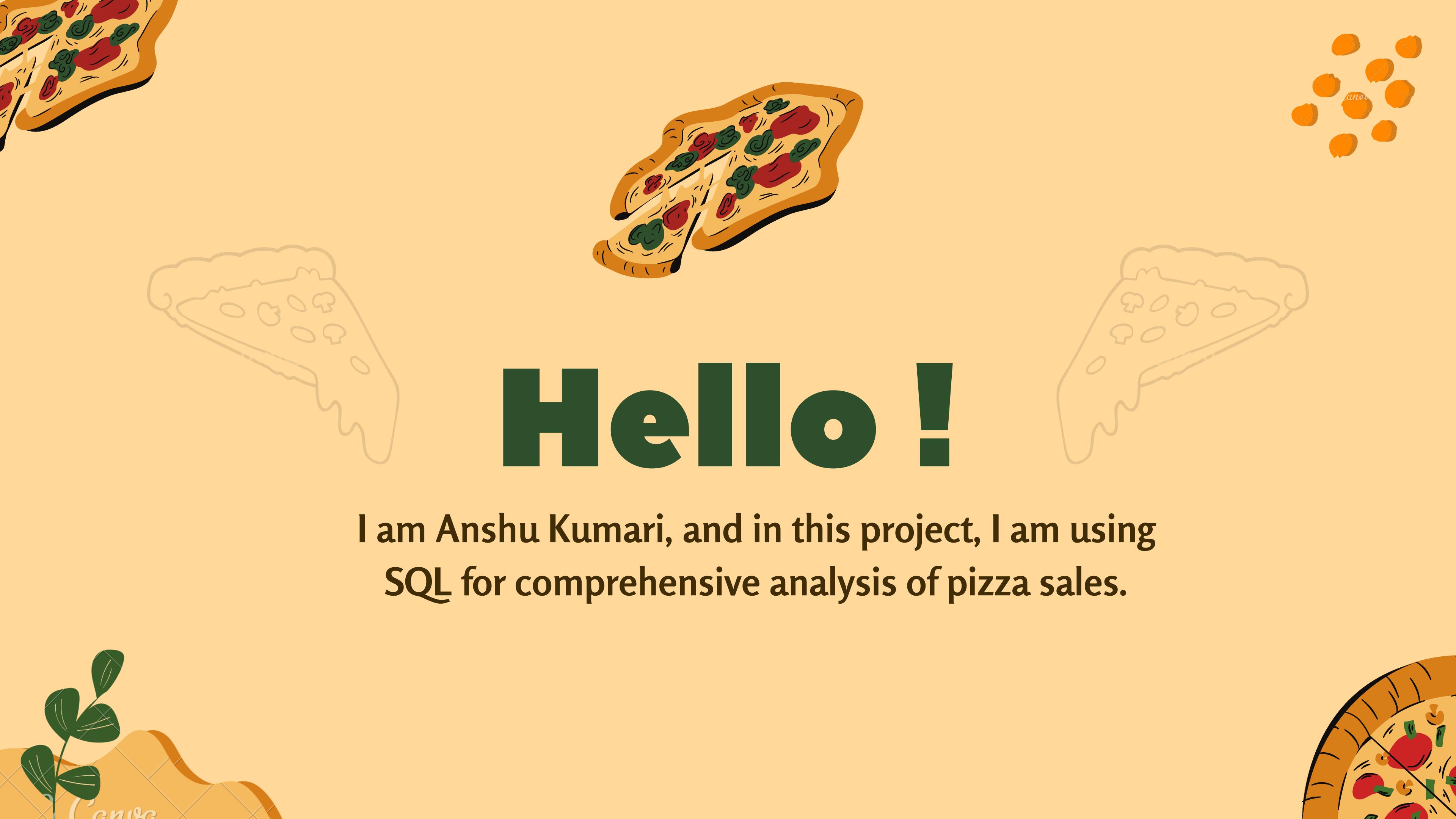


# PIZZA SALES PROJECT

@Anshu Kumari



# Hello !

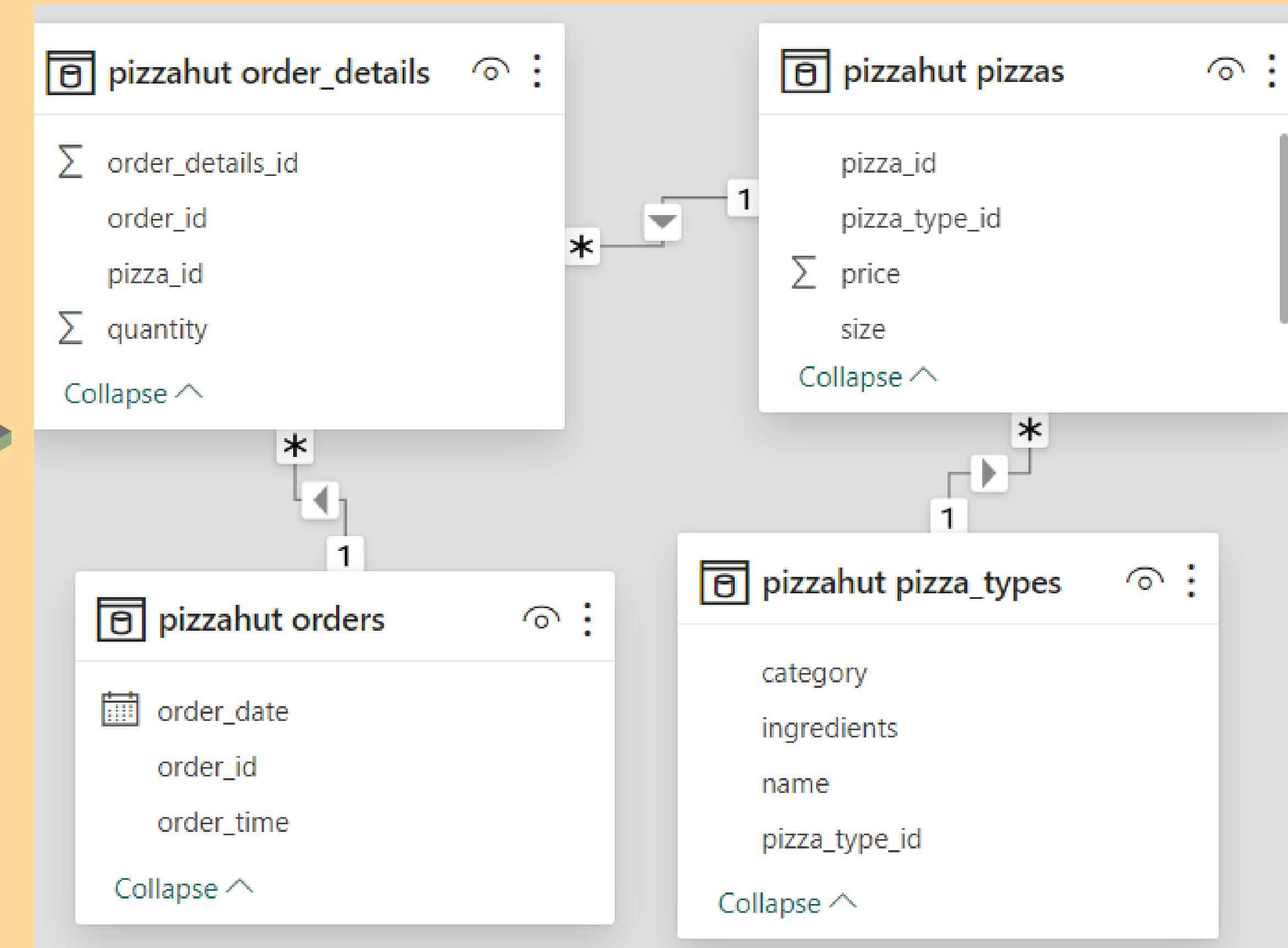
I am Anshu Kumari, and in this project, I am using  
SQL for comprehensive analysis of pizza sales.

