SQL - project on pizza sales

HELLO !!

my name is om salunkhe , i have utilizes sql query to solve questions that are releated to the pizza sales.

problem statement

1. *Retrieve the total number of orders placed.*
2. *Calculate the total revenue generated from pizza sales.*
3. *Identify the highest-priced pizza.*
4. *Identify the most common pizza size ordered.*
5. *List the top 5 most ordered pizza types along with their quantities.*

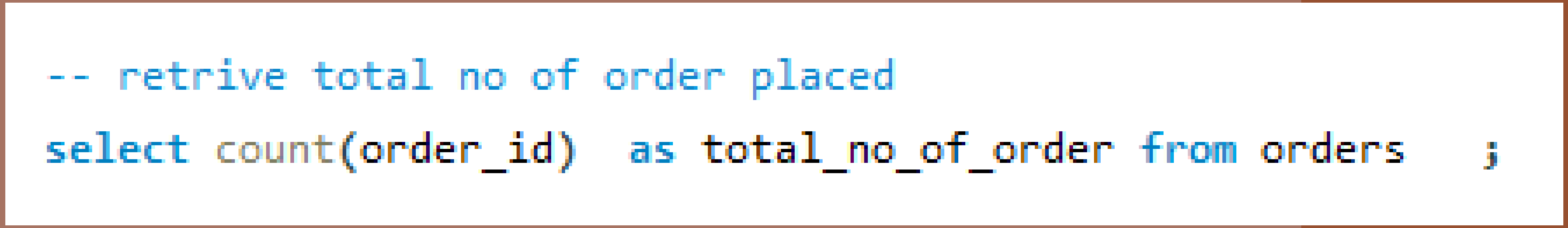
problem statement

6.Join the necessary tables to find the total quantity of each pizza category ordered.
7.Determine the distribution of orders by hour of the day.
8.Join relevant tables to find the category-wise distribution of pizzas.
9.Group the orders by date and calculate the average number of pizzas ordered per day.
10.Determine the top 3 most ordered pizza types based on revenue

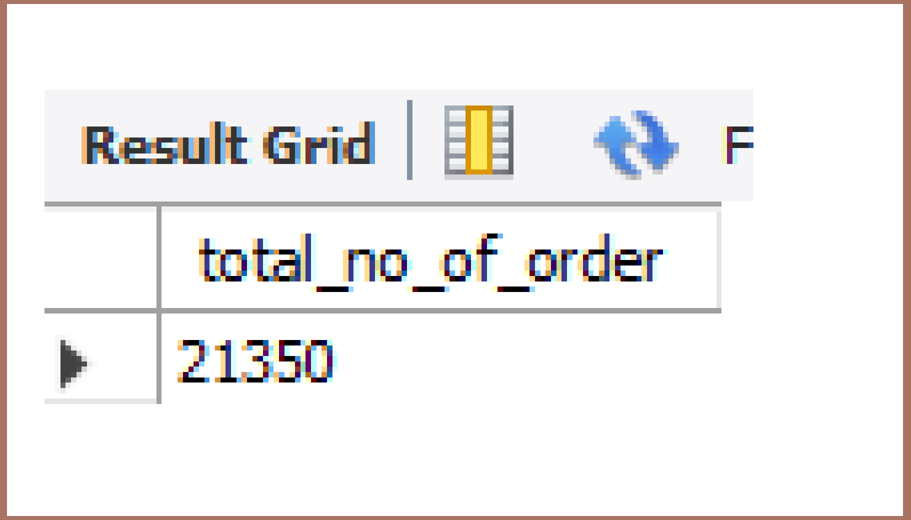
problem statement

11.Calculate the percentage contribution of each pizza type to total revenue.
12.Analyze the cumulative revenue generated over time.
13.Determine the top 3 most ordered pizza types based on revenue for each pizza category.

1.Retrieve the total number of orders placed.

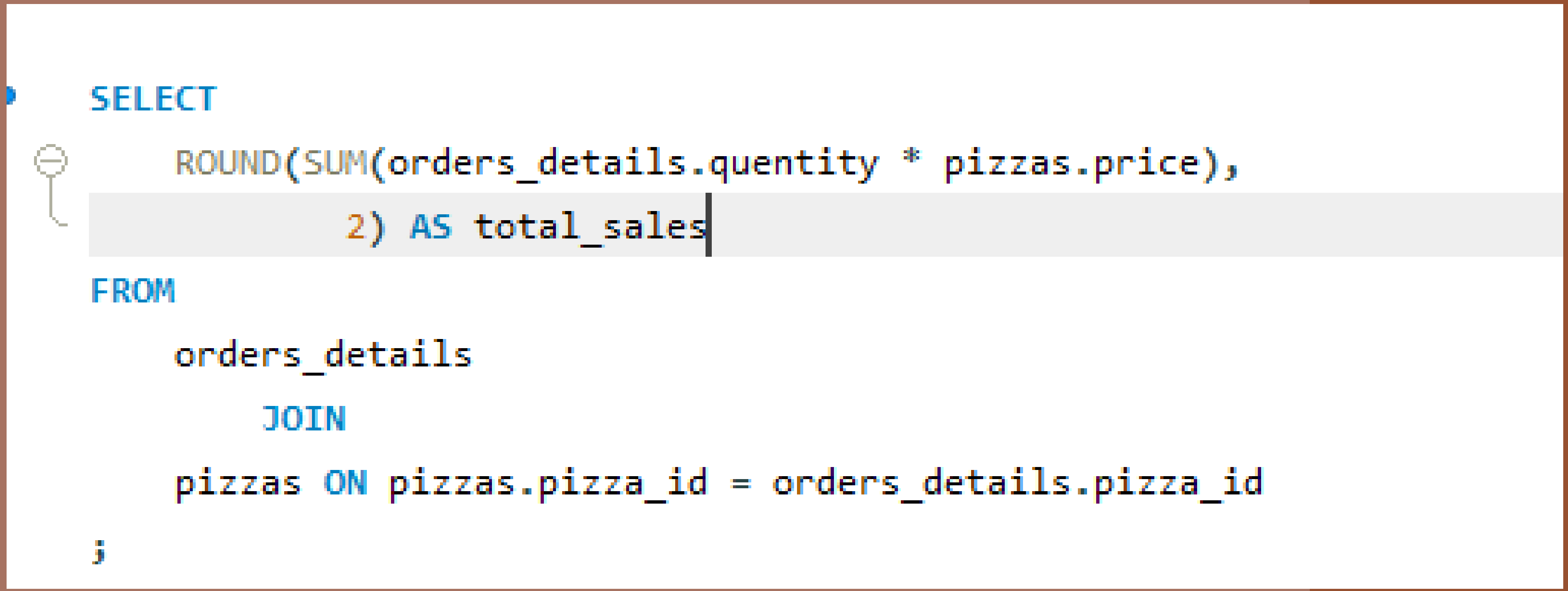


```sql
-- retrive total no of order placed
select count(order_id)  as total_no_of_order from orders   ;
```

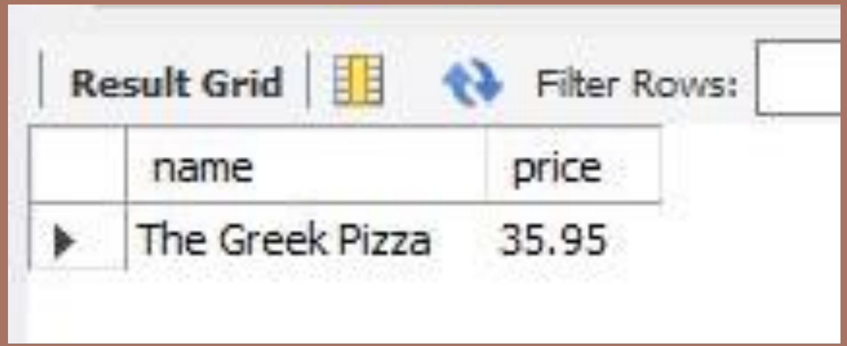


total_no_of_order
21350

2.Calculate the total revenue generated from pizza sales.

