SQL Project Based on PIZZA Sales

ORDER NOW

+123-456-7890

HELLO !!

I AM KAMAL AND IN THIS PROJECT
I UTILIZED SQL QUERIES TO SOLVE THE
SALES BASED QUESTIONS .

DATA :- GITHUB

Next

QUESTIONS

Basic

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

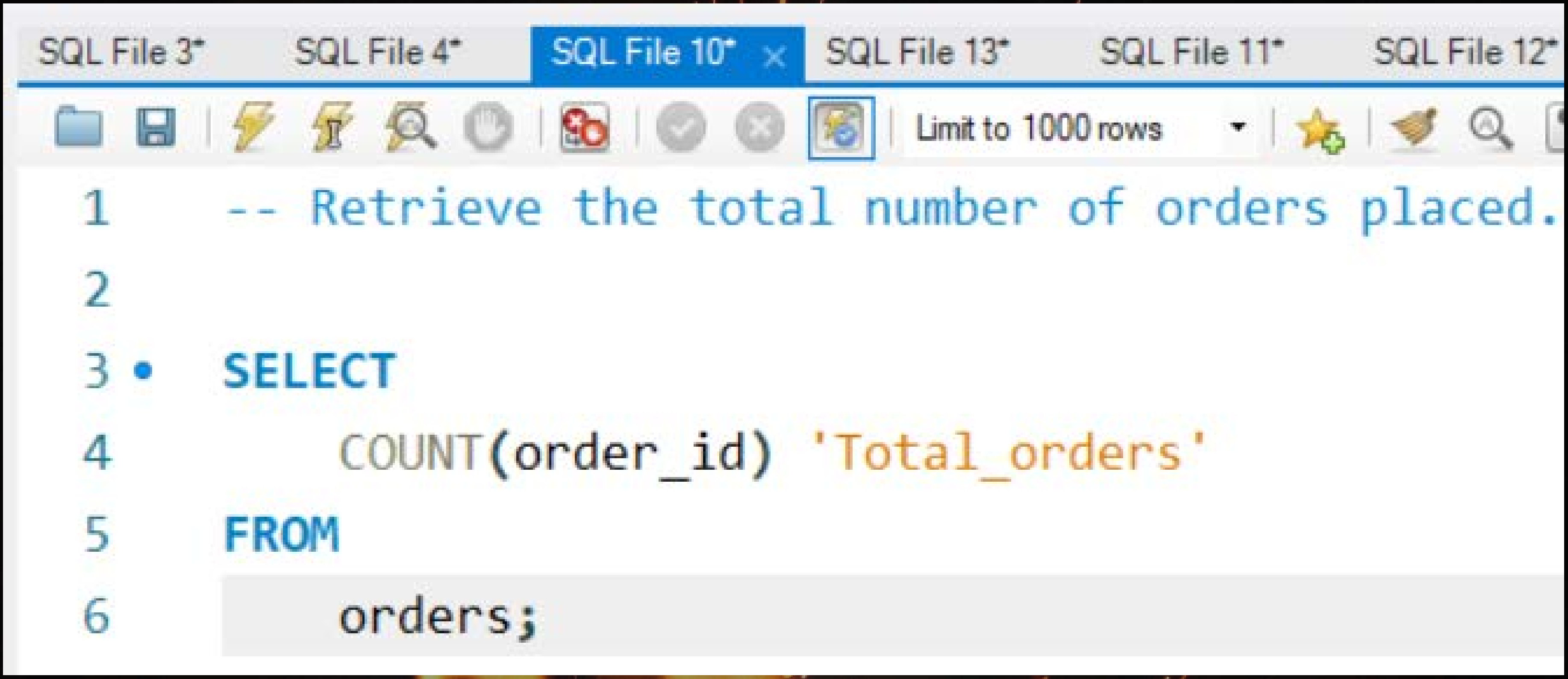
Intermediate

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

Advanced

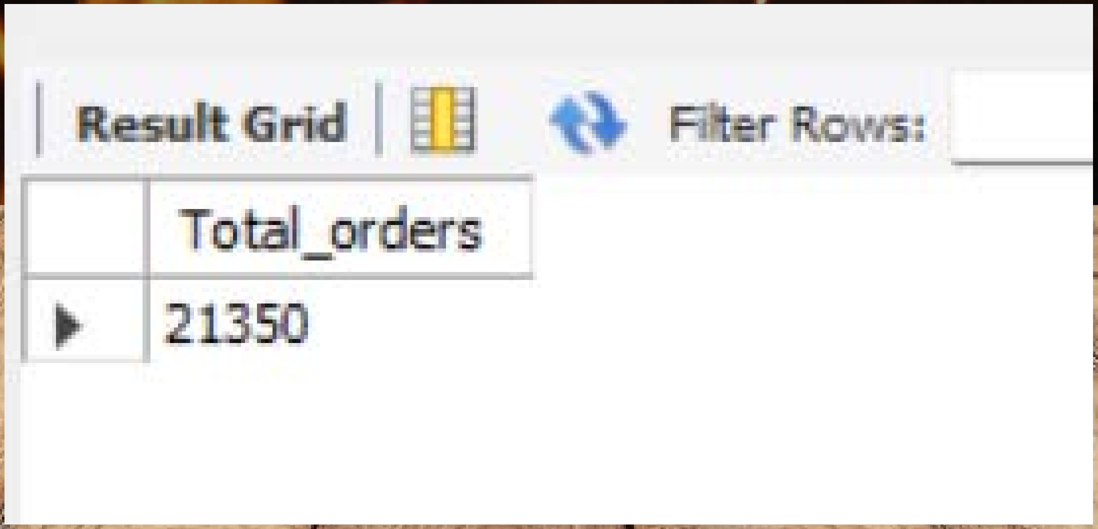
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

1. Retrieve the total number of orders placed.



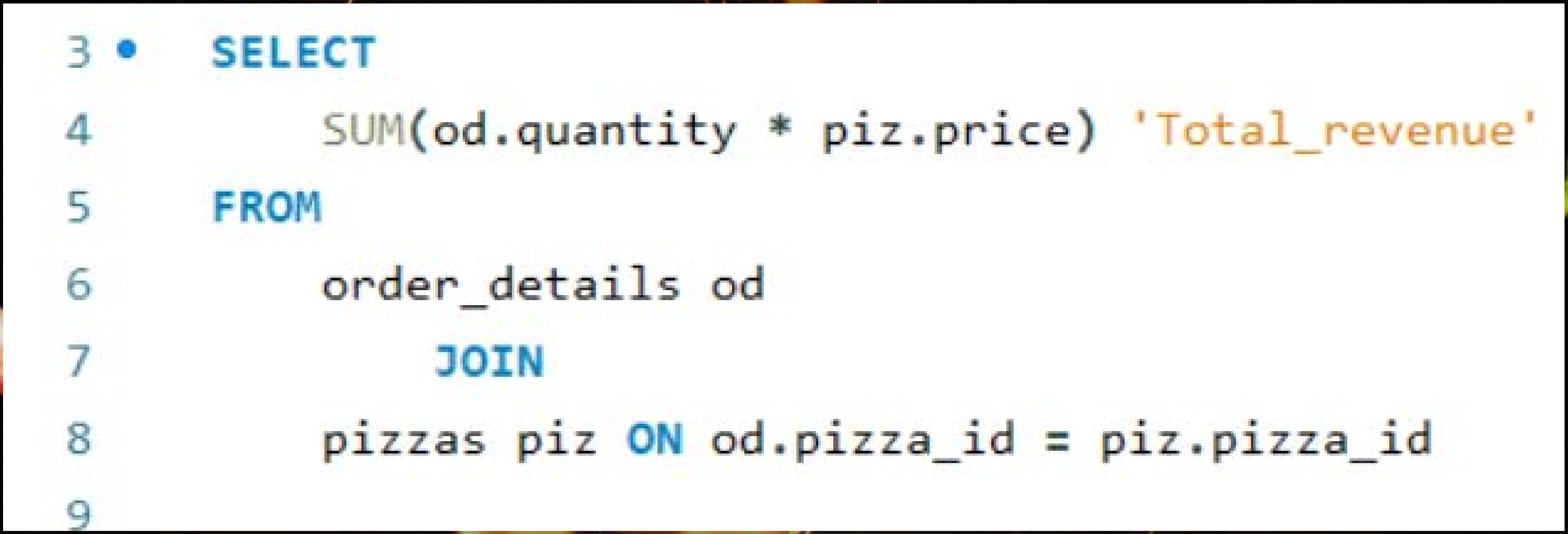
```sql
-- Retrieve the total number of orders placed.

SELECT
    COUNT(order_id) 'Total_orders'
FROM
    orders;
```

