

# Pizza Hut Sales Data Analysis

SQL

# AIM

To analyze pizza sales data,  
focusing on revenue, order trends  
and make business decisions.

# 1. Retrieve the total number of orders placed.

- `SELECT COUNT(order_id) AS Total_Orders FROM orders;`

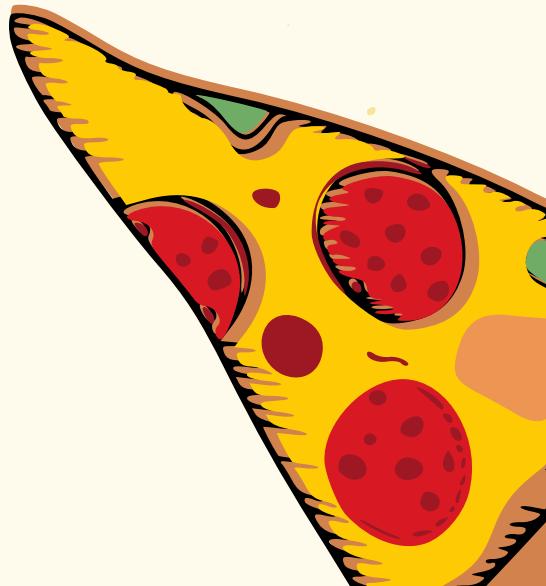
| Result Grid |              |
|-------------|--------------|
|             | Total_Orders |
| ▶           | 21350        |



## 2. Calculate the total revenue generated from pizza sales.

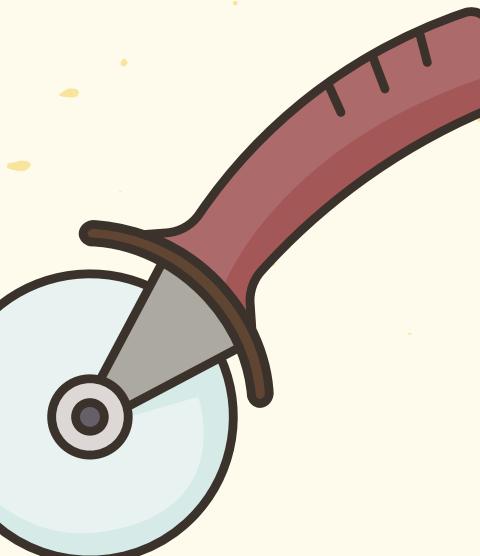
- ```
SELECT ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_Sale
FROM order_details
JOIN pizzas
ON pizzas.pizza_id = order_details.pizza_id;
```

| Result Grid |            |
|-------------|------------|
|             | Total_Sale |
| ▶           | 817860.05  |



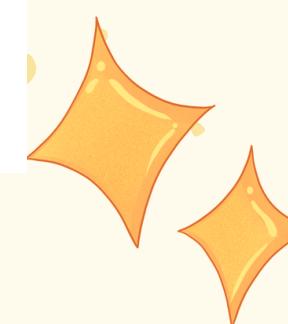
# 3. Identify the highest priced pizza.

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



Result Grid | Filter Rows

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



## 4. Identify the most common pizza size ordered.

- ```
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 LIMIT 1;
```

Result Grid | Filter Rows:

|   | size | Order_COUNT |
|---|------|-------------|
| ▶ | L    | 18526       |

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

- ```
SELECT pizza_types.name, SUM(order_details.quantity)
AS Count
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 count
DESC LIMIT 5;
```



Result Grid | Filter Rows:

|   | name                       | Count |
|---|----------------------------|-------|
| ▶ | The Classic Deluxe Pizza   | 2453  |
|   | The Barbecue Chicken Pizza | 2432  |
|   | The Hawaiian Pizza         | 2422  |
|   | The Pepperoni Pizza        | 2418  |
|   | The Thai Chicken Pizza     | 2371  |

# 6. 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;
```

Result Grid | Filter Rows:

|   | category | Total_Quantity |
|---|----------|----------------|
| ▶ | Classic  | 14888          |
|   | Supreme  | 11987          |
|   | Veggie   | 11649          |
|   | Chicken  | 11050          |



# 7. Determine the distribution of orders by hour of the day.

- ```
SELECT hour(order_time) AS Hours,  
       count(order_id) AS No_of_orders  
  FROM orders  
 GROUP BY hours;
```

Result Grid | Filter Rows

|   | Hours | No_of_orders |
|---|-------|--------------|
| ▶ | 11    | 1231         |
|   | 12    | 2520         |
|   | 13    | 2455         |
|   | 14    | 1472         |
|   | 15    | 1468         |
|   | 16    | 1920         |
|   | 17    | 2336         |
|   | 18    | 2399         |

## 8. Find the category wise distribution of pizzas.

