

PIZZA SALES ANALYSIS

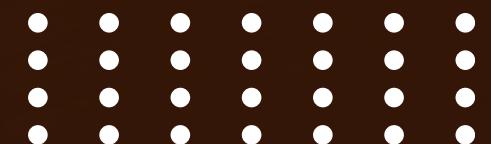


ABOUT THIS PROJECT



Our Passion for Pizza

- My Pizza Sales Analysis Project focuses on solving business problems by analyzing sales data using SQL.
- I wrote a series of SQL queries, ranging from basic to advanced, to extract actionable insights from the dataset.
- Key areas of analysis included identifying best-selling pizzas, understanding peak sales hours, and uncovering customer preferences.
- The project provided data-driven solutions for optimizing inventory management and pricing strategies.
- SQL queries were designed to highlight performance metrics, such as sales trends and underperforming menu items.
- This project demonstrates the power of SQL in addressing real-world business challenges effectively.



SQL QUERY



THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(*) from orders;
```

| | Total_orders_placed |
|---|---------------------|
| ▶ | 21350 |

SQL QUERY



THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(od.quantity * p.price), 0) AS Total_revenue  
FROM  
    order_details AS od  
    INNER JOIN  
    pizzas AS p ON od.pizza_id = p.pizza_id;
```

| | Total_revenue |
|---|---------------|
| → | 817860 |

SQL QUERY



THE HIGHEST-PRICED PIZZA.

```
SELECT
    pt.name, p.price
FROM
    pizza_types AS pt
        INNER JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY 2 DESC
LIMIT 1;
```

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

SQL QUERY

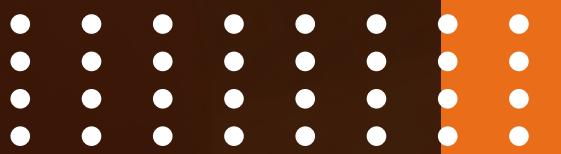


THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    size, COUNT(*) as pizzas  
FROM  
    pizzas  
GROUP BY size  
ORDER BY 2 DESC  
LIMIT 1;
```

| | size | pizzas |
|---|------|--------|
| ▼ | S | 32 |

SQL QUERY



TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT
    pizza_types.name, SUM(quantity) AS quantity
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY 2 DESC
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 |

SQL QUERY



TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT
    pizza_types.category, SUM(quantity) as quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY 1;
```

| | category | quantity |
|--|----------|----------|
| | Classic | 14888 |
| | Veggie | 11649 |
| | Supreme | 11987 |
| | Chicken | 11050 |

SQL QUERY



DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT  
    HOUR(order_time) as Order_time, COUNT(order_id) as orders  
FROM  
    orders  
GROUP BY 1  
ORDER BY 1;
```

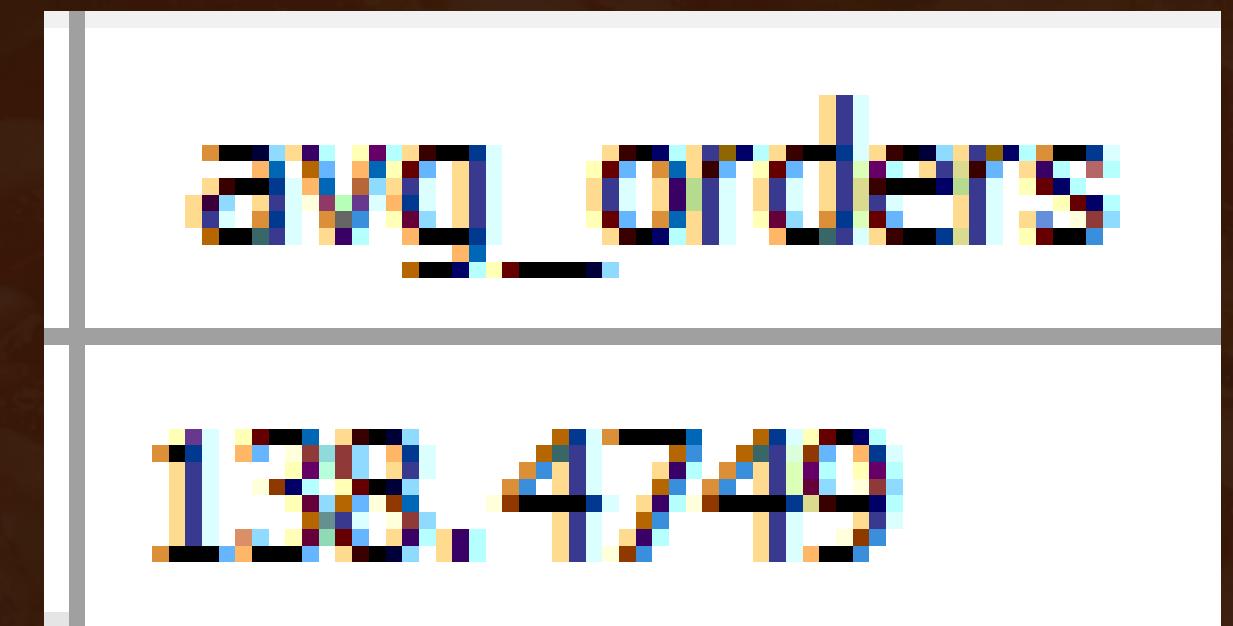
| Order_time | orders |
|------------|--------|
| 9 | 1 |
| 10 | 8 |
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |
| 18 | 2399 |
| 19 | 2009 |

SQL QUERY



AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```
SELECT  
    AVG(quantity)  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON order_details.order_id = orders.order_id  
    GROUP BY 1) AS quantity;
```



SQL QUERY



TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
SELECT
    pizza_types.name, SUM(quantity * price) AS revenue
FROM
    pizzas
        JOIN
    pizza_types 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 2 DESC
LIMIT 5;
```

| name | revenue |
|------------------------------|----------|
| The Thai Chicken Pizza | 43434.25 |
| The Barbecue Chicken Pizza | 42768 |
| The California Chicken Pizza | 41409.5 |

SQL QUERY



PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
SELECT
    pizza_types.name, round(((round(SUM(quantity * price),2))/817860)*100,2) AS revenue_percentage
FROM
    pizzas
        JOIN
    pizza_types 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 2 DESC;
```

| name | revenue_percentage |
|------------------------------|--------------------|
| The Thai Chicken Pizza | 5.31 |
| The Barbecue Chicken Pizza | 5.23 |
| The California Chicken Pizza | 5.06 |
| The Classic Deluxe Pizza | 4.67 |
| The Spicy Italian Pizza | 4.26 |
| The Southwest Chicken Pizza | 4.24 |
| The Italian Supreme Pizza | 4.09 |
| The Hawaiian Pizza | 3.95 |
| The Four Cheese Pizza | 3.95 |
| The Sicilian Pizza | 3.78 |
| The Pepperoni Pizza | 3.69 |
| The Greek Pizza | 3.48 |
| The Mexicana Pizza | 3.27 |

SQL QUERY



THE CUMULATIVE REVENUE GENERATED OVER TIME

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

| order_date | cum_revenue |
|------------|-------------------|
| 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 |
| 2015-01-10 | 23990.35000000002 |
| 2015-01-11 | 25862.65 |
| 2015-01-12 | 27781.7 |
| 2015-01-13 | 29831.30000000003 |

**THANK YOU
FOR ATTENTION**