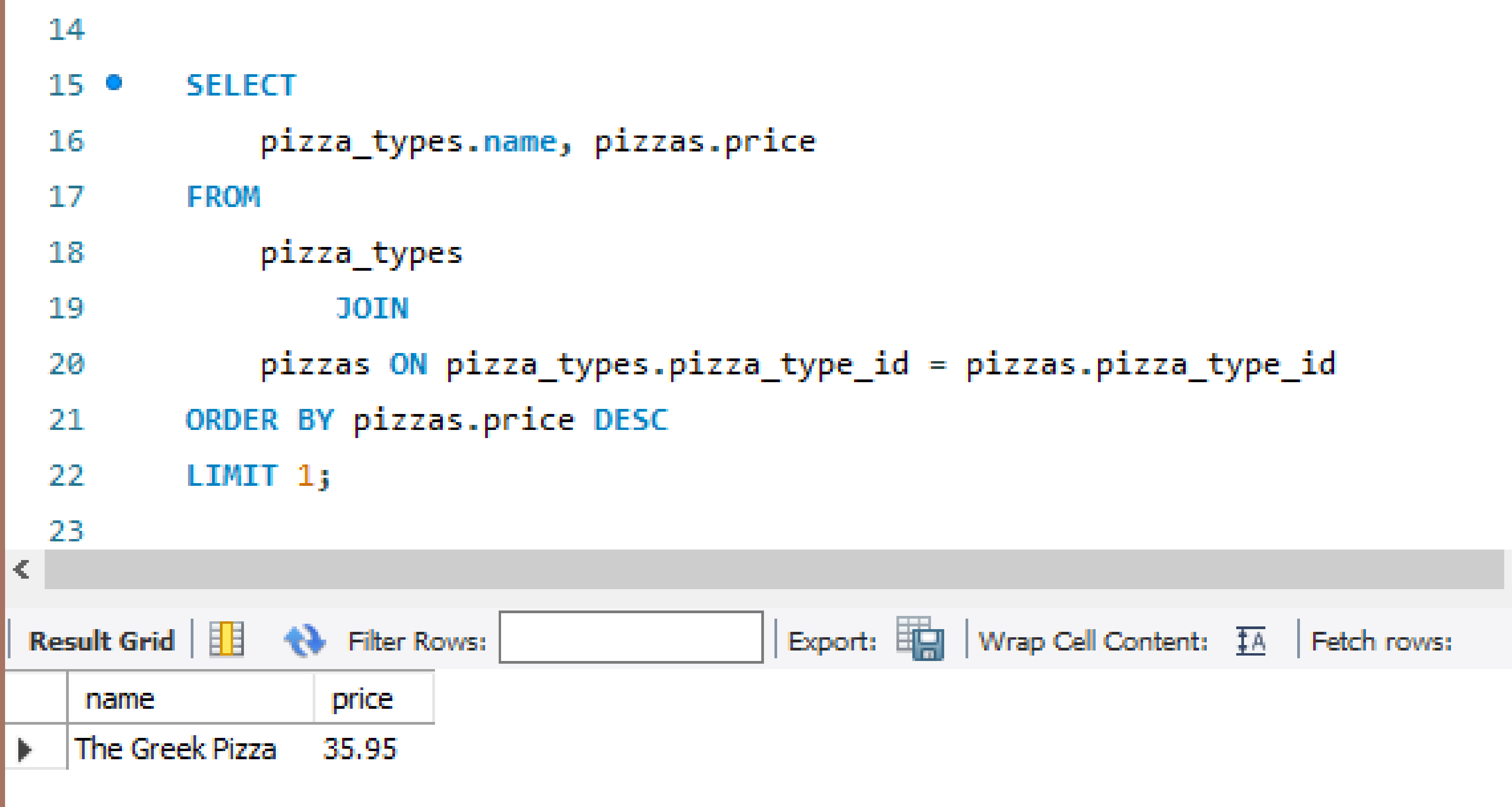


```
SELECT
    ROUND(SUM(orders_details.quentity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
;
```



