

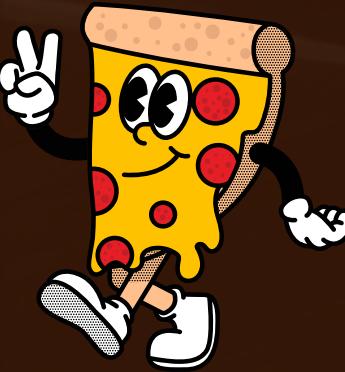


SQL Project  
PIZZA SALES INSIGHTS

AN SQL ANALYSIS OF

# PIZZA SALES

PROJECT



# KUNAL KUMAR

## DATA ANALYST ENTHUSIAST

### ABOUT ME

I am an aspiring data analyst with a strong foundation in data management, analysis, and visualization. Proficient in Excel, Power BI, SQL, Python, and statistics, I bring a well-rounded skill set for transforming complex data into actionable insights. My expertise in these tools and techniques allows me to efficiently clean, analyze, and visualize data to support data-driven decision-making.

### SKILLS

**Programming Languages:** Python, SQL  
**Packages:** NumPy, Pandas, Matplotlib  
**Databases:** MySQL, PostgreSQL,  
**Tools:** Microsoft Power BI, Excel, Tableau  
**Other Tools:** VS Code, Pycharm, Git, Jupyter  
**Hard Skills:** Data Modelling, Data Visualization, Statistical Data Analysis, Data Analysis

# Retrieve the total number of orders placed

## CODE

```
SELECT  
    COUNT(order_id) AS total_numbers_orders  
FROM  
    orders;
```

## RESULT

|   | total_numbers_orders |
|---|----------------------|
| ▶ | 21350                |

# Calculate the total revenue generated from pizza sales

## CODE

```
SELECT  
    ROUND(SUM(orders_deatils.quantity * pizzas.price),  
        2) AS total_revenue  
  
FROM  
    orders_deatils  
    JOIN  
    pizzas ON orders_deatils.pizza id = pizzas.pizza id;
```

## RESULT

|   |               |
|---|---------------|
|   | total_revenue |
| ▶ | 807206.6      |

# Identify the highest-priced pizza

## CODE

```
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;
```

## RESULT

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

# Identify the most common pizza size ordered

## CODE

```
SELECT
    pizzas.size,
    COUNT(orders_deatils.order_details_id) AS order_count
FROM
    pizzas
    JOIN
        orders_deatils ON pizzas.pizza_id = orders_deatils.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

## RESULT

|   | size | order_count |
|---|------|-------------|
| ▶ | L    | 18282       |

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

## CODE

```
SELECT
    pizza_types.name, SUM(orders_deatils.quantity) AS quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_deatils ON pizzas.pizza_id = orders_deatils.pizza_id
GROUP BY pizza_types.name
ORDER BY quantities DESC
LIMIT 5;
```

## RESULT

|   | name                       | quantities |
|---|----------------------------|------------|
| ▶ | The Classic Deluxe Pizza   | 2412       |
|   | The Barbecue Chicken Pizza | 2400       |
|   | The Pepperoni Pizza        | 2390       |
|   | The Hawaiian Pizza         | 2389       |
|   | The Thai Chicken Pizza     | 2338       |

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

## CODE

```
SELECT
    pizza_types.category,
    SUM(orders_deatils.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_deatils ON pizzas.pizza_id = orders_deatils.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC
```

## RESULT

|   | category | quantity |
|---|----------|----------|
| ▶ | Classic  | 14680    |
|   | Supreme  | 11850    |
|   | Veggie   | 11488    |
|   | Chicken  | 10910    |

# Determine the distribution of orders by hour of the day

## CODE

```
SELECT  
    HOUR(orders.order_time) AS hours,  
    COUNT(orders.order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(orders.order_time);
```

## RESULT

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

# Join relevant tables to find the category-wise distribution of pizza

## CODE

```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

## RESULT

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

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

## CODE

```
SELECT  
    ROUND(AVG(quantity), 0) AS average_number  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_deatils.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_deatils ON orders.order_id = orders_deatils.order_id  
    GROUP BY orders.order_date) AS order_data;
```

## RESULT

|   | average_number |
|---|----------------|
| ▶ | 139            |

# Determine the top 3 most ordered pizza types based on revenue

## CODE

```
SELECT
    pizza_types.name,
    SUM(orders_deatils.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_deatils ON pizzas.pizza_id = orders_deatils.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```

## RESULT

|   | name                         | revenue |
|---|------------------------------|---------|
| ▶ | The Thai Chicken Pizza       | 42821.5 |
|   | The Barbecue Chicken Pizza   | 42204   |
|   | The California Chicken Pizza | 40861.5 |

# Calculate the percentage contribution of each pizza type to total revenue

## CODE

```
with category_revenue as (
    select pizza_types.category , round(sum(orders_deatils.quantity* pizzas.price),2) as revenue
    from pizza_types
    join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
    join orders_deatils on pizzas.pizza_id = orders_deatils.pizza_id
    group by pizza_types.category
)
select category, revenue, round((revenue/ (select sum(revenue) from category_revenue) *100),2) as percent_contribution
from category_revenue
order by revenue desc;
```

## RESULT

|    | category | revenue   | percent_contribution |
|----|----------|-----------|----------------------|
| 1. | Classic  | 216992.35 | 26.88                |
| 2. | Supreme  | 205770.85 | 25.49                |
| 3. | Chicken  | 193434.5  | 23.96                |
| 4. | Veggie   | 191008.9  | 23.66                |

# Analyze the cumulative revenue generated over time

## CODE

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

## RESULT

|   | 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           |

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

## CODE

```
select name,category,revenue
from
(select name, category, revenue,
rank() over(partition by category order by revenue desc) as rank_pizza
from
(select pizza_types.name, pizza_types.category, sum(orders_deatils.quantity* pizzas.price)
from pizza_types
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_deatils on pizzas.pizza_id = orders_deatils.pizza_id
group by pizza_types.name, pizza_types.category) as a) as b
where rank_pizza<=3
```

## RESULT

|    | name                         | category | revenue    |
|----|------------------------------|----------|------------|
| 1  | The Thai Chicken Pizza       | Chicken  | 42821.5    |
| 2  | The Barbecue Chicken Pizza   | Chicken  | 42204      |
| 3  | The California Chicken Pizza | Chicken  | 40861.5    |
| 4  | The Classic Deluxe Pizza     | Classic  | 37536.5    |
| 5  | The Hawaiian Pizza           | Classic  | 31841      |
| 6  | The Pepperoni Pizza          | Classic  | 29811.75   |
| 7  | The Spicy Italian Pizza      | Supreme  | 34428.75   |
| 8  | The Italian Supreme Pizza    | Supreme  | 33062.5    |
| 9  | The Sicilian Pizza           | Supreme  | 30612      |
| 10 | The Four Cheese Pizza        | Veggie   | 31656.6500 |
| 11 | The Mexicana Pizza           | Veggie   | 26466      |
| 12 | The Five Cheese Pizza        | Veggie   | 25770.5    |

# THANK YOU



[www.linkedin.com/in/kunal-kumar-python/](https://www.linkedin.com/in/kunal-kumar-python/)



kunalkr696@gmail.com



+91 7491063850