

DATA ANALYST PORTFOLIO PROJECT

POWER BI +SQL

BY:- Dimple Sharma



Intro

Project Overview

Through this project we need to analyze key indicators for our pizza sales data to gain our insights into our business performance. Specially, I calculated the following metrices:

- Total Revenue
- Average Order Value
- Total Pizzas sold
- Total Orders
- Average Pizzas Per Order, etc.

I would also like to visualize our pizza sales data to gain insights & understand key trends such as daily trend, monthly trend, yearly trend with the help of bar graphs and charts and DAX Expressions.

PART-1

MYSQL SERVER

IMPORTING THE DATA



WRITING SQL QUERIES



CREATION OF DATABASE



CREATING REPORT



PART-2

POWER BI

CONNECTING TO MYSQL
SERVER



DATA PROCESSING



CLEANING OF DATA



DATA VISUALIZATION



SOFTWARE USED

MS OFFICE/ EXCEL



POWER BI



MYSQL SERVER 19.0



QUERY-1

Selecting all records from the particular table i.e pizza_sales

result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:				
pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_
1	1	hawaiian_m	1	01-01-2015	11:38:36	13.25	13.25	M
2	2	classic_dlx_m	1	01-01-2015	11:57:40	16	16	M
3	2	five_cheese_l	1	01-01-2015	11:57:40	18.5	18.5	L
4	2	ital_supr_l	1	01-01-2015	11:57:40	20.75	20.75	L
5	2	mexicana_m	1	01-01-2015	11:57:40	16	16	M
6	2	thai_dkn_l	1	01-01-2015	11:57:40	20.75	20.75	L
7	3	ital_supr_m	1	01-01-2015	12:12:28	16.5	16.5	M
8	3	prsc_argla_l	1	01-01-2015	12:12:28	20.75	20.75	L
9	4	ital_supr_m	1	01-01-2015	12:16:31	16.5	16.5	M
10	5	ital_supr_m	1	01-01-2015	12:21:30	16.5	16.5	M
11	6	bbq_dkn_s	1	01-01-2015	12:29:36	12.75	12.75	S
12	6	the_greek_s	1	01-01-2015	12:29:36	12	12	S

SYNTAX:

use pizzadb;

SELECT * FROM pizza_sales;

QUERY-2 (kpers)

Total Revenue- Sum of the total price of all orders

Total_Revenue	
▶	378447.10000000516

SYNTAX:

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

QUERY-3 (kprs)

Average Order value- The average amount spent per order, calculated by dividing the total revenue by total orders

	Avg_order_Value
→	38.33928679971687

SYNTAX:

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

QUERY-4 (kpers)

Total Pizzas Sold- The sum of quantities of all pizzas sold

	Total_pizza_sold
▶	22917

SYNTAX:

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

QUERY-5 (kpers)

Average Pizzas Per Order-
Calculated by dividing the total
Number of pizzas sold by total
number of orders

	Avg_Pizzas_per_order
→	2.32

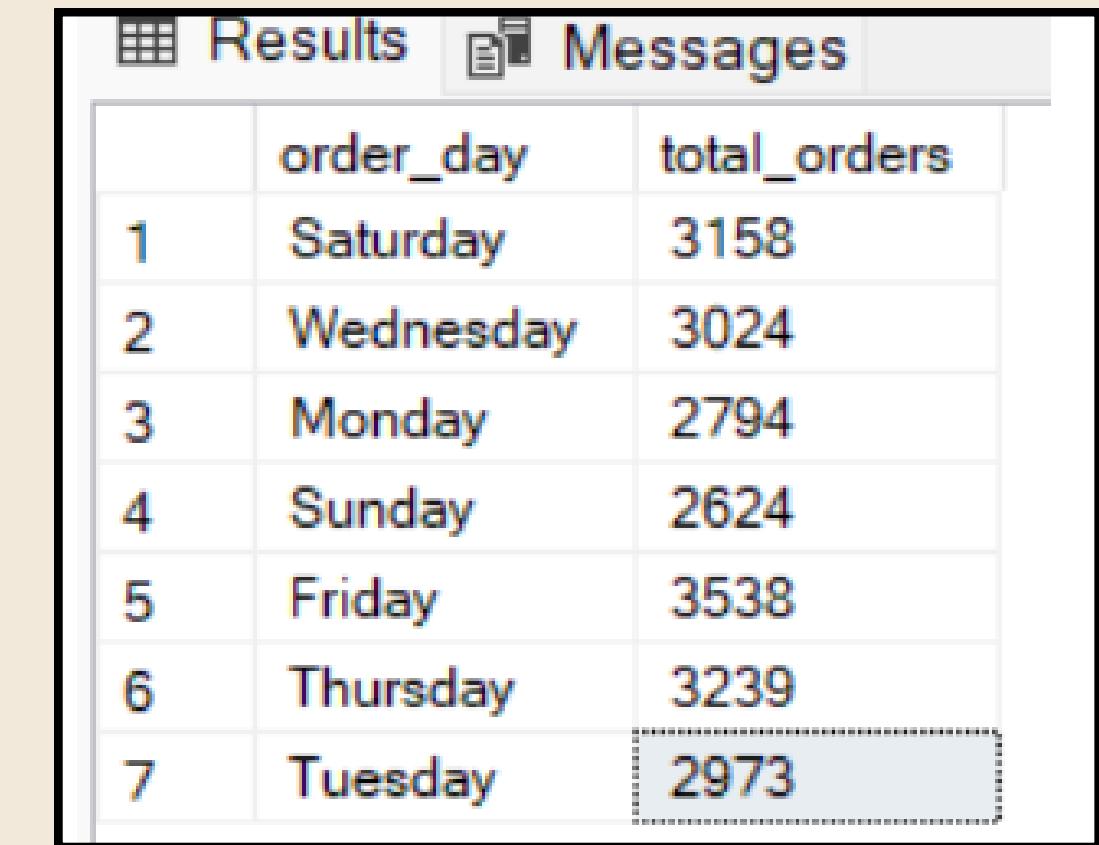
SYNTAX:

```
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;
```

