

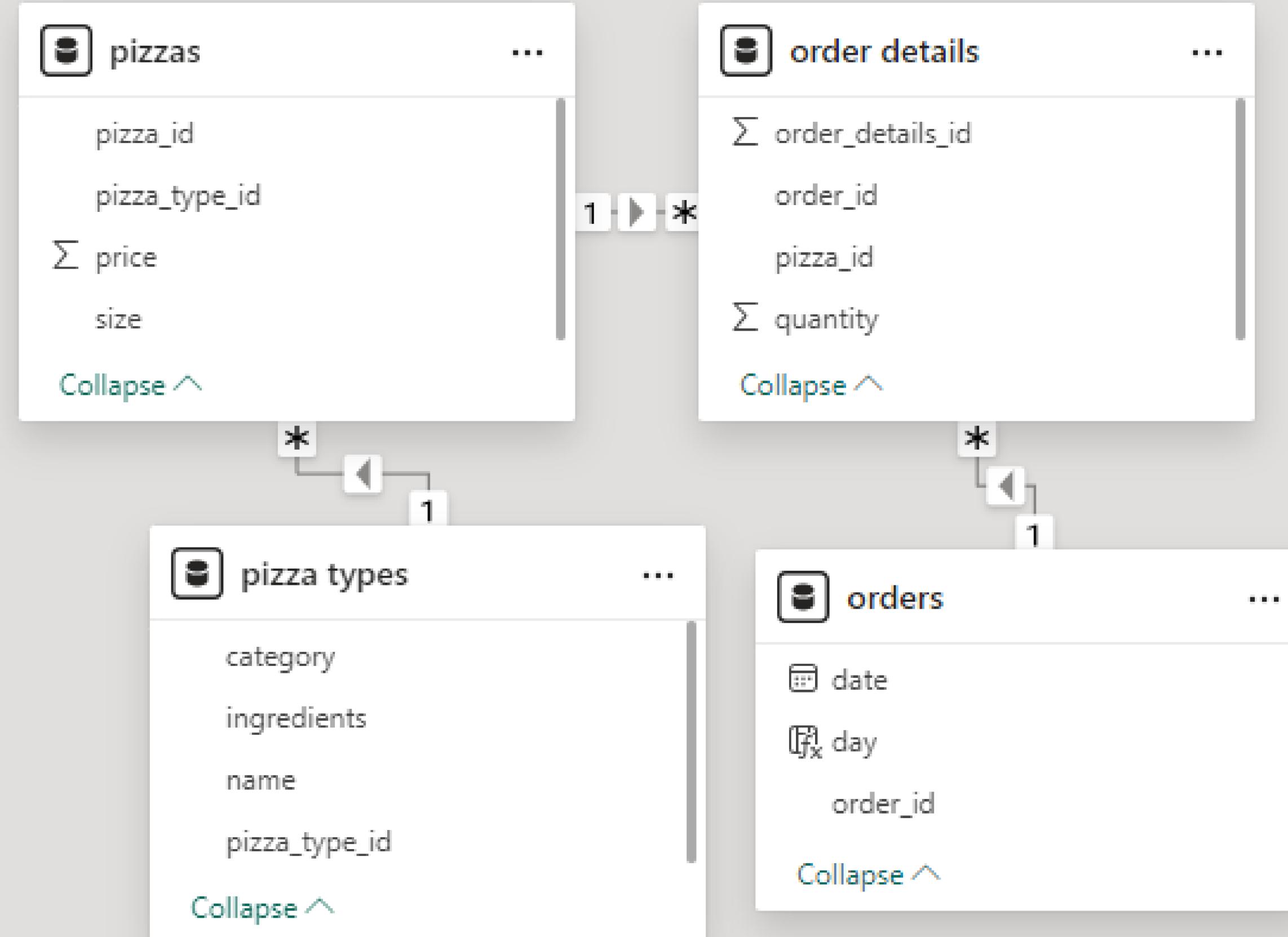
# PIZZA SALES DATA ANALYSIS

I have used MYSQL to analyse key performance indicators and trends and POWERBI for data visualisation

PRESENTED BY PRITAM JANGID

Let's start our adventure in the world of Analysis!

