

PIZZA SALES ANALYSIS



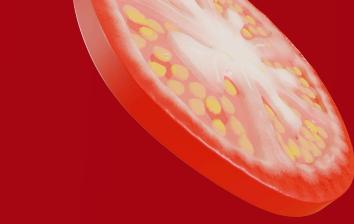
YOGESH
KUSHWAHA

WELCOME TO

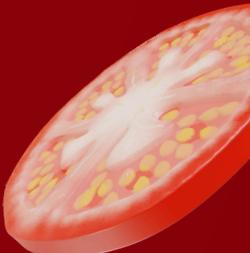
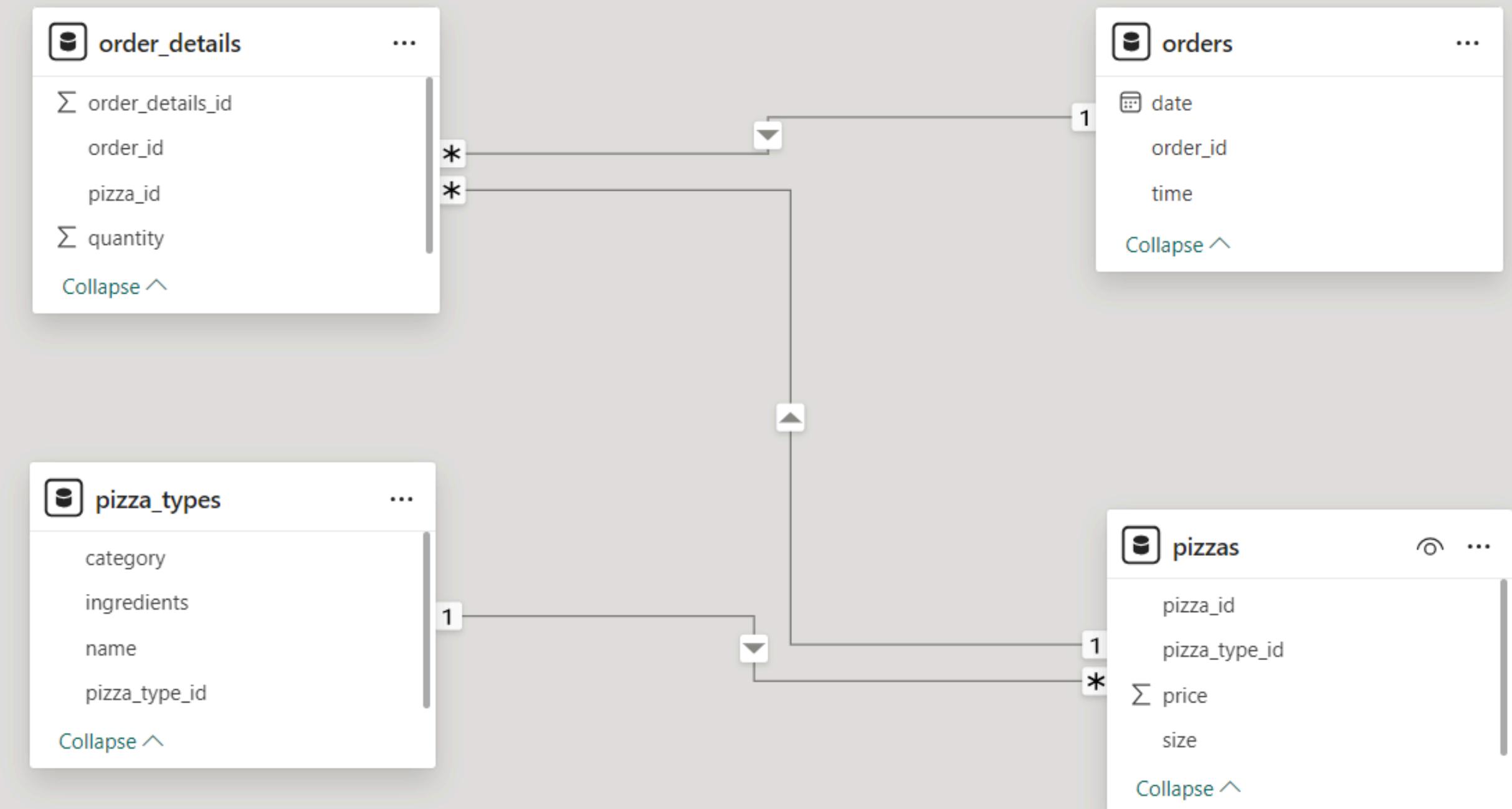
PIZZA SALES ANALYSIS USING MYSQL

This project focuses on analyzing pizza sales data to gain insights and optimize business performance. Analyzed pizza sales to identify top-selling items, busiest hours, revenue trends, and profitable categories using SQL queries. The project showcases the use of SQL for data-driven decision-making in the food service industry.