QUERY-6

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 displays a table with 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

QUERY-1

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)
```

	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

QUERY-8

% of Sales by Pizza Category

	pizza_category	total_revenue	PCT
▶	Classic	101089.50	26.71
	Veggie	90893.35	24.02
	Supreme	96628.50	25.53
	Chicken	89835.75	23.74

```
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
```

QUERY-9

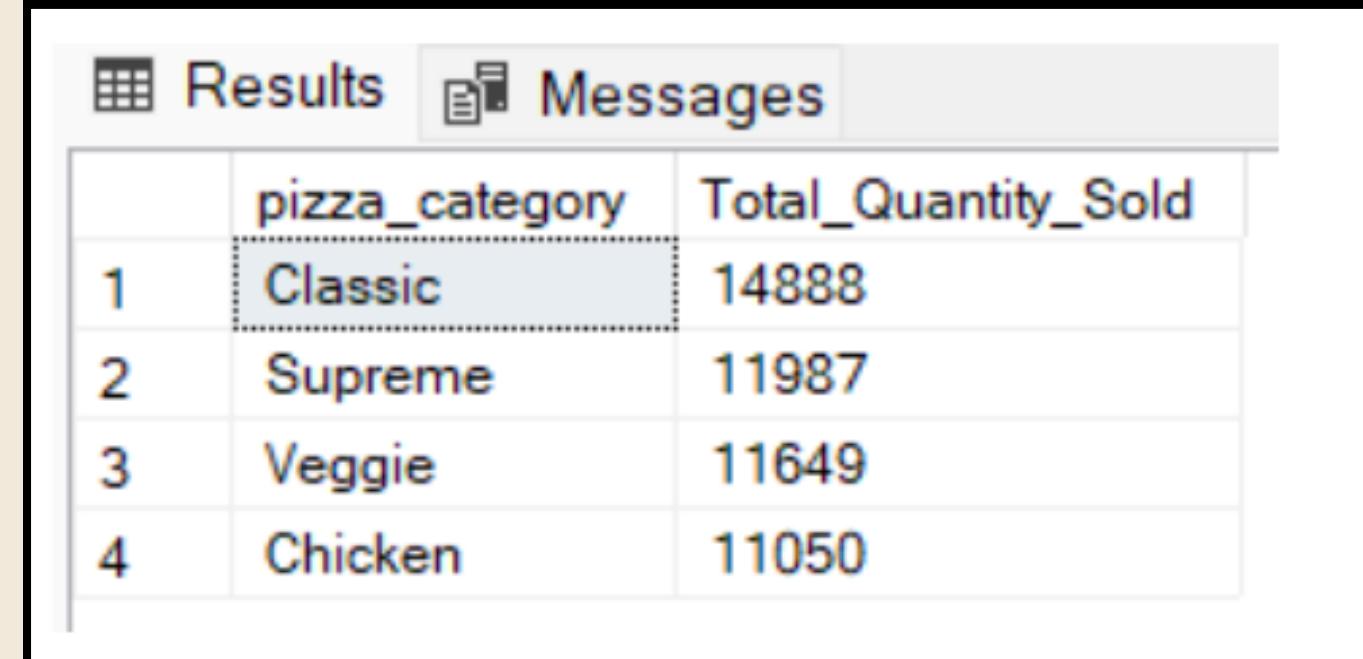
% of Sales by Pizza Size

	pizza_size	total_revenue	PCT
▶	L	174269.10	46.05
	M	114961.25	30.38
	S	81767.00	21.61
	XL	6910.50	1.83
	XXL	539.25	0.14

```
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
```

QUERY-10

Total Pizzas Sold by Pizza Category



The screenshot shows a database interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active and displays a table with four rows. The table has two columns: 'pizza_category' and 'Total_Quantity_Sold'. The data is as follows:

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

```
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
```

QUERY-11

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
```



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. The table has three columns: 'pizza_name' (containing the names of the pizzas), 'Total_Revenue' (containing their respective total revenues), and a row number column (labeled '1' through '5'). The data is as follows:

	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

QUERY-12

Bottom 5 Pizzas by Revenue

pizza_name	Total_Revenue
The Brie Carre Pizza	11588.4998130798
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

```
SELECT Top 5 pizza_name, SUM(total_price) AS Total_Revenue  
FROM pizza_sales  
GROUP BY pizza_name  
ORDER BY Total_Revenue ASC;
```

INSIGHTS

PART-2 (INSIGHTS)