OUTPUT



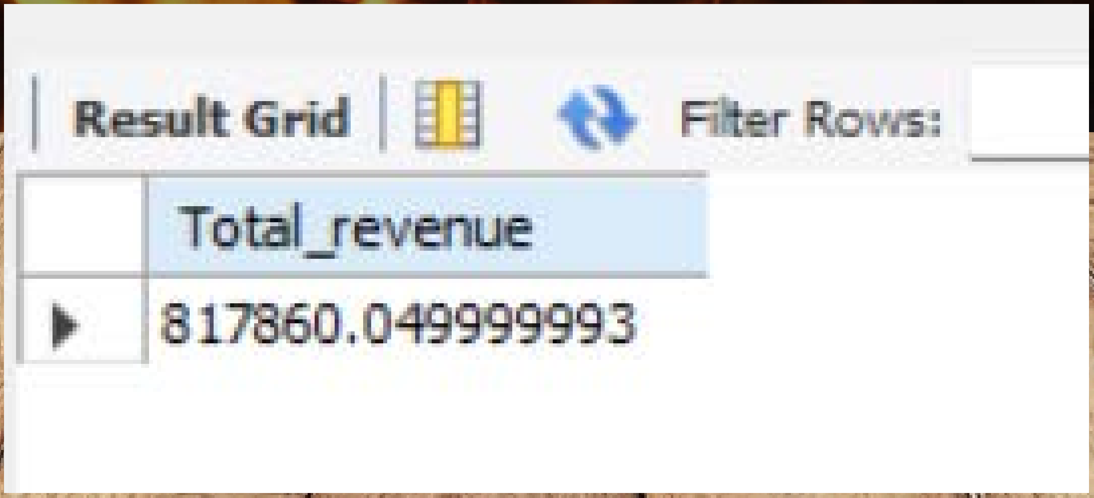
```
Total_orders
21350
```

2. **Calculate the total revenue generated from pizza sales.**



```sql
SELECT
    SUM(od.quantity * piz.price) 'Total_revenue'
FROM
    order_details od
        JOIN
    pizzas piz ON od.pizza_id = piz.pizza_id
```

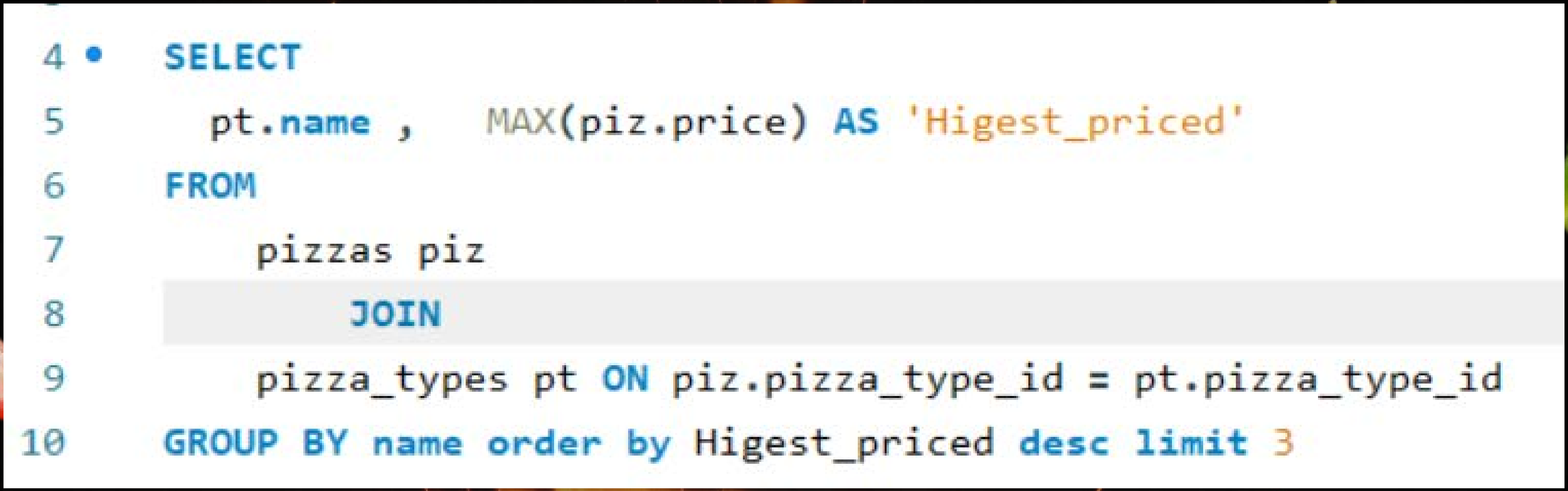
OUTPUT



```
Total_revenue
817860.049999993
```

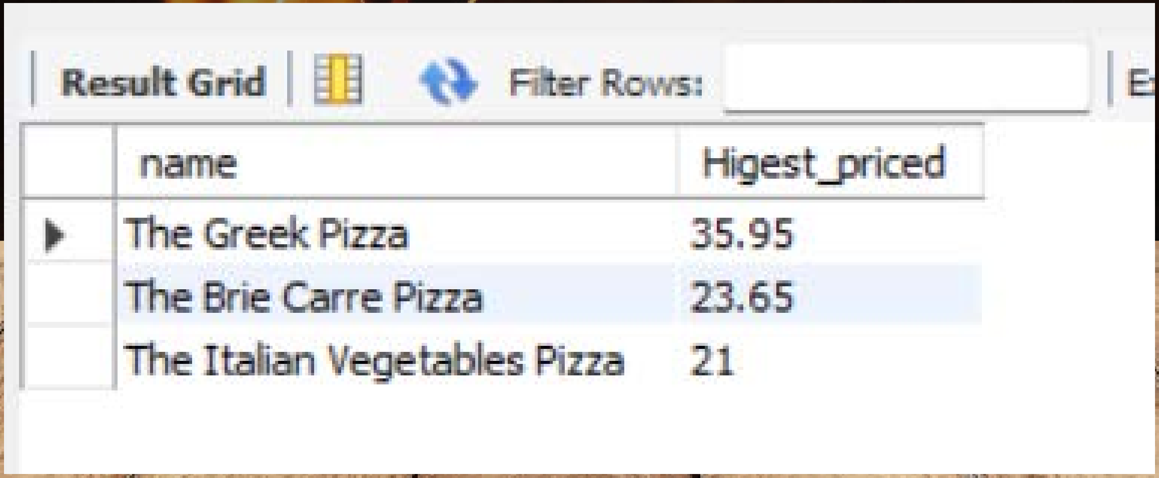
Identify the highest-priced pizza

3. IDENTIFY THE HIGHEST_PRICED PIZZA ?



```
SELECT
  pt.name ,   MAX(piz.price) AS 'Higest_priced'
FROM
    pizzas piz
        JOIN
    pizza_types pt ON piz.pizza_type_id = pt.pizza_type_id
GROUP BY name order by Higest_priced desc limit 3
```

