



Pizza Sales Analysis Using PostgreSQL



Introduction

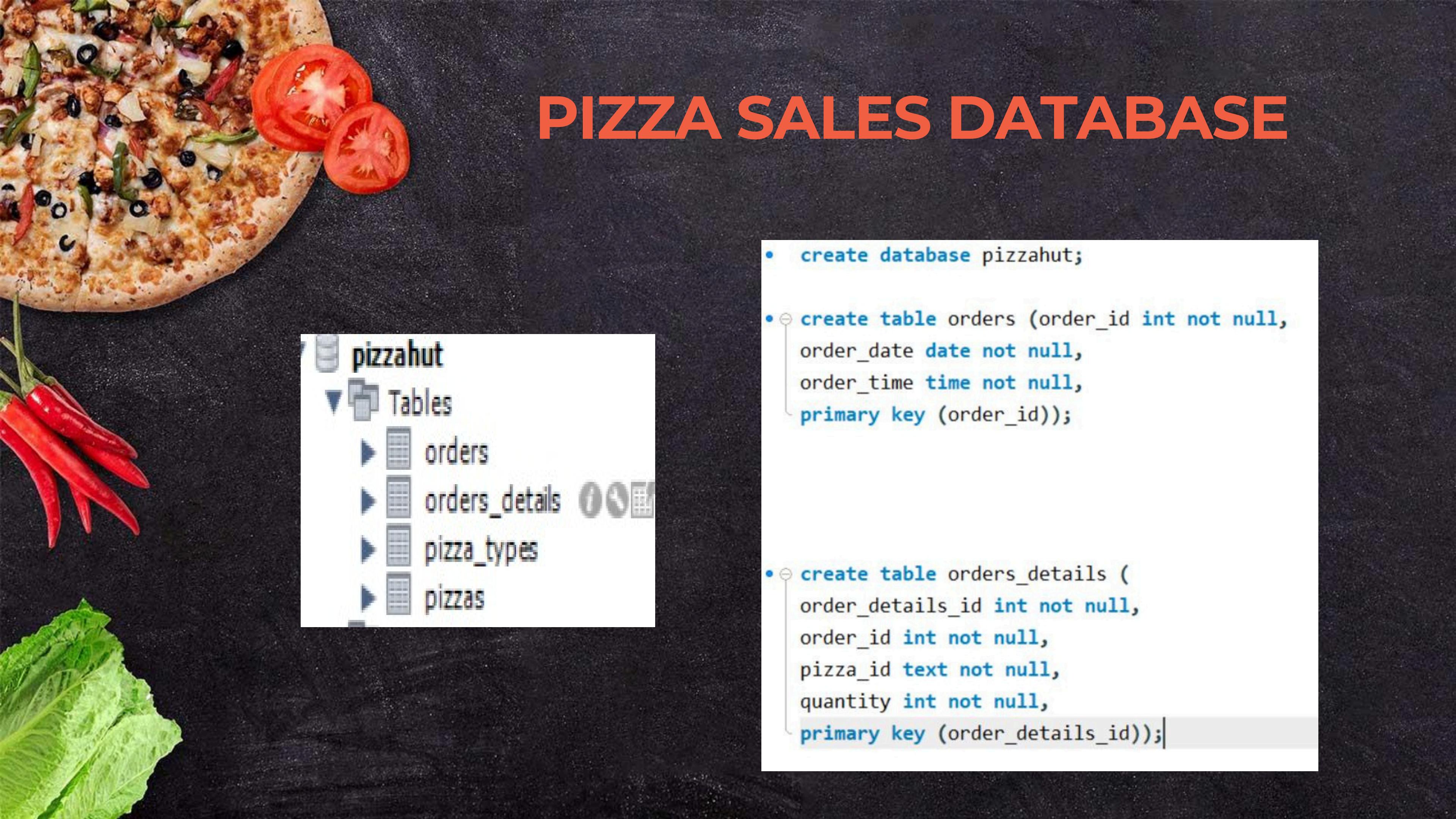
This SQL project analyzes a pizza sales dataset to gain insights into various aspects of the business. The dataset includes information about orders, pizza types, prices, and order timings.

By using SQL queries, the project aims to retrieve meaningful data that can help understand customer preferences, optimize the menu, and improve overall business strategies.

Objective

This pizza sales project aims to analyze the sales data to extract actionable insights that can inform business decisions. By utilizing SQL, the project aims to:

- 1. Understand Sales Performance:** Calculate total orders, and revenue, and identify the best-performing pizzas in terms of sales volume and revenue.
- 2. Customer Preferences:** Identify the most popular pizza sizes and types to understand customer preferences and demand patterns.
- 3. Optimize Inventory and Menu:** Analyze the distribution of orders by time and category to optimize inventory management and tailor the menu to meet customer demand effectively.
- 4. Revenue Analysis:** Assess the contribution of different pizza types to total revenue and evaluate revenue trends over time to support strategic planning and forecasting.



PIZZA SALES DATABASE



- `create database pizzahut;`
- `create table orders (order_id int not null, order_date date not null, order_time time not null, primary key (order_id));`
- `create table orders_details (order_details_id int not null, order_id int not null, pizza_id text not null, quantity int not null, primary key (order_details_id));`

Retrieve the total number of orders placed



```
select count(order_id) as total_orders from orders;
```

	total_orders	lock
1	21350	



Calculate the total revenue generated from pizza sales.



