

KPI and Charts Requirement on Pizza Sales
using SQL

PIZZA SALES ANALYSIS

BY: ABHISHEK PATEL



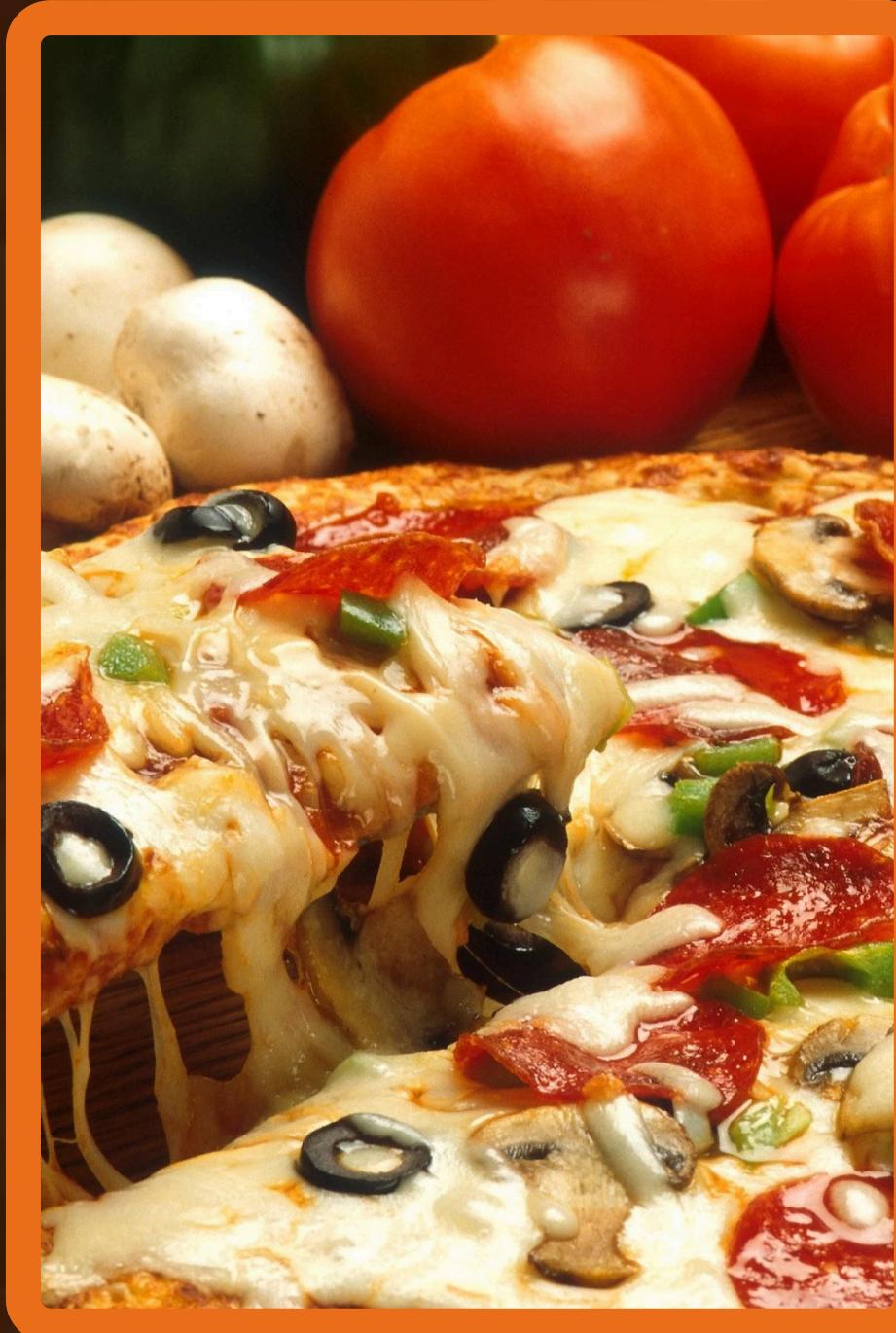
SQL

WHAT'S IN THIS



Hello, I am Abhishek Patel. In this project, I analyzed pizza sales data using MS SQL to identify key trends and insights. I focused on querying and aggregating data to calculate essential metrics and visualizing results to uncover sales patterns. This work provided actionable insights to optimize strategies and improve business performance.





ABOUT OUR PROJECT

- Analyzed pizza sales data using MS SQL Server to address key KPIs, such as Total Revenue, Average Order Value, and Total Pizzas Sold.
- Designed SQL queries to generate trends (daily, hourly, monthly), sales breakdowns by category and size, and rankings of top and bottom sellers.
- Provided actionable insights to boost sales performance and guide strategic decisions.