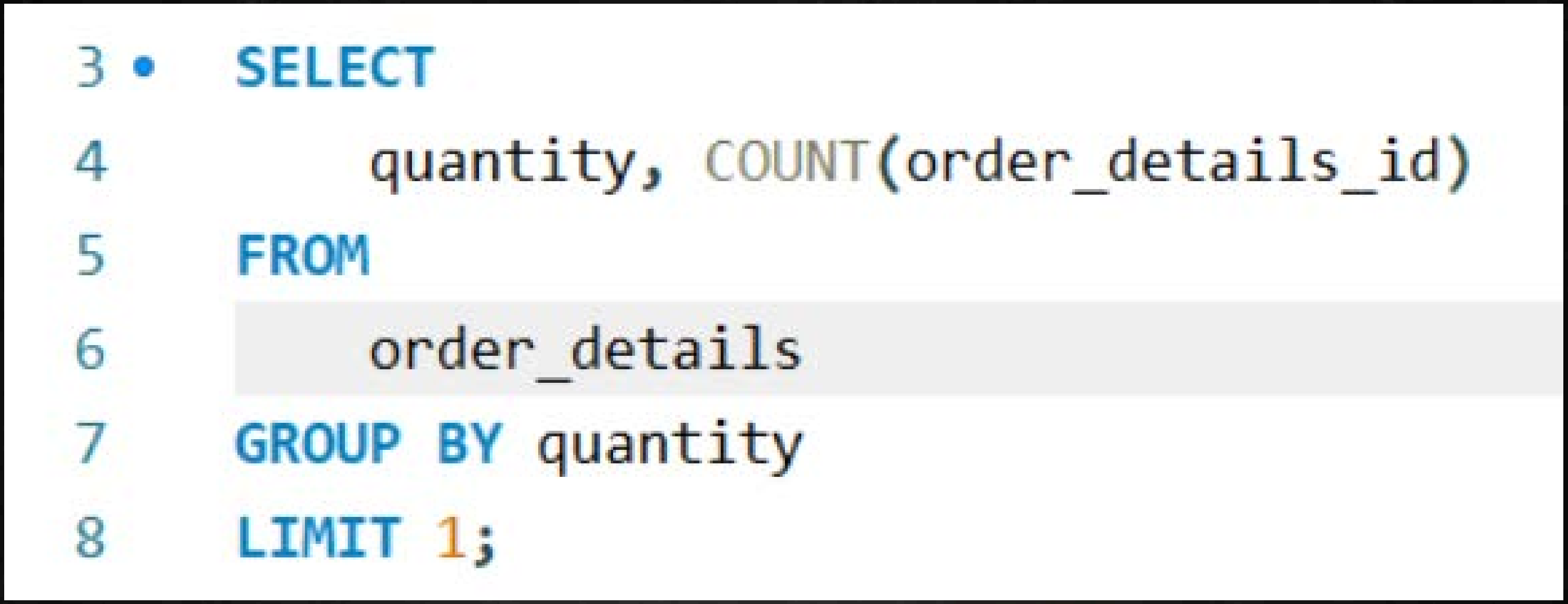
OUTPUT



name	Higest_priced
The Greek Pizza	35.95
The Brie Carre Pizza	23.65
The Italian Vegetables Pizza	21

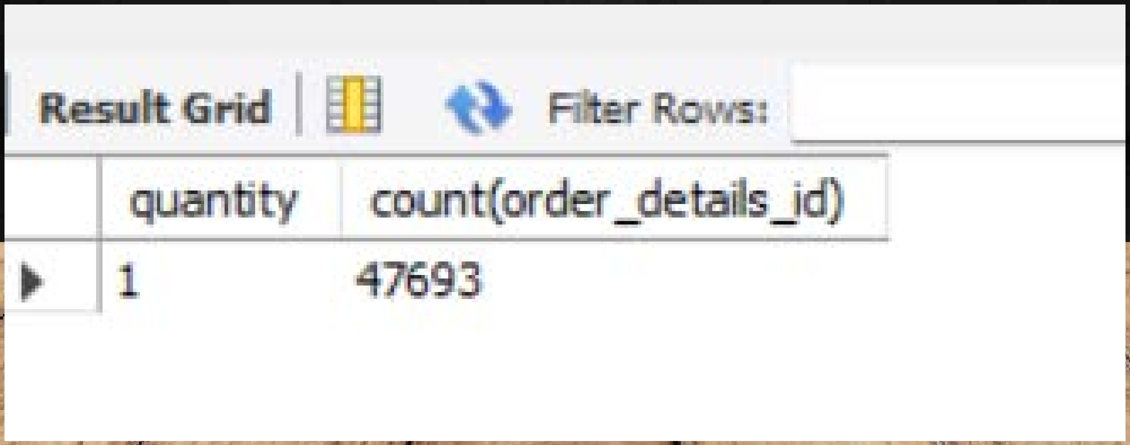
Identify the highest-priced pizza

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED .



```sql
SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity
LIMIT 1;
```