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

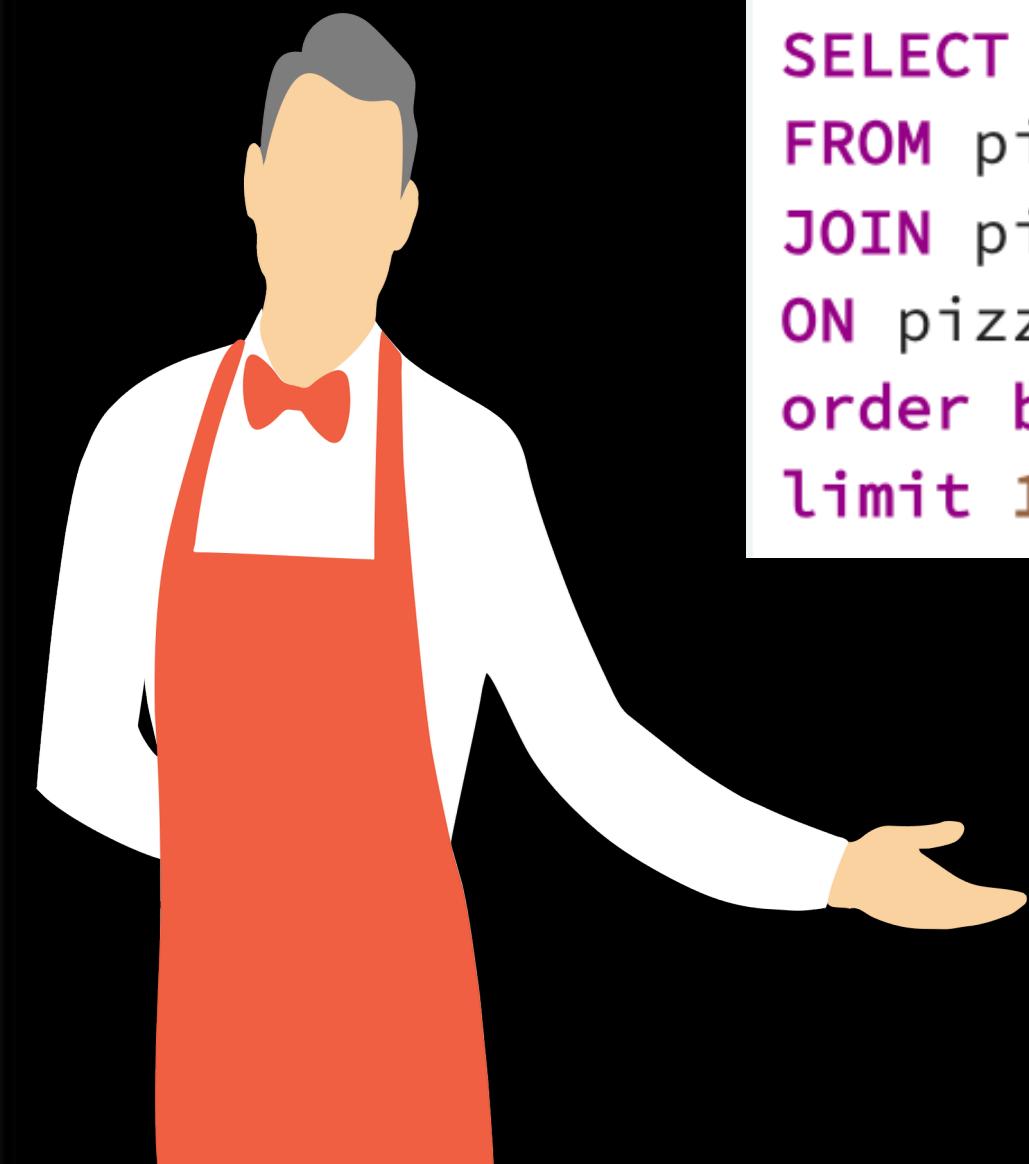
	total_revenue	🔒
1	817860.05	



Identify the highest-priced pizza

```
SELECT pizza_types.name, pizzas.price  
FROM pizzas  
JOIN pizza_types  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by pizzas.price desc  
limit 1;
```