SCHEMA



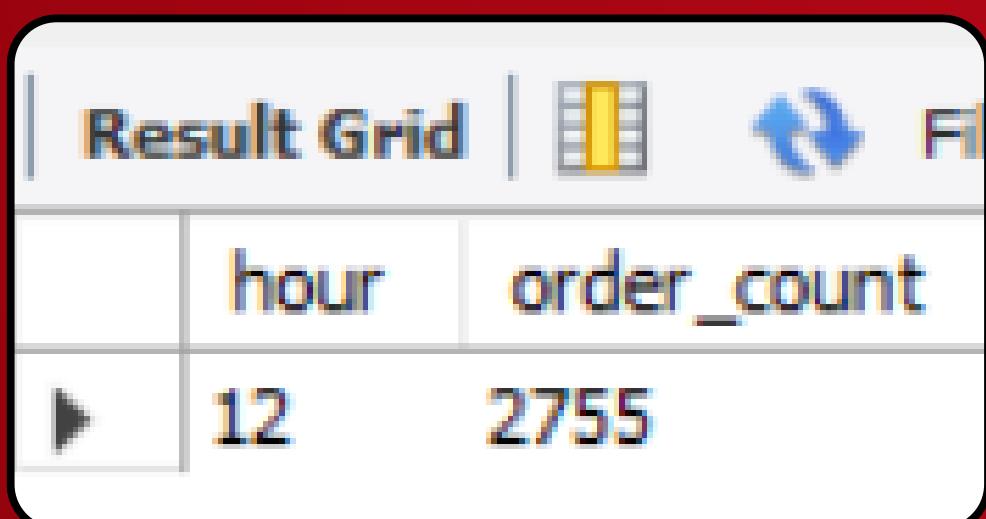


QUESTIONS ANALYZED IN THE PROJECT

1. Determine the busiest hour for pizza orders.
2. Join the necessary tables to find the total quantity of each pizza category ordered.
3. List the top 3 most frequently ordered pizza types by quantity.
4. Identify the least ordered pizza size.
5. Identify the most profitable pizza type based on total revenue.
6. Calculate the monthly trend of pizza sales.
7. Analyze the top pizza categories by revenue.
8. List pizza sizes by category that generated more than \$10,000 in revenue.
9. Predict peak hours for pizza orders using past data.
10. Analyze the top 5 most expensive pizzas and the correlation between pizza price and order volume.

- Determine the busiest hour for pizza orders.

```
SELECT HOUR(order_time) AS hour, COUNT(*) AS order_count  
FROM orders  
GROUP BY HOUR(order_time)  
ORDER BY order_count DESC  
LIMIT 1;
```



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'hour' and 'order_count'. There is one row of data with values 12 and 2755. The grid has a header row with column names and a data row below it. There are navigation icons at the top of the grid.

	hour	order_count
▶	12	2755

← OUTPUT

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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
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.category
ORDER BY total_quantity DESC;
```

OUTPUT ->

	category	total_quantity
▶	Classic	55146
	Supreme	44757
	Veggie	43662
	Chicken	40770

- List the top 3 most frequently ordered pizza types by quantity.

```
3 • SELECT
4      pt.name, SUM(od.quantity) AS total_quantity
5  FROM
6    pizza_types pt
7      JOIN
8    pizzas pz ON pt.pizza_type_id = pz.pizza_type_id
9      JOIN
10   order_details od ON od.pizza_id = pz.pizza_id
11 GROUP BY pt.name
12 ORDER BY total_quantity DESC
13 LIMIT 3;
```

Result Grid | Filter Rows:

	name	total_quantity
▶	The Pepperoni Pizza	9315
	The Barbecue Chicken Pizza	8994
	The Classic Deluxe Pizza	8979

← OUTPUT

- Identify the least ordered pizza size.