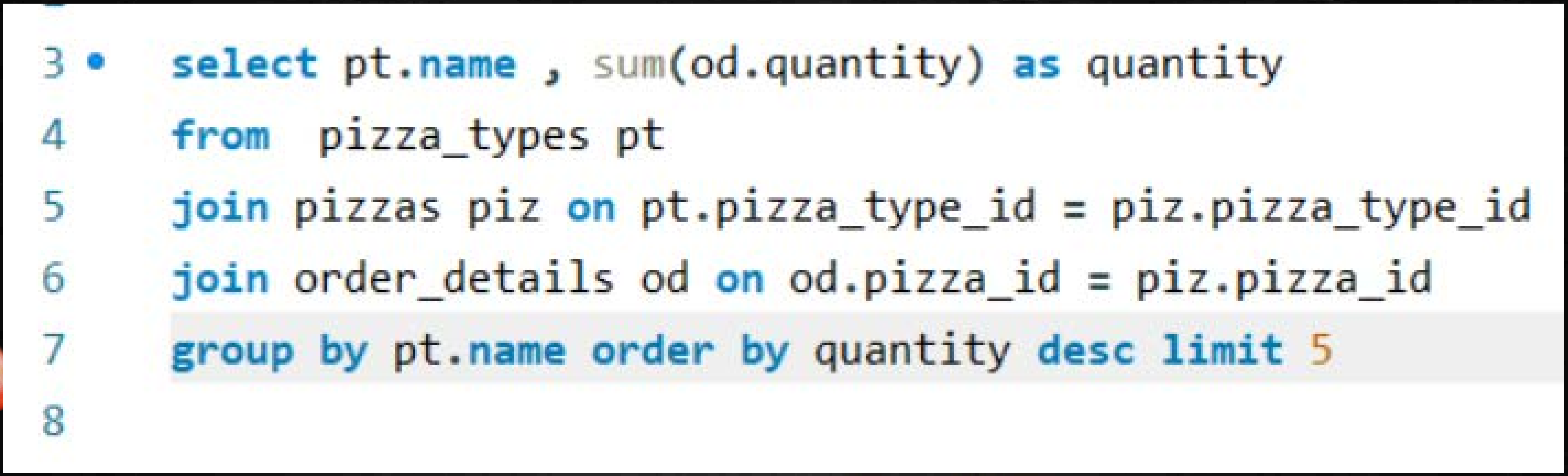
OUTPUT



quantity	count(order_details_id)
1	47693

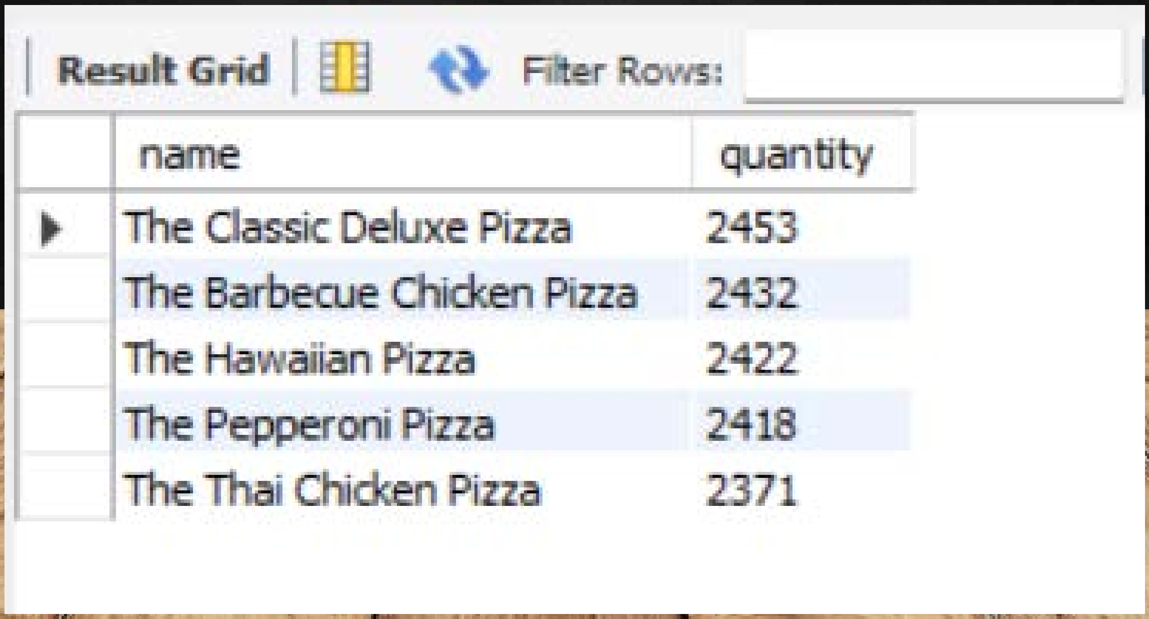
Identify the highest-priced pizza

5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES .



```sql
select pt.name , sum(od.quantity) as quantity
from  pizza_types pt
join pizzas piz on pt.pizza_type_id = piz.pizza_type_id
join order_details od on od.pizza_id = piz.pizza_id
group by pt.name order by quantity desc limit 5
```

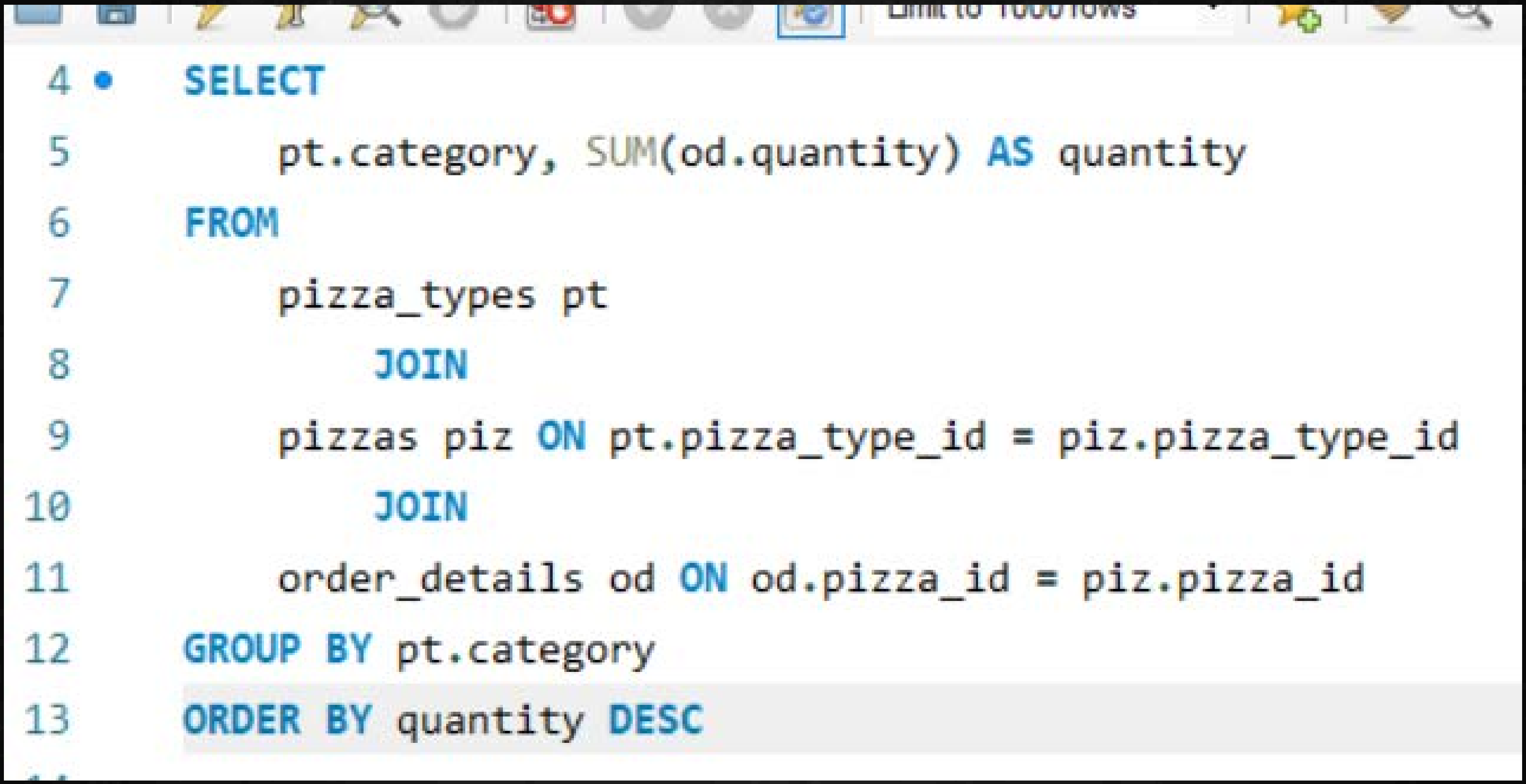
OUTPUT



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

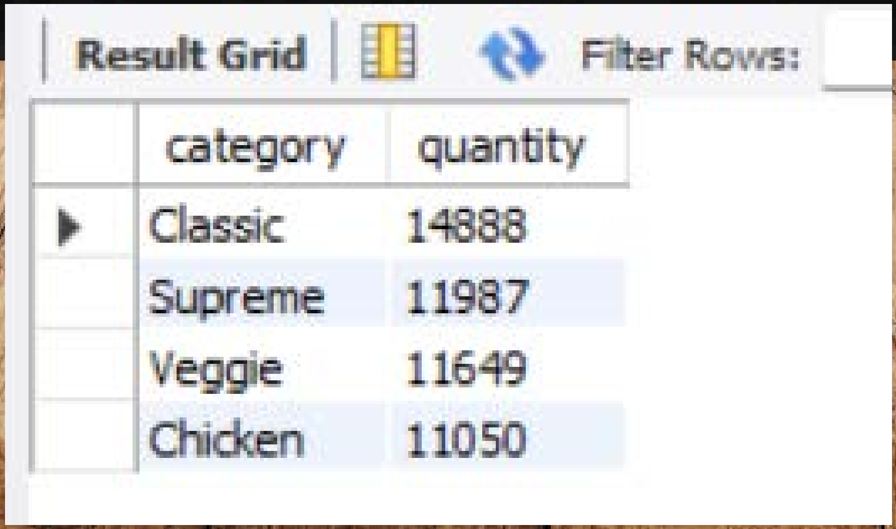
Identify the highest-priced pizza

6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED .



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

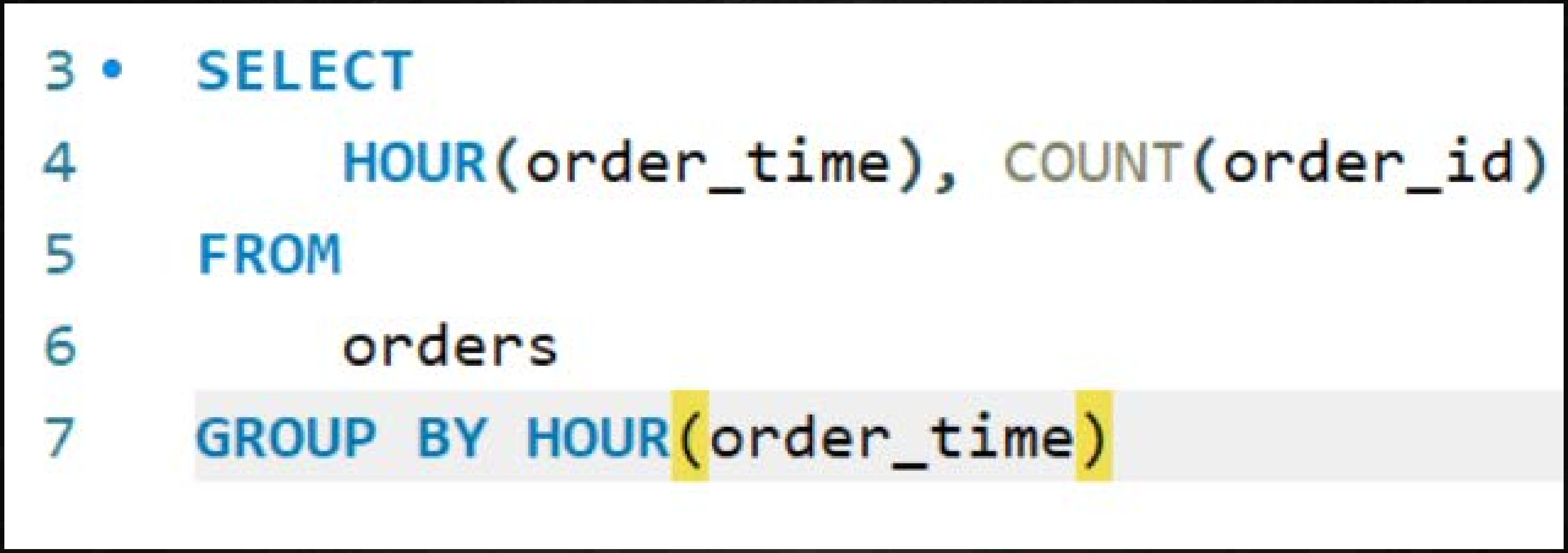
OUTPUT



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

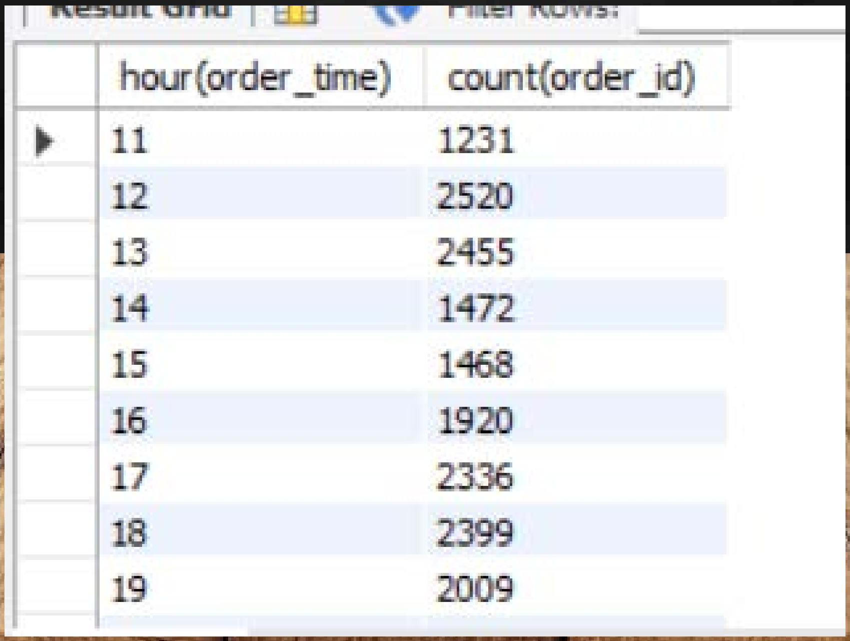
Identify the highest-priced pizza

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY .



```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
```

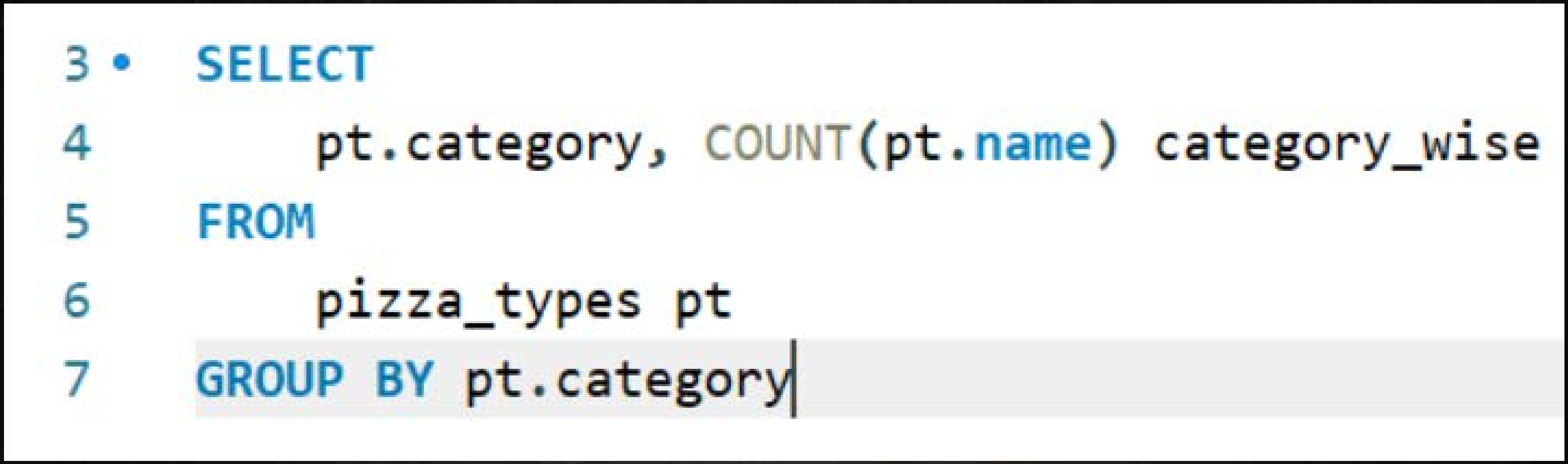
OUTPUT



hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

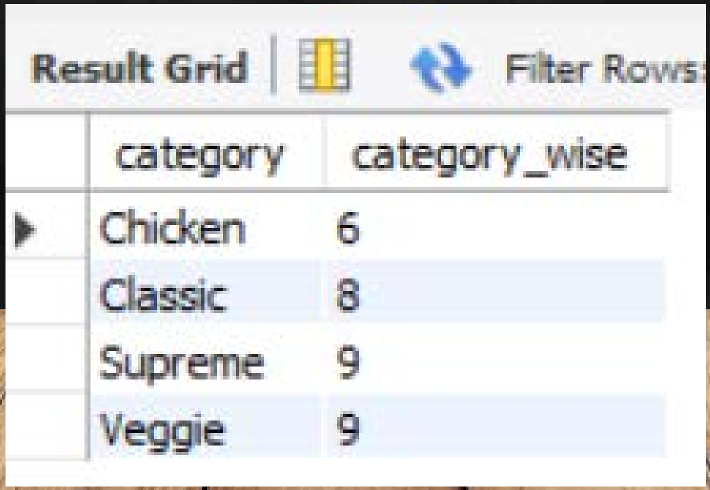
Identify the highest-priced pizza

8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS .



```
SELECT
    pt.category, COUNT(pt.name) category_wise
FROM
    pizza_types pt
GROUP BY pt.category
```

