PIZZA HUT SALES ANALYSIS

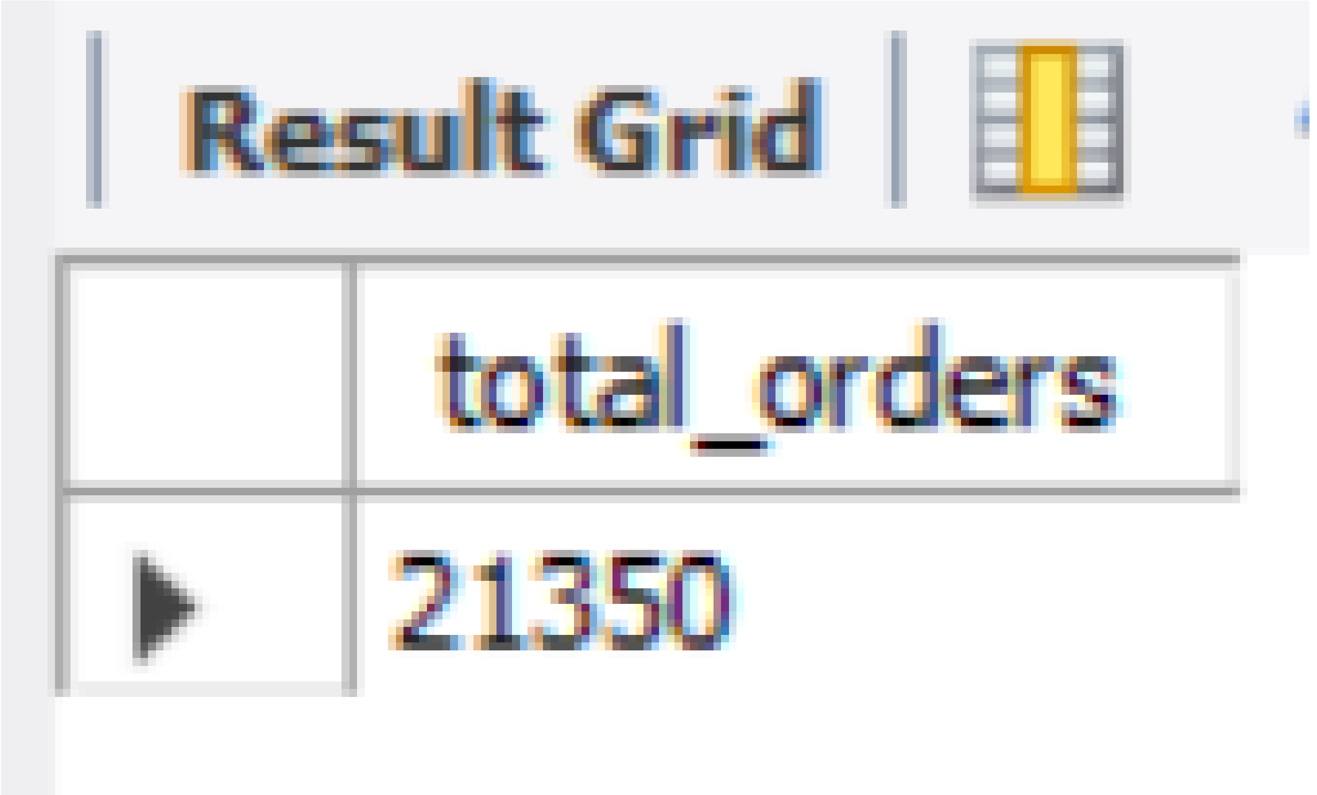
SQL PROJECT

INTRODUCTION

THIS PROJECT INVOLVES ANALYZING PIZZA HUT SALES DATA USING SQL TO EXTRACT MEANINGFUL INSIGHTS AND IMPROVE BUSINESS DECISION-MAKING PROCESSES. THE ANALYSIS COVERS VARIOUS DIMENSIONS SUCH AS SALES PERFORMANCE, CUSTOMER BEHAVIOR, PRODUCT PREFERENCES, AND OPERATIONAL EFFICIENCY.

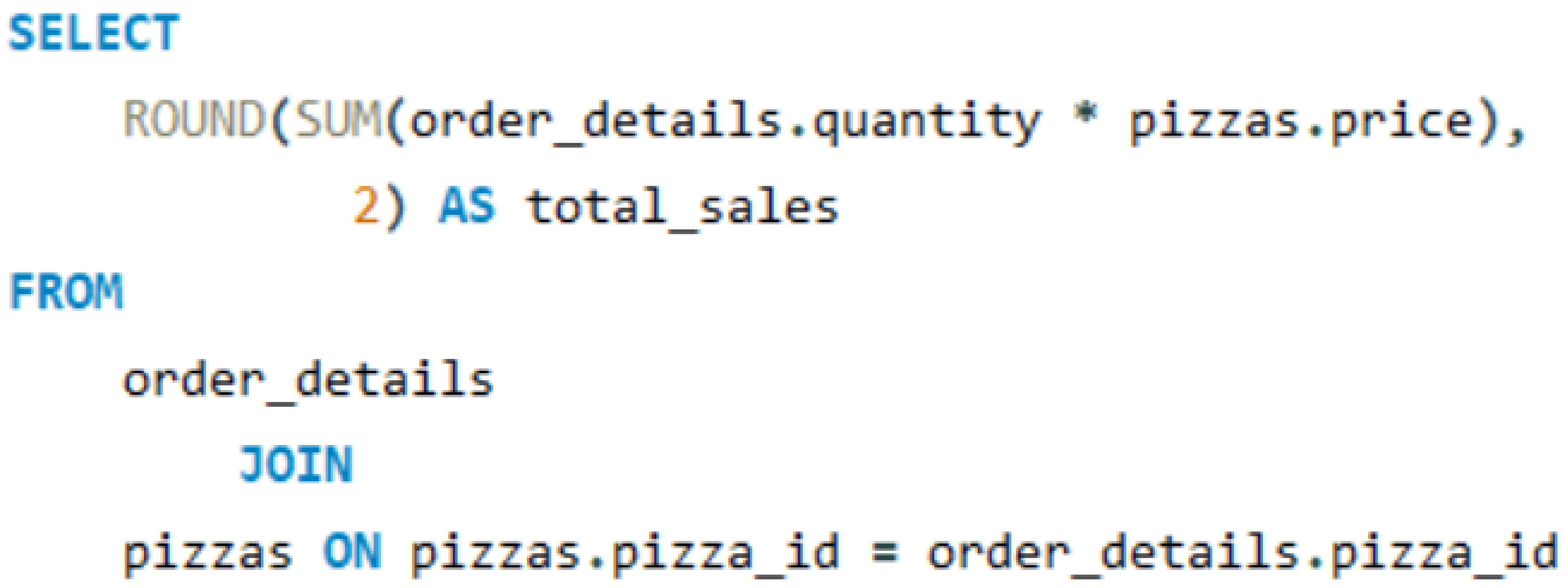
Retrieve the total number of orders placed.

```sql
Select count(order_id) As total_orders from orders
```