# TABLES AND THEIR COLUMNS



# Retrieve the total number of orders placed.

```
1      -- Retrieve the total number of orders placed.  
2 •  SELECT  
3      COUNT(orders.order_id) total_orders  
4  FROM  
5  orders;  
6
```

	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

```
1      -- Calculate the total revenue generated from pizza sales.  
2 •  SELECT  
3      ROUND(SUM(od.quantity * p.price), 2) total_revenue  
4  FROM  
5      order_details od  
6      JOIN  
7      pizzas p ON od.pizza_id = p.pizza_id;
```

	total_revenue
▶	817860.05

# Identify the highest-priced pizza.

```
1      -- Identify the highest-priced pizza.  
2 •  SELECT  
3          pt.name, p.price  
4  FROM  
5          pizza_types pt  
6              JOIN  
7          pizzas p ON pt.pizza_type_id = p.pizza_type_id  
8  ORDER BY p.price DESC  
9  LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

```
1 -- Identify the most common pizza size ordered.  
2 • SELECT  
3     COUNT(od.quantity) total_order, p.size  
4 FROM  
5     order_details od  
6         JOIN  
7     pizzas p ON od.pizza_id = p.pizza_id  
8 GROUP BY p.size;
```

	total_order	size
▶	15385	M
	18526	L
	14137	S
	544	XL
	28	XXL

# List the top 5 most ordered pizza types along with their quantities.

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2 • SELECT
3      pt.name,
4      COUNT(od.order_id) total_order,
5      SUM(od.quantity) total_quantity
6  FROM
7      pizza_types pt
8          JOIN
9      pizzas p ON pt.pizza_type_id = p.pizza_type_id
10         JOIN
11     order_details od ON p.pizza_id = od.pizza_id
12  GROUP BY pt.name
13 ORDER BY total_quantity DESC
14 LIMIT 5;
```

	name	total_order	total_quantity
▶	The Classic Deluxe Pizza	2416	2453
	The Barbecue Chicken Pizza	2372	2432
	The Hawaiian Pizza	2370	2422
	The Pepperoni Pizza	2369	2418
	The Thai Chicken Pizza	2315	2371

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

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2 • SELECT  
3      pt.category, SUM(od.quantity) total_quantity  
4  FROM  
5      pizza_types pt  
6          JOIN  
7      pizzas p ON pt.pizza_type_id = p.pizza_type_id  
8          JOIN  
9      order_details od ON p.pizza_id = od.pizza_id  
10     GROUP BY pt.category  
11     ORDER BY total_quantity;
```

	category	total_quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

# Determine the distribution of orders by hour of the day.

```
1  -- Determine the distribution of orders by hour of the day.  
2 • SELECT  
3      HOUR(time) time_in_hour, COUNT(order_id) order_count  
4  FROM  
5    orders  
6 GROUP BY HOUR(time)  
7 ORDER BY time_in_hour;
```

	time_in_hour	order_count
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472

# Join relevant tables to find the category-wise distribution of pizzas.

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2 • SELECT  
3      category, COUNT(name) pizza_count  
4  FROM  
5      pizza_types  
6  GROUP BY category;
```

	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
1      -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2 •  SELECT  
3      ROUND(AVG(count), 0) avg_pizzas_opd  
4  FROM  
5  (SELECT  
6      o.date, SUM(od.quantity) count  
7  FROM  
8      orders o  
9  JOIN order_details od ON o.order_id = od.order_id  
10     GROUP BY date) AS a;
```

	avg_pizzas_opd
▶	138

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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.  
2 • SELECT  
3      pt.name, SUM(p.price * od.quantity) revenue  
4  FROM  
5      pizza_types pt  
6          JOIN  
7      pizzas p ON pt.pizza_type_id = p.pizza_type_id  
8          JOIN  
9      order_details od ON p.pizza_id = od.pizza_id  
10 GROUP BY pt.name  
11 ORDER BY revenue DESC  
12 LIMIT 3
```

	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.

```
SELECT
```

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

	category	per_cont
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# Analyze the cumulative revenue generated over time.

```
-- Analyze the cumulative revenue generated over time.
```

```
SELECT
```

```
    date,  
    SUM(daily_sales) OVER (ORDER BY date) AS cumulative_sales
```

```
FROM (
```

```
    SELECT
```

```
        o.date AS date,  
        SUM(od.quantity * p.price) AS daily_sales
```

```
    FROM orders o
```

```
    JOIN
```

```
        order_details od ON o.order_id = od.order_id
```

```
    JOIN
```

```
        pizzas p ON od.pizza_id = p.pizza_id
```

```
    GROUP BY o.date
```

```
) AS aggregated_sales ORDER BY date;
```

	date	cumulative_sales
▶	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

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

```
1    -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
2 • select category , name , revenue from  
3   (select pt.name , pt.category, sum(od.quantity*p.price) revenue,  
4    rank() over(partition by category order by sum(od.quantity*p.price) desc) rnk  
5    from order_details od join pizzas p on od.pizza_id = p.pizza_id  
6    join pizza_types pt on pt.pizza_type_id=p.pizza_type_id  
7    group by pt.name , pt.category )as a where rnk <=3;
```