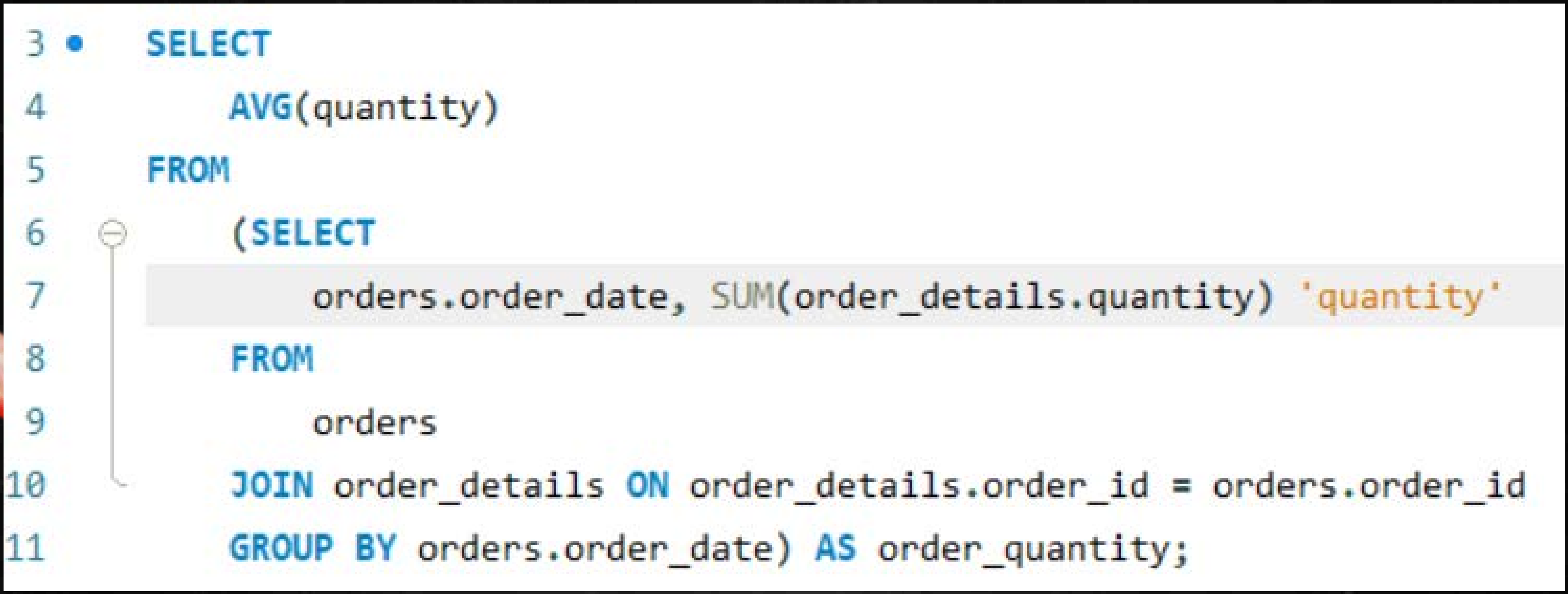
OUTPUT



category	category_wise
Chicken	6
Classic	8
Supreme	9
Veggie	9

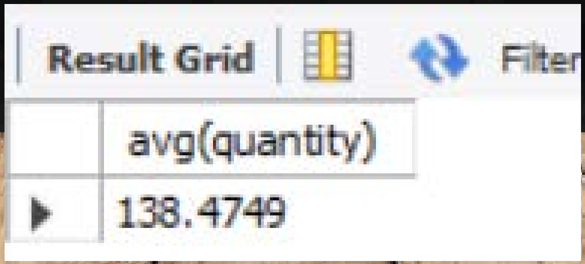
Identify the highest-priced pizza

9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY .



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

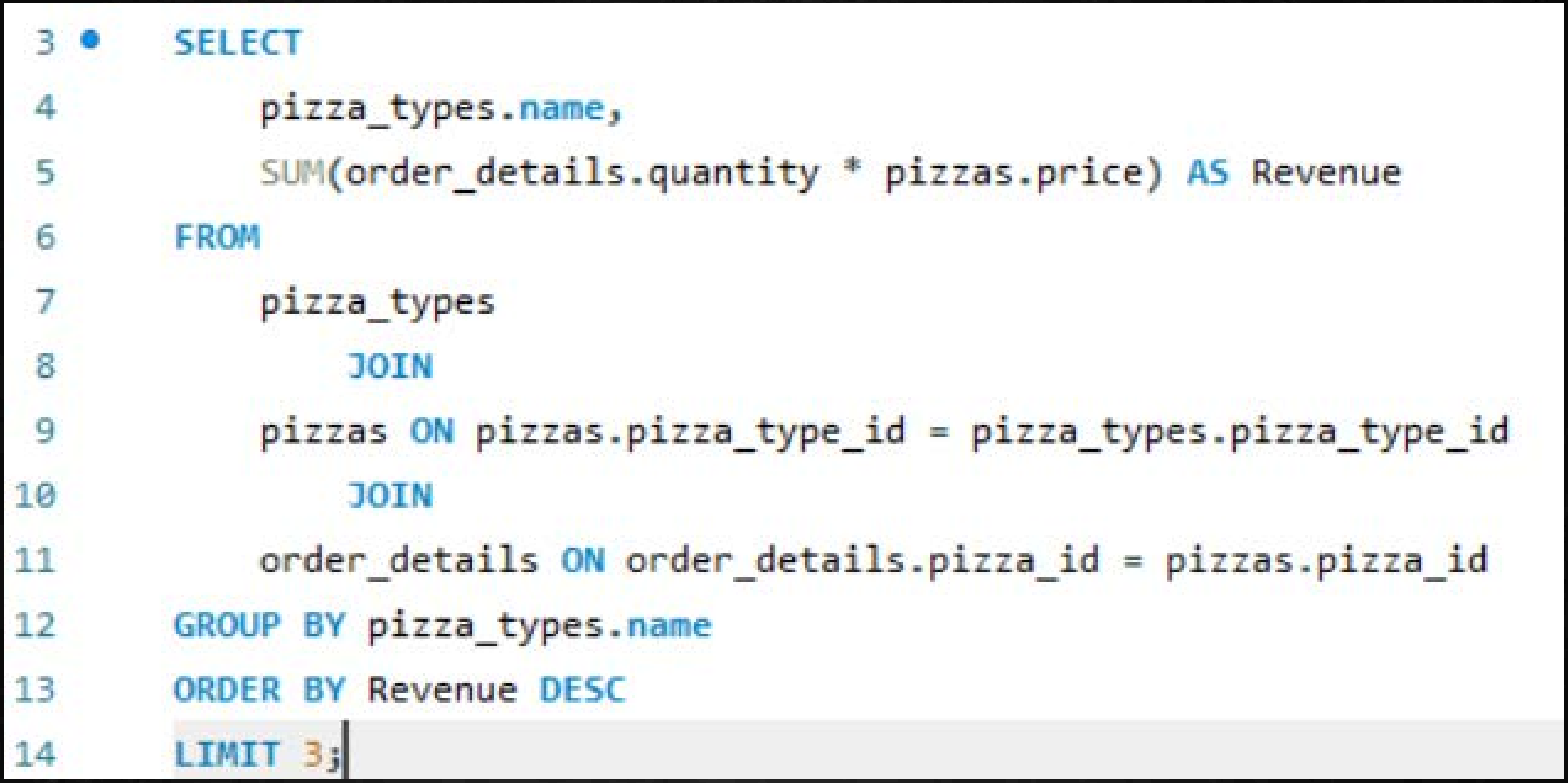
OUTPUT



```
avg(quantity)
138.4749
```

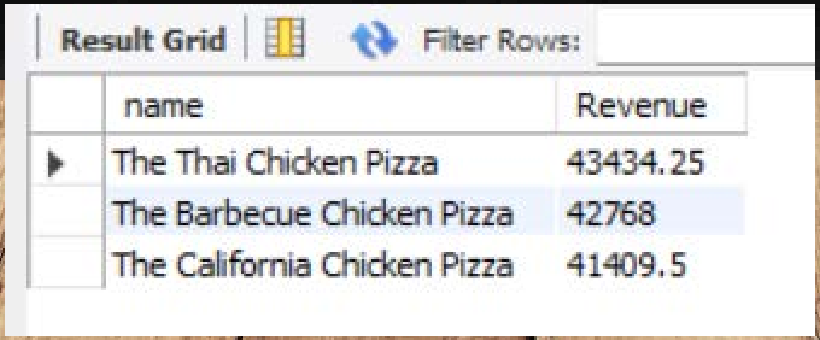
Identify the highest-priced pizza

10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE .