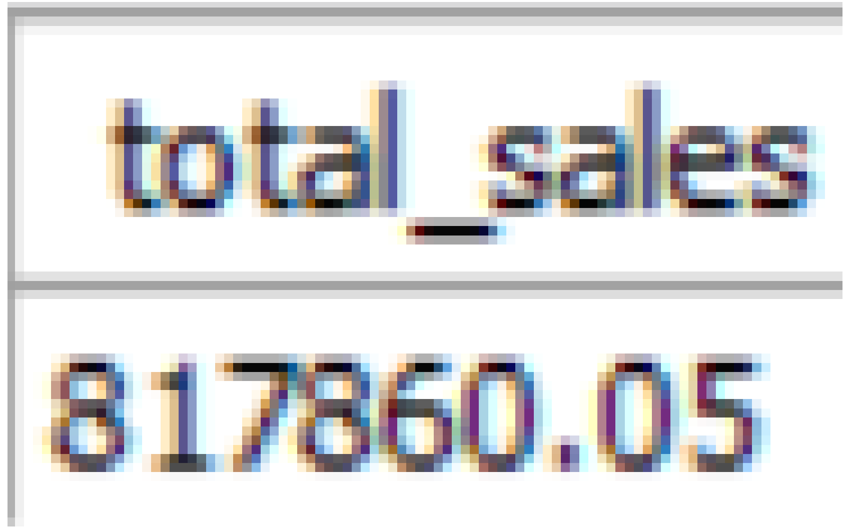


	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

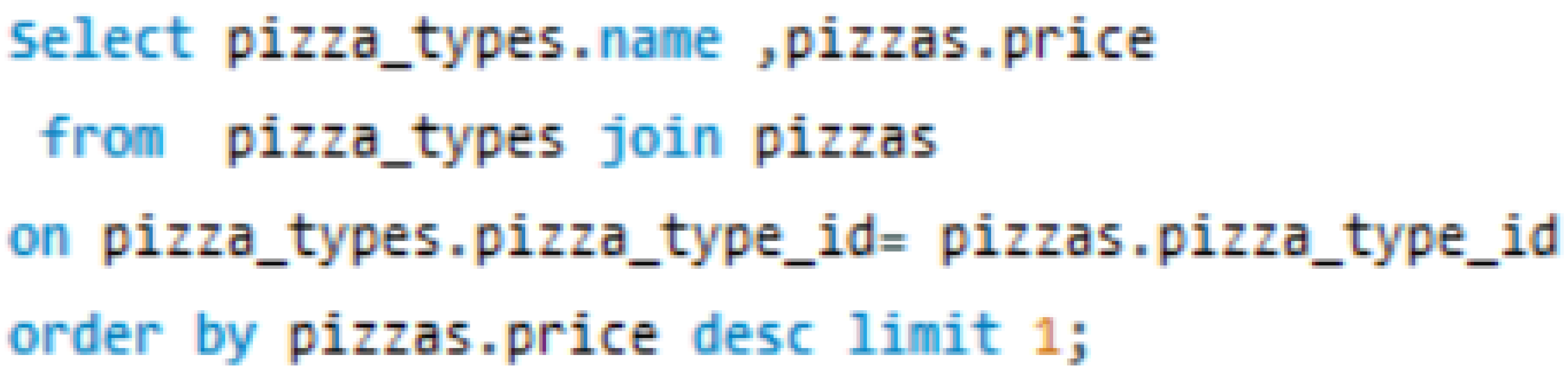


```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

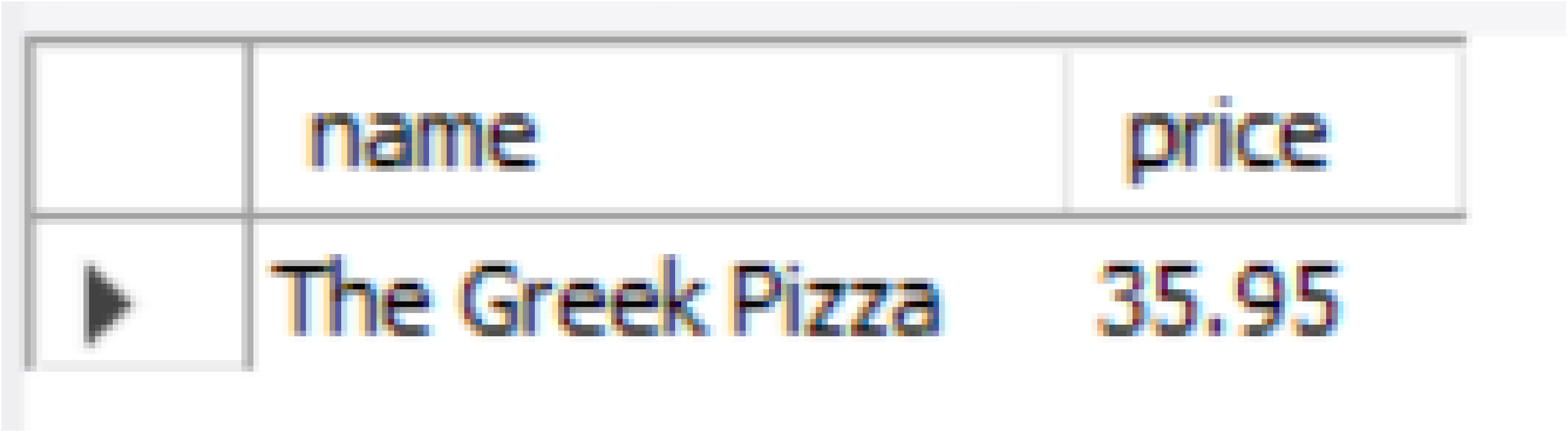


total_sales
817860.05

Identify the highest-priced pizza.

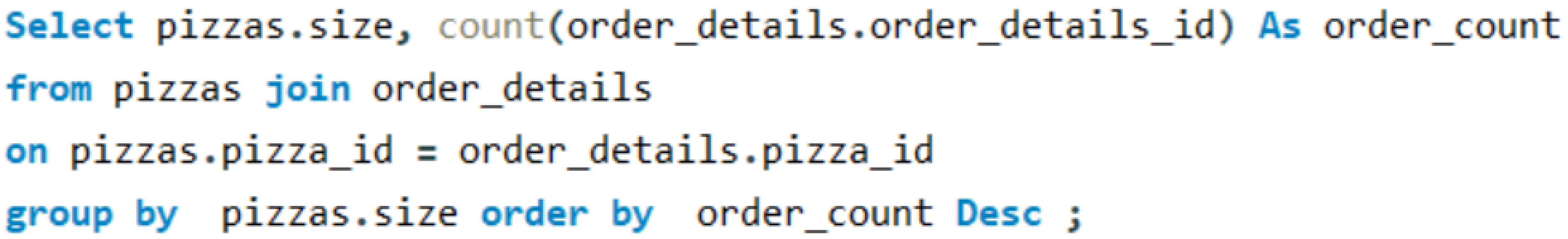


```
Select pizza_types.name ,pizzas.price
 from  pizza_types join pizzas
on pizza_types.pizza_type_id= pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

