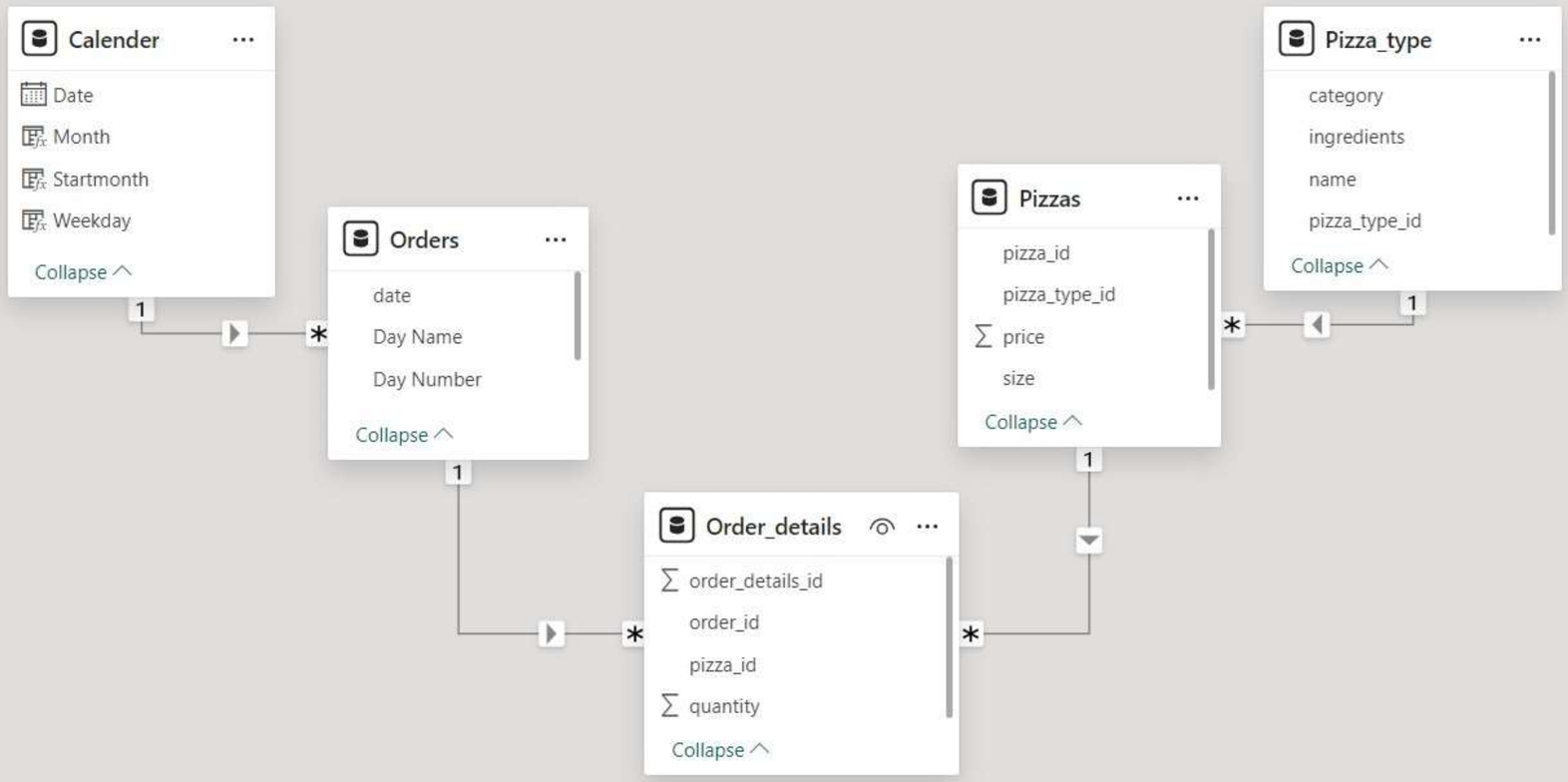
DOMINO'S

P I Z Z A

Name : Shraddhanath Kar

HELLO

My name is Shraddhanath Kar. In this project I have utilized SQL queries to solve question related to pizza sales and also visualize in Power BI where its help to understand the insights of the data very clearly.