- ```
SELECT pizza_types.category AS Category,
       count(pizza_types.name) AS Number_of_pizzas
  FROM pizza_types
 GROUP BY Category;
```

|   | Category | Number_of_pizzas |
|---|----------|------------------|
| ▶ | Chicken  | 6                |
|   | Classic  | 8                |
|   | Supreme  | 9                |
|   | Veggie   | 9                |

# 9. Calculate the average number of pizzas ordered per day.

- ```
SELECT round(avg(Total_orders),0) AS Avg_orders_per_day
FROM
(SELECT orders.order_date, sum(order_details.quantity) AS Total_orders
FROM order_details
JOIN orders
ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS Orders_qunatity;
```

The screenshot shows a database query results grid with one row and two columns. The first column is labeled 'Avg\_orders\_per\_day' and contains the value '138'. The grid has a light gray background with a white header row. There are navigation arrows at the bottom left of the grid.

| Avg_orders_per_day |
|--------------------|
| 138                |

## 9. Calculate the average number of pizzas ordered per day.

- ```
SELECT round(avg(Total_orders),0) AS Avg_orders_per_day
FROM
(SELECT orders.order_date, sum(order_details.quantity)
AS Total_orders
FROM order_details
JOIN orders
ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS Orders_qunatity;
```

| Result Grid        | Filter |
|--------------------|--------|
| Avg_orders_per_day |        |
| 138                |        |

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

- ```
SELECT 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.name
 ORDER BY Revenue DESC LIMIT 3;
```

Result Grid | Filter Rows:

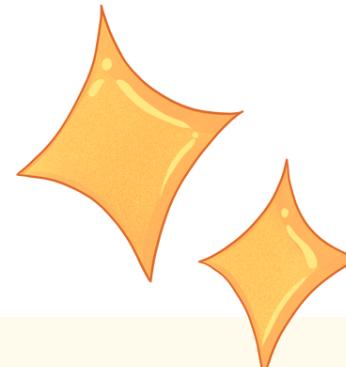
|   | name                         | Revenue  |
|---|------------------------------|----------|
| ▶ | The Thai Chicken Pizza       | 43434.25 |
| ▶ | The Barbecue Chicken Pizza   | 42768    |
|   | The California Chicken Pizza | 41409.5  |

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

```
• SELECT pizza_types.category,  
      ROUND(SUM(order_details.quantity * pizzas.price)/ (SELECT  
      ROUND(SUM(order_details.quantity * pizzas.price),2) AS Total_Sale  
      FROM order_details  
      JOIN pizzas  
      ON pizzas.pizza_id = order_details.pizza_id)*100,2) 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  
      ORDER BY Revenue DESC LIMIT 3;
```

Result Grid | Filter

| category | Revenue |
|----------|---------|
| Classic  | 26.91   |
| Supreme  | 25.46   |
| Chicken  | 23.96   |



# 12. Analyze 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 order_details.pizza_id = pizzas.pizza_id
     JOIN orders
       ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS Sales;
```

Result Grid | Filter Rows:

|   | order_date | Cum_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            |

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

| name                                       | category | revenue           |
|--------------------------------------------|----------|-------------------|
| The Chicken Pesto Pizza                    | Chicken  | 16701.75          |
| The Chicken Alfredo Pizza                  | Chicken  | 16900.25          |
| The Southwest Chicken Pizza                | Chicken  | 34705.75          |
| The Pepperoni, Mushroom, and Peppers Pizza | Classic  | 18834.5           |
| The Big Meat Pizza                         | Classic  | 22968             |
| The Napolitana Pizza                       | Classic  | 24087             |
| The Brie Carre Pizza                       | Supreme  | 11588.49999999999 |
| The Spinach Supreme Pizza                  | Supreme  | 15277.75          |
| The Calabrese Pizza                        | Supreme  | 15934.25          |
| The Green Garden Pizza                     | Veggie   | 13955.75          |
| The Mediterranean Pizza                    | Veggie   | 15360.5           |
| The Spinach Pesto Pizza                    | Veggie   | 15596             |

- ```
SELECT name,category, revenue FROM
(SELECT category,name, revenue,
rank() over(partition by category order by revenue) as rn
FROM
(SELECT pizza_types.category, 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.category, pizza_types.name) as A) as B
WHERE rn<=3;
```



# Conclusion

- **Classic pizzas generate the most revenue, followed by Supreme and Chicken.**
- **Top-selling pizza: "The Thai Chicken Pizza."**
- **Peak order time: 12–1 PM.**
- **Large pizzas are the most popular size.**
- **Focus marketing on bestsellers and adjust inventory for peak times.**

**Thank you for  
your attention**