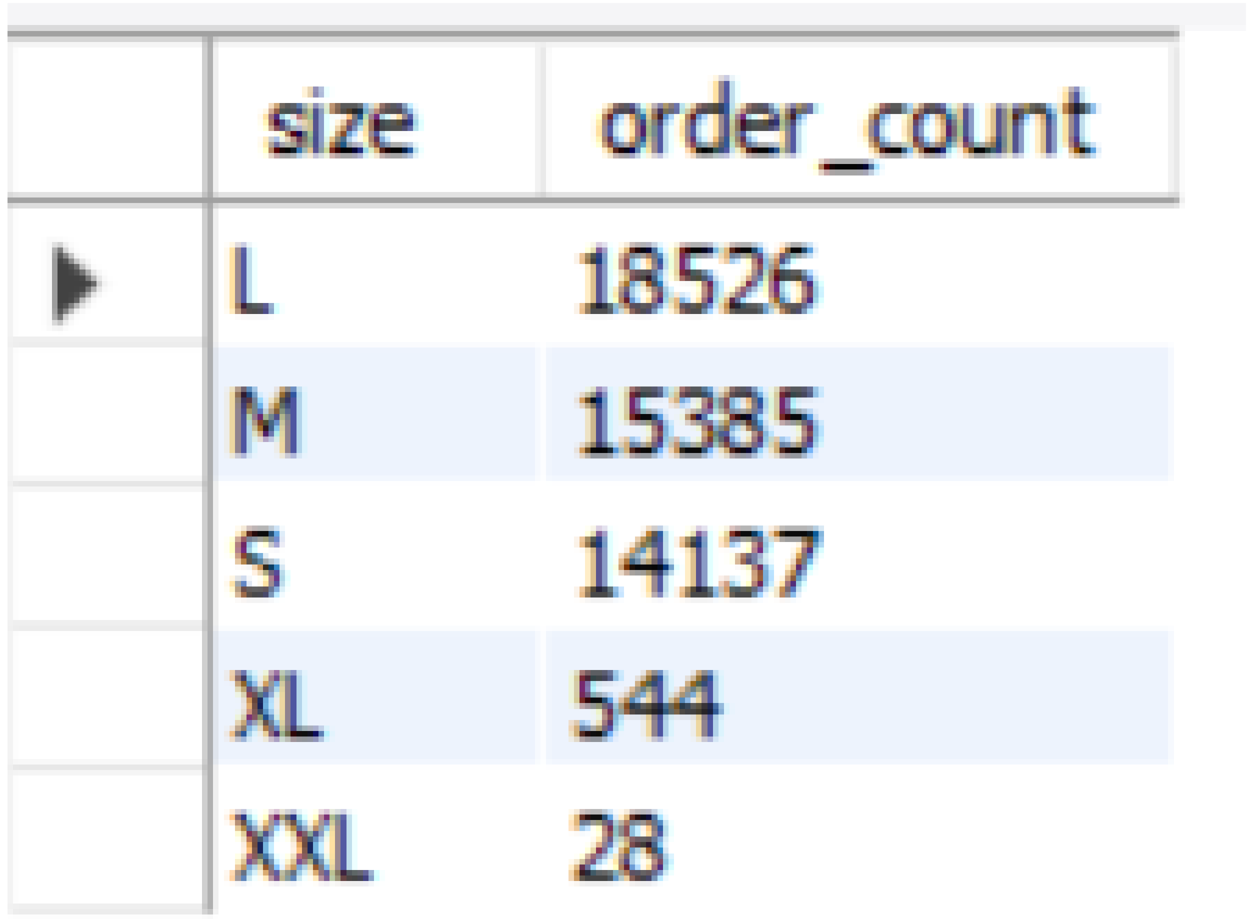


	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

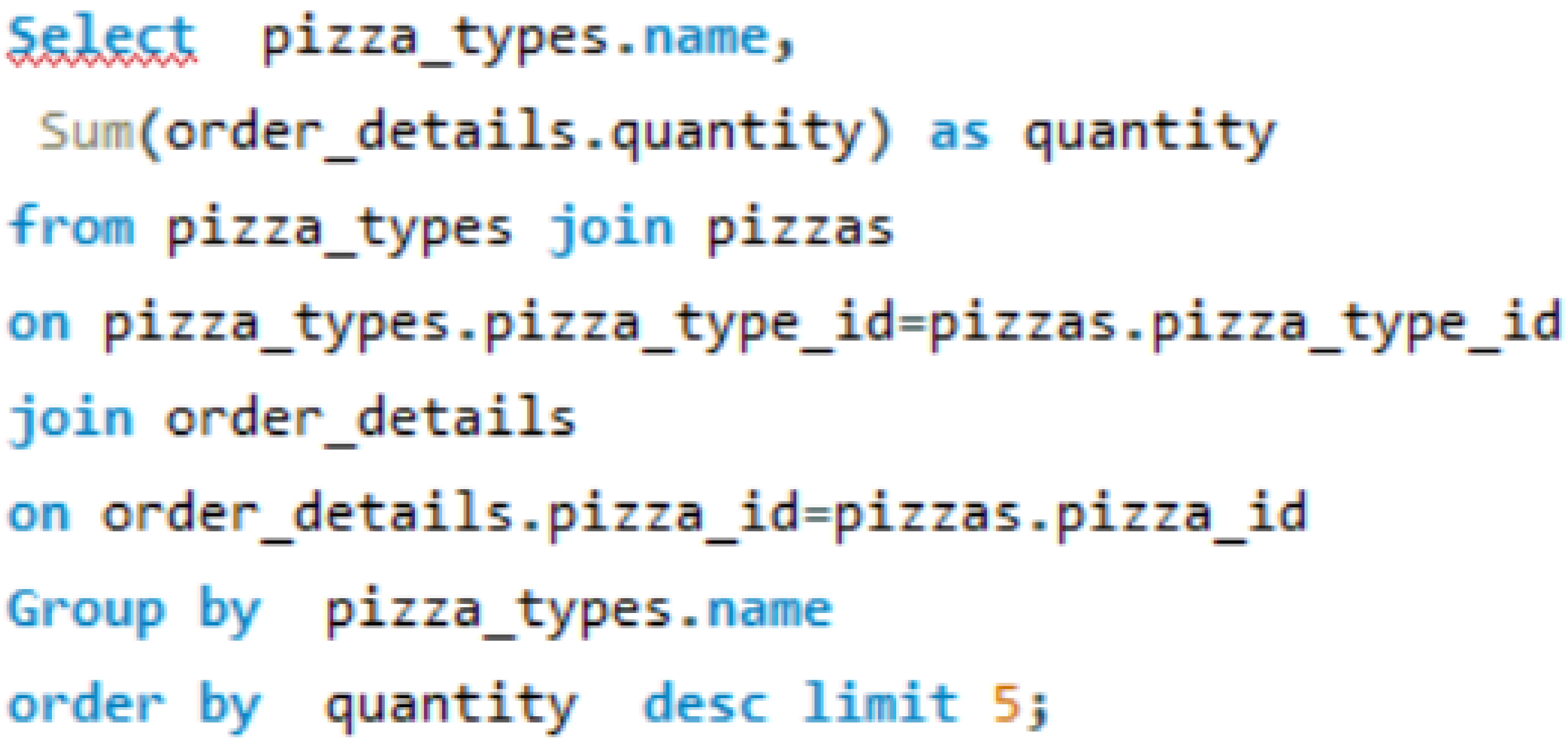


```sql
Select pizzas.size, count(order_details.order_details_id) As order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by  pizzas.size order by  order_count Desc ;
```

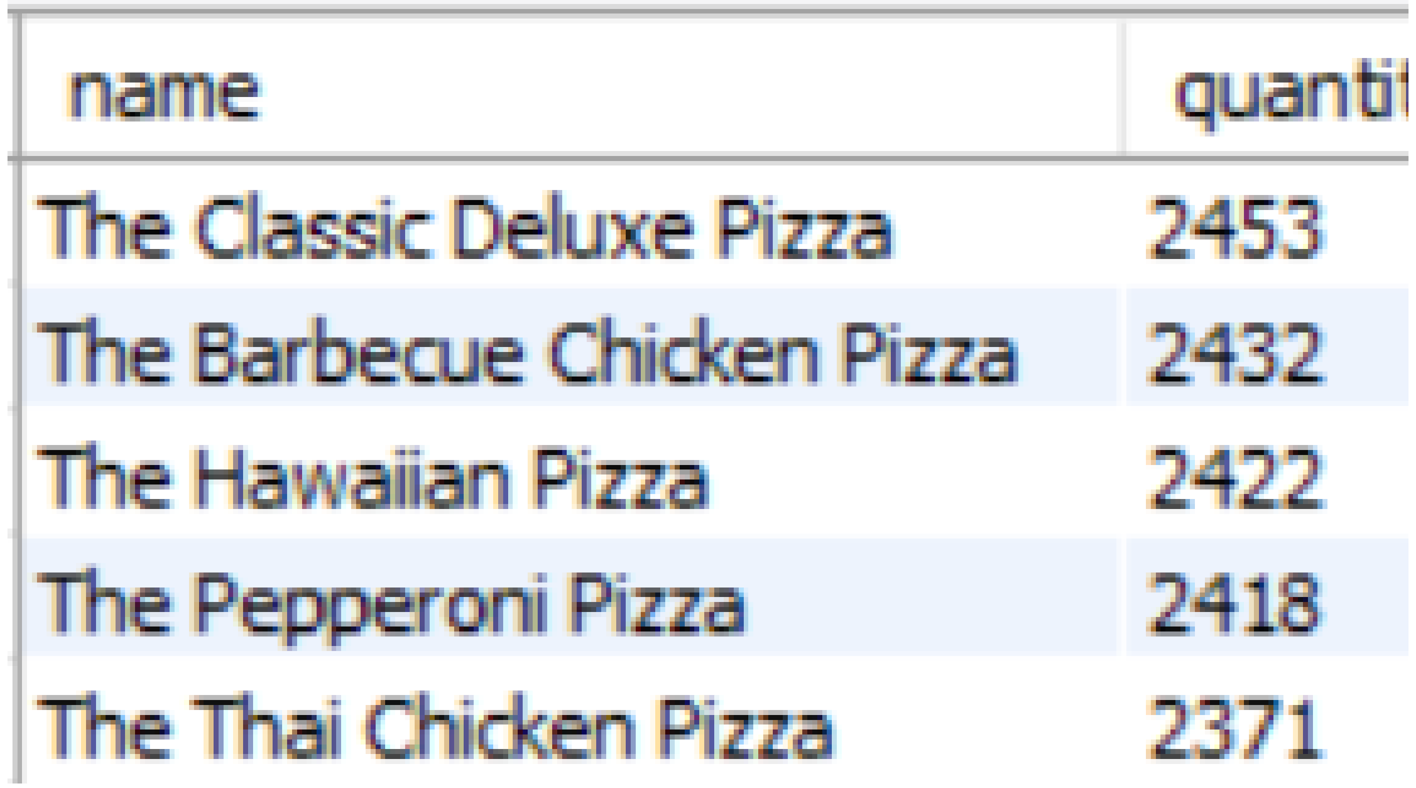


	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

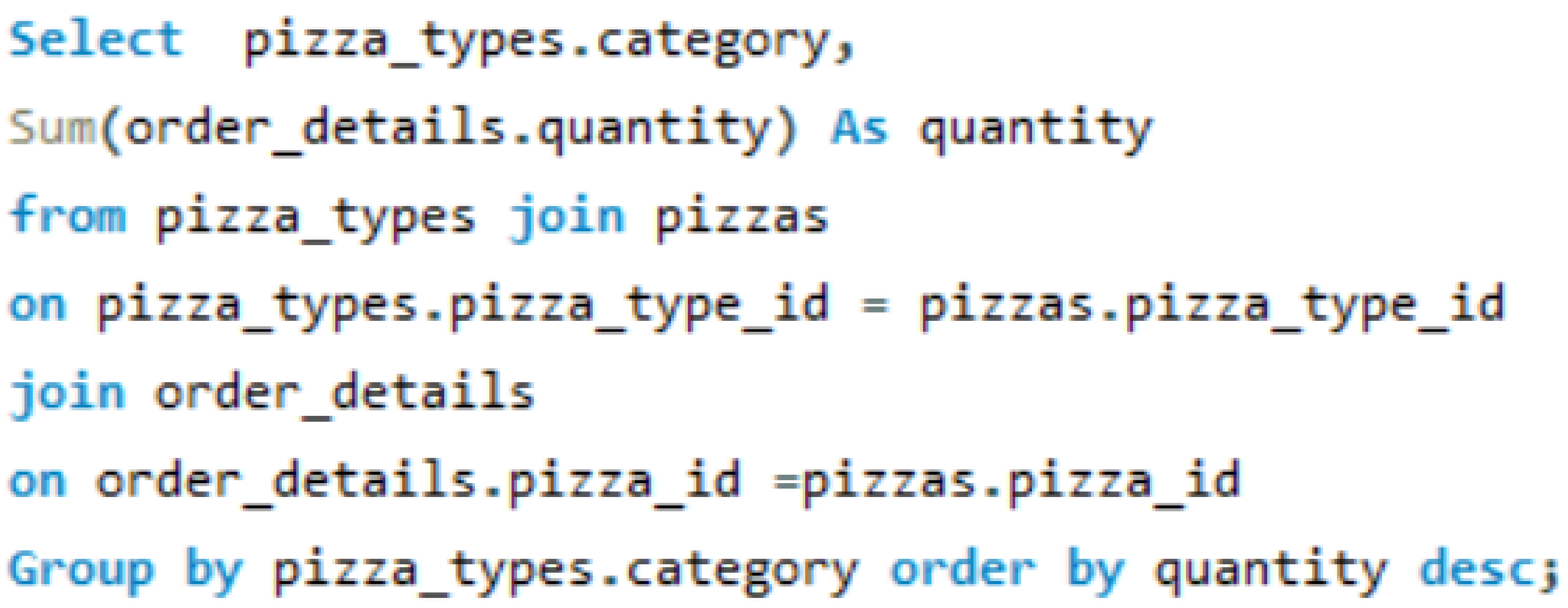


```
Select  pizza_types.name,
 Sum(order_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
Group by  pizza_types.name
order by  quantity  desc limit 5;
```