	name character varying (500) 	price double precision 
1	The Greek Pizza	35.95

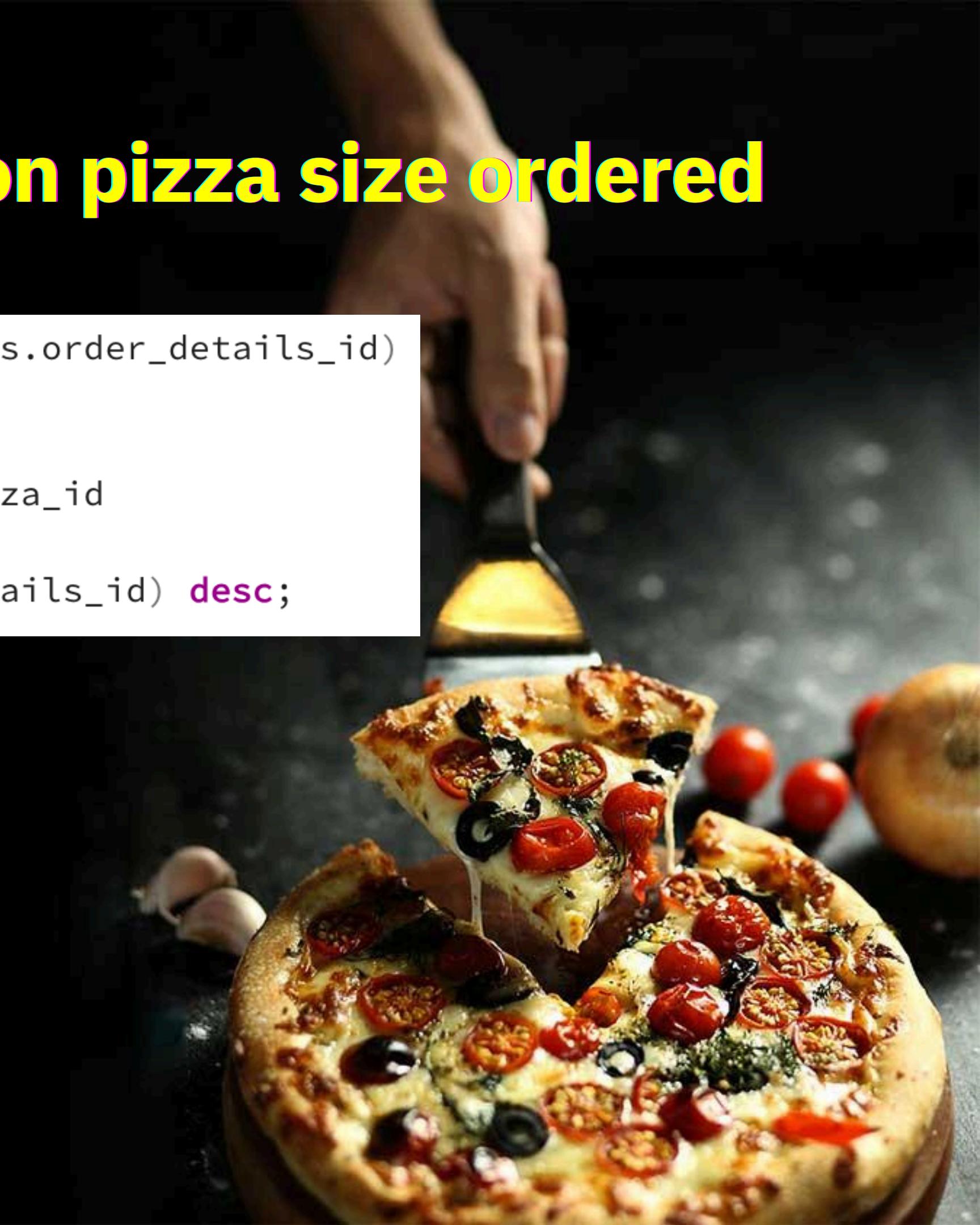


Identify the most common pizza size ordered

```
SELECT pizzas.size, count(order_details.order_details_id)
FROM pizzas
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by count(order_details.order_details_id) desc;
```



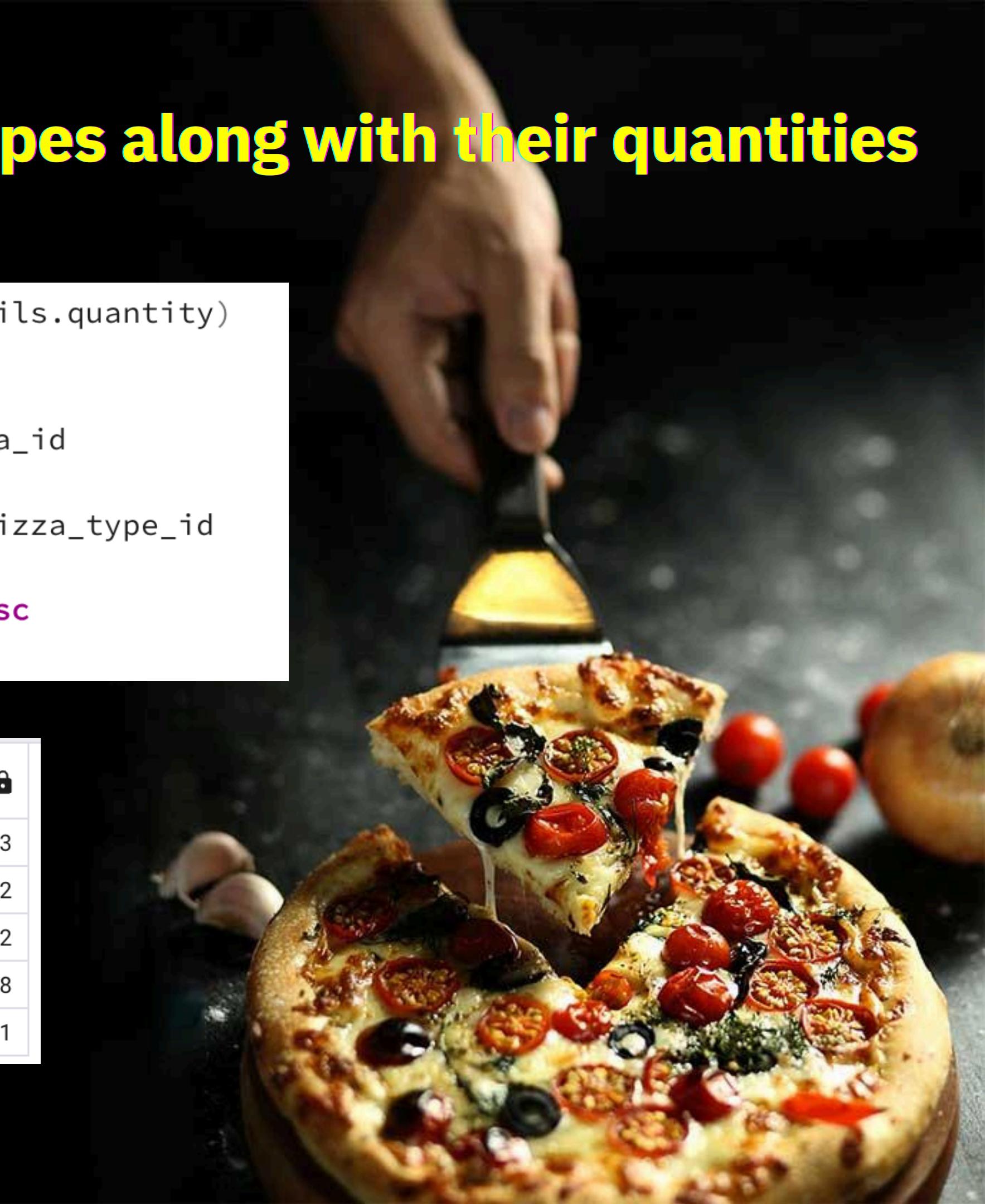
	size character varying (50)	count bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28



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

```
SELECT pizza_types.name, sum(order_details.quantity)
FROM pizzas
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.name
order by sum(order_details.quantity) desc
limit 5;
```

	name character varying (500)	sum bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

