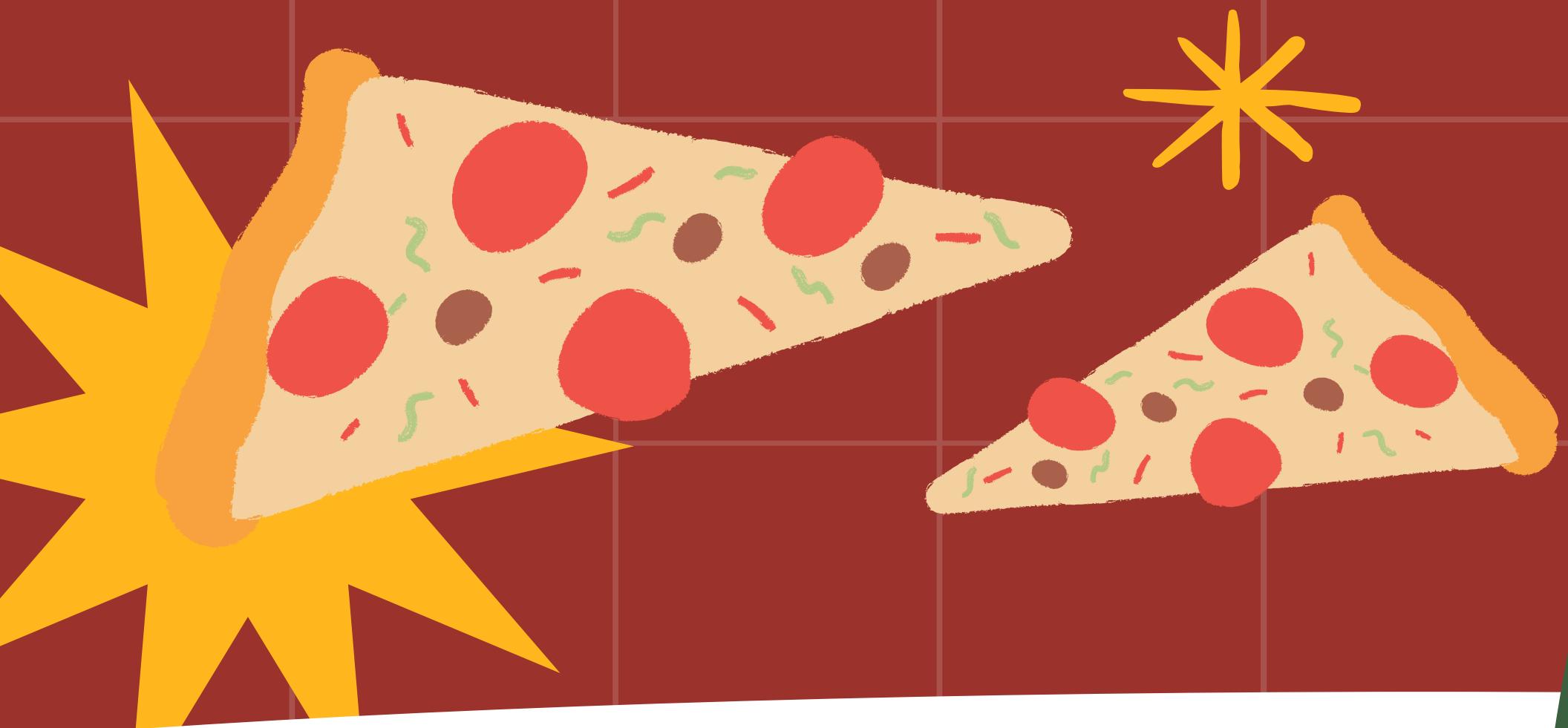


# PIZZA SALES SQL/QUERY



**Retrieve the total number  
of orders placed.**



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

# Calculate the total revenue generated from pizza sales



```
SELECT  
    SUM(order_details.quantity * pizzas.price) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

# Identify the highest-priced pizza name

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



# Identify the highest-priced pizza name

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



**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;
```