name	price
The Greek Pizza	35.95

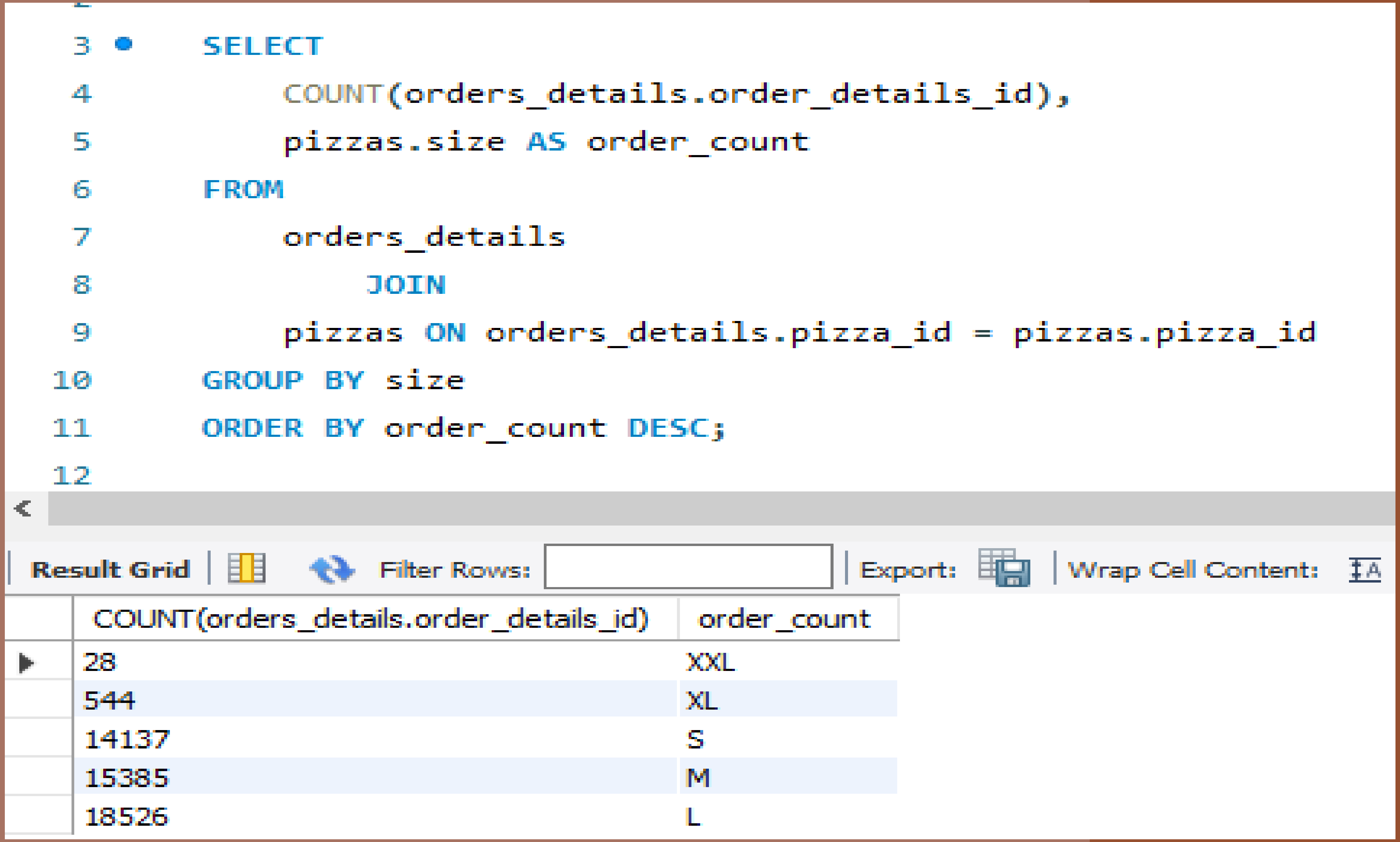
3.Identify the highest-priced pizza.



```
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

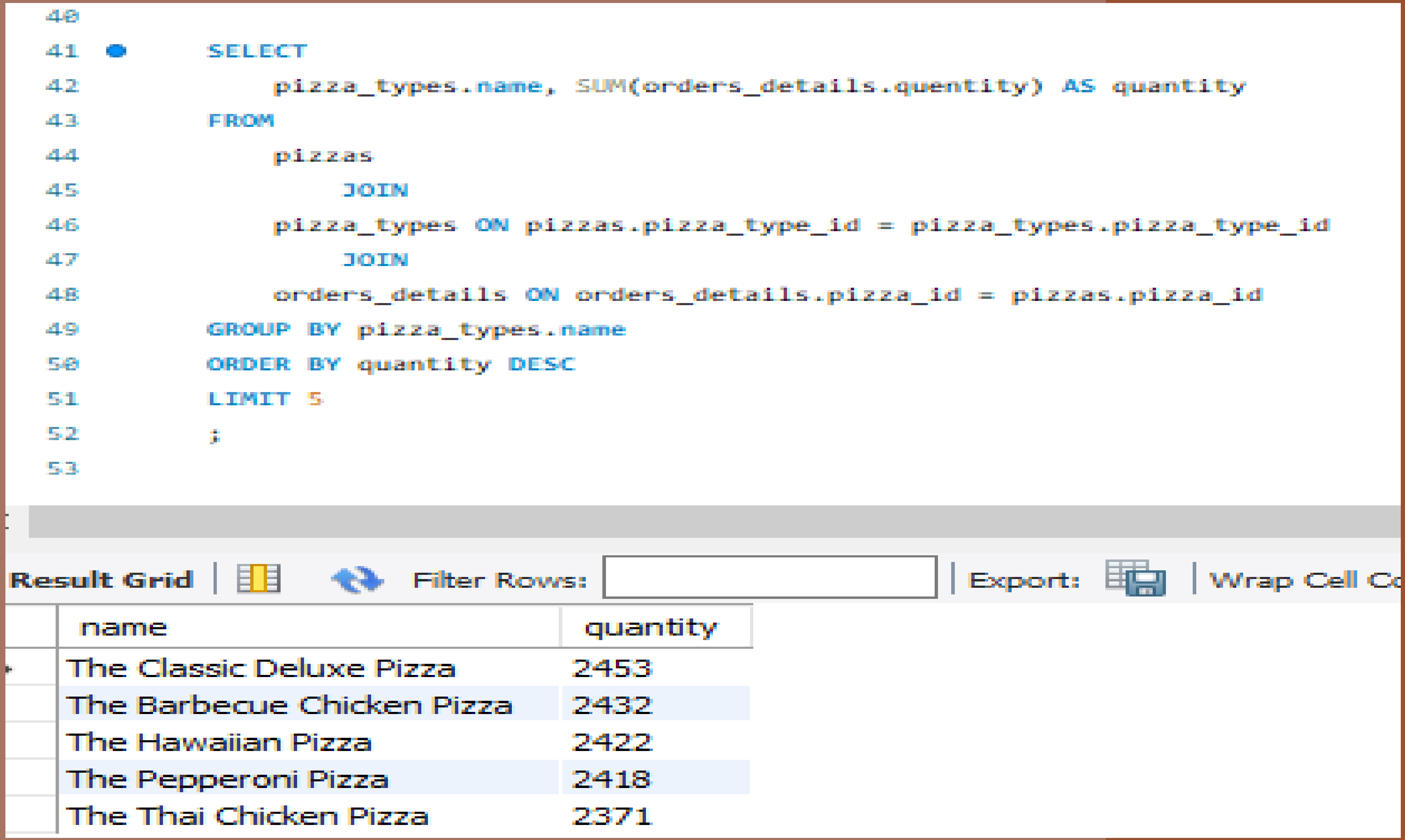
4.Identify the most common pizza size ordered.



```sql
SELECT
    COUNT(orders_details.order_details_id),
    pizzas.size AS order_count
FROM
    orders_details
        JOIN
    pizzas ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY size
ORDER BY order_count DESC;
```

COUNT(orders_details.order_details_id)	order_count
28	XXL
544	XL
14137	S
15385	M
18526	L

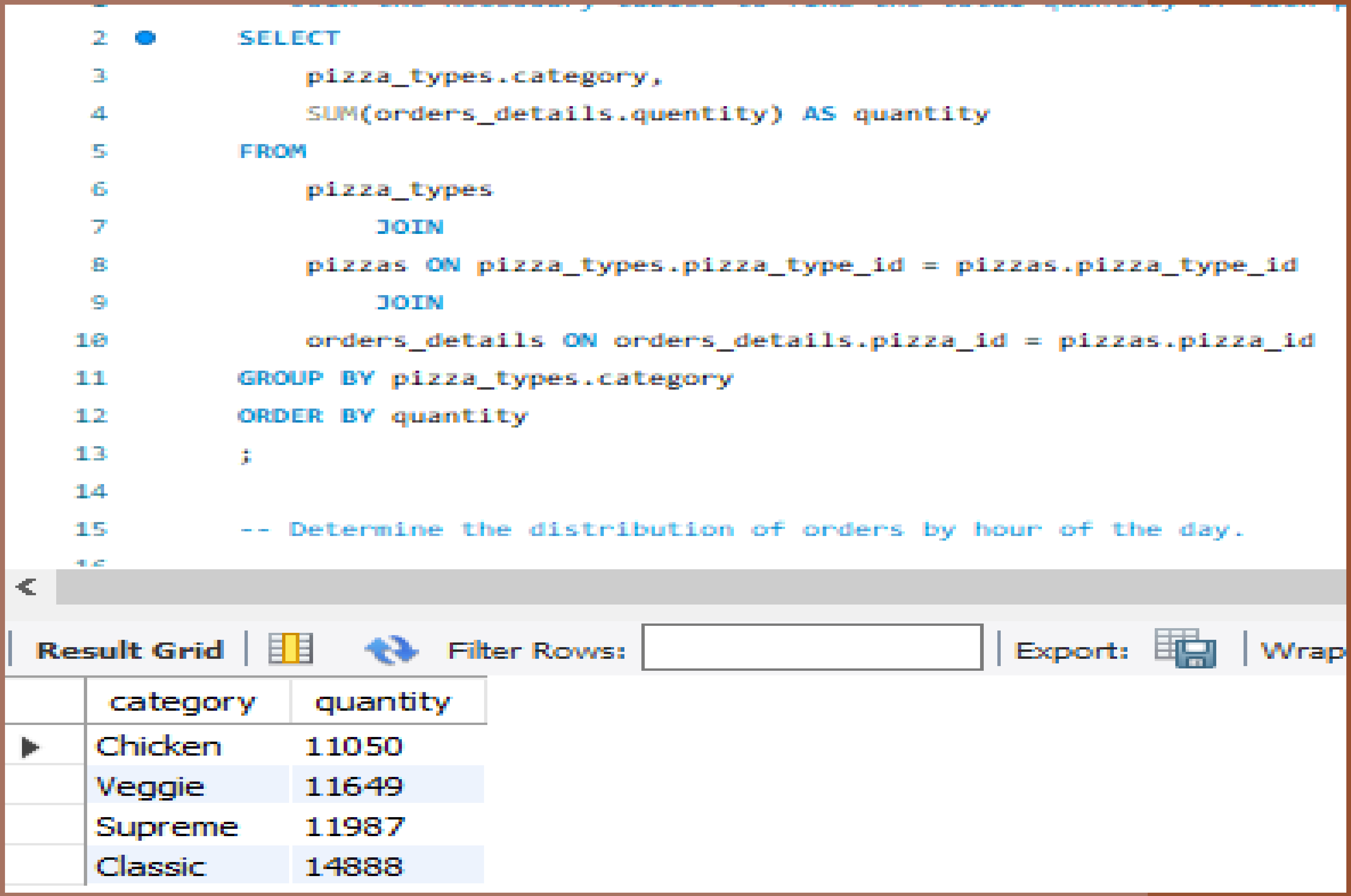
5.List the top 5 most ordered pizza types along with their quantities.