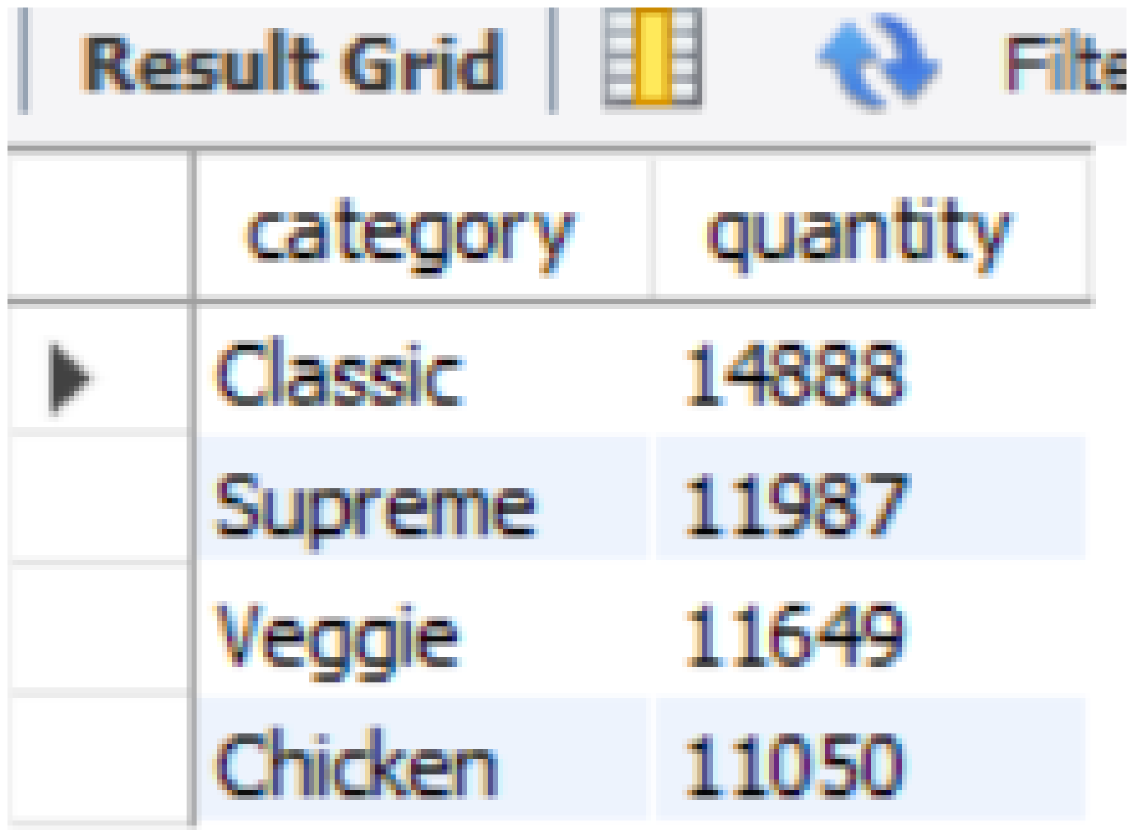


name	quanti
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

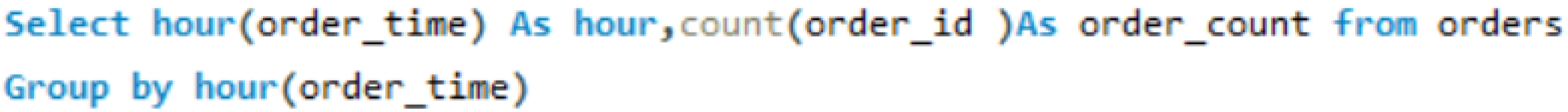


```
Select  pizza_types.category,
Sum(order_details.quantity) As quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id =pizzas.pizza_id
Group by pizza_types.category order by quantity desc;
```

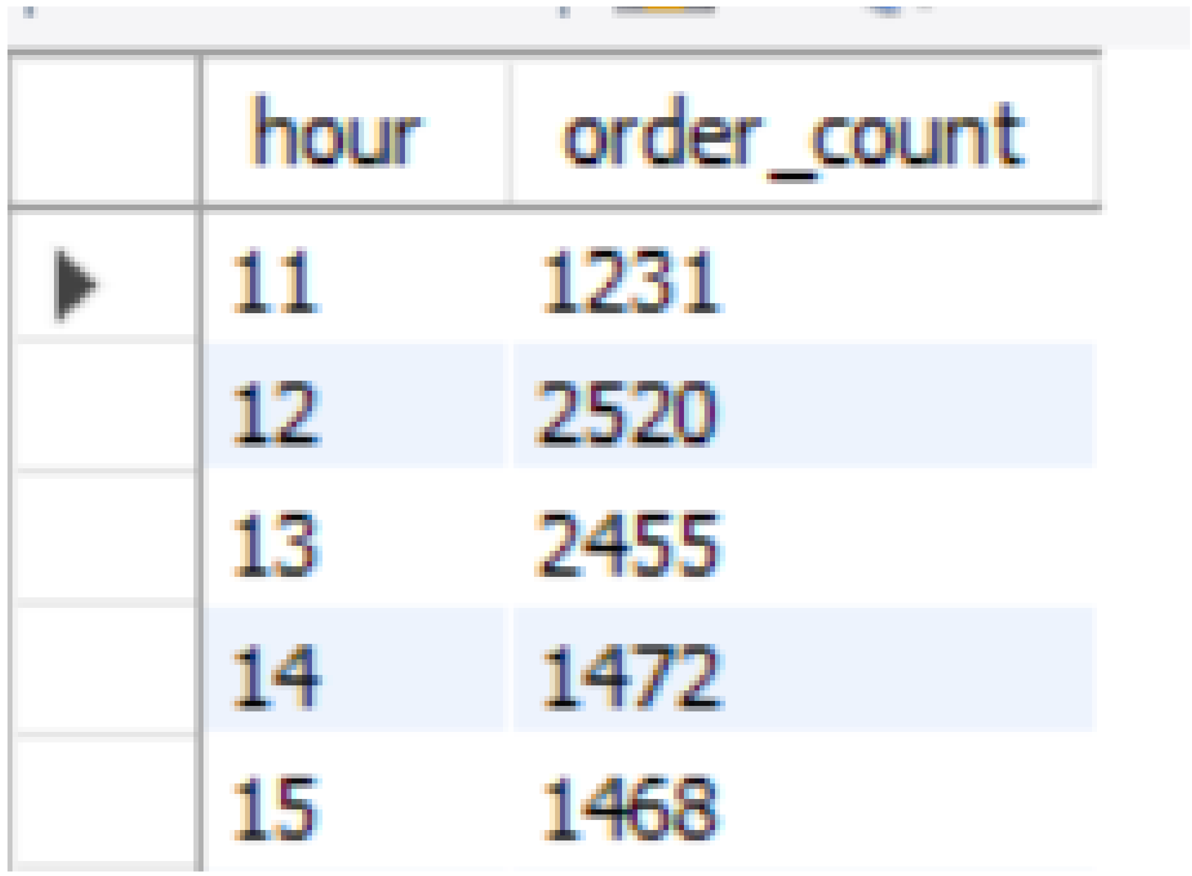


category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

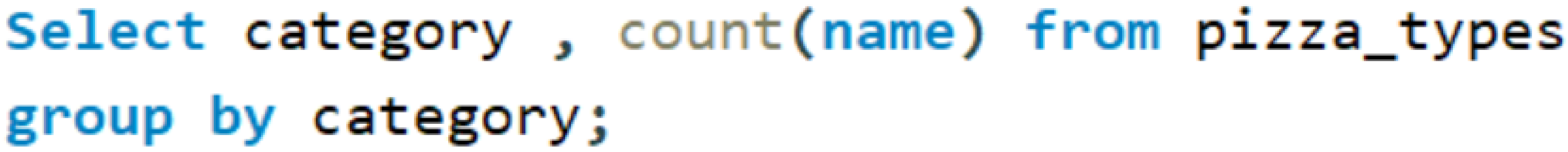


```
Select hour(order_time) As hour,count(order_id )As order_count from orders
Group by hour(order_time)
```