# Objective

This project aims to analyze pizza sales data, revealing performance and trends within a restaurant. By examining the data, it seeks to uncover valuable insights for informed decision-making and strategic planning.



# Data Model Overview



# Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) Total_Order  
FROM  
    orders;
```

	Total_Order
↓	21350



# Calculate the total revenue generated from pizza sales.



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

Total_Revenue
817860.05



# Identify the highest-priced pizza.

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

	name	price
▶	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

```
SELECT p.size, COUNT(od.order_details_id) as order_count  
FROM order_details od  
JOIN pizzas p  
ON od.pizza_id = p.pizza_id  
GROUP BY p.size;
```

	size	order_count
▶	M	15385
	L	18526
	S	14137
	XL	544
	XXL	28

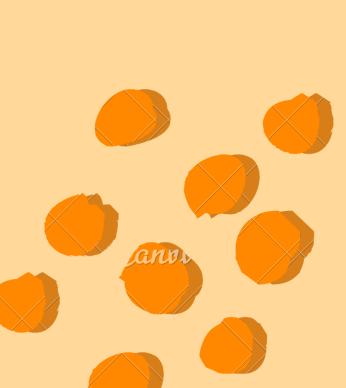


# List the top 5 most ordered pizza types along with their quantities.



```
SELECT pt.name, COUNT(od.quantity) AS quantity  
FROM order_details od  
JOIN pizzas p  
ON od.pizza_id = p.pizza_id  
JOIN pizza_types pt  
ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY pt.name ORDER BY quantity DESC LIMIT 5;
```

name	quantity
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315



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