```
SELECT
    pizza_types.name, SUM(orders_details.quentity) AS quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
;
```

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

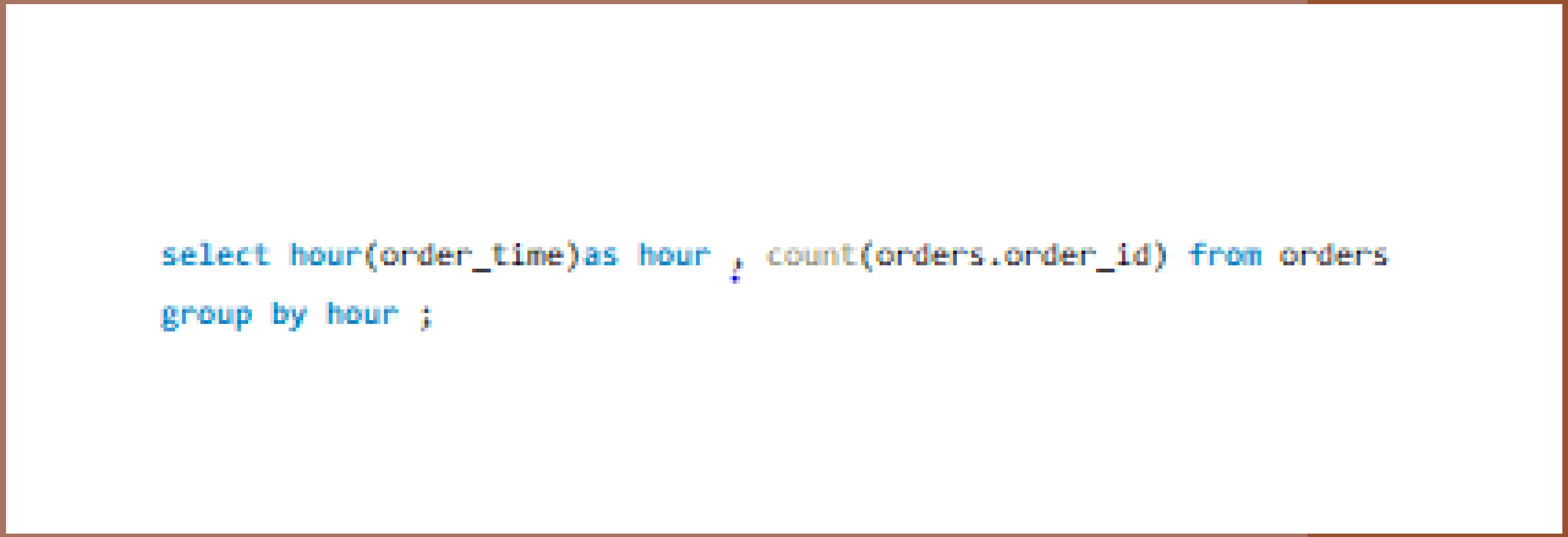
6.Join the necessary tables to find the total quantity of each pizza category ordered.



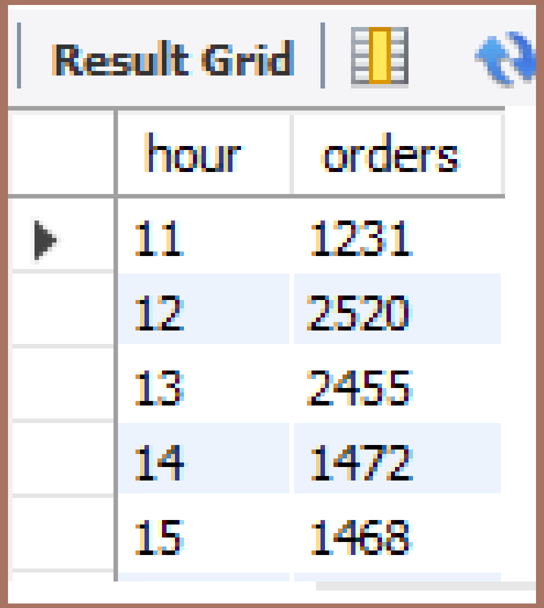
```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quentity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity
;