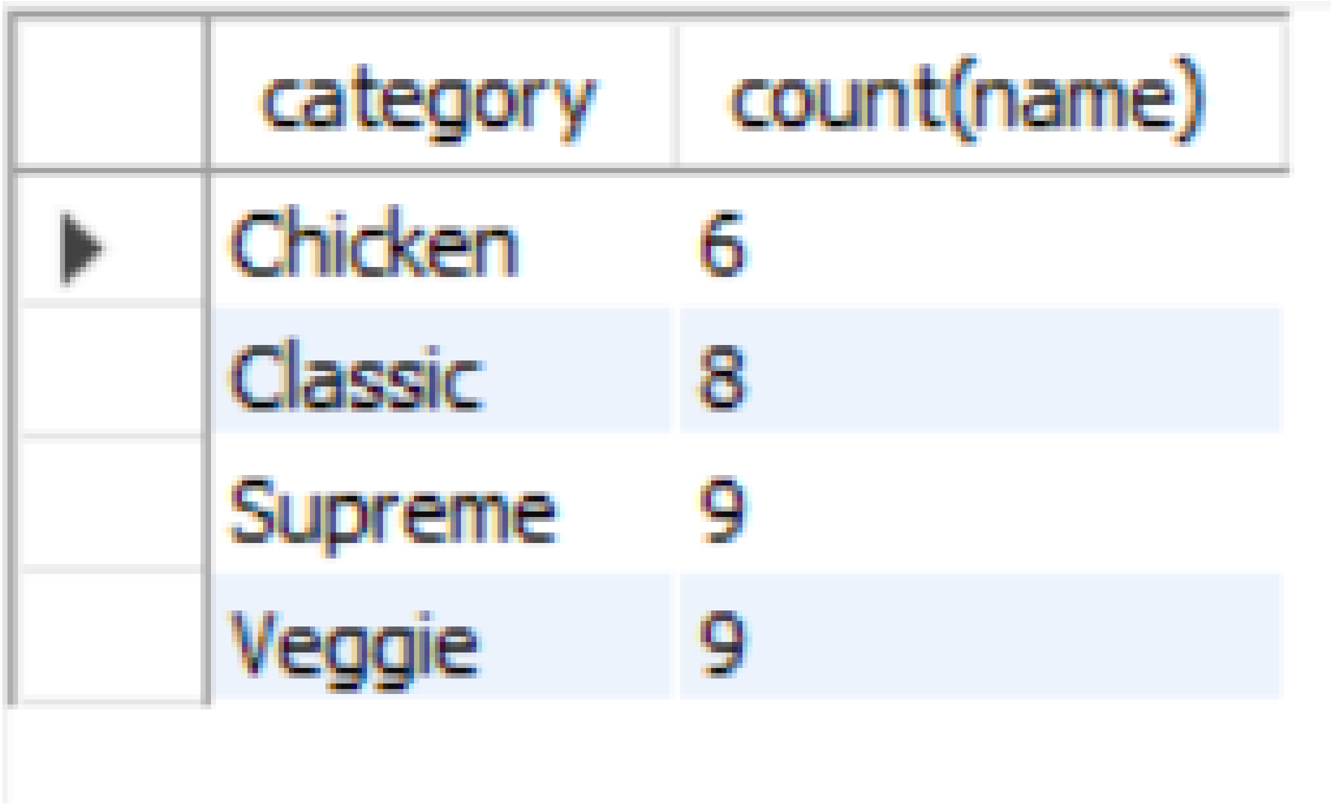


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

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

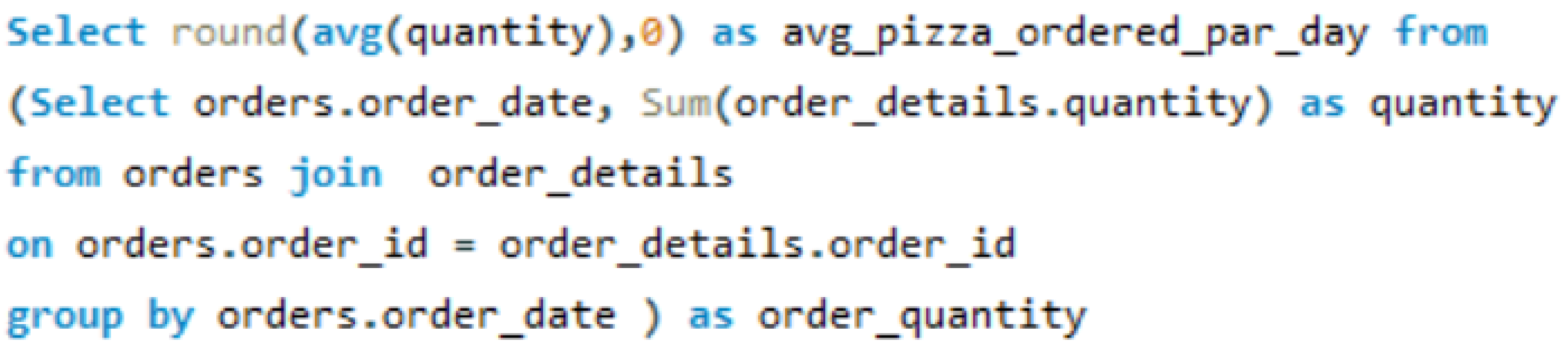


```sql
Select category , count(name) from pizza_types
group by category;
```

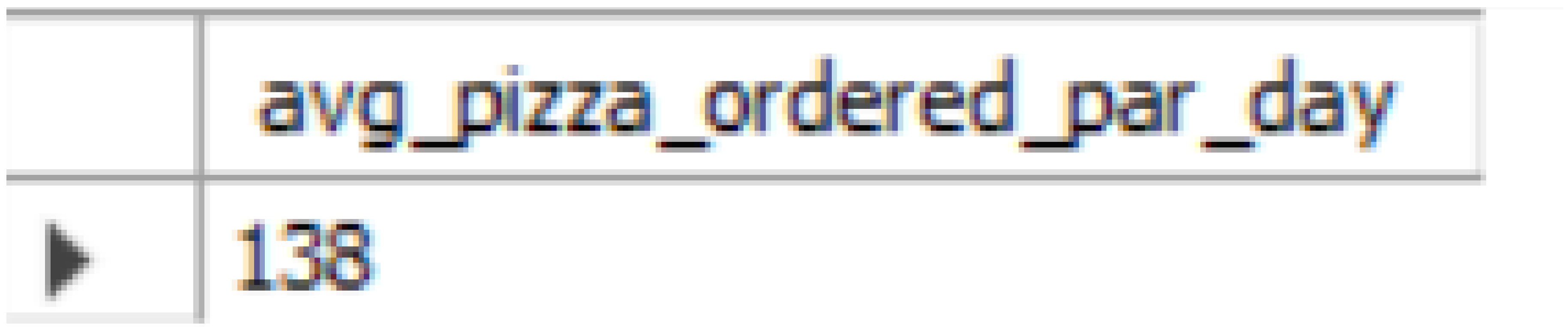


	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

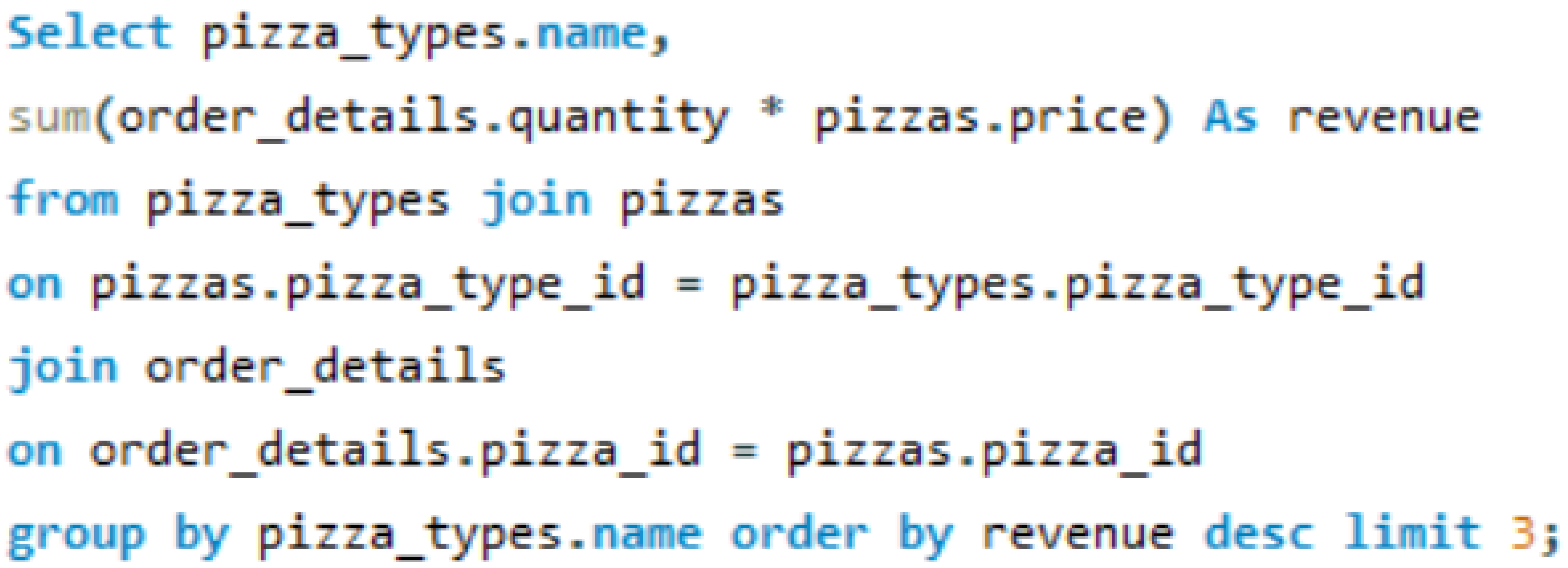


```
Select round(avg(quantity),0) as avg_pizza_ordered_par_day from
(Select orders.order_date, Sum(order_details.quantity) as quantity
from orders join  order_details
on orders.order_id = order_details.order_id
group by orders.order_date ) as order_quantity
```

