

# PIZZA



PIZZA SALES ANALYSIS



INGOODE COMPANY

+250

Restaurant

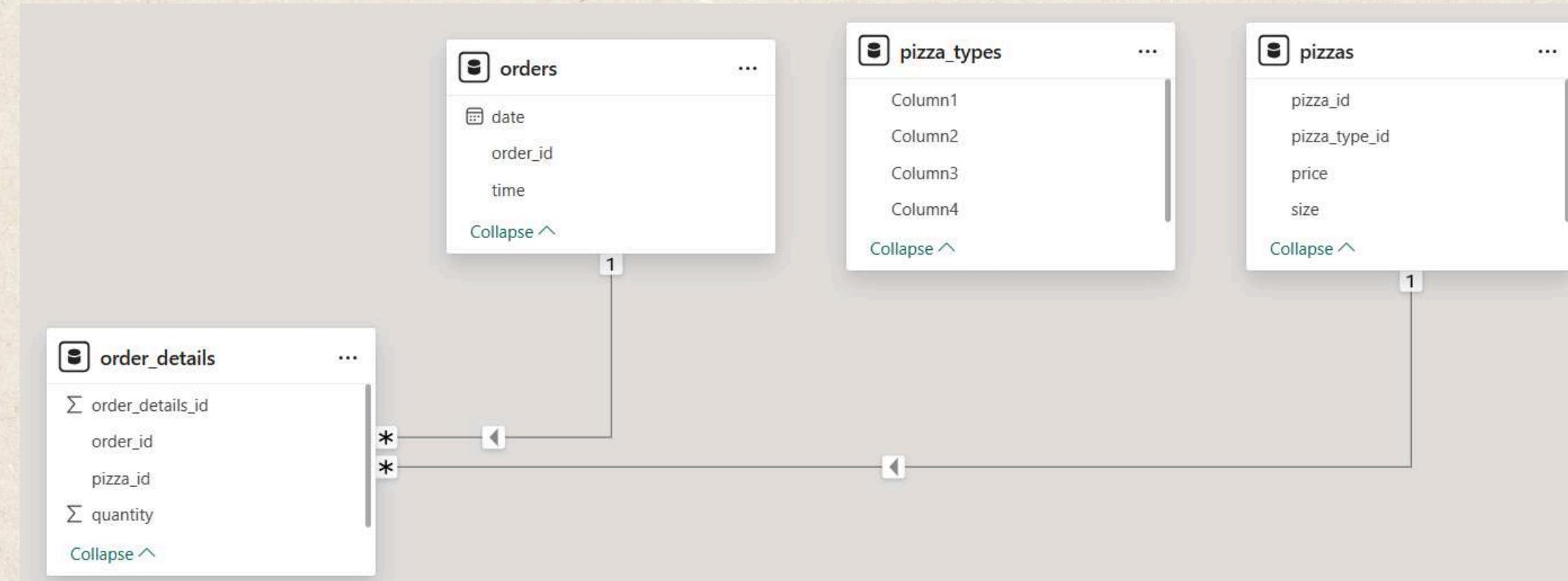
4.9

Good Rate

+100

Pizza menu

# Welcome To Pizza



-- Retrieve the total number of orders placed

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

| Result Grid |              |
|-------------|--------------|
|             | total_orders |
| ▶           | 21350        |