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.

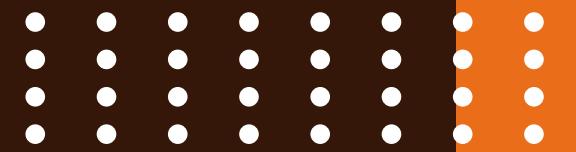
PROBLEM STATEMENT



CHARTS REQUIREMENT

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

KPI'S REQUIREMENT



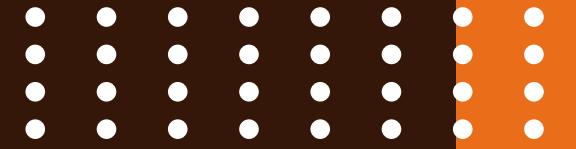
1. TOTAL REVENUE: THE SUM OF THE TOTAL PRICE OF ALL PIZZA ORDERS.

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

The screenshot shows a database query results window. At the top, there are two tabs: 'Results' (highlighted in blue) and 'Messages'. Below the tabs is a table with one row. The table has two columns: the first column is labeled 'Total_Revenue' and the second column contains the value '817860.05083847'. The value in the second column is highlighted with a red border.

Total_Revenue	817860.05083847
1	817860.05083847

KPI'S REQUIREMENT



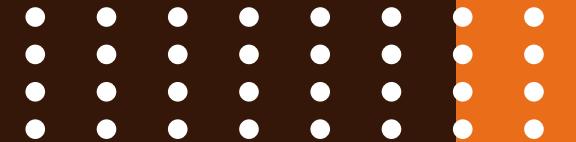
2. AVERAGE ORDER VALUE: THE AVERAGE AMOUNT SPENT PER ORDER, CALCULATED BY DIVIDING THE TOTAL REVENUE BY THE TOTAL NUMBER OF ORDERS.

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

The screenshot shows a database query results window. At the top, there are two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with one row. The table has two columns: the first column is labeled 'Avg_Order_Value' and the second column contains the value '38.3072623343546'. The entire table is enclosed in a dashed border.

	Avg_Order_Value
1	38.3072623343546

KPI'S REQUIREMENT



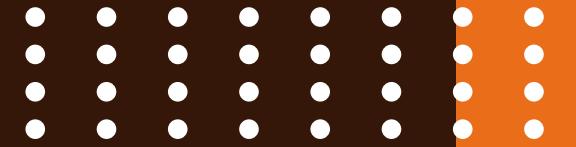
3. TOTAL PIZZAS SOLD: THE SUM OF THE QUANTITIES OF ALL PIZZAS SOLD.

```
SELECT SUM(quantity) AS Total_Pizzas_Sold FROM pizza_sales
```

The screenshot shows a database query results window. At the top, there are two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with one row of data. The table has two columns: the first column is labeled 'Total_Pizzas_Sold' and the second column contains the value '49574'. The 'Results' tab is highlighted with a blue border.

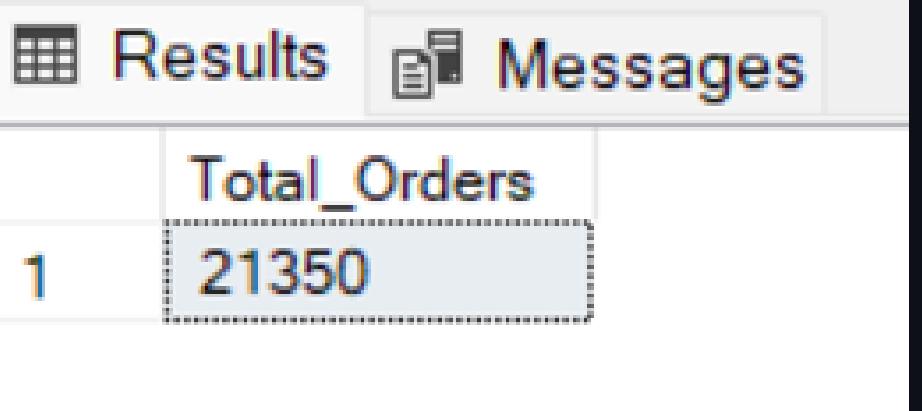
Total_Pizzas_Sold	
1	49574

KPI'S REQUIREMENT



4. TOTAL ORDERS: THE TOTAL NUMBER OF ORDERS PLACED.

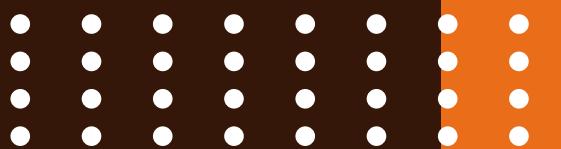
```
SELECT COUNT(DISTINCT order_id) AS Total_Orders FROM pizza_sales
```



The screenshot shows a database query results window. At the top, there are two tabs: "Results" (selected) and "Messages". The results table has one column labeled "Total_Orders". There is one row with the value "21350".