Calender
Date
Month
Startmonth
Weekday
Collapse
Orders
date
Day Name
Day Number
Collapse
Order_details
order_details_id
order_id
pizza_id
quantity
Collapse
Pizzas
pizza_id
pizza_type_id
price
size
Collapse
Pizza_type
category
ingredients
name
pizza_type_id
Collapse
1
*

Retrieve total number of pizzas sold.

```sql
SELECT
    SUM(quantity) AS pizza_sold
FROM
    order_details;
```

Result Grid

	pizza_sold
▶	49574

Average order values

```sql
SELECT
    ROUND(SUM(order_details.quantity * pizza_details.price) / COUNT(DISTINCT (order_details.order_id)),
            2) AS avg_order_value
FROM
    order_details
        JOIN
    pizza_details ON pizza_details.pizza_id = order_details.pizza_id;
```

Result Grid

avg_order_value
38.31

Average number of pizza per order

```sql
SELECT
    ROUND(SUM(order_details.quantity) / COUNT(DISTINCT (order_details.order_id)),
            0) AS avg_no_pizza_per_order
FROM
    order_details;
```

Result Grid | Filter Rows

avg_no_pizza_per_order
2

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

Calculate the total revenue and number of order per category.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales,
    COUNT(DISTINCT (order_details.order_id)) AS total_orders
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY total_sales DESC;
```

Result Grid | Filter Rows:

category	total_sales	total_orders
Classic	220053.1	10859
Supreme	208197	9085
Chicken	195919.5	8536
Veggie	193690.45	8941

Total revenue and number of order per size.

```sql
SELECT
    pizza_details.size,
    SUM(order_details.quantity * pizza_details.price) AS total_revenue,
    COUNT(DISTINCT (order_details.order_id)) AS total_orders
FROM
    order_details
        JOIN
    pizza_details ON order_details.pizza_id = pizza_details.pizza_id
GROUP BY pizza_details.size;
```

Result Grid | Filter Rows:

size	total_revenue	total_orders
L	375318.7000000083	12736
M	249382.25	11159
S	178076.4999999984	10490
XL	14076	544
XXL	1006.6000000000005	28

Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size, COUNT(order_details.quantity) AS order_count
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid | Filter Row

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

List the most ordered pizza types along with their quantity.

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Ro

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hourly,monthly,daily in orders and revenue of pizzas.

```sql
SELECT
    CASE
        WHEN HOUR(orders.order_time) BETWEEN 9 AND 12 THEN 'Morning'
        WHEN HOUR(orders.order_time) BETWEEN 12 AND 15 THEN 'Afternoon'
        WHEN HOUR(orders.order_time) BETWEEN 15 AND 18 THEN 'Evening'
        WHEN HOUR(orders.order_time) BETWEEN 18 AND 21 THEN 'Dinner'
        WHEN HOUR(orders.order_time) BETWEEN 21 AND 23 THEN 'Late Night'
        ELSE 'Others'
    END AS meal_time,
    COUNT(DISTINCT (order_details.order_id)) AS total_orders
FROM
    order_details
        JOIN
    orders ON order_details.order_id = orders.order_id
GROUP BY meal_time
ORDER BY total_orders DESC;
```

Result Grid | Filter Rows:

meal_time	total_orders
Evening	6655
Afternoon	5395
Dinner	4849
Morning	3760
Late Night	691

Weekdays orders of pizzas.

```sql
SELECT
    DAYNAME(orders.order_date) AS day_name,
    COUNT(DISTINCT (order_details.order_id)) AS total_orders
FROM
    order_details
        JOIN
    orders ON order_details.order_id = orders.order_id
GROUP BY DAYNAME(orders.order_date)
ORDER BY total_orders DESC;
```

Result Grid | Filter Rows:

day_name	total_orders
Friday	3538
Thursday	3239
Saturday	3158
Wednesday	3024
Tuesday	2973
Monday	2794
Sunday	2624

Monthwise orders of pizzas.

