

# DATA SLICE & INSIGHTS: PIZZA STORE SALES ANALYSIS USING SQL



Analyzed a pizza store dataset using SQL to uncover sales patterns, revenue drivers, and product performance.





# PROBLEM STATEMENT

The pizza business wants to understand:

- Sales performance
- Peak order times
- Best-selling items
- Revenue contribution by type & category
- Customer order behavior

**Yawar Ali**  
**Head Chef Analytics**



# ABOUT OUR DELICIOUS PIZZA DATASET

Tables used:

- order\_details
- orders
- pizza\_types
- pizzas



[Find dataset here.](#)

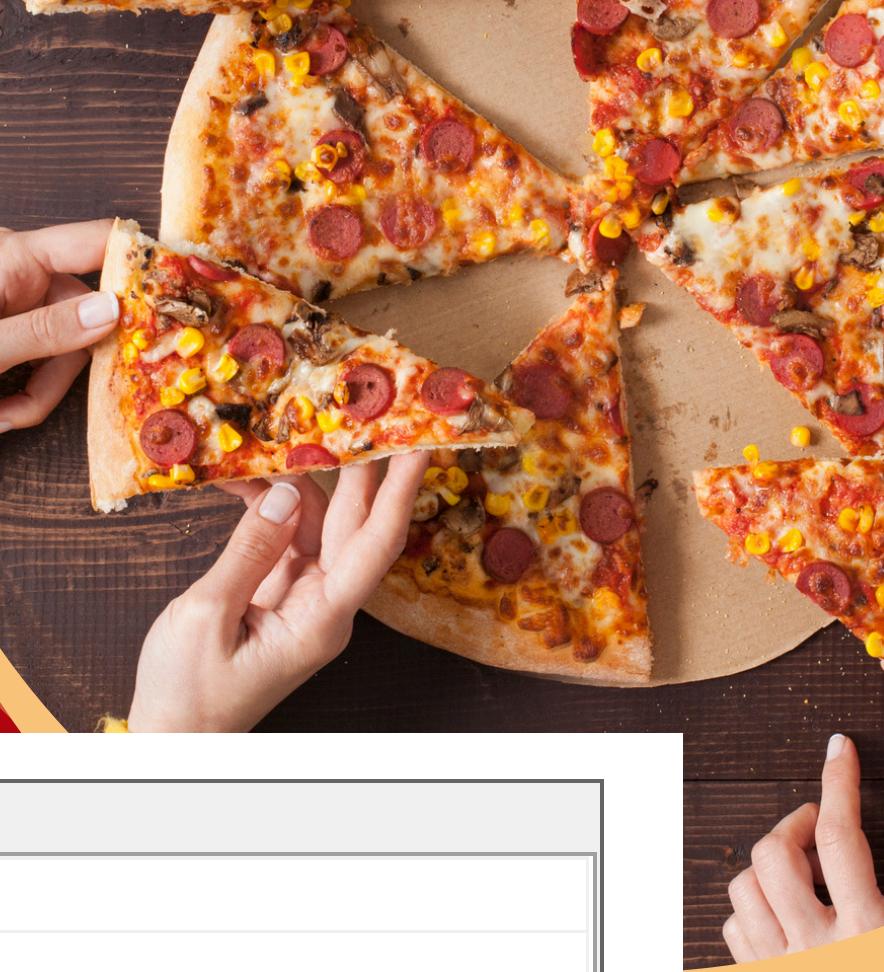
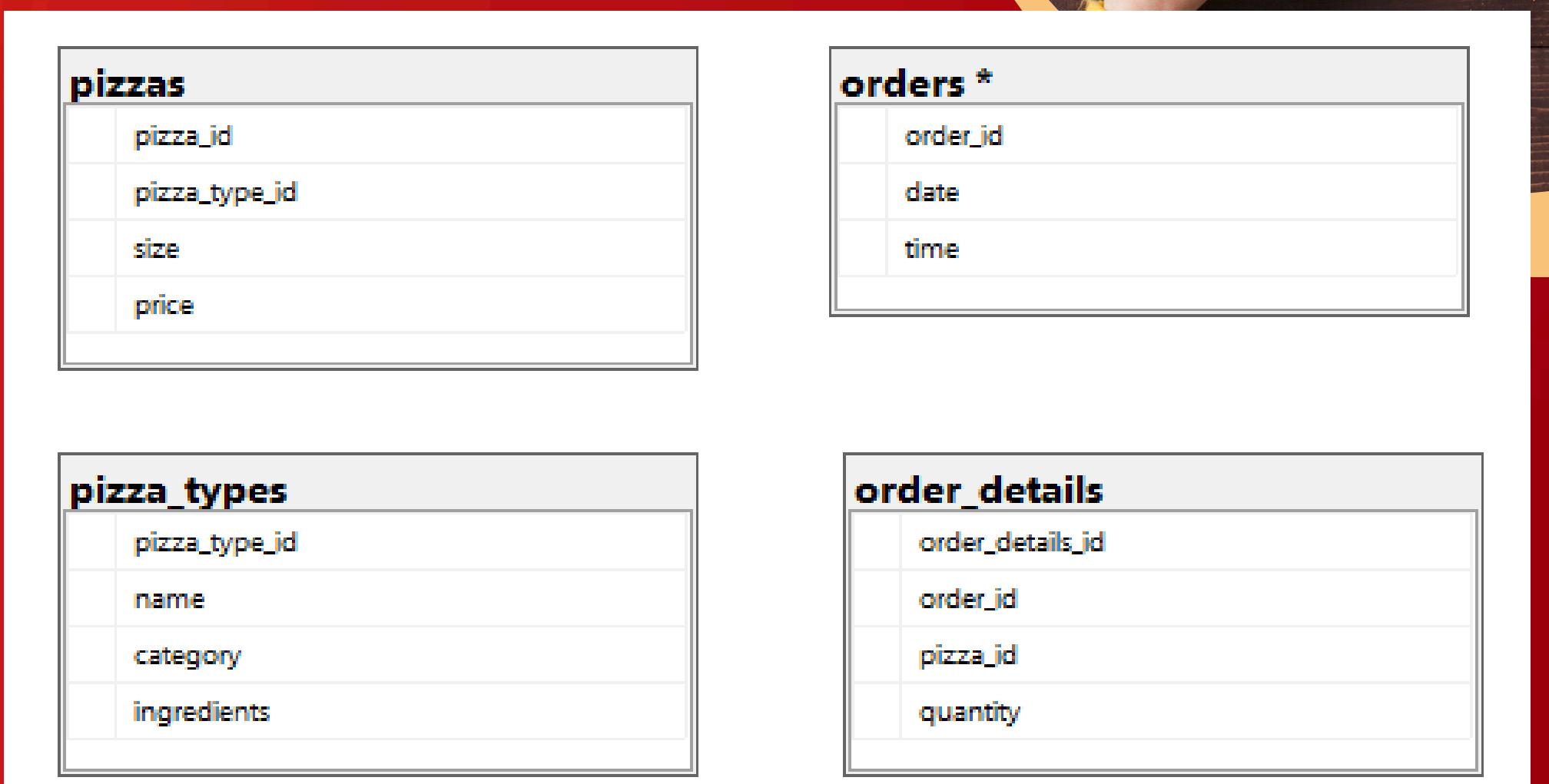