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

category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

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

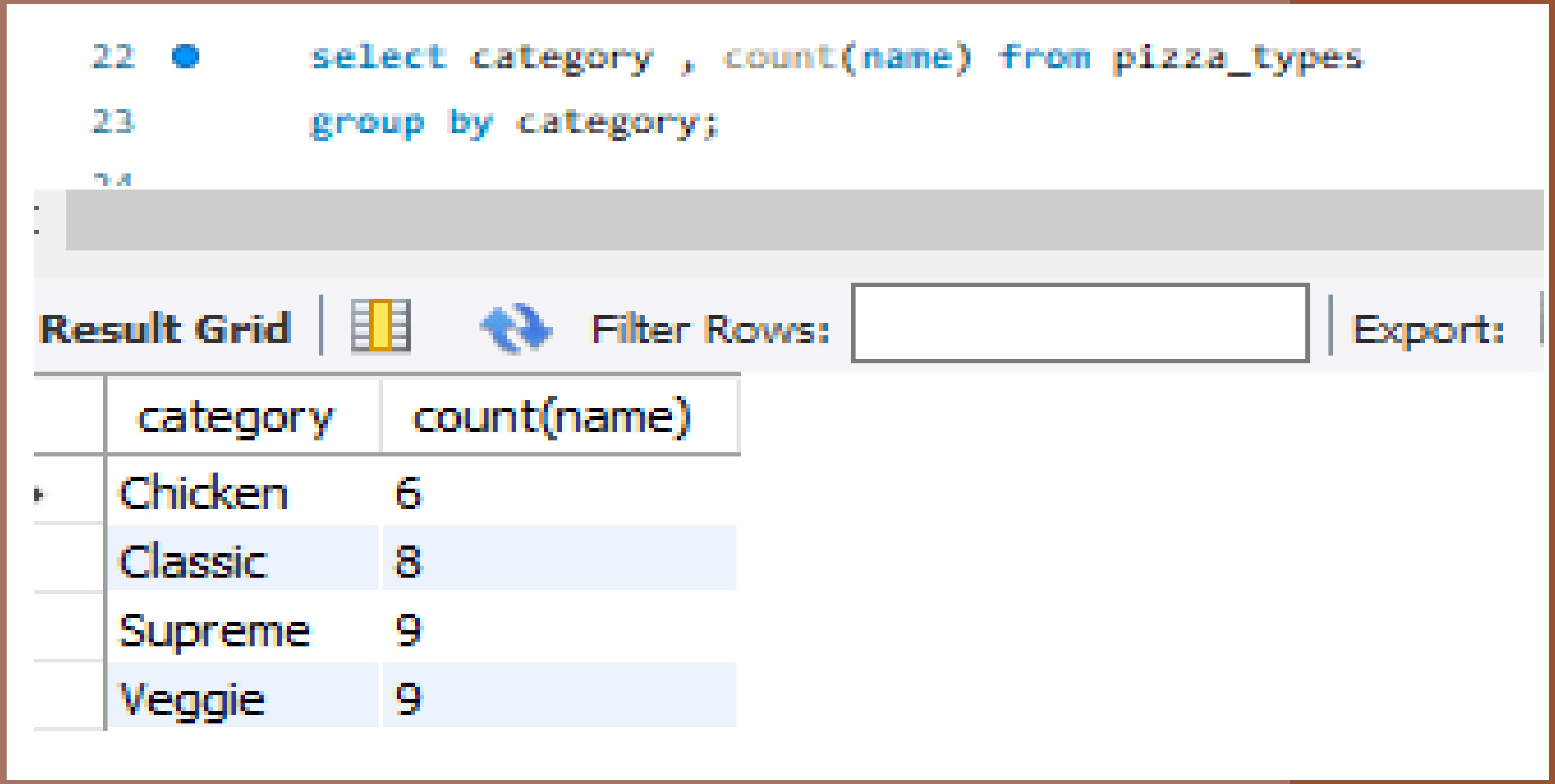


```
select hour(order_time)as hour , count(orders.order_id) from orders
group by hour ;
```



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

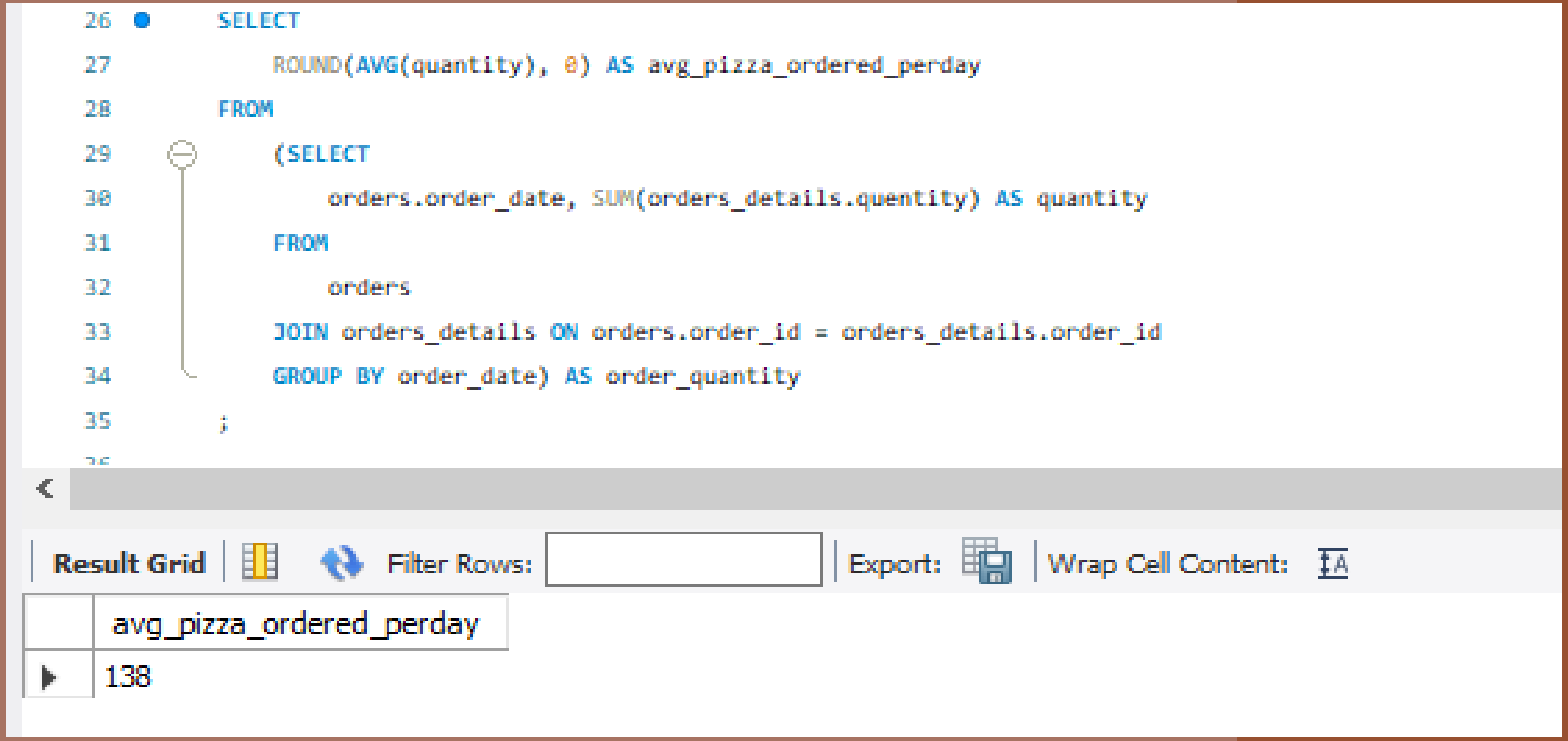
8.Join relevant tables to find the category-wise distribution of pizzas.