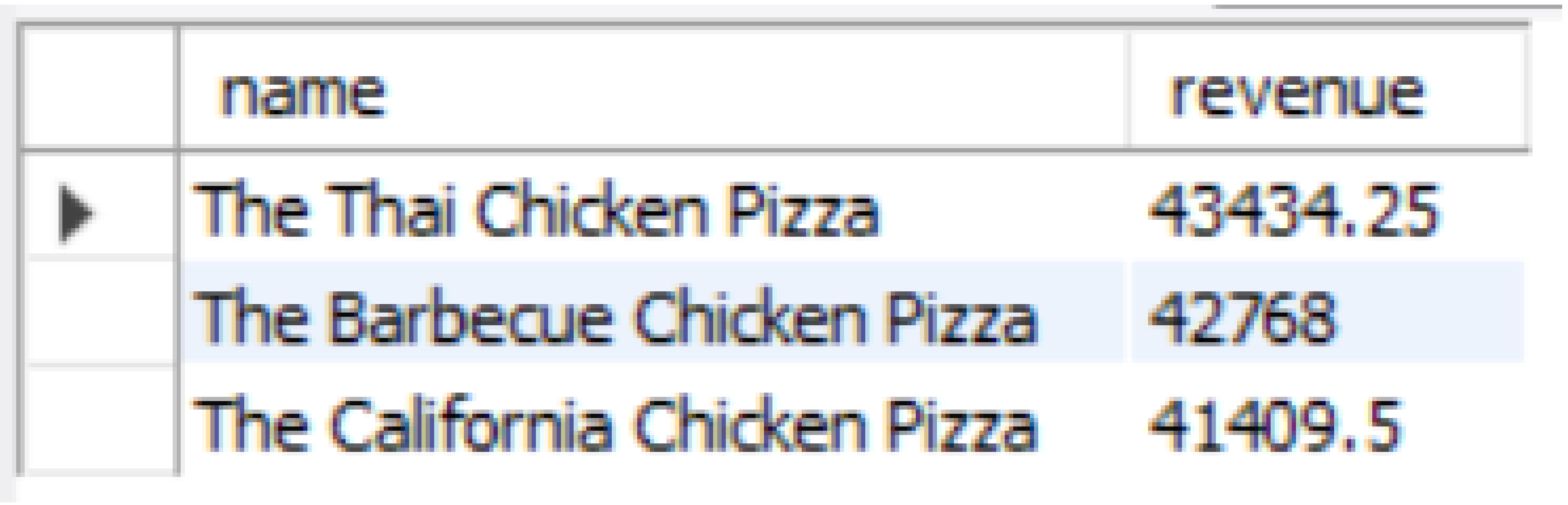


	avg_pizza_ordered_par_day
▶	138

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

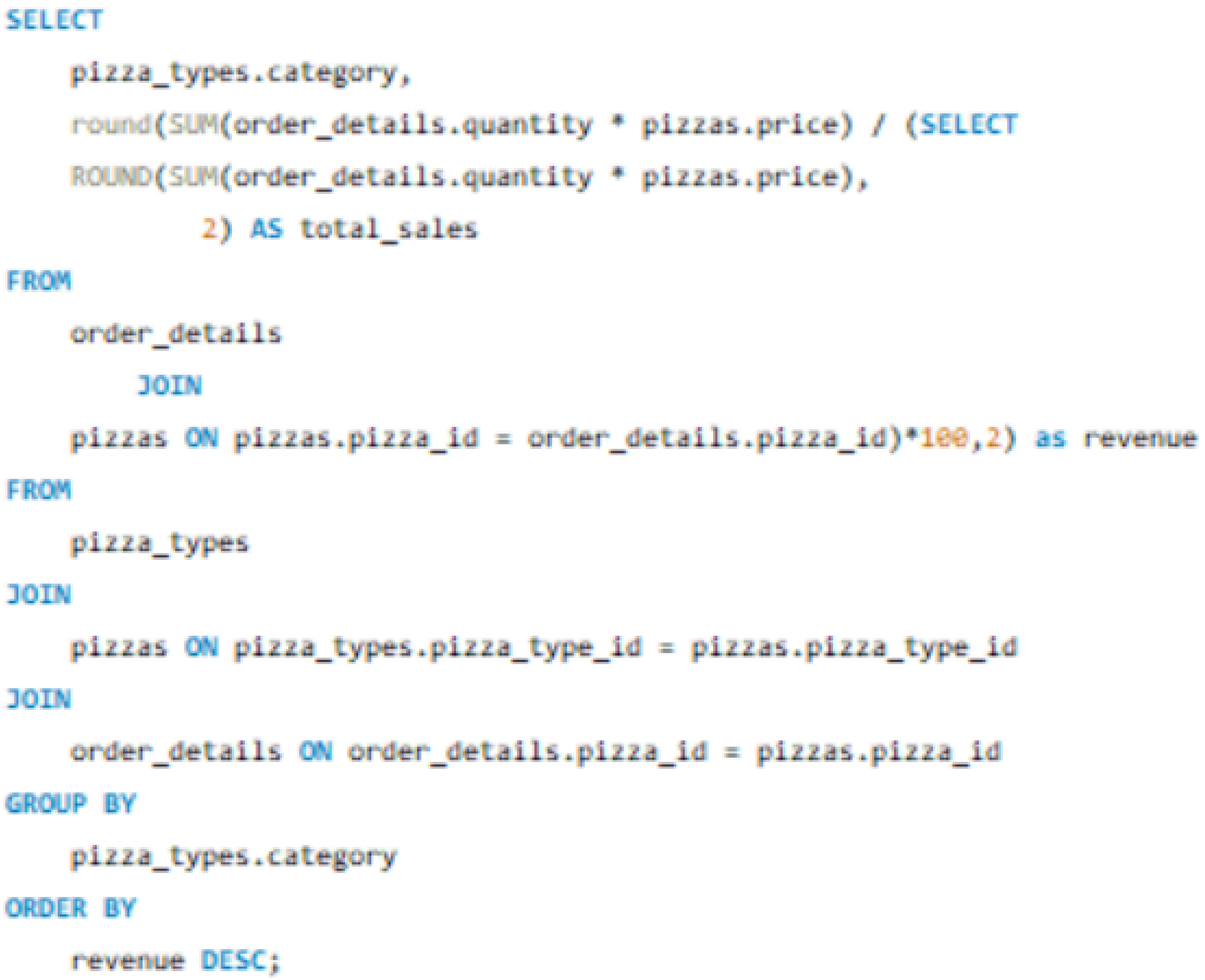


```
Select pizza_types.name,
sum(order_details.quantity * pizzas.price) As revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

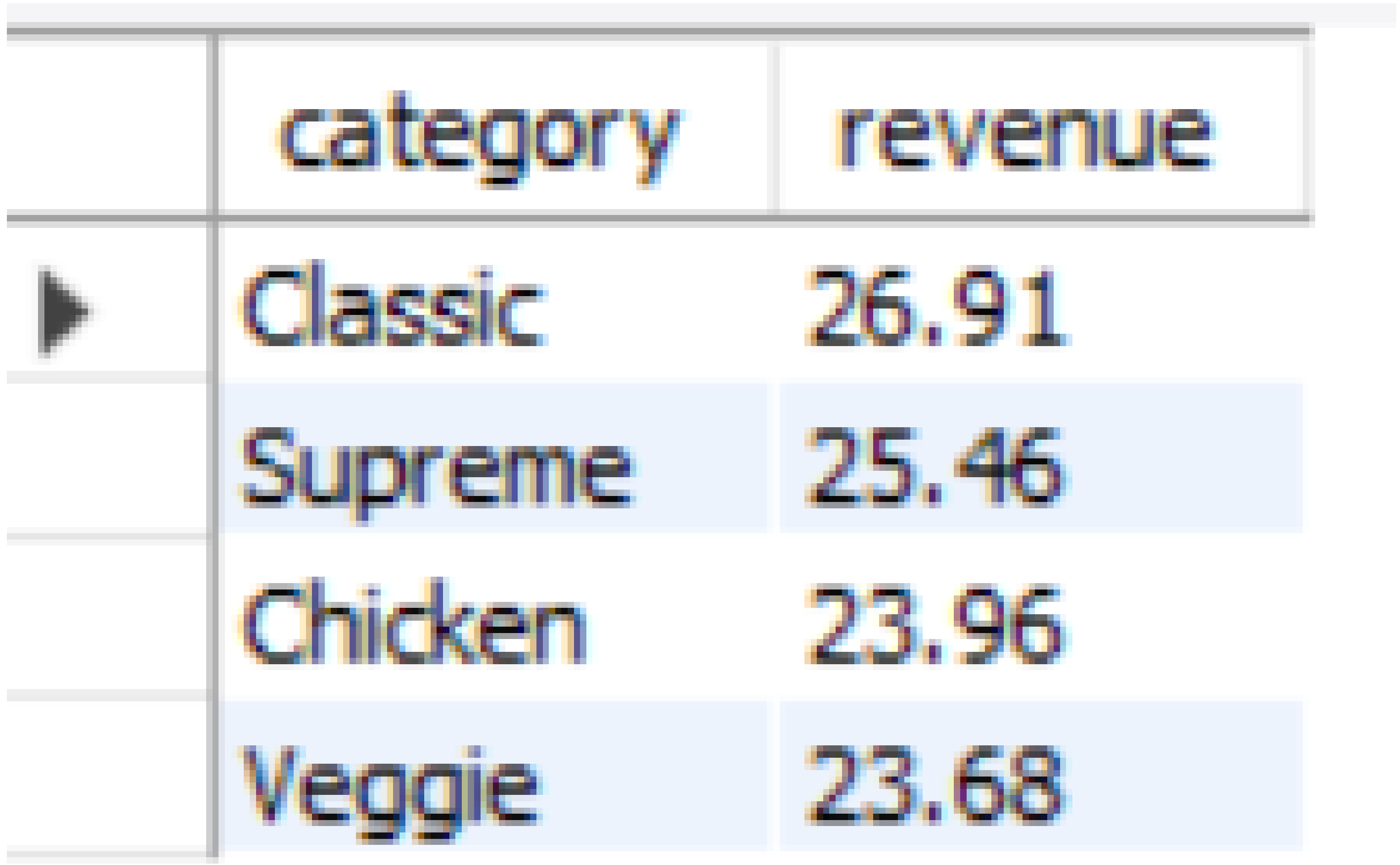


	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

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

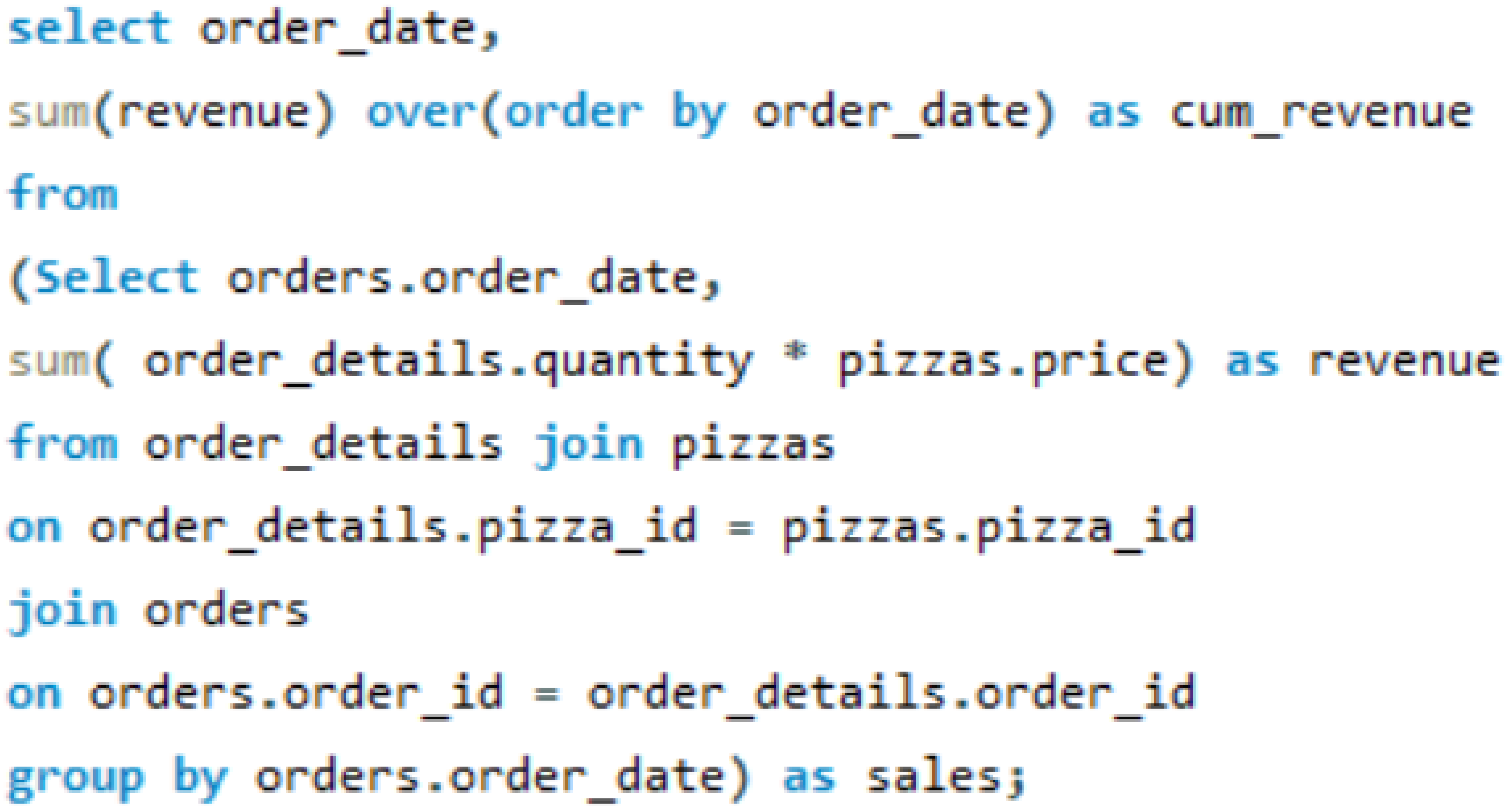


