

Aim: The tutorial describes the project that it is a 2 way confirmation project like using sql and also through power bi.

The power BI output has donut charts, bar charts, scatter plots and some kinda information.

Problem Statement is divided into 2 parts

1. KPI requirements : Key performance Indicator

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.

2.

PROBLEM STATEMENT

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. **Monthly 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.

PROBLEM STATEMENT

CHARTS REQUIREMENT

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.

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.

6. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders

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

7. Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders

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

Tools I am using : PgAdmin, power BI

I tried creating a table and importing file in pgadmin but the date column in pizzas file has data in formats like 2/2/2023 and also 2-2-2023, but postgresql accepting only one type of format.

```
create table pizza_sales(
    pizza_id int,
    order_id int,
    pizza_name_id varchar,
    quantity int,
    order_date date,
    order_time time,
    unit_price float,
    total_price float,
    pizza_size varchar,
    pizza_category varchar,
    pizza_ingredients varchar,
    pizza_name varchar
)
```

So used sqlalchemy and connected to postgresql server and created a table.

```
from sqlalchemy import create_engine
from sqlalchemy import create_engine
conn_string = 'postgresql://postgres:0427@localhost/pizza_db'
db = create_engine(conn_string)
```

```
conn = db.connect()  
df = pd.read_csv('pizza_sales.csv')  
df.to_sql('pizza_sales', con = conn, if_exists='replace', index = False)
```

Querying the table to get the results:

1. select sum(total_price) as total_revenue from pizza_sales

A screenshot of a database query interface. The top bar has several icons: a grid, a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a save icon, a download icon, and a refresh icon. Below this is a table with one row of data. The table has two columns: the first is empty, and the second is labeled "total_revenue" with the type "double precision". A small lock icon is next to the column name. The data row shows a value of 1 and a value of 817860.0499999993.

	total_revenue double precision
1	817860.0499999993

- 2.select sum(total_price)/count(distinct order_id) as average_order_value
from pizza_sales

A screenshot of a database query interface. The top bar has several icons: a grid, a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a save icon, a download icon, and a refresh icon. Below this is a table with one row of data. The table has two columns: the first is empty, and the second is labeled "average_order_value" with the type "double precision". A small lock icon is next to the column name. The data row shows a value of 1 and a value of 38.307262295081635.

	average_order_value double precision
1	38.307262295081635

3. select sum(quantity) as pizzas_sold
from pizza_sales

A screenshot of a database query interface. The top bar has several icons: a grid, a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a save icon, a download icon, and a refresh icon. Below this is a table with one row of data. The table has two columns: the first is empty, and the second is labeled "pizzas_sold" with the type "numeric". A small lock icon is next to the column name. The data row shows a value of 1 and a value of 49574.

	pizzas_sold numeric
1	49574

4. select count(distinct order_id) as total_orders
from pizza_sales

	total_orders	bigint
1	21350	

5. select round(sum(quantity)/count(distinct order_id), 2) as avg_pizza_ordered_per_order
from pizza_sales

	avg_pizza_ordered_per_order	numeric
1	2.32	

Chart Outputs:

Finding the total number of orders per each day

```
1. SELECT TO_CHAR(TO_DATE(ORDER_DATE, 'DD-MM-YYYY'), 'Day') AS
   day_of_week, count(distinct order_id)
   FROM PIZZA_SALES
   group by TO_CHAR(TO_DATE(ORDER_DATE, 'DD-MM-YYYY'), 'Day')
```

	day_of_week	count
	text	bigint
1	Friday	3538
2	Monday	2794
3	Saturday	3158
4	Sunday	2624
5	Thursday	3239
6	Tuesday	2973
7	Wednesday	3024

Finding orders for each month

```
2. SELECT TO_CHAR(TO_DATE(ORDER_DATE, 'DD-MM-YYYY'), 'Month') AS
   Month_of_year, count(distinct order_id)
   FROM PIZZA_SALES
   group by TO_CHAR(TO_DATE(ORDER_DATE, 'DD-MM-YYYY'), 'Month')
```

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

Percentage of each category

3. select pizza_name, sum(quantity) as pizza_quantity
from pizza_sales

	pizza_category	sum	pct
	text	double precision	double precision
1	Supreme	208196.9999999822	25.45631126009884
2	Chicken	195919.5	23.955137556847493
3	Veggie	193690.45000000298	23.682590927384783
4	Classic	220053.1000000001	26.9059602556699

Percentage of each size

4.select pizza_size, sum(total_price), sum(total_price)*100 / (select sum(total_price) from pizza_sales) as PCT

```
from pizza_sales  
group by pizza_size
```

A screenshot of a database query results window. The top bar contains various icons for file operations like save, open, and export. The main area is a table with four columns: pizza_size (text), sum (double precision), pct (double precision), and an unlabeled fourth column. The data rows are as follows:

	pizza_size	sum	pct
1	S	178076.49999999843	21.773468455880682
2	XXL	1006.6000000000005	0.12307729176892906
3	XL	14076	1.7210768517181052
4	M	249382.25	30.492044451859723
5	L	375318.7000000087	45.8903329487743

For first quarter

```
5.select pizza_size, sum(total_price), sum(total_price)*100 / (select sum(total_price) from pizza_sales) as PCT  
from pizza_sales  
WHERE DATE_PART('quarter',TO_DATE(ORDER_DATE, 'DD-MM-YYYY'))=1  
group by pizza_size
```