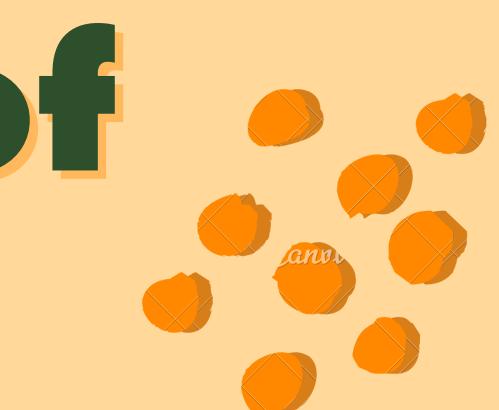
```
SELECT pt.category, SUM(od.quantity) AS quantity
FROM order_details od
JOIN pizzas p
ON od.pizza_id = p.pizza_id
JOIN pizza_types pt
ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category ORDER BY quantity DESC;
```

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

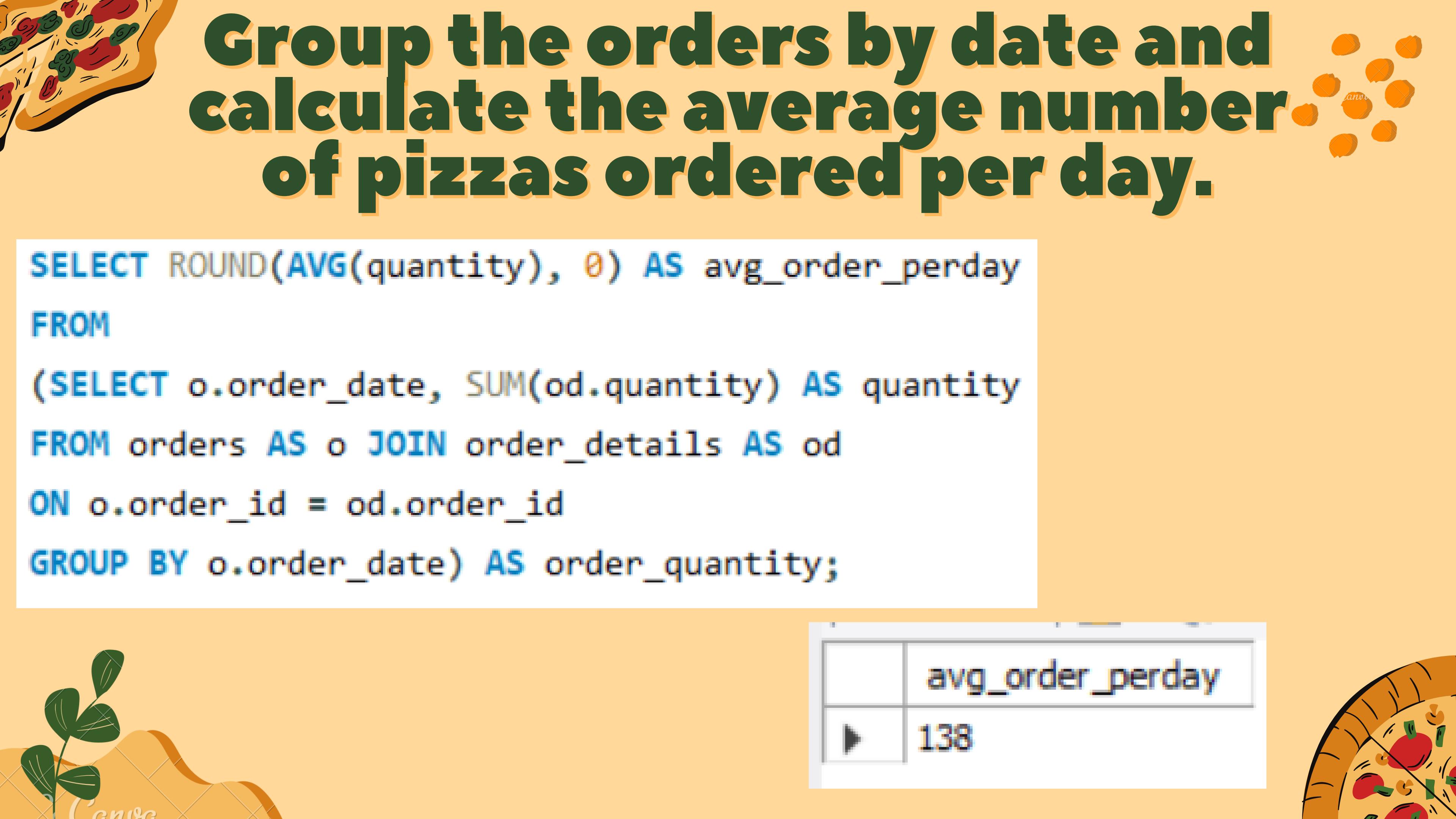
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



# Find the category-wise distribution of pizzas.

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

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



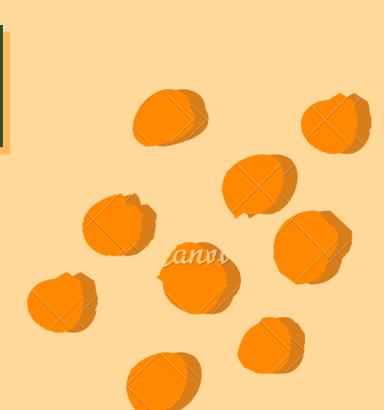
# Group the orders by date and calculate the average number of pizzas ordered per day.

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

	avg_order_perday
▶	138



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



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

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



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

```
SELECT pt.category, ROUND(SUM(od.quantity * p.price) /  
(SELECT SUM(od.quantity * p.price) FROM  
order_details AS od JOIN pizzas AS p ON od.pizza_id = p.pizza_id) * 100,2) AS revenue  
FROM order_details od  
JOIN pizzas p ON od.pizza_id = p.pizza_id  
JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY pt.category ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