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

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

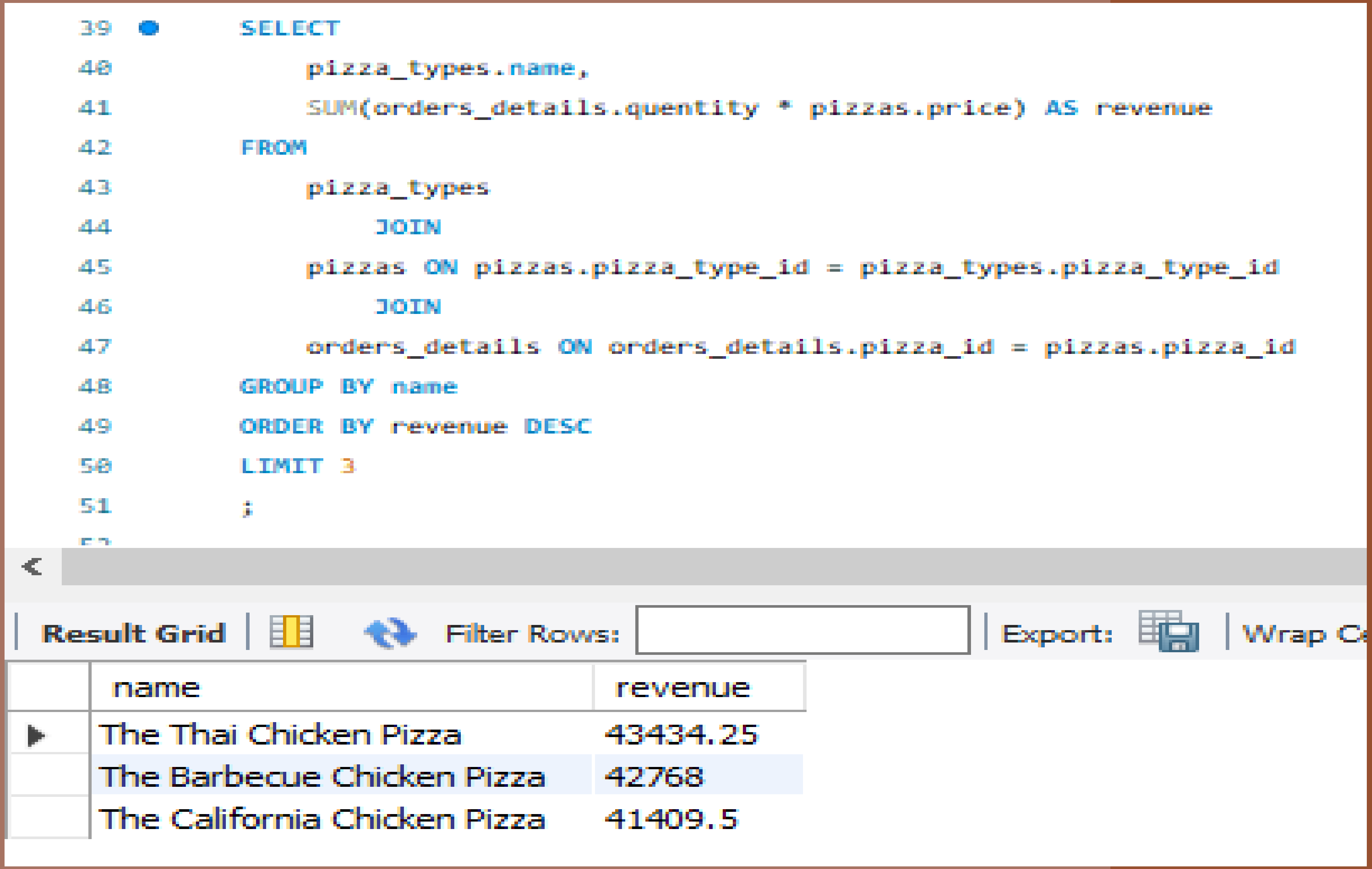
9.Group the orders by date and calculate the average number of pizzas ordered per day.



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

avg_pizza_ordered_perday
138

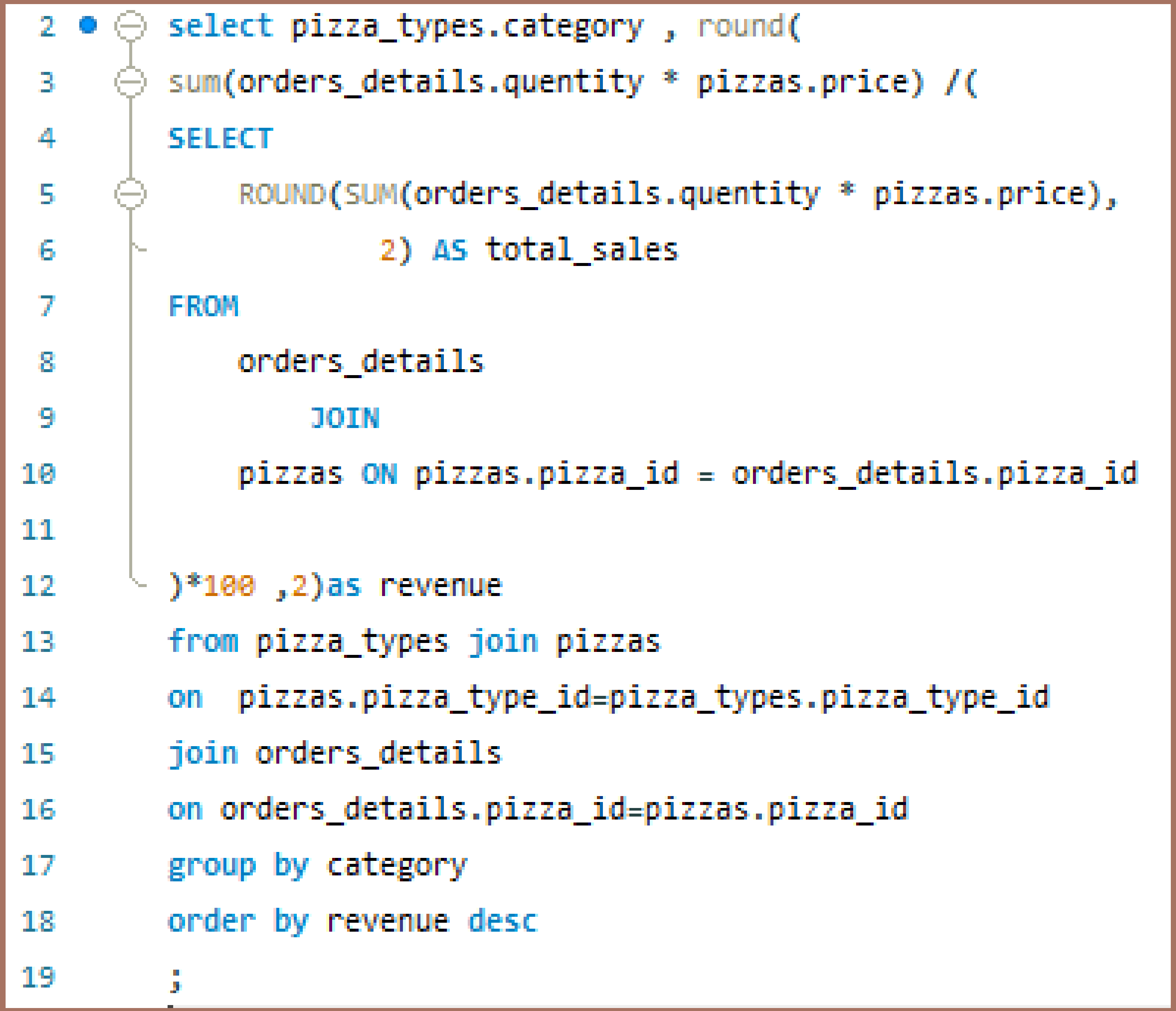
10.Determine the top 3 most ordered pizza types based on revenue



