

SQL PROJECT ON PIZZA SALES



- WHERE EVERY SLICE TELLS A STORY



PROJECT OVERVIEW

About

Contact

Goal: Analyze pizza shop sales data to find trends, top sellers, and revenue patterns

- **Dataset:**
- **Tables:** orders, order_details, pizzas, pizza_types
- **Time Period:** Jan 2015
- **Tools:** SQL (MySQL)

ANALYSIS ROADMAP

[Home](#)[About](#)[Contact](#)

1. Sales & Revenue Overview

- 📦 Total number of orders placed
- 💰 Total revenue generated
- 🏆 Highest-priced pizza
- 📏 Most common pizza size ordered

2. Product Popularity Insights

- ⬆️_{TOP} Top 5 most ordered pizza types by quantity
- 💰 Top 3 pizza types by revenue
- 🏅 Top 3 pizza types by revenue within each category

ANALYSIS ROADMAP

[Home](#)[About](#)[Contact](#)

3. Category & Size Trends

- Total quantity sold per pizza category
- Category-wise distribution of pizzas

4. Time-Based Trends

- Order distribution by hour of the day
- Average number of pizzas ordered per day
- Cumulative revenue over time

5. Revenue Contribution

- Percentage contribution of each pizza type to total revenue



DATABASE SCHEMA

[Home](#)[About](#)[Contact](#)

1. orders

- **order_id (PK)**
- **order_date**
- **order_time**

2. order_details

- **order_details_id (PK)**
- **order_id (FK → orders)**
- **pizza_id (FK → pizzas)**
- **quantity**

3. pizzas

- **pizza_id (PK)**
- **pizza_type_id (FK → pizza_types)**
- **size**
- **price**

4. pizza_types

- **pizza_type_id (PK)**
- **name**
- **category**
- **ingredients**



SQL APPROACH

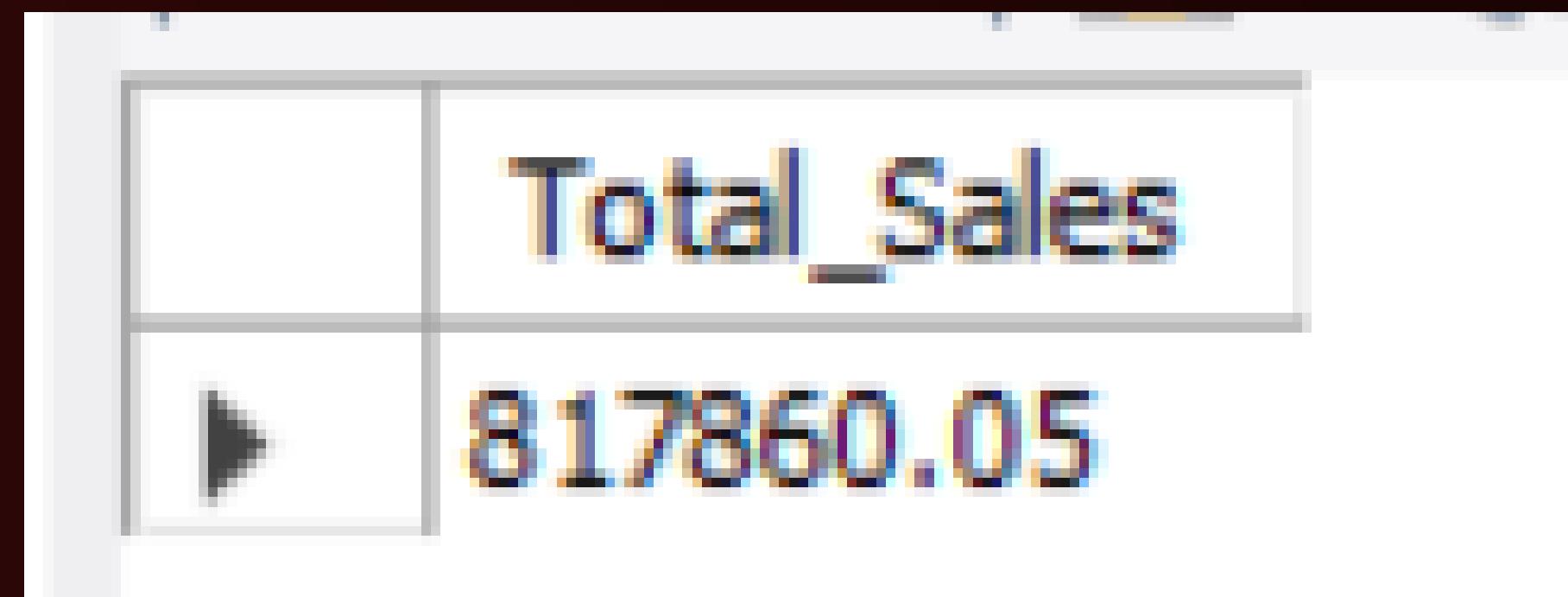
[Home](#)[About](#)[Contact](#)

```
-- Retrieve the total number of orders placed.  
select count(order_id) as Total_Orders from orders;
```

| Total_Orders |
|--------------|
| 21350 |



```
-- Calculate the total revenue generated from pizza sales.  
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS Total_Sales  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id  
;
```