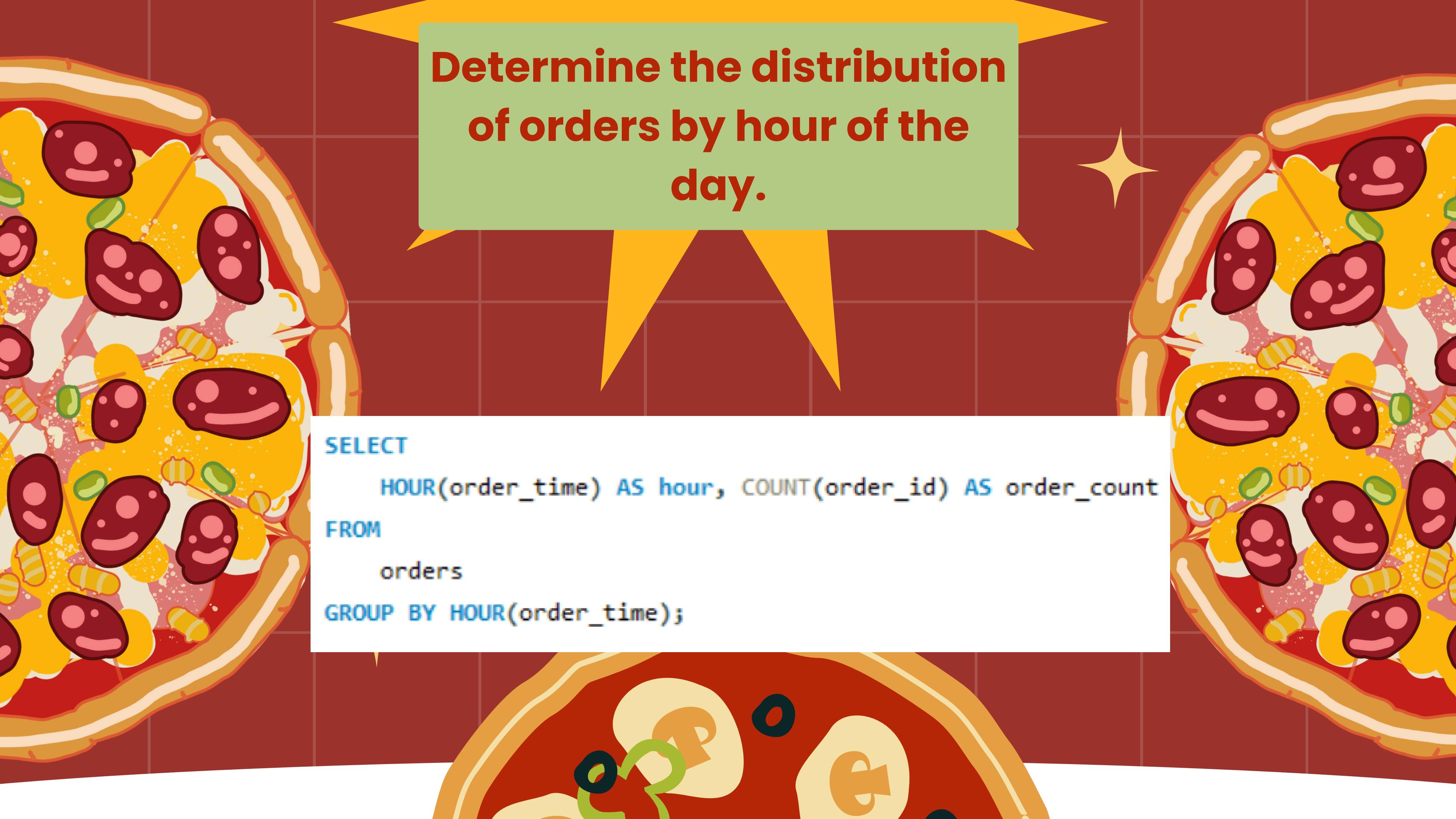
List the top 5 most ordered pizza types along with their quantities.

```
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 SUM(order_details.quantity) DESC  
LIMIT 5;
```