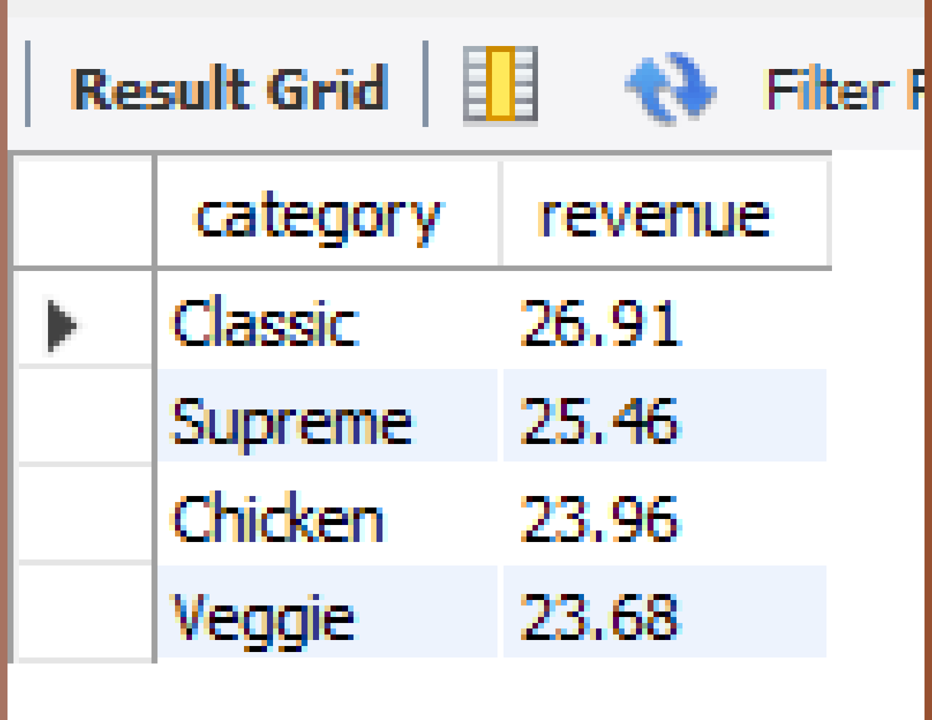
```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quentity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY name
ORDER BY revenue DESC
LIMIT 3
;
```

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

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

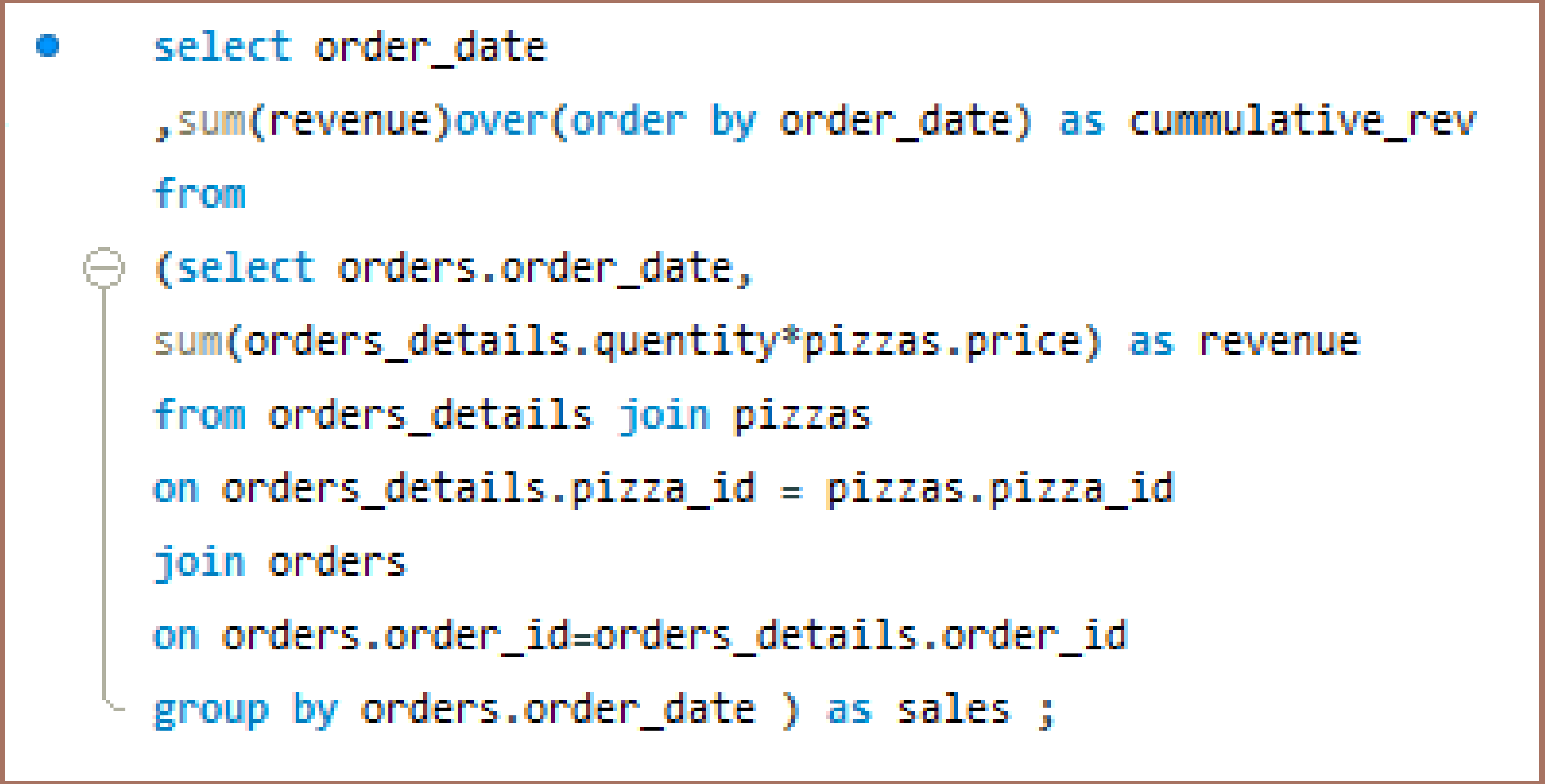


```
select pizza_types.category , round(
sum(orders_details.quentity * pizzas.price) /(
SELECT
    ROUND(SUM(orders_details.quentity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id

)*100 ,2)as revenue
from pizza_types join pizzas
on  pizzas.pizza_type_id=pizza_types.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by category
order by revenue desc
;
```

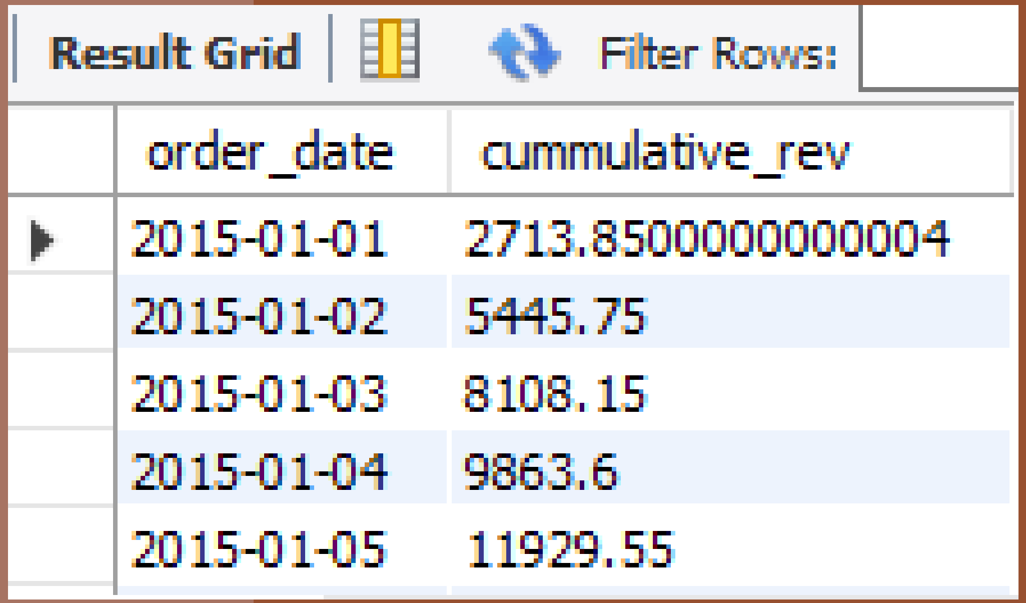


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12.Analyze the cumulative revenue generated over time.