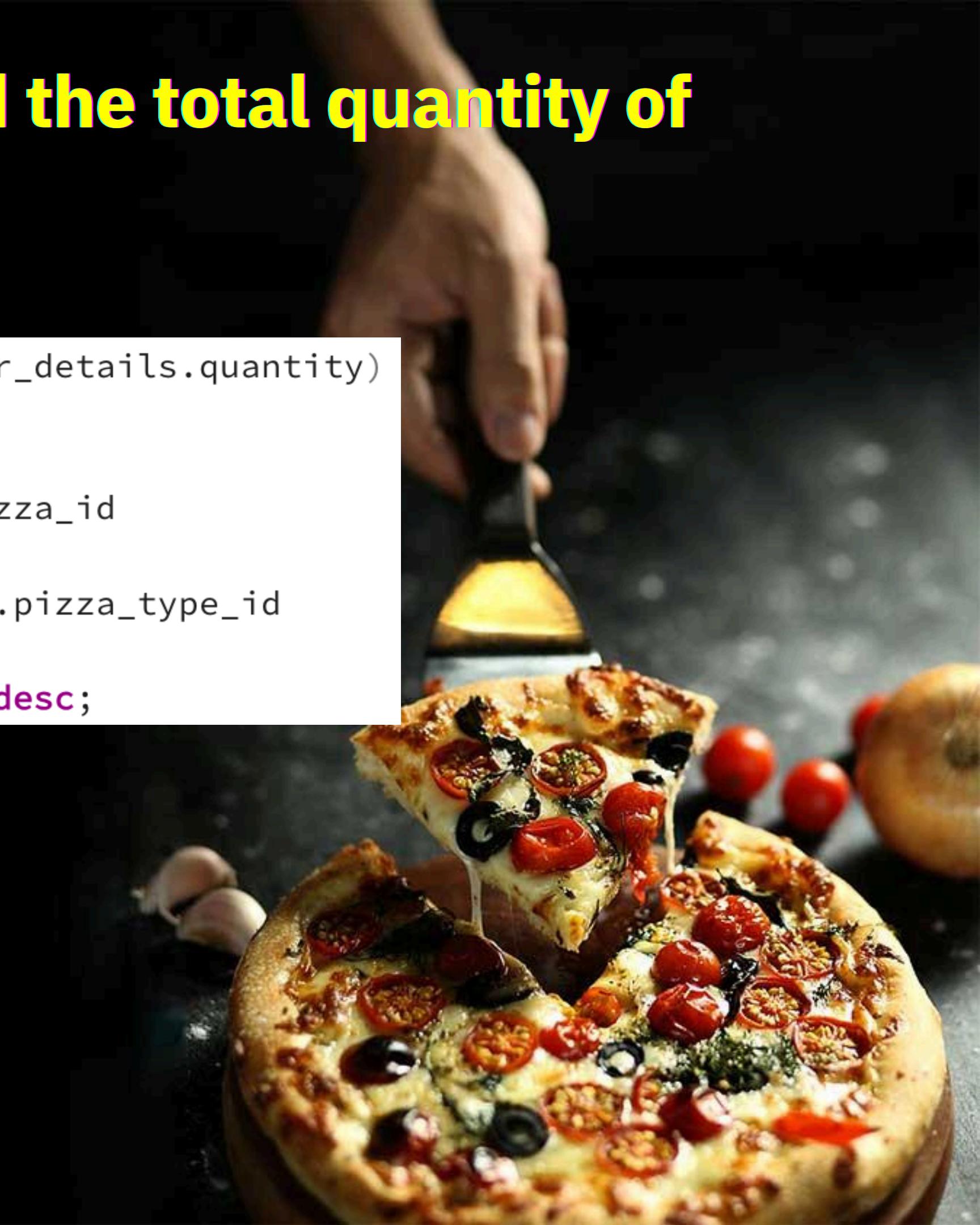


Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT pizza_types.category, sum(order_details.quantity)
FROM pizzas
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category
order by sum(order_details.quantity) desc;
```



	category character varying (100)	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

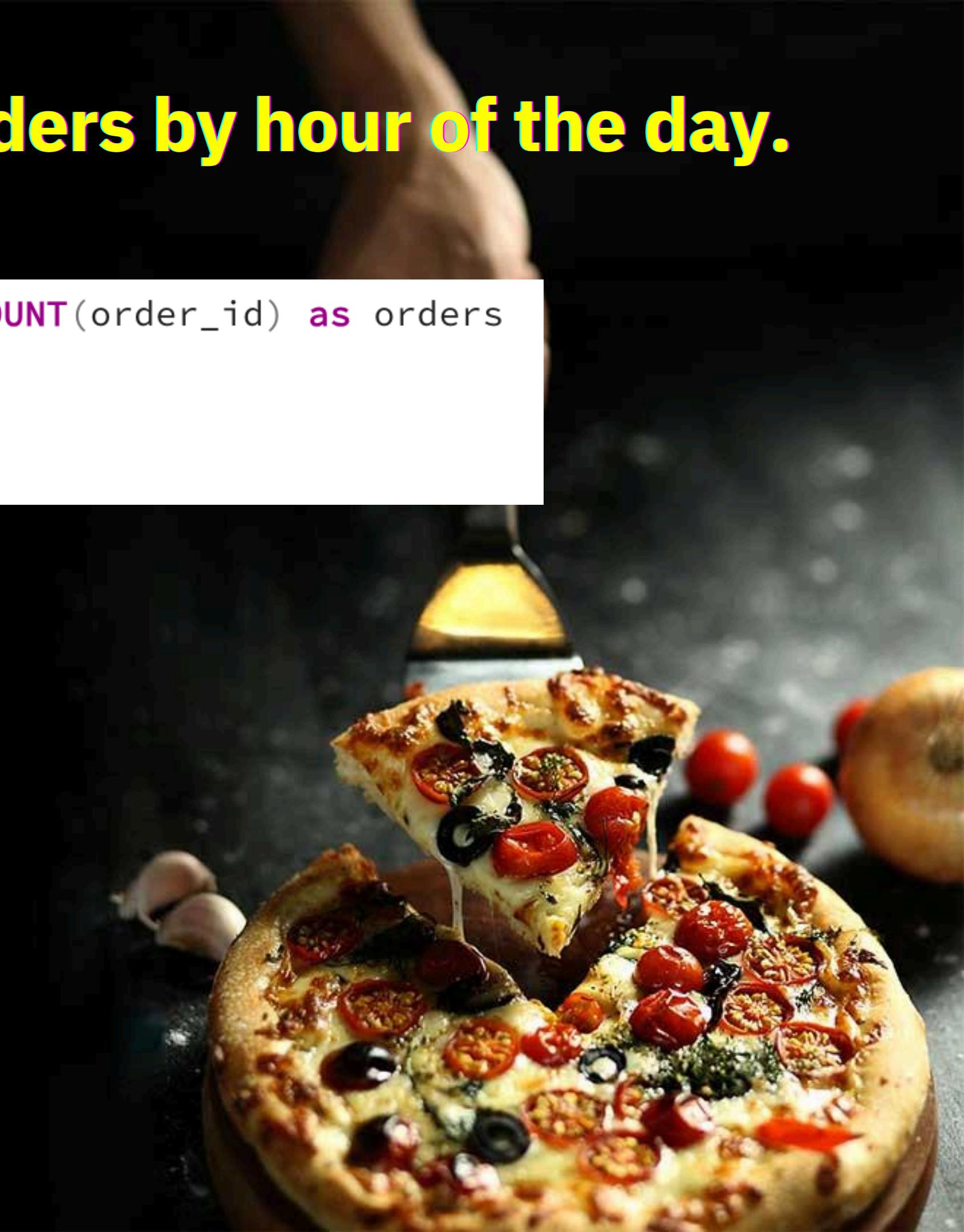


Determine the distribution of orders by hour of the day.