[About](#)[Contact](#)



```
-- Identify the highest-priced pizza.  
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  
LIMIT 1  
;
```

[About](#)[Contact](#)

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |



```
-- Identify the most common pizza size ordered.  
select pizzas.size,count(orders_details.order_details_id) as Order_Count  
from pizzas join orders_details  
on orders_details.pizza_id=pizzas.pizza_id  
group by pizzas.size order by Order_Count desc limit 1
```

[About](#)[Contact](#)

| | size | Order_Count |
|--|------|-------------|
| | L | 18526 |



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

SELECT

pizza_types.name,
SUM(orders_details.quantity) AS Pizza_types_Quantity

FROM

pizza_types

JOIN

pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN

orders_details ON orders_details.pizza_id = pizzas.pizza_id

GROUP BY pizza_types.name

ORDER BY Pizza_types_Quantity DESC

LIMIT 5;

| name | Pizza_types_Quantity |
|----------------------------|-----------------------------|
| The Classic Deluxe Pizza | 2453 |
| The Barbeque Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |
| The Thai Chicken Pizza | 2371 |

About

Contact



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

```
Home About Contact
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS pizza_category_quanitity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY pizza_category_quanitity DESC;
```

| category | pizza_category_quanitity |
|----------|--------------------------|
| Classic | 1488 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

[Home](#)[About](#)[Contact](#)

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

SELECT

COUNT(order_id) AS Order_count, HOUR(order_time) AS Hour

FROM

orders

GROUP BY HOUR(order_time)

| Order_count | Hour |
|-------------|------|
| 1231 | 11 |
| 2520 | 12 |
| 2455 | 13 |
| 1472 | 14 |
| 1468 | 15 |
| 1920 | 16 |
| 2336 | 17 |
| 2399 | 18 |
| 2009 | 19 |
| 1642 | 20 |
| 1198 | 21 |
| 663 | 22 |
| 28 | 23 |
| 8 | 10 |
| 1 | 9 |

[Home](#)[About](#)[Contact](#)

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

SELECT

COUNT(name), category

FROM

pizza_types

GROUP BY category

| COUNT(name) | category |
|--------------------|-----------------|
| 6 | Chicken |
| 8 | Classic |
| 9 | Supreme |
| 9 | Veggie |



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

[Home](#) [About](#) [Contact](#)

```
SELECT
    ROUND(AVG(quantity), 0) AS Average_pizzas_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS total_pizzas_per_day
```

| | Average_pizzas_per_day |
|---|------------------------|
| ▶ | 138 |



```
-- Determine the top 3 most ordered pizza types based on revenue.  
select pizza_types.name,  
sum(pizzas.price*orders_details.quantity) as revenue  
from pizza_types join pizzas  
on pizza_types.pizza_type_id=pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id=pizzas.pizza_id  
group by pizza_types.name  
order by revenue desc limit 3
```

[Home](#)[About](#)[Contact](#)

| 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
    category,
    revenue,
    ROUND((revenue * 100.0) / total_revenue, 2) AS revenue_percentage
FROM (
    SELECT
        pizza_types.category,
        SUM(pizzas.price * orders_details.quantity) AS revenue,
        (SELECT SUM(pizzas.price * orders_details.quantity)
         FROM pizza_types
         JOIN pizzas
             ON pizza_types.pizza_type_id = pizzas.pizza_type_id
         JOIN orders_details
             ON orders_details.pizza_id = pizzas.pizza_id
        ) AS total_revenue
        FROM pizza_types
        JOIN pizzas
            ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN orders_details
            ON orders_details.pizza_id = pizzas.pizza_id
        GROUP BY pizza_types.category
    ) AS t
ORDER BY revenue_percentage DESC;
```

Home

About

Contact

| category | revenue | revenue_percentage |
|----------|---------------------|--------------------|
| Classic | 220053.1000000001 | 26.91 |
| Supreme | 208196.99999999822 | 25.46 |
| Chicken | 195919.5 | 23.96 |
| Veggie | 193690.450000000298 | 23.68 |



-- Analyze the cumulative revenue generated over time.

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

[Home](#)

[About](#)

[Contact](#)

| | order_date | cumulative_revenue |
|---|------------|--------------------|
| ▶ | 2015-01-01 | 2713.8500000000004 |
| | 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 | 23990.35000000002 |

[About](#)[Contact](#)

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
SELECT name, category, revenue  
FROM (  
    SELECT  
        pt.name,  
        pt.category,  
        SUM(od.quantity * p.price) AS revenue,  
        ROW_NUMBER() OVER (  
            PARTITION BY pt.category  
            ORDER BY SUM(od.quantity * p.price) DESC  
        ) AS rn  
    FROM pizza_types pt  
    JOIN pizzas p  
        ON pt.pizza_type_id = p.pizza_type_id  
    JOIN orders_details od  
        ON od.pizza_id = p.pizza_id  
    GROUP BY pt.name, pt.category  
) AS ranked  
WHERE rn <= 3  
ORDER BY category, revenue DESC;
```

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



KEY INSIGHTS

Home

About

Contact

- Top 3 pizzas by revenue: Thai Chicken Pizza, Barbecue Chicken Pizza, California Pizza
- Peak sales day: Friday evenings
- Most popular category: Classic
- Total Sales: 817860.05