```
SELECT
    pizza_types.category,
    round(SUM(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id)*100,2) as revenue
FROM
    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;
```

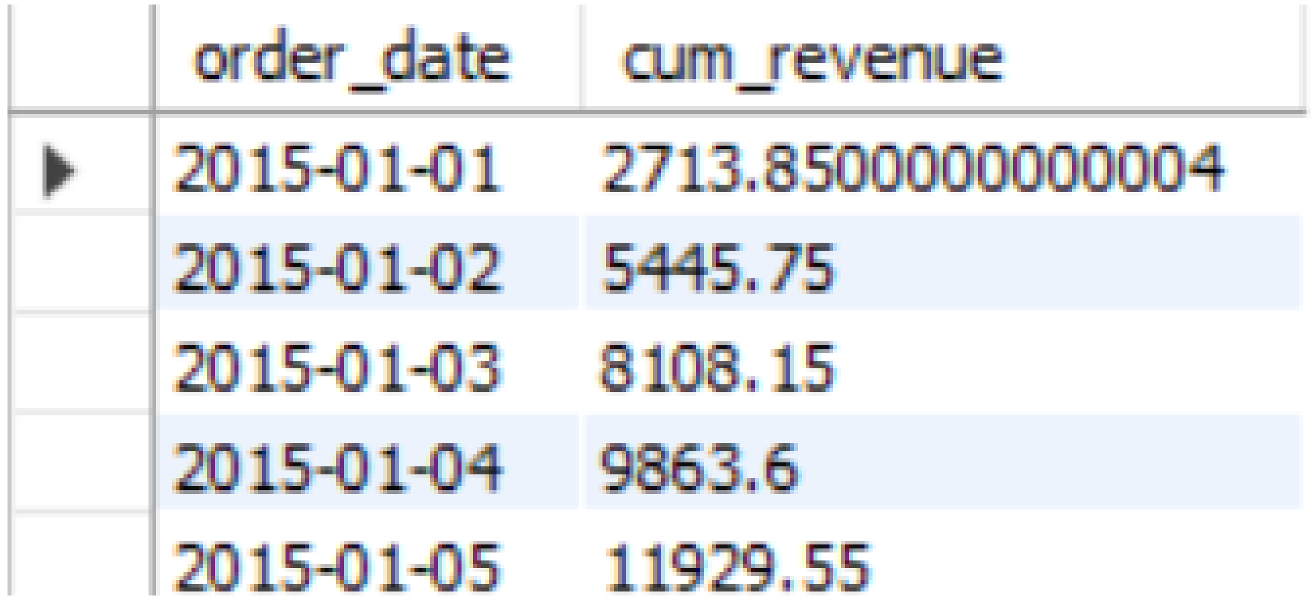


	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

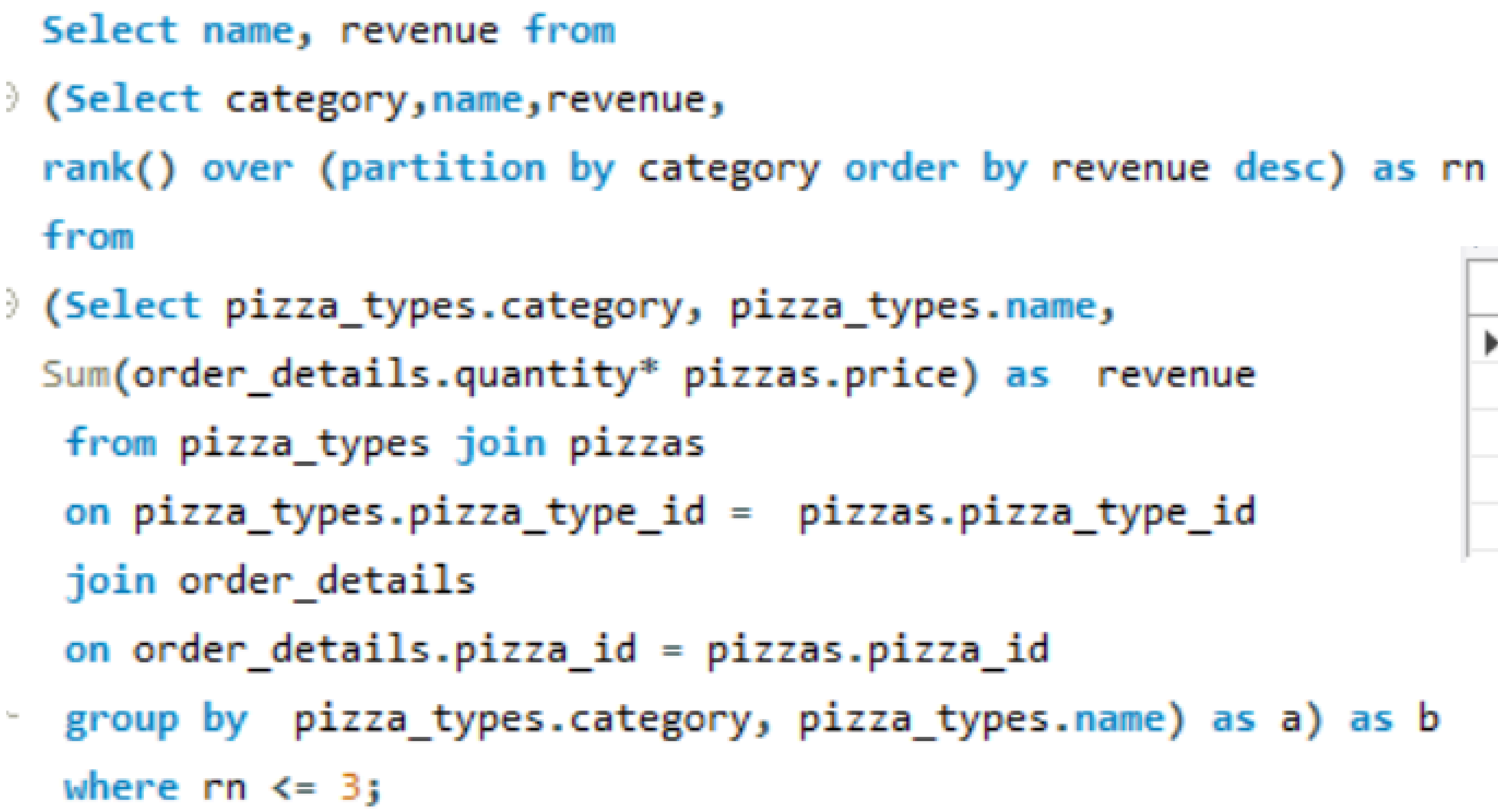