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

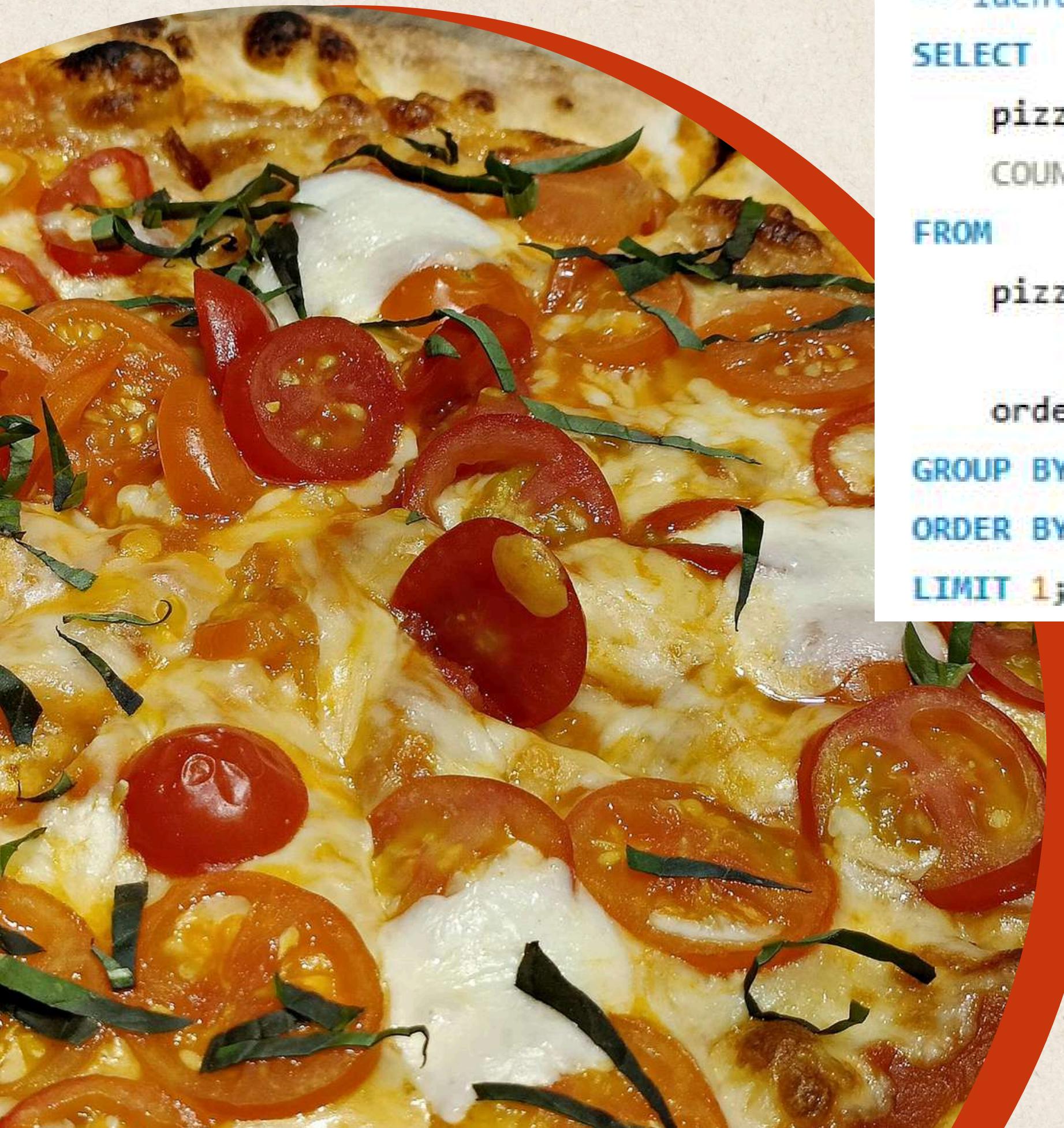
| Result Grid |             |
|-------------|-------------|
|             | total_sales |
| →           | 817860.05   |

-- 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;
```



| Result Grid |                 | Filter Rx |
|-------------|-----------------|-----------|
|             | name            | price     |
| ▶           | The Greek Pizza | 35.95     |



-- Identify the most common pizza size ordered

**SELECT**

pizzas.size,

COUNT(order\_details.order\_details\_id) **AS** total\_orders

**FROM**

pizzas

**JOIN**

order\_details **ON** pizzas.pizza\_id = order\_details.pizza\_id

**GROUP BY** pizzas.size

**ORDER BY** total\_orders **DESC**

**LIMIT** 1;

| Result Grid |

|   | size | total_orders |
|---|------|--------------|
| ▶ | L    | 18526        |

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

SELECT

```
    pizza_types.name,  
    SUM(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  
LIMIT 5;
```

| Result Grid |                            | Filter Rows: |
|-------------|----------------------------|--------------|
|             | 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         |



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