```
SELECT EXTRACT(HOUR FROM time) AS hour, COUNT(order_id) as orders  
FROM orders  
GROUP BY EXTRACT(HOUR FROM time)  
ORDER BY COUNT(order_id) desc;
```



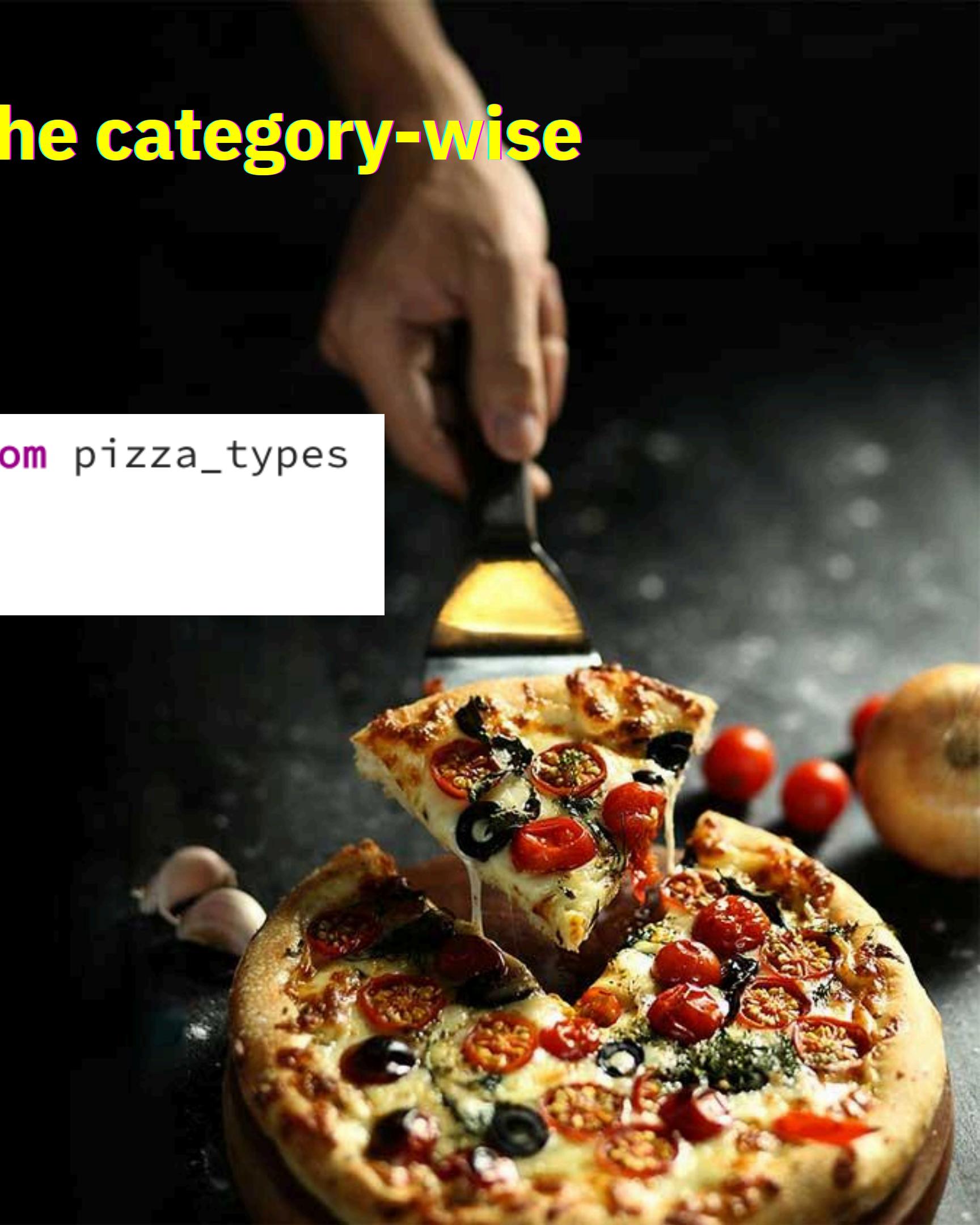
hour numeric	orders bigint
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1



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