```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(Select orders.order_date,
sum( order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group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

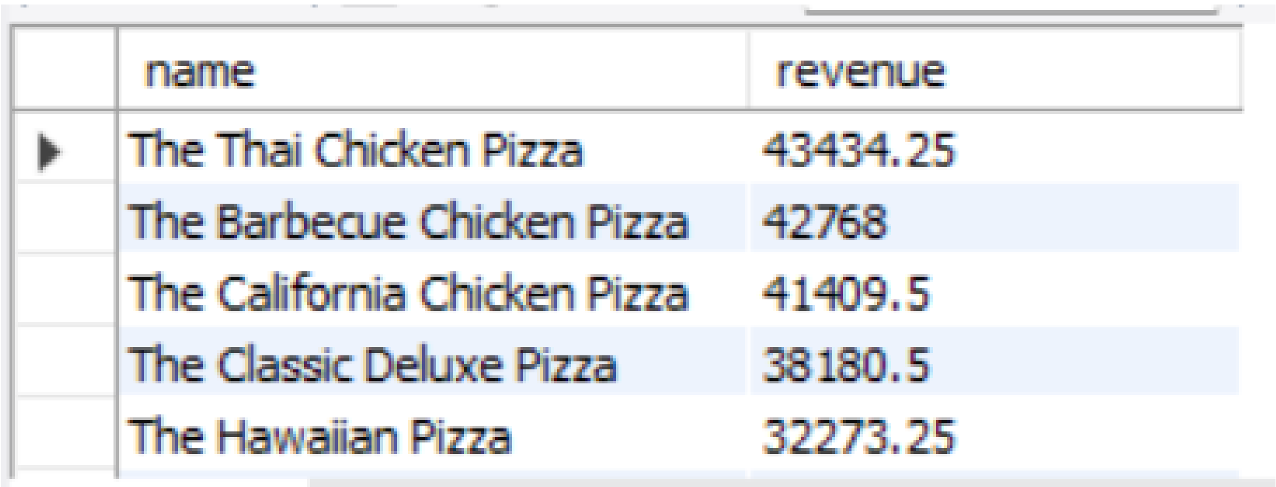


	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

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



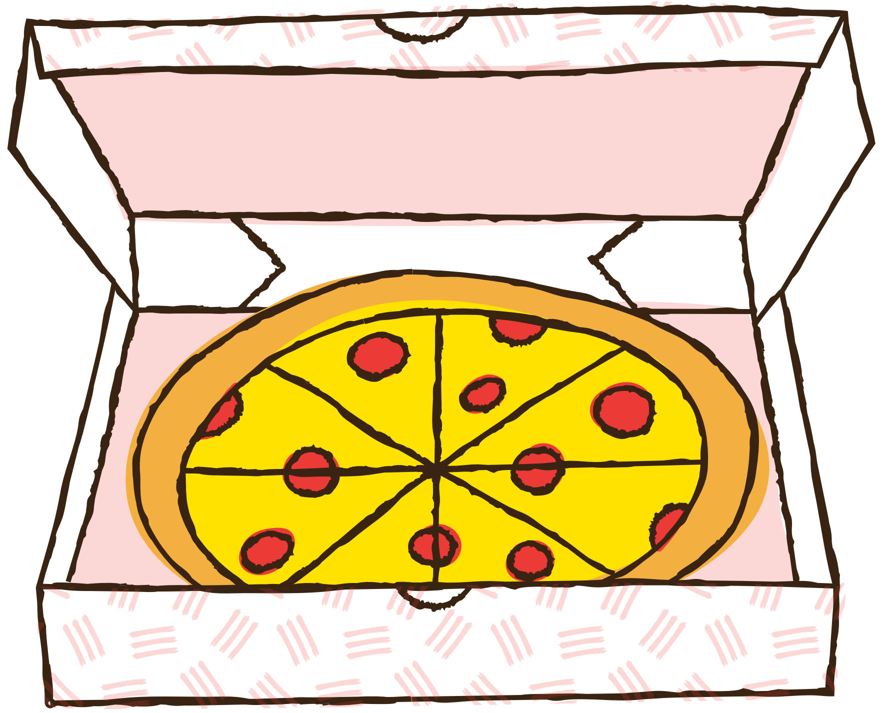
```
Select name, revenue from
(Select category,name,revenue,
rank() over (partition by category order by revenue desc) as rn
from
(Select pizza_types.category, pizza_types.name,
Sum(order_details.quantity* pizzas.price) as  revenue
 from pizza_types join pizzas
 on pizza_types.pizza_type_id =  pizzas.pizza_type_id
 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;
```

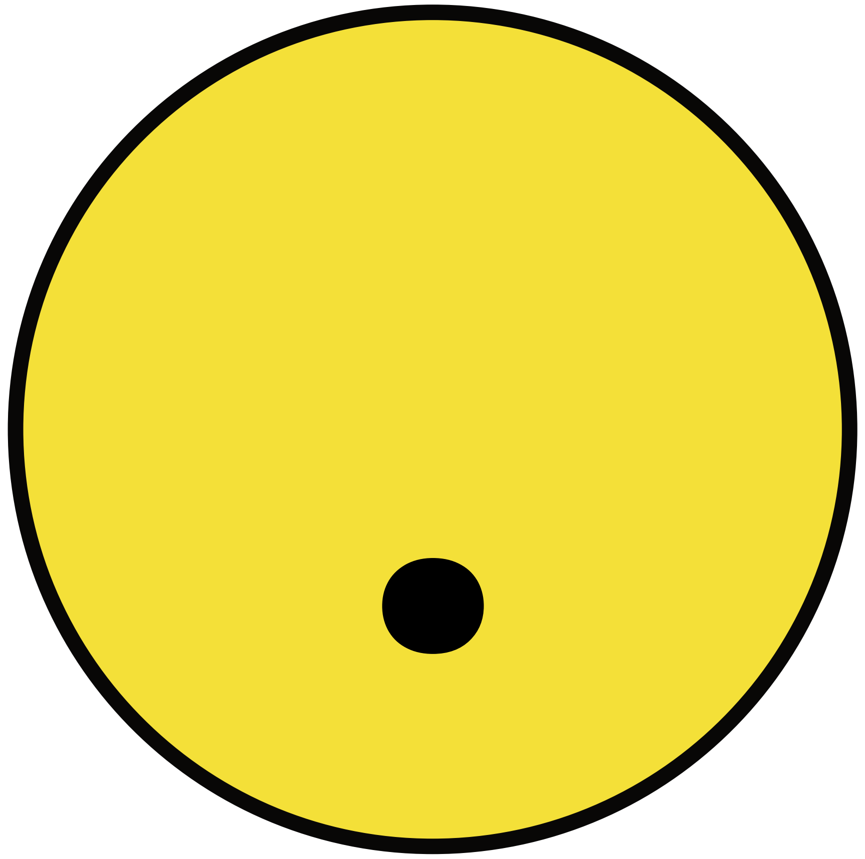


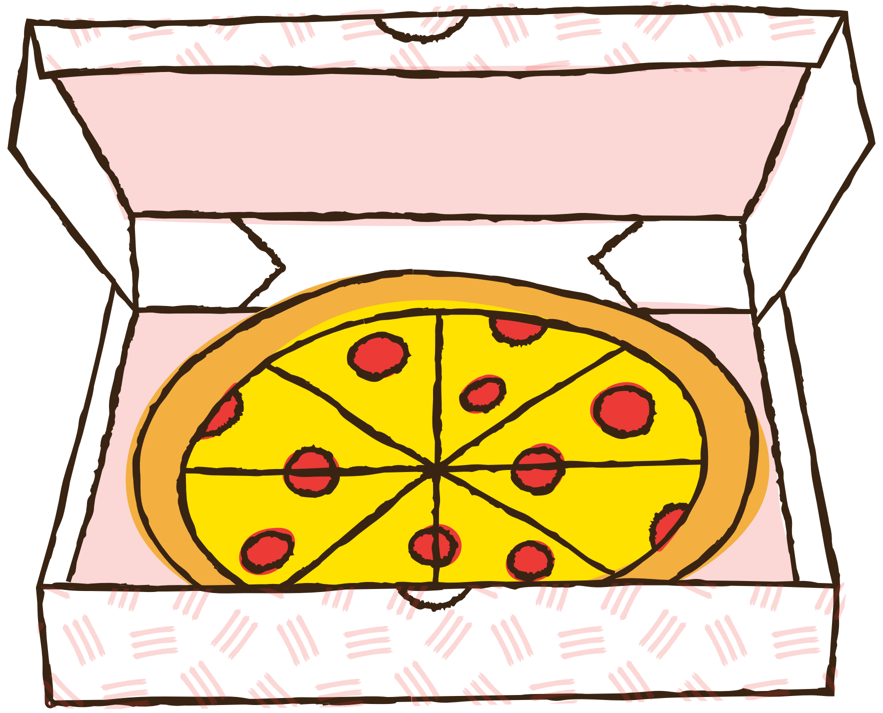
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

Conclusion.

- Improve inventory management.
- Tailor marketing strategies.
- Enhance customer experience.
- Optimize operational efficiency.







Thank you.