

# Pizza Sales Analysis



This presentation explores the analysis of pizza sales data to uncover insights and identify opportunities for improvement. I have used the tools MySQL workbench for querying and analysis and Microsoft PowerPoint for Presentation ,By examining sales trends, customer preferences, and other key metrics, we can optimize operations, enhance customer satisfaction, and drive revenue growth, which will help to the pizza hut company.



by Gitanjali Pekamwar



# Table of Contents

- 1) Sales Performance Analysis
- 2) Customer behavior Insights
- 3) Revenue Trends
- 4) Operational Efficiency Metrics



# Introduction to the Dataset

## Pizza Sales Dataset

There are 4 datasets(Tables)

1)orders

2)order details

3)pizzas

4)pizza types

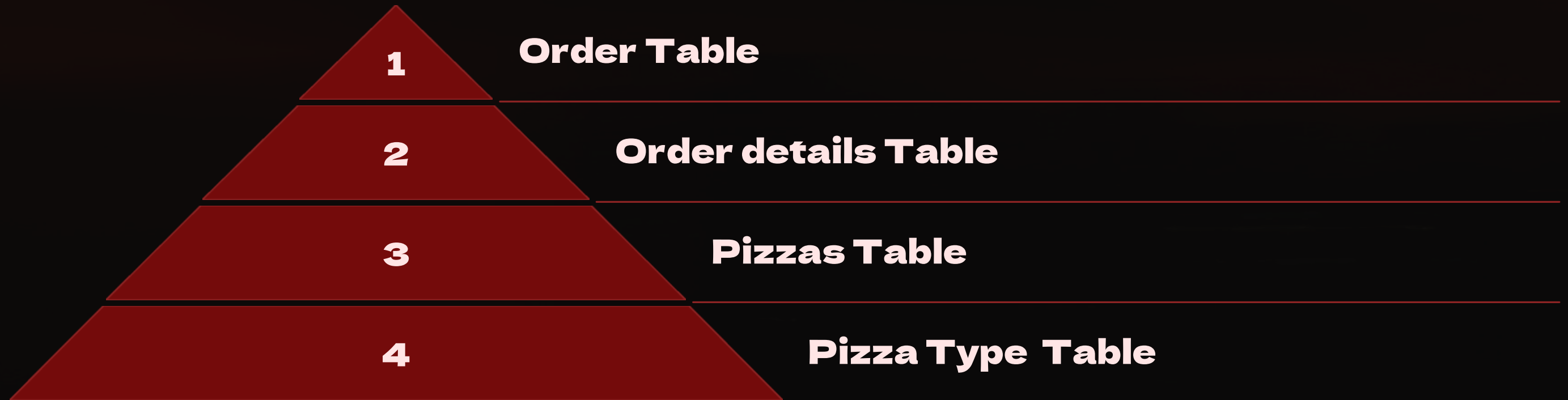
## Key Fields

- Order ID
- Order Date
- Pizza Type ID
- Pizza ID
- Category
- Price
- Quantity
- Name



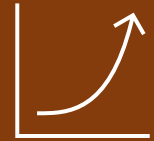


# Data Modeling



A relational database model was designed to organize the pizza sales data efficiently, ensuring data consistency and facilitating analysis.

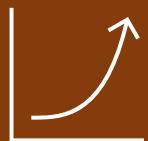
# 1) Retrieve the total number of orders placed.



```
1  -- Retrieve the total number of orders placed.
2
3  • SELECT
4      COUNT(order_id) AS total_orders
5  FROM
6      orders;
7
```

Result Grid

	total_orders
▶	21350



2) Calculate the total revenue generated from pizza sales.


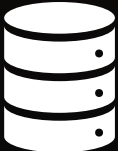
```
-- Calculate the total revenue generated from pizza sales.

• SELECT
    ROUND(SUM(pizzas.price * order_details.quantity),2)
      AS total_revenue
FROM
    pizzas
    INNER JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id;
```

	total_revenue
▶	817860.05

### 3) Identify the highest-priced pizza.

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



	name	highest_priced_pizza
▶	The Greek Pizza	35.95

## 4) Identify the most common pizza size ordered.

```
1  -- Identify the most common pizza size ordered.
2  • SELECT
3      pizzas.size AS size,
4      COUNT(order_details.order_details_id) as max_orders
5  FROM
6      pizzas
7      INNER JOIN
8      order_details ON pizzas.pizza_id = order_details.pizza_id
9  GROUP BY pizzas.size
10 order by max_orders DESC;
```

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



# 5) 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      pizza_types.name, SUM(order_details.quantity) AS quantity
4  FROM
5      pizza_types
6      JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9      order_details ON order_details.pizza_id = pizzas.pizza_id
10 GROUP BY pizza_types.name
11 ORDER BY quantity DESC
12 LIMIT 5;
```



	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

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



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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050





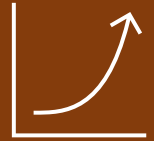
## 7) Determine the distribution of orders by hour of the day.