**Join the necessary tables  
to 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;
```



Determine the distribution  
of orders by hour of the  
day.

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

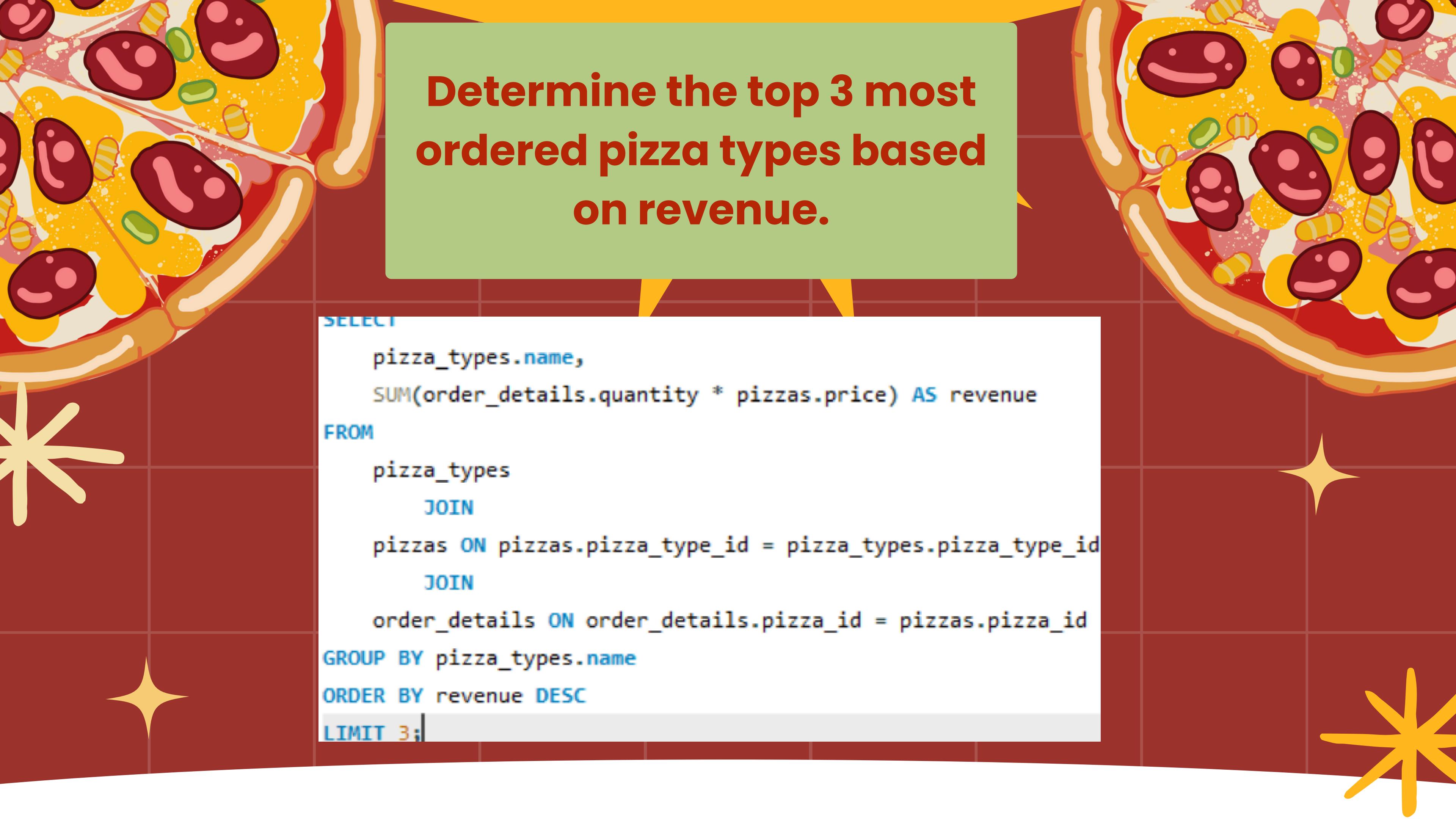
**Join relevant tables to find  
the category-wise  
distribution of pizzas.**



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

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

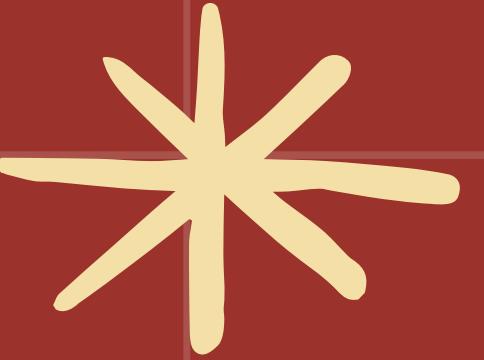
```
SELECT  
    AVG(quantity)  
FROM  
    (SELECT  
        orders.order_data, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_data) AS order_quantity;
```



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;
```

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



03



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

Analyze the cumulative revenue generated over time.



```
SELECT order_data, SUM(revenue) over(order by order_data) as cum_revenue
FROM
(SELECT orders.order_data,
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_data) as sales;
```

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

```
SELECT name, revenue FROM
(SELECT category, name, revenue,rank() over(partition by category order by revenue desc) as rn
FROM
(SELECT pizza_types.category,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.category, pizza_types.name) as a) as b
where rn <= 3;
```



**THANK YOU**