Figure 1: This is an ERD for database.



# FOUNDATIONAL METRICS

## KPIs Analyzed

- ✓ Total Orders: 21,350
- ✓ Total Revenue: \$817,860.05
- ✓ Highest-Priced Pizza: The Greek Pizza (\$35.95)
- ✓ Most Ordered Size: Large (18,526 orders)
- ✓ Top 5 Pizza Types by Orders:

*Classic Deluxe, BBQ Chicken, Hawaiian, Pepperoni, Thai Chicken*

```
-- 1. Retrieve the total number of orders placed.  
Select * from orders;
```

```
Select count(order_id) as total_orders from orders;  
-- Answer: 21350
```

```
-- 2. Calculate the total revenue generated from pizza sales.  
Select * from order_details; -- for quantity
```

```
Select * from pizzas; -- for pricing
```

```
Select  
round(sum (order_details.quantity * pizzas.price),2) as total_sales -- round off 2 dec places  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id  
-- Answer: 817860.05
```

```
-- 3. Identify the highest-priced pizza.
```

```
Select * from pizzas;
```

```
Select pizza_types.name, pizzas.price  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by pizzas.price desc;
```

```
-- Answer: The Greek Pizza 35.9500007629395
```

```
-- 4. Identify the most common pizza size ordered.
```

```
Select * from order_details;
```

```
Select pizzas.size, count(order_details.order_details_id) as order_count  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size  
order by order_count desc;
```

```
-- Answer: Large with 18526 order_count
```

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

```
Select * from order_details;
```

```
Select pizza_types.name, count(order_details.quantity) as 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.name  
order by quantity desc;
```

```
-- Answer: The Classic Deluxe Pizza with 2416 orders
```

```
-- The Barbecue Chicken Pizza 2372 orders
```

```
-- The Hawaiian Pizza 2370 orders
```

```
-- The Pepperoni Pizza 2369 orders
```

```
-- The Thai Chicken Pizza 2315 orders
```

These results help understand overall demand and high-level sales behavior.