```sql
SELECT
    MONTHNAME(orders.order_date) AS day_name,
    COUNT(DISTINCT (order_details.order_id)) AS total_orders
FROM
    order_details
        JOIN
    orders ON order_details.order_id = orders.order_id
GROUP BY MONTHNAME(orders.order_date)
ORDER BY total_orders DESC;
```

Result Grid Filter Rows:

day_name	total_orders
July	1935
May	1853
January	1845
August	1841
March	1840
April	1799
November	1792
June	1773
February	1685
December	1680
September	1661
October	1646

Result 14

Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizza_order_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizza_order_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue_percentage
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage DESC;
```

Result Grid Filter Rows:

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
select order_date,sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,sum(order_details.quantity * pizzas.price) as revenue
from order_details
join orders on order_details.order_id = orders.order_id
join pizzas on order_details.pizza_id = pizzas.pizza_id
group by orders.order_date)
as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.850000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002

Result 20

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select name,revenue,ranks from
(select category,name,revenue,rank() over(partition by category order by revenue desc) as ranks
from
(select pizza_types.category,pizza_types.name,
sum(order_details.quantity * pizzas.price) as revenue
from
pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category,pizza_types.name) as a) as b
where ranks <= 3;
```

Result Grid | Filter Rows: | Export:

name	revenue	ranks
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

Result 26

Most order pizza.