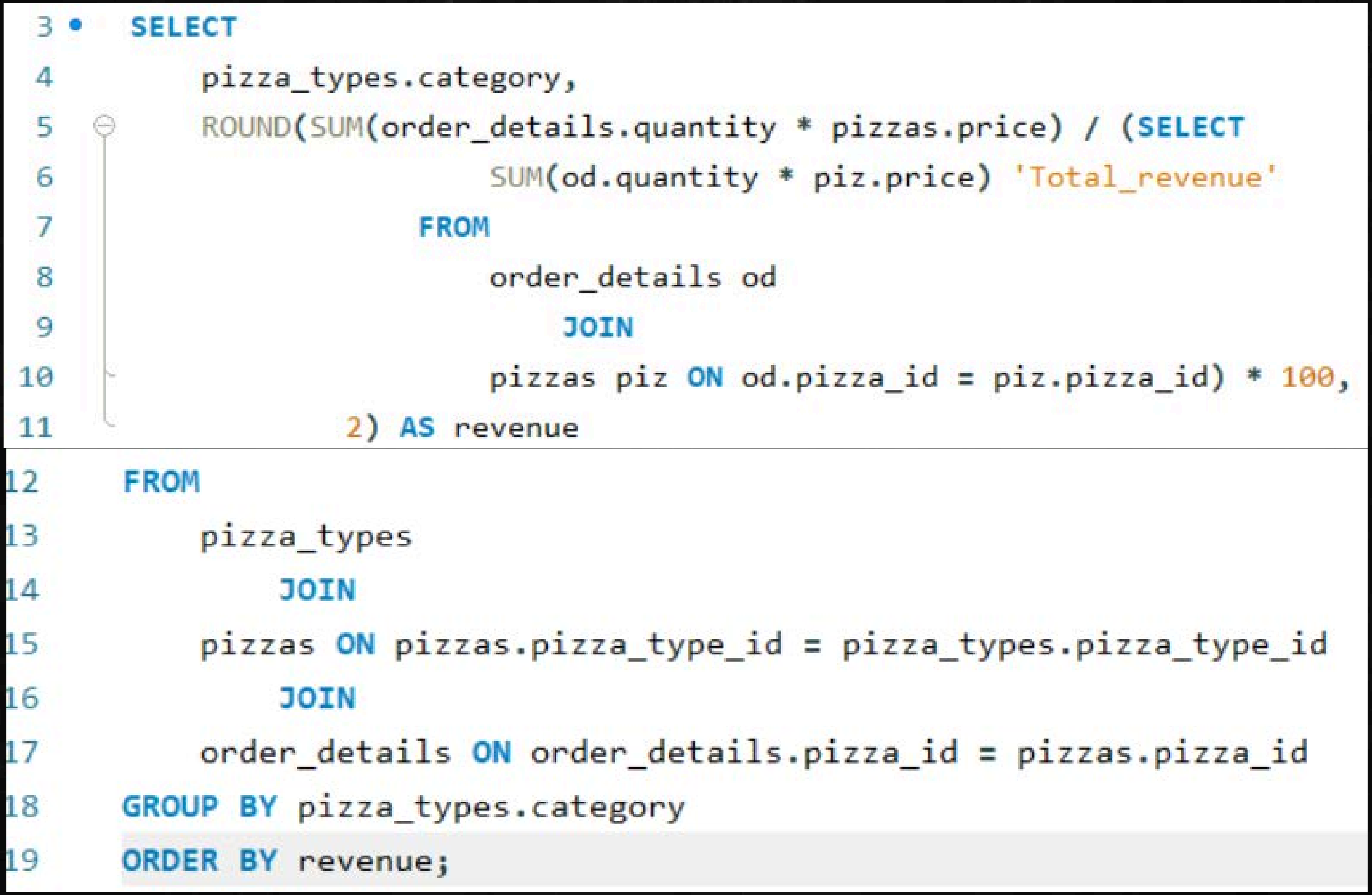
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY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

OUTPUT



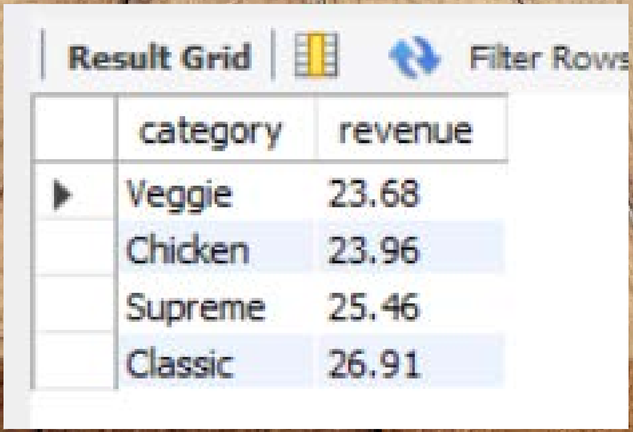
name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE .



```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(od.quantity * piz.price) 'Total_revenue'
                FROM
                    order_details od
                        JOIN
                    pizzas piz ON od.pizza_id = piz.pizza_id) * 100,
            2) 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.category
ORDER BY revenue;
```

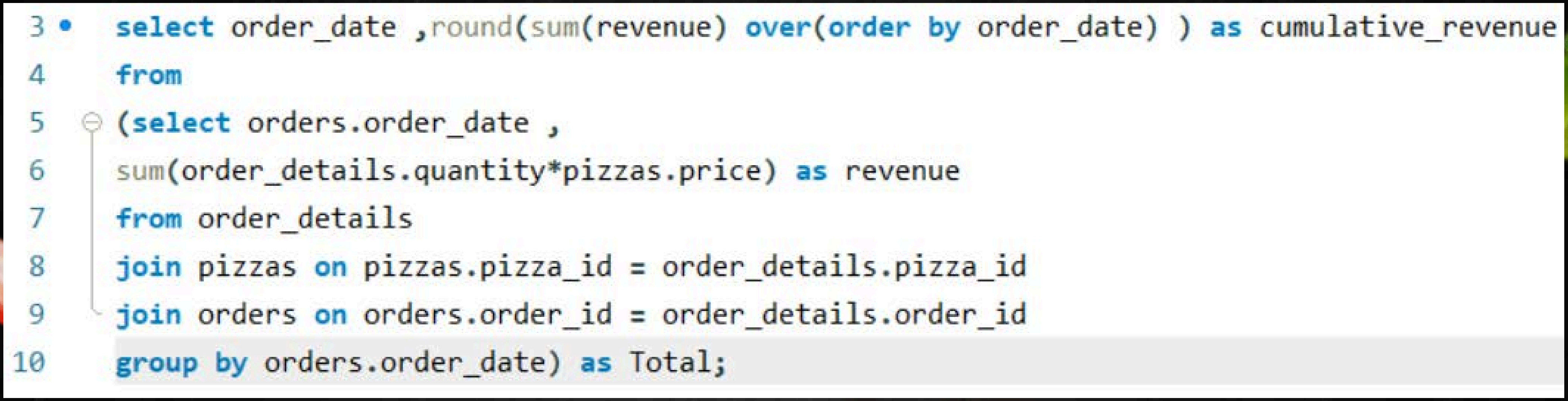
OUTPUT



```
category   revenue
Veggie     23.68
Chicken    23.96
Supreme    25.46
Classic    26.91
```