# OPERATIONAL ANALYSIS

Order volume peaks between 6-8 PM. Classic and Veggie categories perform strongest.



```
-- 1. Join the necessary tables to find the total quantity of each pizza category.  
Select * from pizza_types;
```

```
Select pizza_types.category,  
count(order_details.quantity) as 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 quantity desc;  
-- Answer: Classic 14579 orders  
-- Supreme 11777 orders  
-- Veggie 11449 orders  
-- Chicken 10815 orders
```

```
-- 2. Determine the distribution of orders by hour of the day.  
Select * from orders;
```

```
SELECT DATEPART(HOUR, orders.time) AS order_hour, COUNT(order_id) AS order_count  
FROM orders  
GROUP BY DATEPART(HOUR, orders.time)  
ORDER BY order_count desc;  
-- Answer: 12 2520  
-- 13 2455  
-- 18 2399
```

```
-- 3. Join relevant tables to find the category-wise distribution of pizzas.  
Select * from pizza_types;
```

```
Select category, count(name) as items from pizza_types  
group by category;  
-- Answer: Chicken 6  
-- Classic 8  
-- Supreme 9  
-- Veggie 9
```

```
Select * from orders;
```

```
SELECT TOP 3  
pizza_types.name,  
SUM(order_details.quantity * pizzas.price) AS revenue  
FROM pizza_types  
JOIN pizzas  
ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
JOIN order_details  
ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC;  
-- Answer: The Thai Chicken Pizza 43434.25  
-- The Barbecue Chicken Pizza 42768  
-- The California Chicken Pizza 41409.5
```

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

```
Select round(avg(quantity),0) as average_order_per_day from -- making subquery  
(Select orders.date, count(order_details.quantity) as quantity  
from orders join order_details  
on orders.order_id = order_details.order_id  
group by orders.date) as order_quantity;  
-- Answer: 135
```

## KPIs Analyzed

- ✓ Total Quantity Sold by Pizza Category
- ✓ Peak Order Hours
- ✓ Category-Wise Pizza Variety
- ✓ Average Number of Pizzas Ordered per Day
- ✓ Top 3 Pizza Types by Revenue



## KPIs Analyzed

- ✓ Contribution % to Total Revenue by Pizza Category
- ✓ Cumulative Revenue Over Time
- ✓ Top 3 Pizza Types by Revenue for Each Category

These insights help identify high-value products that drive profitability.

# STRATEGIC FINDINGS

```
-- 1. Calculate the percentage contribution of each pizza type to total revenue.  
Select * from pizzas;
```

```
SELECT  
    pizza_types.category,  
    (round (SUM(order_details.quantity * pizzas.price) /  
        (SELECT ROUND(SUM(order_details.quantity * pizzas.price), 2)  
        FROM order_details  
        JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id  
        ) * 100,2)  
    ) AS revenue_percentage  
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 revenue_percentage DESC;  
-- Answer: Classic 26.91  
-- Supreme 25.46  
-- Chicken 23.96  
-- Veggie 23.68
```

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

```
Select * from orders;  
  
SELECT  
    sales.date,  
    SUM(revenue) OVER (ORDER BY sales.date) AS cum_revenue  
FROM (SELECT  
    orders.date,  
    SUM(order_details.quantity * pizzas.price) AS revenue  
    FROM order_details  
    JOIN pizzas  
        ON order_details.pizza_id = pizzas.pizza_id  
    JOIN orders  
        ON orders.order_id = order_details.order_id  
    GROUP BY orders.date  
) AS sales;  
-- Answer: 2015-01-01 2713.85000228882  
-- 2015-01-02 5445.7500038147  
-- 2015-01-03 8108.15000724792
```

```
-- 3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
Select * from pizza_types;
```

```
SELECT name, revenue  
FROM (SELECT  
    category,  
    name,  
    revenue,  
    RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn  
    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 order_details.pizza_id = pizzas.pizza_id  
        GROUP BY pizza_types.category, pizza_types.name  
) AS a  
) AS b  
WHERE rn <= 3;  
-- Answer: 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 30940.5  
-- The Four Cheese Pizza 32265.7010040283  
-- The Mexicana Pizza 26780.75  
-- The Five Cheese Pizza 26066.5
```

# BUSINESS IMPACT

- Identified **peak ordering times** for staffing and inventory planning.
- Highlighted **high-revenue pizza types and categories** for targeted promotions.
- Revealed **most popular pizza sizes and flavors**, helping optimize menu design.
- Showed **daily revenue trends**, supporting forecasting and sales strategy.

## WHAT I LEARNED

01

Writing optimized SQL queries to answer real business questions.

03

Using joins, aggregations, and window functions for advanced analysis.

05

Strengthening data storytelling skills to communicate results effectively.

02

Translating raw data into actionable business insights.

04

Visualizing and summarizing findings for reports and presentations.

06

Helps optimize menu



**THANKS  
FOR YOUR  
ATTENTION**