```
3 • SELECT
4     pz.size, SUM(od.quantity) AS total_quantity
5 FROM
6     pizzas pz
7     JOIN
8         order_details od ON pz.pizza_id = od.pizza_id
9 GROUP BY pz.size
.0 ORDER BY total_quantity ASC
1 LIMIT 1;
```

OUTPUT ->

	size	total_quantity
▶	XXL	30

- Identify the most profitable pizza type based on total revenue.

```
3 • SELECT
4     pt.name AS pizza_name,
5     ROUND(SUM(od.quantity * pz.price), 0) AS total_revenue
6 FROM
7     order_details od
8     JOIN
9     pizzas pz ON od.pizza_id = pz.pizza_id
10    JOIN
11    pizza_types pt ON pt.pizza_type_id = pz.pizza_type_id
12 GROUP BY pt.name
13 ORDER BY total_revenue DESC
14 LIMIT 1;
```

Result Grid		Filter Rows:
	pizza_name	total_revenue
▶	The Thai Chicken Pizza	160571

← OUTPUT

- Calculate the monthly trend of pizza sales.

```
2
3 • SELECT
4     DATE_FORMAT(o.order_date, '%Y-%m') AS month,
5     SUM(od.quantity) AS total_pizzas_sold
6 FROM
7     orders o
8     JOIN
9     order_details od ON o.order_id = od.order_id
10 GROUP BY DATE_FORMAT(o.order_date, '%Y-%m')
11 ORDER BY month ASC;
```

OUTPUT ->

	month	total_pizzas_sold
▶	2015-01	33854
	2015-02	4457
	2015-03	4261
	2015-04	4151
	2015-05	4328
	2015-06	4107
	2015-07	4392
	2015-08	4168
	2015-09	3890
	2015-10	3883
	2015-11	4266
	2015-12	3935

- Analyze the top pizza categories by revenue.

```
3 • SELECT
4     pt.category,
5         ROUND(SUM(od.quantity * pz.price), 0) AS revenue
6 FROM
7     pizza_types pt
8         JOIN
9     pizzas pz ON pz.pizza_type_id = pt.pizza_type_id
10        JOIN
11    order_details od ON od.pizza_id = pz.pizza_id
12 GROUP BY pt.category
13 ORDER BY revenue DESC;
```

Result Grid | Filter

	category	revenue
▶	Classic	814288
	Supreme	775697
	Veggie	726551
	Chicken	722222

← OUTPUT

- List pizza sizes by category that generated more than \$10,000 in revenue.

```
3 • SELECT
4     pz.size,
5     pt.category,
6     ROUND(SUM(od.quantity * pz.price), 2) AS total_revenue
7   FROM
8     order_details od
9       JOIN
10    pizzas pz ON pz.pizza_id = od.pizza_id
11      JOIN
12    pizza_types pt ON pz.pizza_type_id = pt.pizza_type_id
13   GROUP BY pt.category , pz.size
14   HAVING total_revenue > 10000
15 ORDER BY total_revenue DESC;
```

OUTPUT ->

	size	category	total_revenue
▶	L	Veggie	392035.35
	L	Chicken	375678.75
	L	Supreme	350692.5
	L	Classic	276903
S	Classic	259359	
M	Supreme	245918.25	
M	Chicken	241049.25	
M	Classic	222158.25	
M	Veggie	214138.5	
S	Supreme	179086.65	
S	Veggie	120377.25	
S	Chicken	105493.5	
XL	Classic	52632	

- Predict peak hours for pizza orders using past data.

```
• SELECT  
    HOUR(o.order_time) AS hour, COUNT(o.order_id) AS order_count  
FROM  
    orders o  
GROUP BY hour  
ORDER BY order_count DESC  
LIMIT 3;
```

	hour	order_count
▶	12	2755
	13	2666
	18	2609

← OUTPUT

- Analyze the top 5 most expensive pizzas and the correlation between pizza price and order volume

```
2 • SELECT
3     pt.name AS pizza_name,
4     p.price,
5     SUM(od.quantity) AS total_quantity
6 FROM
7     order_details od
8     JOIN
9     pizzas p ON od.pizza_id = p.pizza_id
10    JOIN
11    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
12 GROUP BY pt.name , p.price
13 ORDER BY p.price DESC
14 LIMIT 5;
```

OUTPUT ->

	pizza_name	price	total_quantity
▶	The Greek Pizza	35.95	90
	The Greek Pizza	25.5	2064
	The Brie Carre Pizza	23.65	1776
	The Italian Vegetables Pizza	21	804
	The California Chicken Pizza	20.75	3294

THANK YOU!