# Analyze the cumulative revenue generated over time.

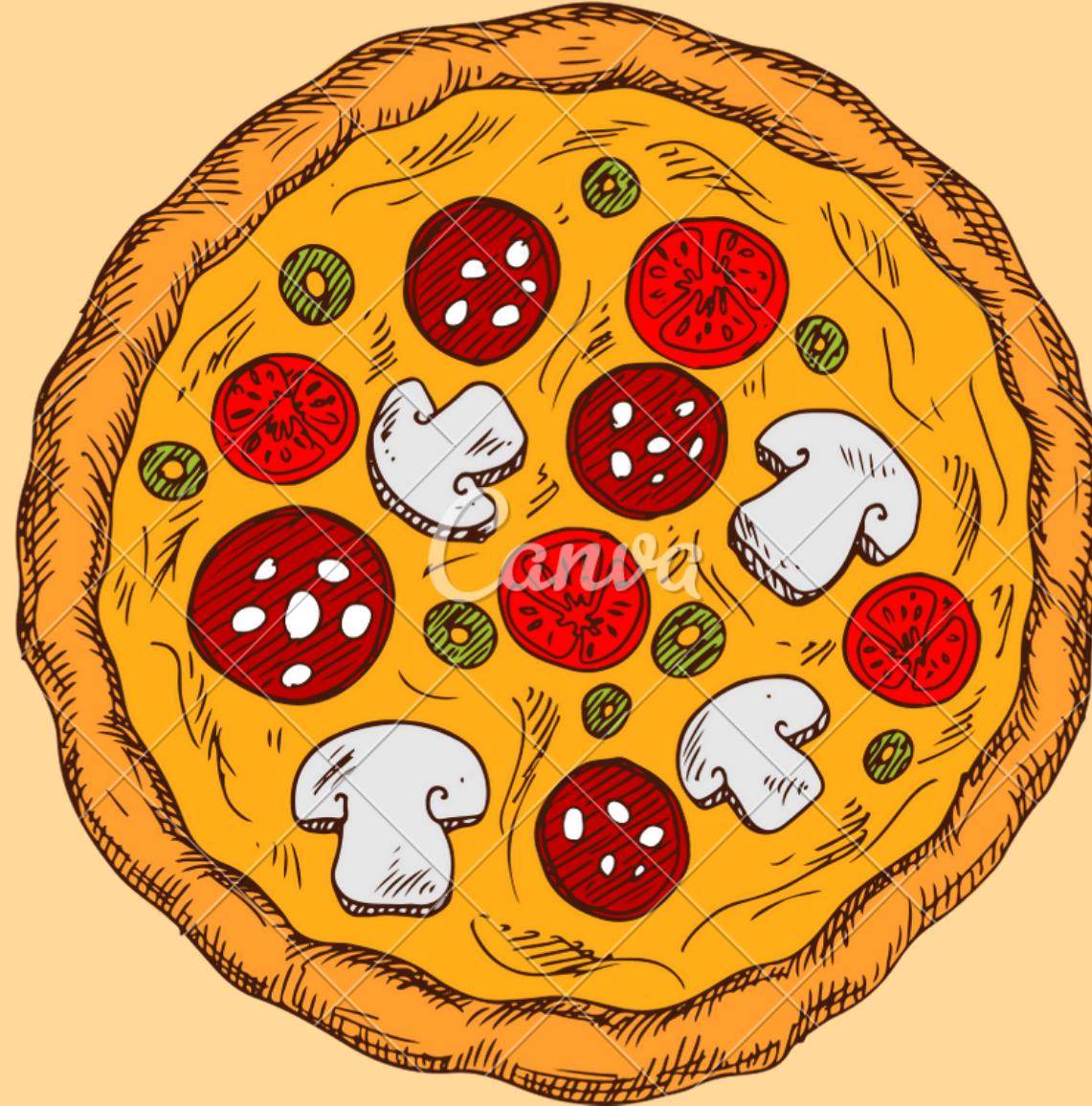
```
select order_date,  
round(sum(revenue) over(order by order_date),2) as cum_revenue  
from (select o.order_date, sum(od.quantity*p.price)as revenue  
from order_details as od join pizzas as p  
on od.pizza_id=p.pizza_id join orders as o  
on o.order_id=od.order_id group by o.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2713.85
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	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2

# 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 pt.category, pt.name, sum(od.quantity*p.price) as revenue  
from order_details od join pizzas p on od.pizza_id=p.pizza_id  
join pizza_types pt on p.pizza_type_id=pt.pizza_type_id  
group by pt.category, pt.name)as a) as b where rn <=3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



Pizza Sales Report

**THANK  
YOU**