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

[Clear all slicers](#)

date



size

 L
  M
  S
  XL
  XXL

Total Orders

49K

Total Revenue

818K

Most Ordered Pizza

The Classic Deluxe Pizza

Orders By Day

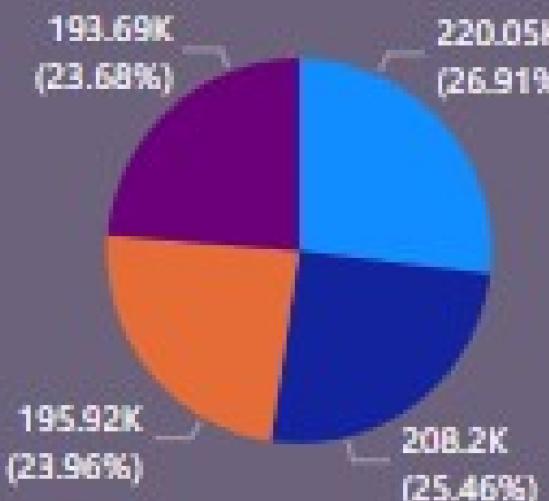
revenue by Month



category

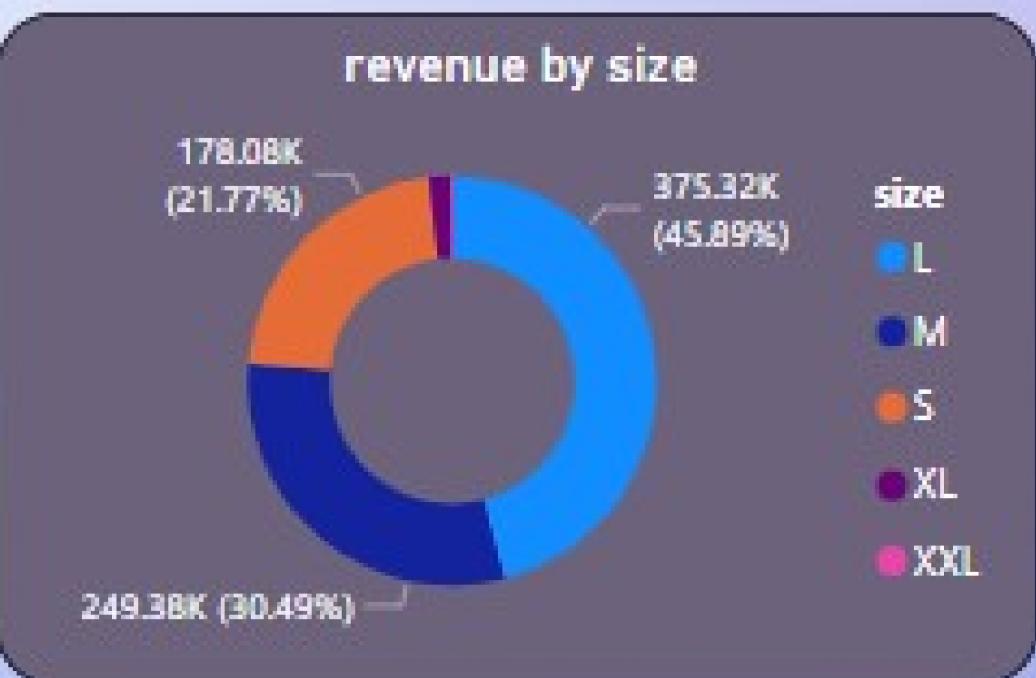
 Chicken
  Classic
  Supreme
  Veggie

revenue by category



- category
- Classic
- Supreme
- Chicken
- Veggie

revenue by size



name

name	category	revenue
The Hawaiian Pizza	Classic	32,273.25
The Italian Supreme Pizza	Supreme	33,476.75
The Southwest Chicken Pizza	Chicken	34,705.75
The Spicy Italian Pizza	Supreme	34,831.25
The Classic Deluxe Pizza	Classic	38,180.50
The California Chicken Pizza	Chicken	41,409.50
The Barbecue Chicken Pizza	Chicken	42,768.00
The Thai Chicken Pizza	Chicken	43,434.25



# THANK YOU

COMBINING BOTH MYSQL AND  
POWERBI LED TO VALUABLE  
INSIGHTS FOR MAKING BETTER  
DECISIONS AND IMPROVE SALES  
STRATEGIES. PLEASE DO COMMENT  
FOR ANY IMPROVEMENT