```
SELECT category, count(name) from pizza_types  
group by category  
order by count(name) desc;
```

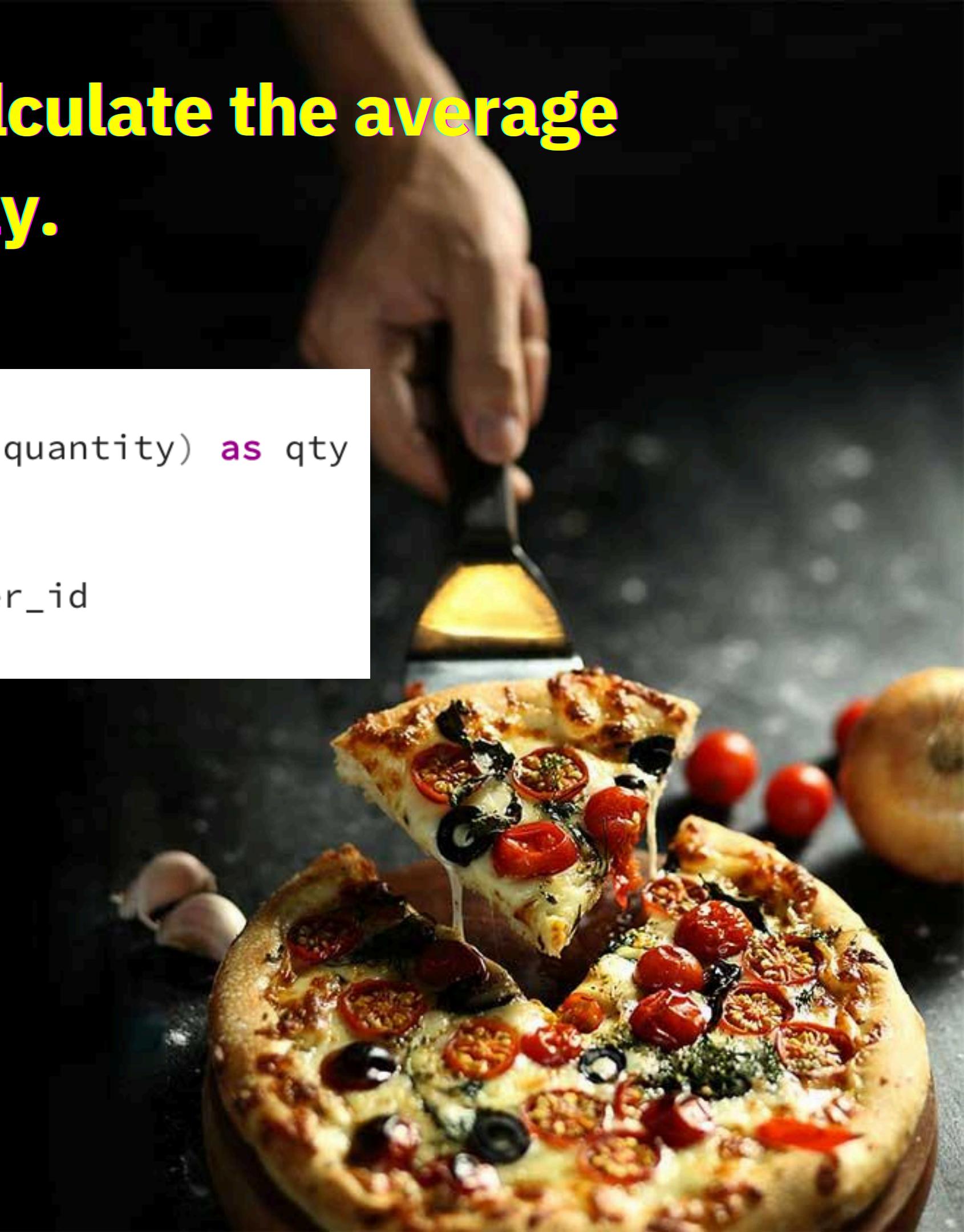
category	count
character varying (100)	bigint
Veggie	9
Supreme	9
Classic	8
Chicken	6



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

```
select round(avg(qty),0) from  
(SELECT orders.date, sum(order_details.quantity) as qty  
from orders  
join order_details  
on orders.order_id = order_details.order_id  
group by orders.date) as orders_qty;
```

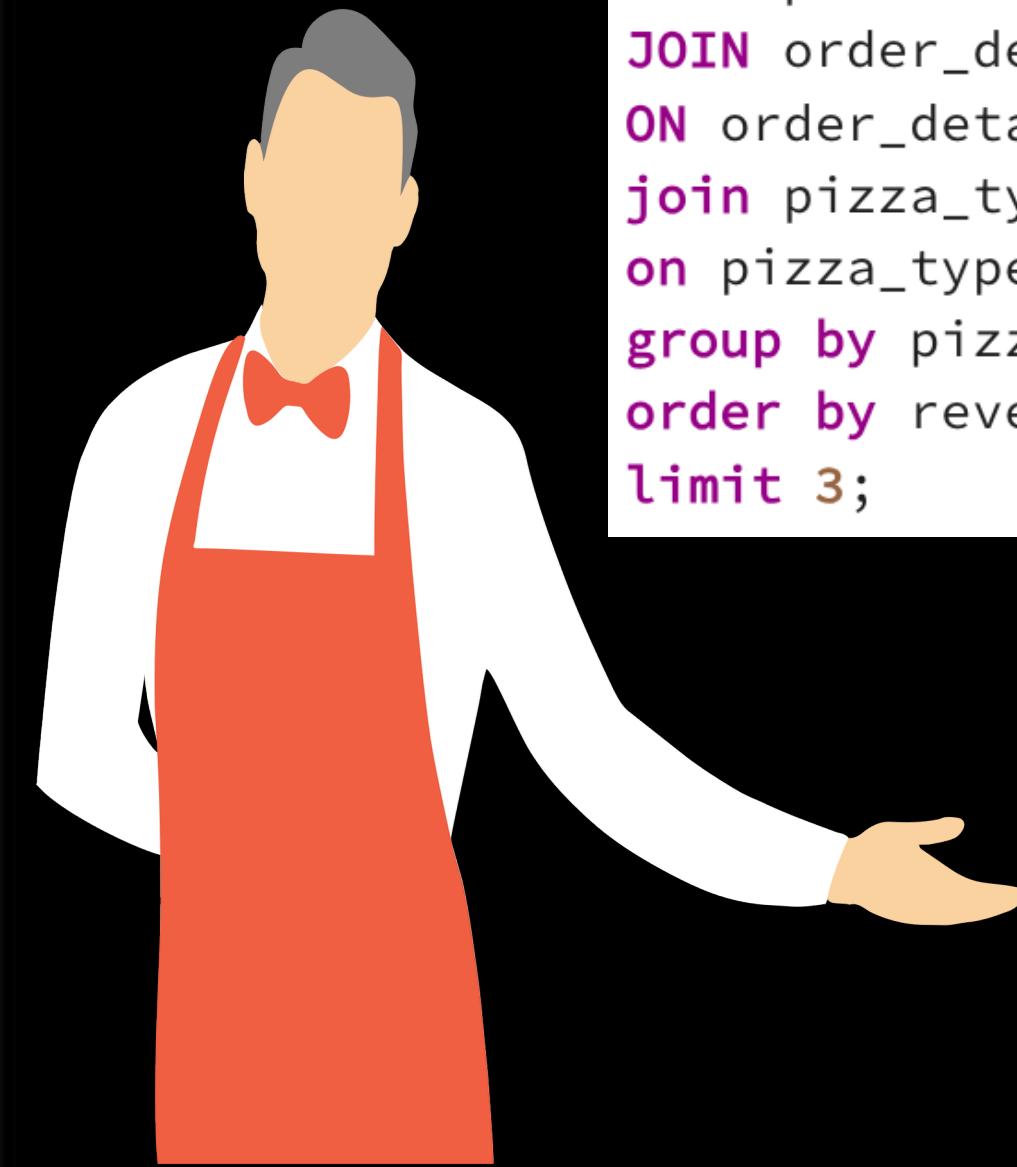
round	numeric	🔒
138		



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