```
SELECT
```

```
    pizza_types.category,  
    SUM(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;
```

Result Grid | Filter

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

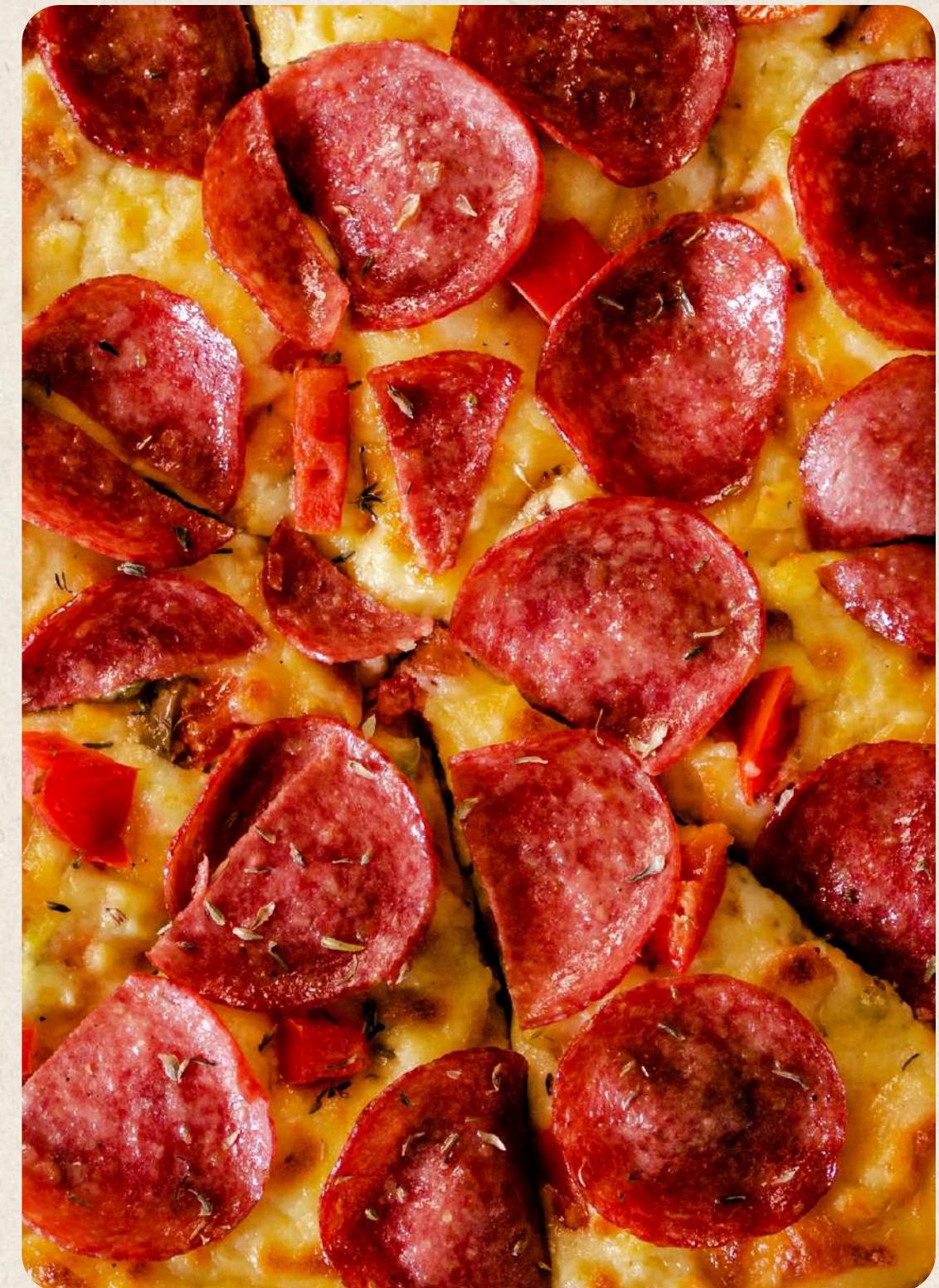
-- Determine the distribution of orders by hour of the day.

SELECT

```
HOUR(order_time) AS order_hour,  
COUNT(order_id) AS total_orders  
FROM orders  
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows

|   | order_hour | total_orders |
|---|------------|--------------|
| ▶ | 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            |



-- Find category-wise distribution of pizzas.