Total_Orders
21350

KPI'S REQUIREMENT



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.

```
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 database query results window. At the top, there are two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with one row. The table has two columns: the first column is labeled 'Avg_Pizzas_Per_Order' and the second column contains the value '2.32'. The number '1' is also present in the first column, indicating the count of rows.

	Avg_Pizzas_Per_Order
1	2.32

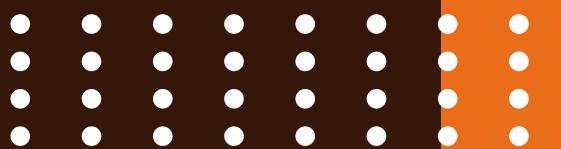
PROBLEM STATEMENT



CHARTS REQUIREMENT

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

CHARTS REQUIREMENT



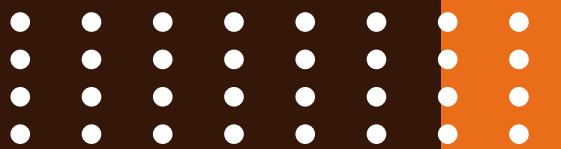
1. DAILY TREND FOR TOTAL ORDERS

```
SELECT DATENAME(DW, order_date) AS Order_Day,  
COUNT(DISTINCT order_id) AS Total_Orders  
FROM pizza_sales  
GROUP BY DATENAME(DW, order_date)
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with seven rows, each representing a day of the week and its corresponding total number of orders. The table has two columns: 'Order_Day' and 'Total_Orders'. The data is as follows:

	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

CHARTS REQUIREMENT



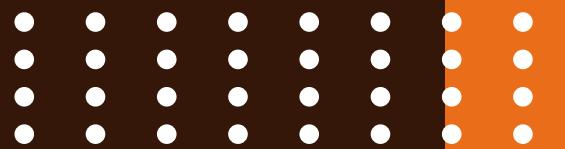
2. HOURLY TREND FOR TOTAL ORDERS

```
SELECT DATEPART(HOUR, order_time) AS Order_Hours,  
COUNT(DISTINCT order_id) AS Total_Orders  
FROM pizza_sales  
GROUP BY DATEPART(HOUR, order_time)  
ORDER BY DATEPART(HOUR, order_time)
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with two columns: 'Order_Hours' and 'Total_Orders'. The data is as follows:

	Order_Hours	Total_Orders
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

CHARTS REQUIREMENT



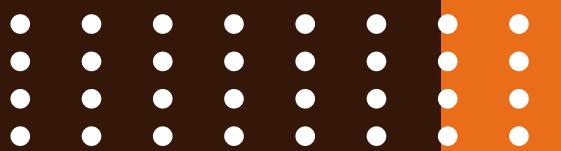
3. MONTHLY TREND FOR TOTAL ORDERS

```
SELECT DATENAME(MONTH, order_date) AS Month_Name,  
COUNT(DISTINCT order_id) AS Total_Orders  
FROM pizza_sales  
GROUP BY DATENAME(MONTH, order_date)
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with 12 rows, each representing a month and its total number of orders. The table has three columns: a row number, the month name, and the total order count. The data is as follows:

	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

CHARTS REQUIREMENT



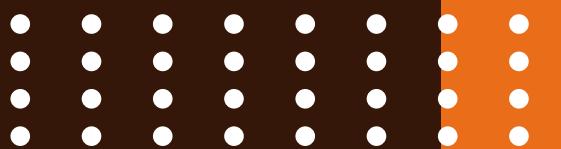
4. PERCENTAGE OF SALES BY PIZZA CATEGORY

```
SELECT pizza_category,
       SUM(total_price) AS Total_Sales,
       SUM(total_price) * 100 /
       (SELECT SUM(total_price) FROM pizza_sales) AS PCT_Total_Sales
  FROM pizza_sales
 GROUP BY pizza_category
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with four columns: pizza_category, Total_Sales, and PCT_Total_Sales. The table has four rows, each corresponding to a pizza category: Classic, Chicken, Veggie, and Supreme. The data is as follows:

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

CHARTS REQUIREMENT



4. PERCENTAGE OF SALES BY PIZZA CATEGORY (JAN MONTH)

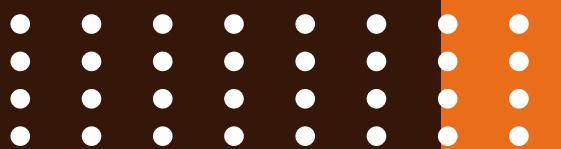
Jan Month:

```
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_Total_Sales
FROM pizza_sales
WHERE MONTH(order_date) = 1
GROUP BY pizza_category
```

Results Messages

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

CHARTS REQUIREMENT



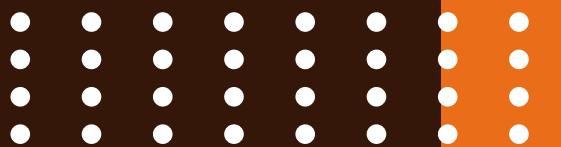
5. PERCENTAGE OF SALES BY PIZZA SIZE

```
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) AS DECIMAL(10,2)) AS PCT_Total_Sales
  FROM pizza_sales
 GROUP BY pizza_size
 ORDER BY PCT_Total_Sales DESC
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with five rows of data. The table has four columns: 'pizza_size', 'Total_Sales', 'PCT_Total_Sales', and a row number column. The data is as follows:

	pizza_size	Total_Sales	PCT_Total_Sales
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

CHARTS REQUIREMENT



5. PERCENTAGE OF SALES BY PIZZA SIZE (FIRST QUARTER)

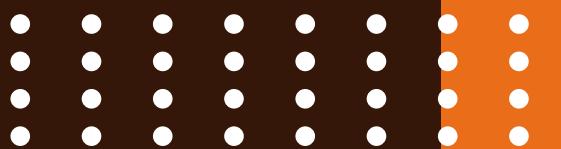
First Quarter:

```
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_Total_Sales
    FROM pizza_sales
   WHERE DATEPART (QUARTER, order_date) =1
  GROUP BY pizza_size
 ORDER BY PCT_Total_Sales DESC
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with four columns: pizza_size, Total_Sales, and PCT_Total_Sales. The data is as follows:

	pizza_size	Total_Sales	PCT_Total_Sales
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

CHARTS REQUIREMENT



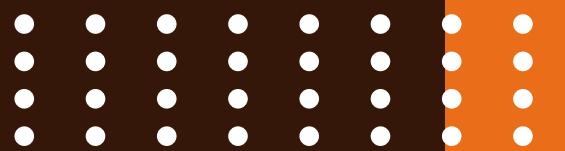
6. TOTAL PIZZAS SOLD BY PIZZA CATEGORY

```
SELECT pizza_category,  
       SUM(quantity) AS Total_Pizzas_Sold  
  FROM pizza_sales  
 GROUP BY pizza_category
```

The screenshot shows a database query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table with four rows of data. The table has two columns: 'pizza_category' and 'Total_Pizzas_Sold'. The data is as follows:

	pizza_category	Total_Pizzas_Sold
1	Classic	14388
2	Chicken	11050
3	Veggie	11649
4	Supreme	11987

CHARTS REQUIREMENT



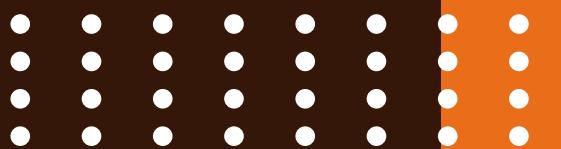
7. TOP 5 BEST SELLERS BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS

```
SELECT TOP 5 pizza_name,
SUM(quantity) AS Total_Pizzas_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY SUM(quantity) DESC
```

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and displays a table with five rows, each representing a pizza name and its total sales quantity. The table has two columns: 'pizza_name' and 'Total_Pizzas_Sold'. The data is as follows:

	pizza_name	Total_Pizzas_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

CHARTS REQUIREMENT



8. BOTTOM 5 BEST SELLERS BY REVENUE, TOTAL QUANTITY AND TOTAL ORDERS

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

The screenshot shows a SQL query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table with five rows. The table has two columns: 'pizza_name' and 'Total_Pizzas_Sold'. The data is as follows:

	pizza_name	Total_Pizzas_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

THANK YOU FOR ATTENTION

About Me:

GitHub: <https://github.com/abhishekpatel16>

LinkedIn: <https://www.linkedin.com/in/abhishekpatel16/>

Email ID: abhishek.skpatel@gmail.com

Phone No: +91 6232120827