```
select pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue
FROM pizzas
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.name
order by revenue desc
limit 3;
```

name character varying (500)	revenue double precision
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
    pizza_types.category,
    ROUND(
        (SUM(order_details.quantity * pizzas.price) /
        (SELECT SUM(order_details.quantity * pizzas.price) FROM pizzas
        JOIN order_details ON order_details.pizza_id = pizzas.pizza_id) * 100)::numeric, 2
    ) AS revenue_percentage
FROM pizzas
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage DESC;
```



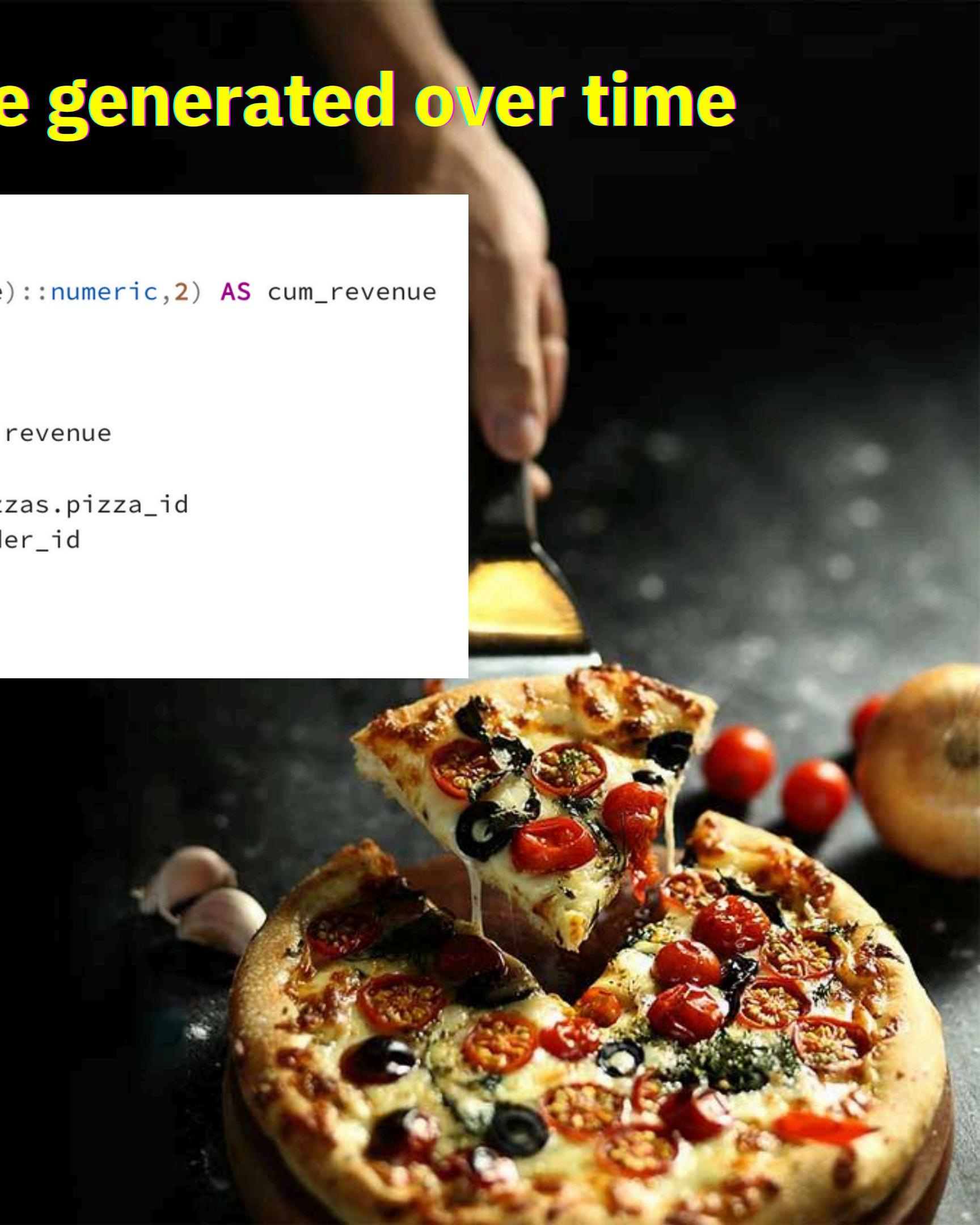
category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