```sql
SELECT
    pizza_types.name, count(order_details.order_id) AS count_pizzas
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY count_pizzas DESC
LIMIT 1;
```

Result Grid | Filter Rows:

name	count_pizzas
The Classic Deluxe Pizza	2416

1/1/2015 | 12/31/2015

PIZZA SALES DASHBOARD

Pizza Category

Chicken | Classic | Supreme | Veggie

$817.86K	21,350	49,574	2	$38.31
Total Revenue	Total Orders	Pizza Sold	Avg Pizza Per Order	Avg Order Value

Busiest Days & Times

Days

Orders are highest on weekend

Friday/Thursday evenings.

Monthly

Maximum orders from month are July & May.

Sales Performance

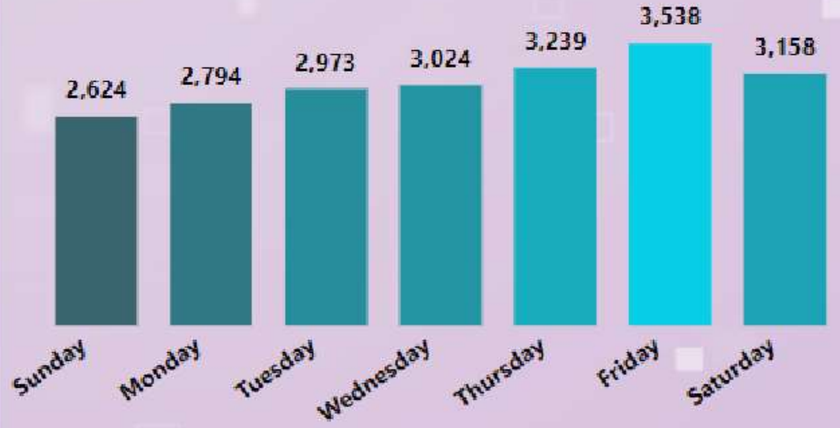
Category

Classic Category contributes maximum sales and orders.

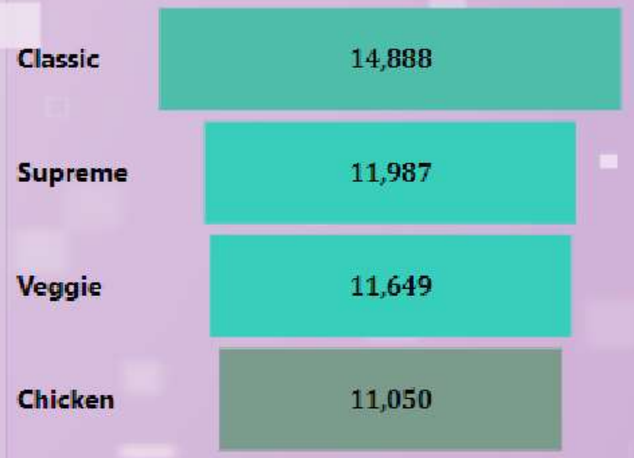
Size

'L' size pizza contributes to maximum sales.

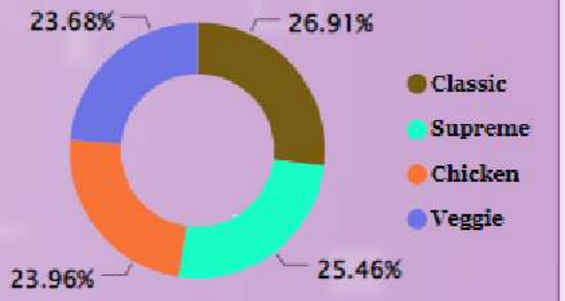
Total orders by Week Day



Pizza Sold by category

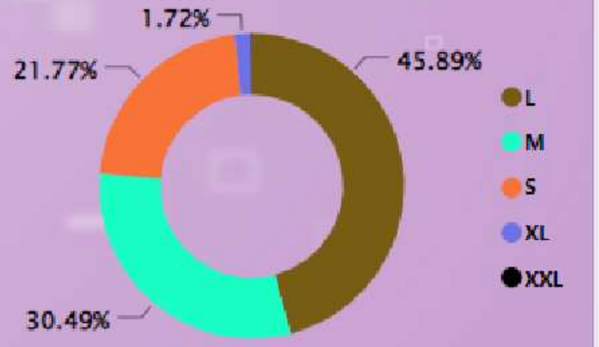


Total Revenue by category



Total Orders by Month

Total Revenue by size



1/1/2015 12/31/2015

PIZZA SALES DASHBOARD

Pizza Category

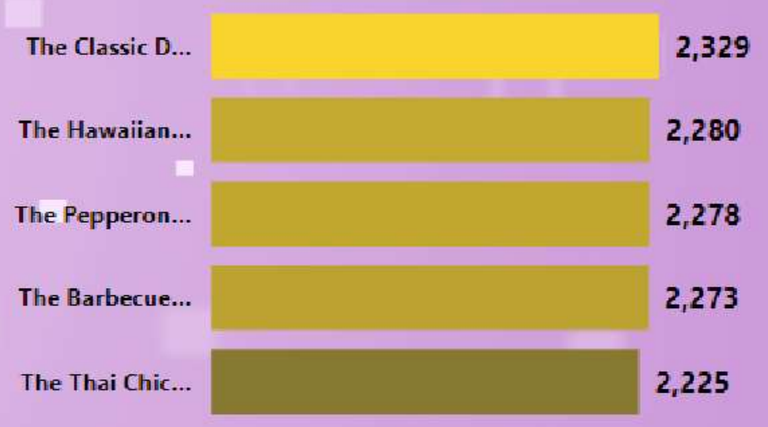
Chicken | Classic | Supreme | Veggie

$817.86K	21,350	49,574	2	$38.31
Total Revenue	Total Orders	Pizza Sold	Avg Pizza Per Order	Avg Order Value

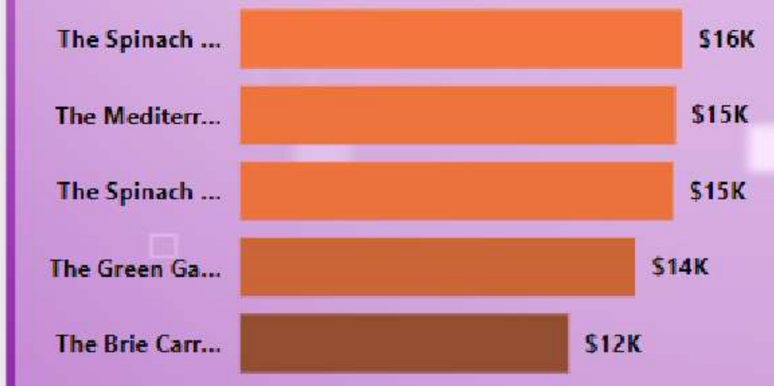
Top 5 Pizzas By Revenue

Top 5 Pizzas By Quantity

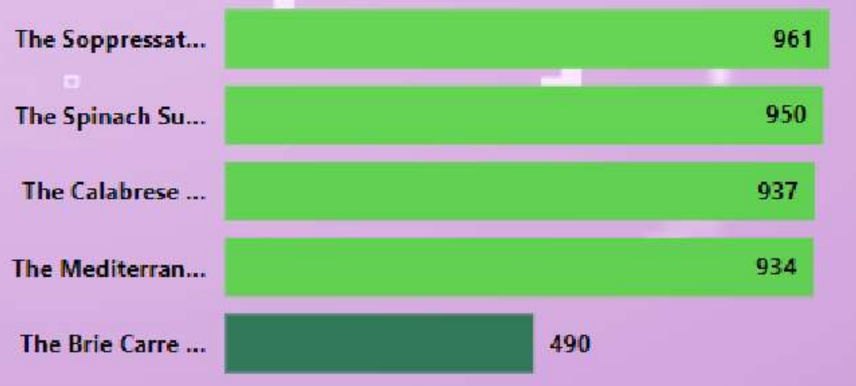
Top 5 Pizzas By Total Orders



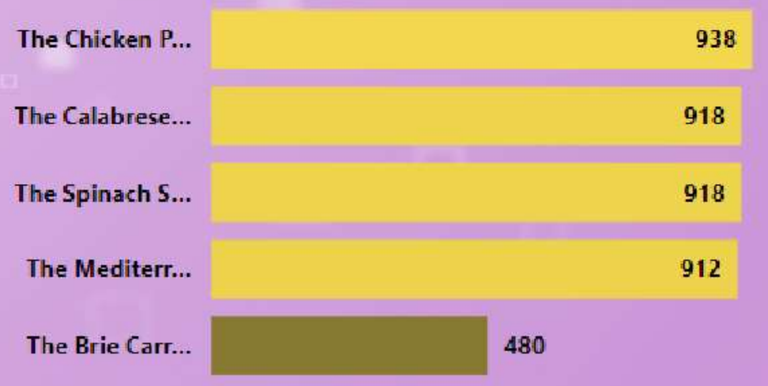
Bottom 5 Pizzas By Revenue



Bottom 5 Pizzas By Quantity



Bottom 5 Pizzas By Total Orders



Problem Statement

We need to analyze key indicator 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 pizza with quantity.

2. Total Orders : The total number of order placed.

3. Pizza Sold : The sum of the quantities of pizzas sold.

4.Average Pizza Per Order : The average number of pizza sold per order, Calculate by dividing the total number of pizza sold by total order placed.

5.Average Order Value : The average amount per order, Calculate the total revenue by total number of orders.