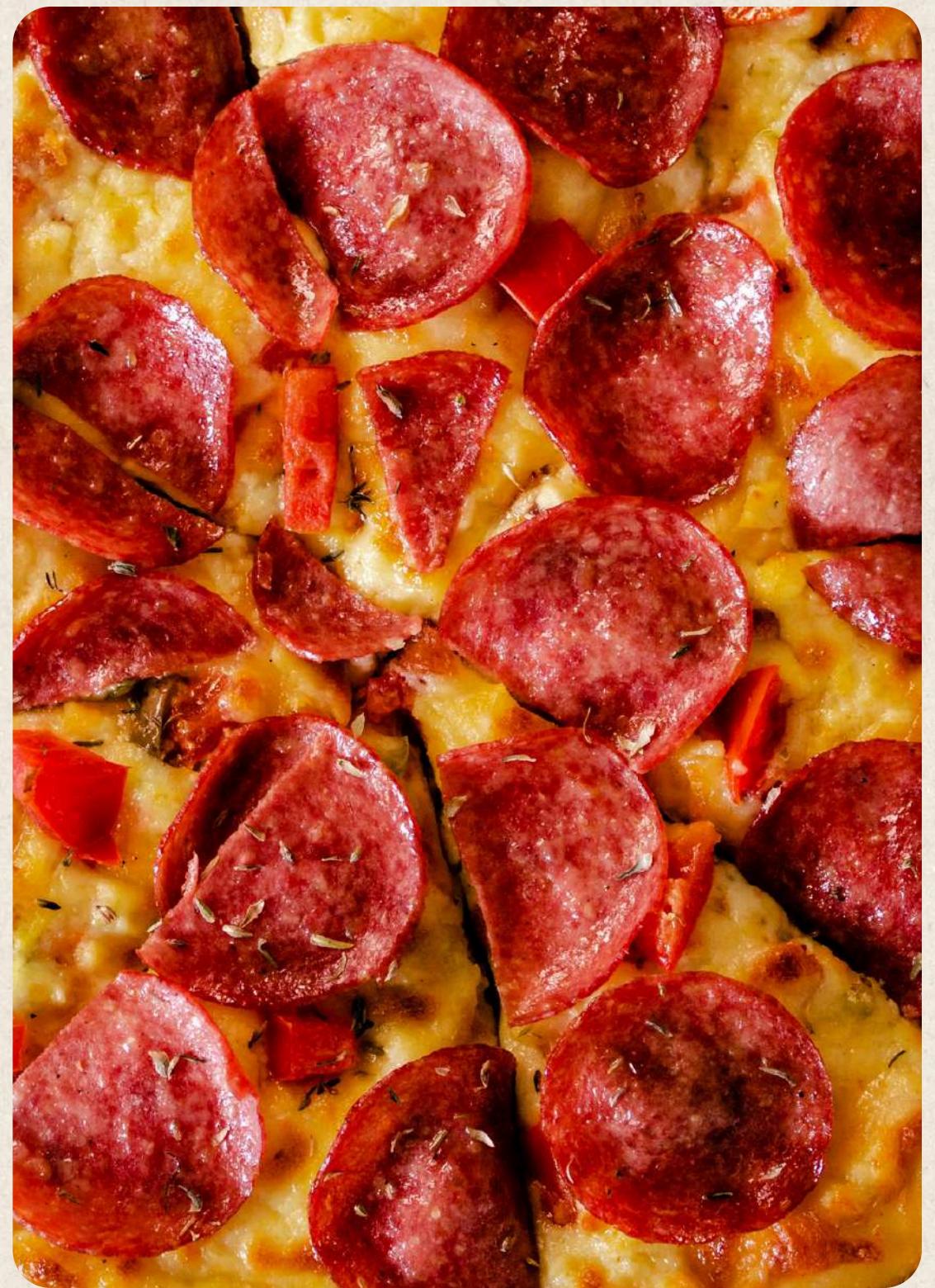
SELECT

category,  
COUNT(name) AS total\_pizzas

FROM pizza\_types

GROUP BY category;

|   | category | total_pizzas |
|---|----------|--------------|
| ▶ | Chicken  | 6            |
| ▶ | Classic  | 8            |
| ▶ | Supreme  | 9            |
| ▶ | Veggie   | 9            |



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

```
SELECT ROUND(AVG(quantity),0) AS avg_pizzas_per_day
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity) AS quantity
    FROM orders
    JOIN order_details
        ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date
) AS order_quantity;
```

| Result Grid |                    |
|-------------|--------------------|
|             | avg_pizzas_per_day |
| ▶           | 138                |



-- 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 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

**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      |



Home

About

Service

Product

```
-- pizza category to total revenue.
```

```
SELECT
```

```
    pizza_types.category,  
    SUM(order_details.quantity * pizzas.price) AS revenue,  
    ROUND(  
        SUM(order_details.quantity * pizzas.price) * 100 /  
        (SELECT SUM(order_details.quantity * pizzas.price)  
         FROM order_details  
        JOIN pizzas  
          ON order_details.pizza_id = pizzas.pizza_id),  
        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 DESC;
```

Result Grid | Filter Rows: | Export

| category | revenue            | revenue_percentage |
|----------|--------------------|--------------------|
| Classic  | 220053.1000000001  | 26.91              |
| Supreme  | 208196.99999999822 | 25.46              |
| Chicken  | 195919.5           | 23.96              |
| Veggie   | 193690.45000000298 | 23.68              |

-- 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;
```

|   | 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            |
|   | 2015-01-10 | 23990.350000000002 |
|   | 2015-01-11 | 25862.65           |
|   | 2015-01-12 | 27781.7            |

-- Top 3 pizzas by revenue within each category

```
SELECT
    category,
    name,
    revenue
FROM (
    SELECT
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue,
        ROW_NUMBER() OVER (
            PARTITION BY pizza_types.category
            ORDER BY SUM(order_details.quantity * pizzas.price) DESC
        ) AS rn
    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
        ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    GROUP BY pizza_types.category, pizza_types.name
) AS ranked_pizzas
WHERE rn <= 3
ORDER BY category, revenue DESC;
```

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

*Thank  
you*

**WORKED  
ON  
MYSQL**