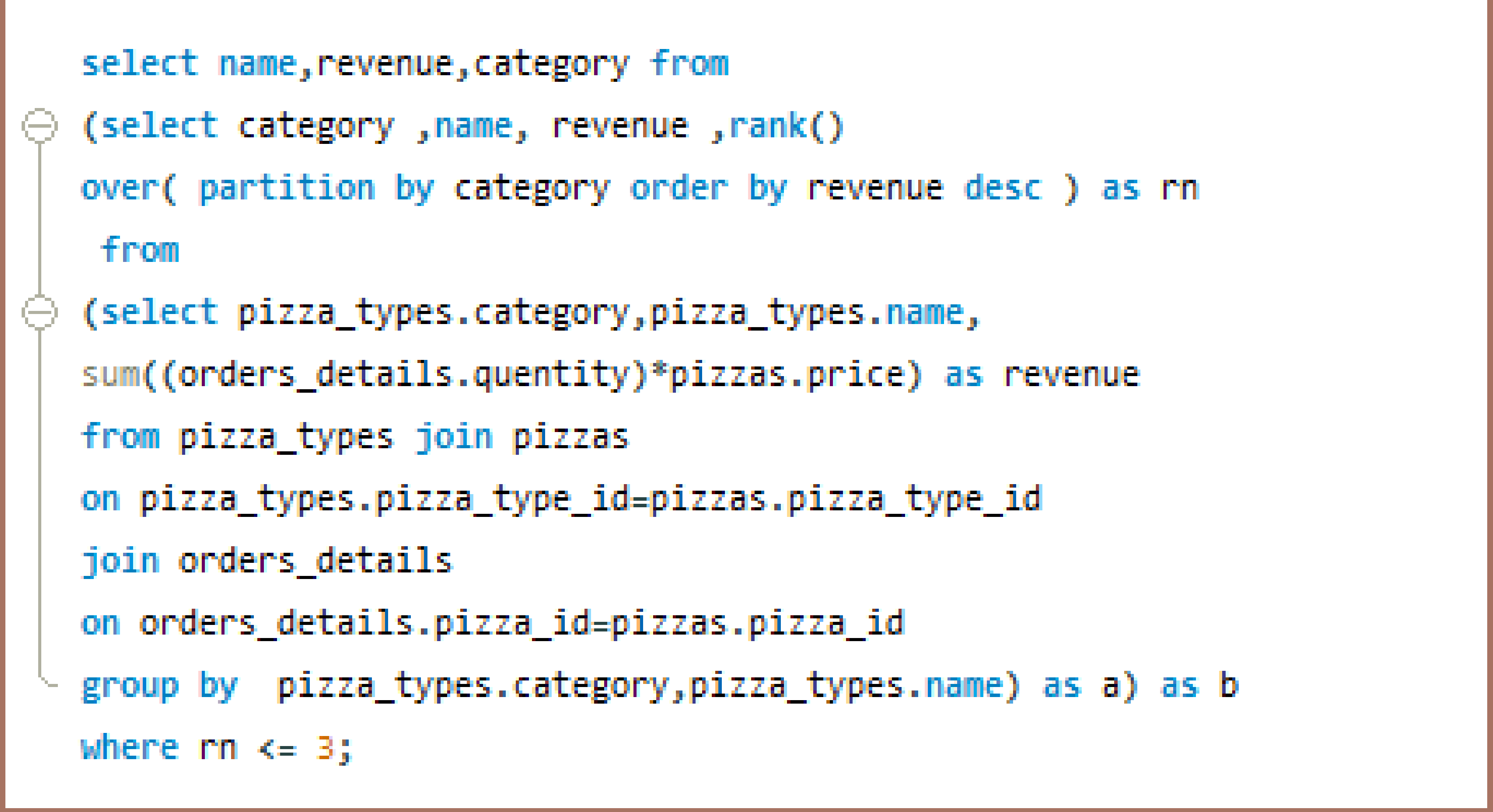


```
select order_date
,sum(revenue)over(order by order_date) as cummulative_rev
from
(select orders.order_date,
sum(orders_details.quentity*pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id=orders_details.order_id
group by orders.order_date ) as sales ;
```

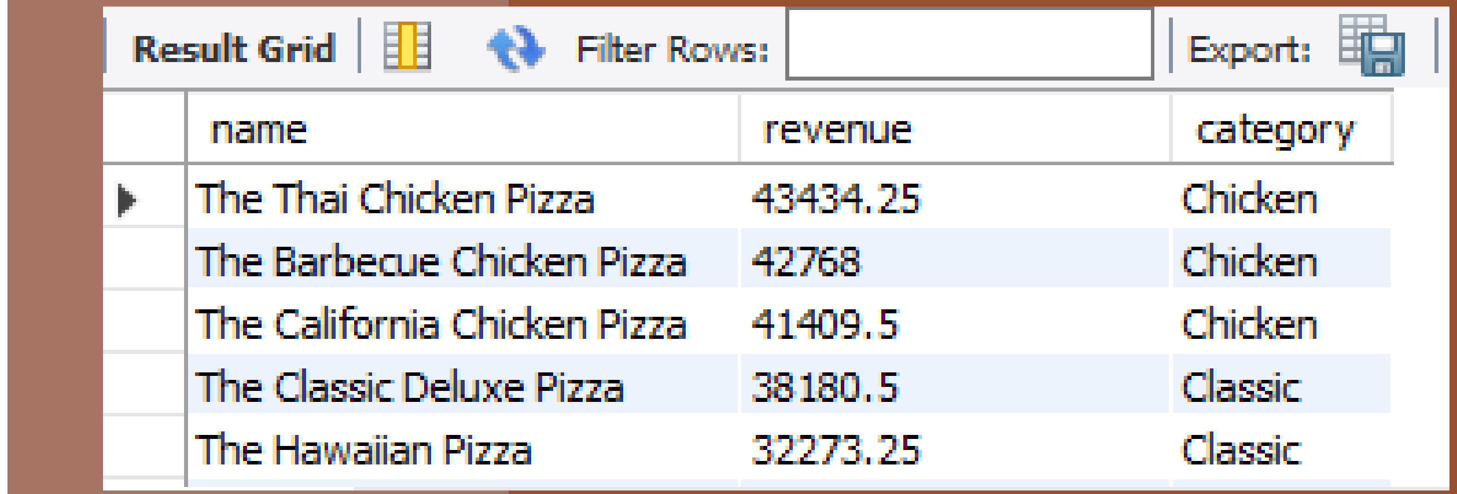


order_date	cummulative_rev
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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



```sql
select name,revenue,category 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((orders_details.quentity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by  pizza_types.category,pizza_types.name) as a) as b
where rn <= 3;
```



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

THANK YOU

✉ salunkheomo2@gmail.com

☎ 9561983565