A screenshot of a database query results window. The top bar contains various icons for file operations like save, open, and export. The main area is a table with four columns: pizza_size (text), sum (double precision), pct (double precision), and an unlabeled fourth column. The data rows are as follows:

	pizza_size	sum	pct
1	L	95229.6499999946	11.643758611268552
2	M	61159	7.477929750939727
3	S	45384.2500000008	5.549146213951944
4	XL	3289.5	0.40220817730368763
5	XXL	287.5999999999997	0.03516494050540828

Top 5 pizzas based on total_revenue

```
6.select pizza_name, sum(total_price) as total_revenue  
from pizza_sales
```

```
group by pizza_name  
order by sum(total_price) desc limit 5
```

	pizza_name text	total_revenue double precision
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

Least 5 pizzas based on total_revenue

```
6..select pizza_name, sum(total_price) as total_revenue  
from pizza_sales  
group by pizza_name  
order by sum(total_price) limit 5
```



	pizza_name text	total_revenue double precision
1	The Brie Carre Pizza	11588.4999999999
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

Top 5 pizzas based on total quantity

```
7.select pizza_name, sum(quantity) as total_quantity  
from pizza_sales  
group by pizza_name  
order by sum(quantity) desc limit 5
```

	pizza_name text	total_quantity numeric
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 pizzas based on total quantity

```
8. select pizza_name, sum(quantity) as total_quantity
from pizza_sales
group by pizza_name
order by sum(quantity) limit 5
```

	pizza_name text	total_quantity numeric
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

Top 5 pizzas based on total orders

```
9.select pizza_name, count(distinct order_id) as total_orders
from pizza_sales
group by pizza_name
order by count(distinct order_id) desc limit 5
```

	pizza_name	total_orders
	text	bigint
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

Bottom 5 pizzas based on total orders

```
10.select pizza_name, count(distinct order_id) as total_orders
from pizza_sales
group by pizza_name
order by count(distinct order_id) limit 5
```

	pizza_name	total_orders
	text	bigint
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Calabrese Pizza	918
4	The Spinach Supreme Pizza	918
5	The Chicken Pesto Pizza	938

Now we have to prove our results using Power BI

1. Started with cleaning the data like changing sizes to full forms like s-Regular, M-Medium, L-Large, XL = X-Large, XXL = XXL-Large
2. Created measures like Total_revenue, Total_price, Total_orders, Average_revenue, Total_pizzas, average num of pizza per order
3. Formatted these by selecting the measure cards and selecting format your visual and applying the properties like layouts, alignment, images etc., and using general for setting height and width.
4. Then inserted shapes using insert shapes and selected rounded square and selected color using style (#573B92)
5. Added image and again created text box and added date with no background effect.
6. Now applied transformation and extracted day from the date column and closed teh transformation.

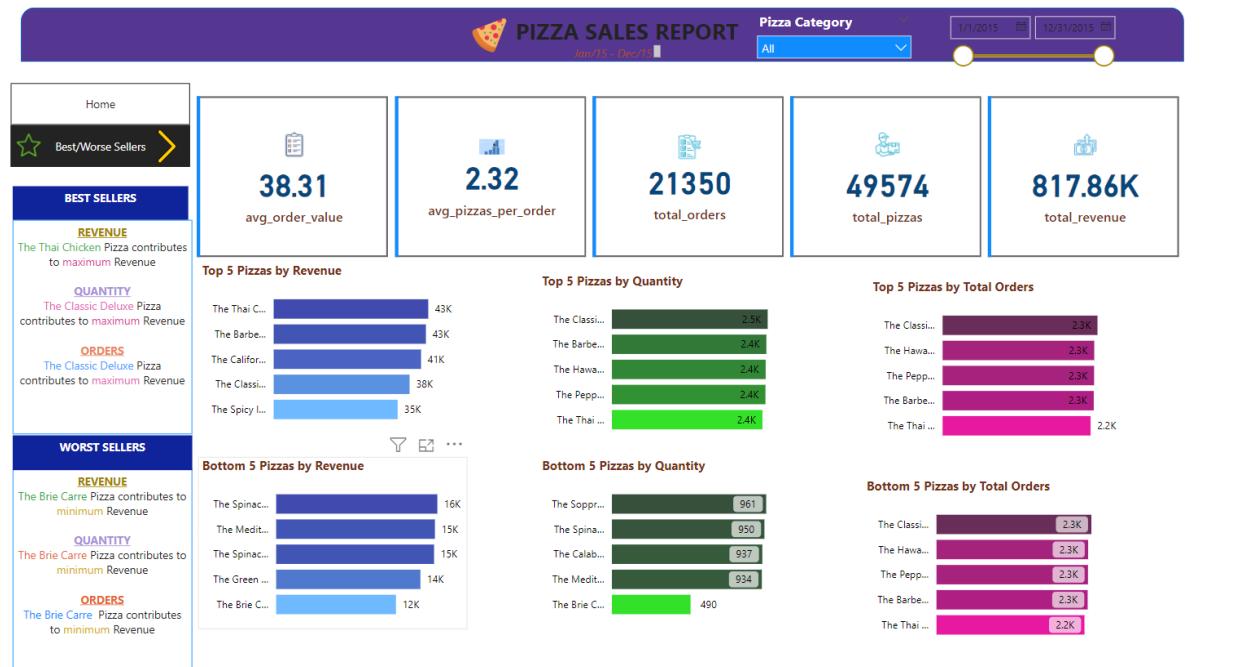
7. Then created the new column with function “`order_day = UPPER(LEFT('public pizza_sales'[Day Name], 3))`”

Power BI Report

Page : 1



Page : 2



8. Also created new columns for month as well using power bi features and generated teh reports using it.
9. Created text boxes for the text for best sellers and worst sellers and inserted buttons using page navigator.
10. Created slicer and applied different styling like dropdown and relative time for the two slicers we have on header.

The End