Problem Statement

Charts Requirement

We would like to visualize various aspect of our pizzas sales data to gain insights and understand key trends. We have identified the following requirements to create charts:

1. **Total Orders By Weekday :** Create a column charts that show daily trends of total orders over weekdays. This chart will help us to identify any pattern or fluctuation in order volume on a daily basis.
2. **Total Orders By Month :** Create an area chart that illustrate total orders by month wise. This charts allows to knowing that which month has highest sale and which month has orders value less according to the whole year.
3. **Pizza Sold By Category:** Create a funnel chart to show pizza sold according to category. By this chart we can compare the sales performance of different pizza categories.

Problem Statement

Charts Requirement

4.**Total Revenue By Category :** Create a donut chart that shows distribution of sales across different pizza categories. this chart provide insights into popularity of different pizza categories and their contribution to overall sales.

5. **Total Revenue By Size :** Create a donut chart that represent the percentage of sales attributes to different pizza size. This will understand customer performance for pizza sizes and its impact on sales.

6.**Top 5 best seller by Revenue, Quantity and Total Orders :** Create a bar chart highlighting the top 5 best selling pizza based on the Revenue, Total Quantity and Total Orders. This will help us to identify the most popular pizzasby option.

7.**Bottom 5 worst seller by Revenue, Quantity and Total Orders :** Create a bar chart highlighting the bottom 5 worst selling pizzas based on Revenue, Total Quantity and total Orders. This will help us to identify underperforming or less popular pizza options.

Software Used

MS Office/ Excel : Version 2021

My SQL Workbench : Version 8.0

Power BI Version: 2.128.1177.0 64-bit (April 2024)

THANK YOU

I appreciate the opportunity to share this project with you.