```
1  #Determine the distribution of orders by hour of the day.
2
3  • SELECT
4      HOUR(order_time) AS hours, COUNT(order_id) AS quantity
5  FROM
6      orders
7  GROUP BY HOUR(order_time)
8  ORDER BY quantity DESC
```

Result Grid		Filter Rows:
	hours	quantity
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1



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



```
1  -- join relevant tables to find the
2  -- category-wise distribution of pizzas.
3
4 • SELECT
5     category, COUNT(name)
6 FROM
7     pizza_types
8 GROUP BY category;
9
```

Result Grid			Filter Rows:	Exp
	category	COUNT(name)		
▶	Chicken	6		
	Classic	8		
	Supreme	9		
	Veggie	9		



## 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
2  -- average number of pizzas ordered per day.
3
4 • SELECT
5     ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
6 FROM
7     (SELECT
8         orders.order_date, SUM(order_details.quantity) AS quantity
9     FROM
10        orders
11     INNER JOIN order_details ON orders.order_id = order_details.order_id
12     GROUP BY orders.order_date) AS order_quantity;
```

Result Grid		Filter Rows:
	avg_pizza_ordered_per_day	
▶	138	

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

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

Result Grid		Filter Rows:	Export:
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

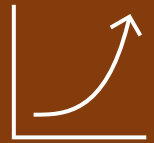


# 11) Calculate the percentage contribution of each pizza Type to total revenue.

```
1  -- Calculate the percentage contribution
2  -- of each pizza type to total revenue.
3
4 • select pizza_types.category,
5 round(sum(order_details.quantity*pizzas.price) / (SELECT
6     ROUND(SUM(order_details.quantity* pizzas.price),2)
7     AS total_revenue
8 FROM
9     order_details
10     INNER JOIN
11     pizzas ON pizzas.pizza_id = order_details.pizza_id)*100,2) as revenue
12 from pizza_types
13 join pizzas
14 on pizza_types.pizza_type_id = pizzas.pizza_type_id
15 join order_details
16 on order_details.pizza_id = pizzas.pizza_id
17 group by pizza_types.category
18 order by revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

## 12) Analyze the cumulative revenue generated over time..



```
1  -- Analyze the cumulative revenue generated over time.
2
3 • select order_date, sum(revenue)
4   over (order by order_date) as cum_revenue
5 from (select
6   orders.order_date,
7   sum(order_details.quantity * pizzas.price) as revenue
8   from order_details
9   join pizzas
10  on order_details.pizza_id = pizzas.pizza_id
11  join orders
12  on orders.order_id = order_details.order_id
13  group by orders.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2713.85000000000004
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



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

```
1  -- Determine the top 3 most ordered pizza
2  -- types based on revenue for each pizza category.
3 • select name, revenue from
4  (select category, name, revenue,
5   rank() over(partition by category order by revenue desc) as rn
6   from
7   (select pizza_types.category, pizza_types.name,
8    sum(order_details.quantity*pizzas.price) as revenue
9    from pizza_types
10   join pizzas
11   on pizza_types.pizza_type_id = pizzas.pizza_type_id
12   join order_details
13   on order_details.pizza_id = pizzas.pizza_id
14   group by pizza_types.category, pizza_types.name) as a) as b
15 where rn <= 3;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content
	name	revenue			
▶	The Thai Chicken Pizza	43434.25			
	The Barbecue Chicken Pizza	42768			
	The California Chicken Pizza	41409.5			
	The Classic Deluxe Pizza	38180.5			
	The Hawaiian Pizza	32273.25			
	The Pepperoni Pizza	30161.75			
	The Spicy Italian Pizza	34831.25			
	The Italian Supreme Pizza	33476.75			
	The Sicilian Pizza	20040.5			



# Conclusion

Working on the "Pizza Sales" project was such an exciting and insightful journey for me! It wasn't just about crunching numbers or writing SQL queries—it was about diving into a real-world dataset and truly understanding the story behind the data. From calculating simple metrics to tackling more advanced revenue analysis, I discovered just how powerful SQL can be in data-backed decisions. Along the way, I learned to write clean and efficient queries, interpret results with precision, and translate numbers into actionable insights that could help a business optimize its operations.

I also to sharpen key SQL skills like working with joins, aggregations, subqueries and time-based analysis. But more importantly and understood how these techniques can be used to identify trends, improve processes, and even maximize profitability. Overall, this project gave me the confidence to tackle real-world challenges and showed me the immense potential of data analytics in transforming businesses.



**Thank You!**