Analyze the cumulative revenue generated over time

```
SELECT
    sales.date,
    round(SUM(sales.revenue) OVER (ORDER BY sales.date)::numeric,2) AS cum_revenue
FROM (
    SELECT
        orders.date,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM pizzas
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
    JOIN orders ON orders.order_id = order_details.order_id
    GROUP BY orders.date
) AS sales
ORDER BY sales.date;
```

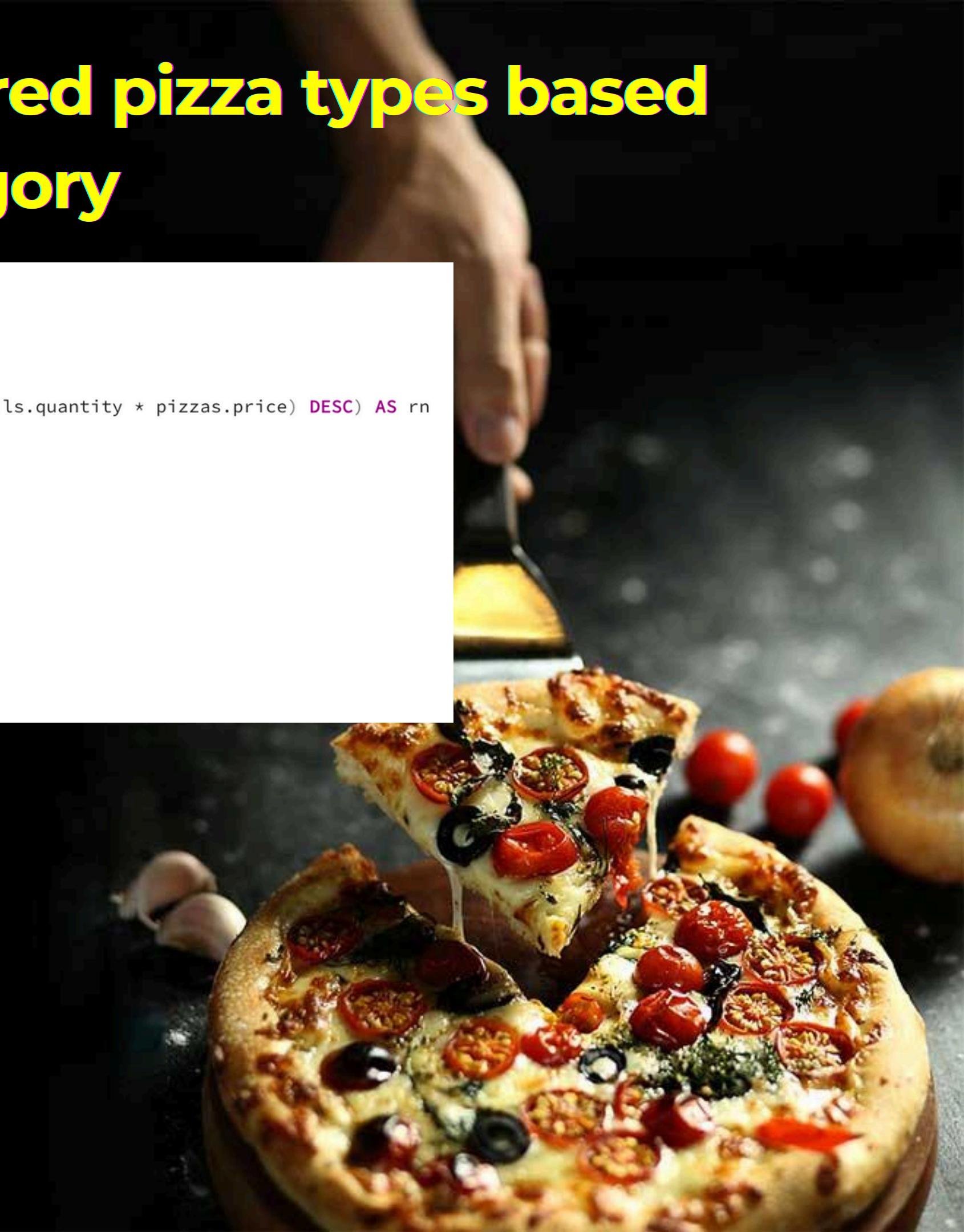
date date	locked	cum_revenue numeric	locked
2015-01-01		2713.85	
2015-01-02		5445.75	
2015-01-03		8108.15	
2015-01-04		9863.60	
2015-01-05		11929.55	
2015-01-06		14358.50	
2015-01-07		16560.70	
2015-01-08		19399.05	
2015-01-09		21526.40	
2015-01-10		23990.35	
2015-01-11		26459.70	



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

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

name character varying (500)	category character varying (100)	revenue double precision
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5





Suggestion and solution

- **Promote Top-Selling Pizzas:** Focus on marketing and promotions for the top 3 revenue-generating pizza types in each category.
 - **Use Revenue Trends:** Analyze revenue trends over time to optimize inventory, staffing, and marketing efforts during peak and off-peak times.
 - **Optimize Pizza Sizes:** Streamline the menu by focusing on the most popular pizza sizes, potentially reducing costs and enhancing customer satisfaction.
 - **Enhance Operational Efficiency:** Align staffing and preparation with peak order times to reduce wait times and improve service.
 - **Refine Product Mix:** Focus on high-margin, high-demand pizzas while phasing out less popular options.
 - **Maximize Customer Engagement:** Create loyalty programs or promotions based on the top 5 most ordered pizzas to encourage repeat purchases.
 - **Strategic Pricing:** Introduce premium versions of popular pizzas and assess pricing strategies to maximize revenue.
 - **Data-Driven Decision-Making:** Regularly update data analysis to monitor sales performance and customer preferences for informed decision-making.
- 

THANK YOU

Open for Freelancing Project

vinayjdc1@gmail.com