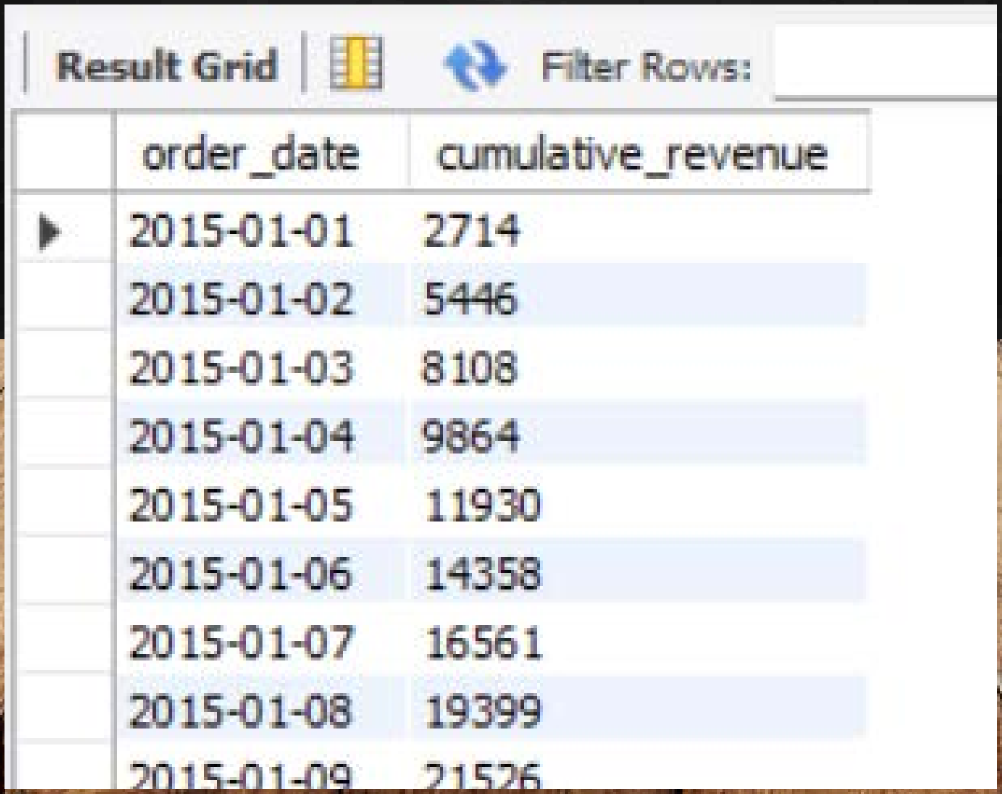
Identify the highest-priced pizza

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME .



```sql
select order_date ,round(sum(revenue) over(order by order_date) ) as cumulative_revenue
from
(select orders.order_date ,
sum(order_details.quantity*pizzas.price) as revenue
from order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id
join orders on orders.order_id = order_details.order_id
group by orders.order_date) as Total;
```

OUTPUT

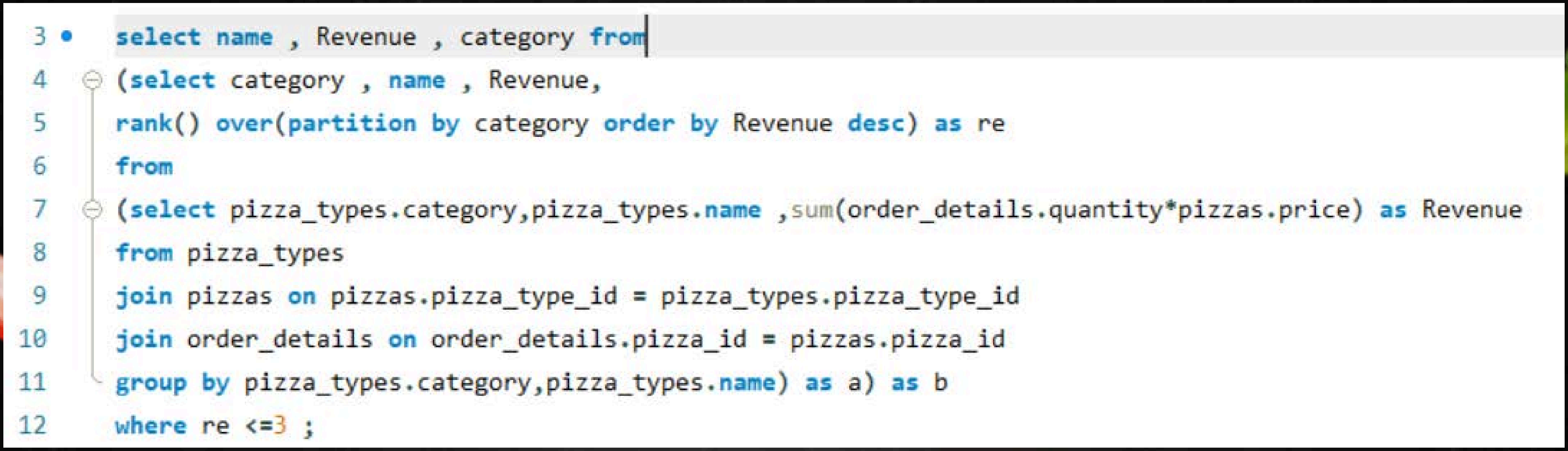


Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9864
2015-01-05	11930
2015-01-06	14358
2015-01-07	16561
2015-01-08	19399
2015-01-09	21526

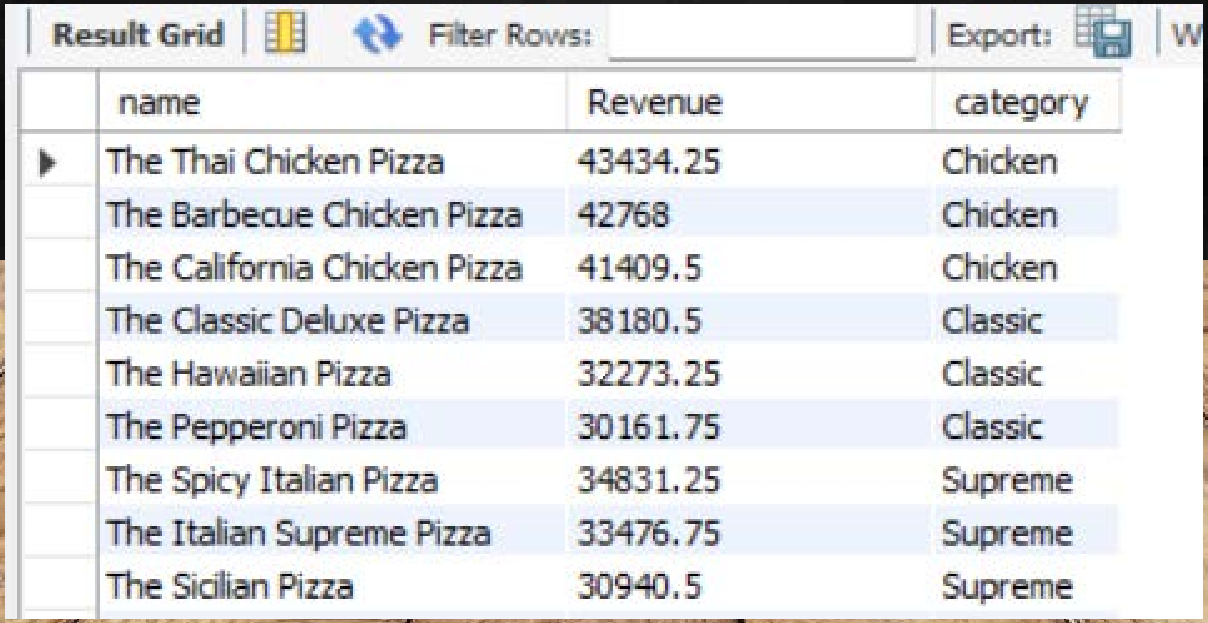
Identify the highest-priced pizza

13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY .



```
select name , Revenue , category from
(select category , name , Revenue,
rank() over(partition by category order by Revenue desc) as re
from
(select pizza_types.category,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.category,pizza_types.name) as a) as b
where re <=3 ;
```

OUTPUT



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
The Pepperoni Pizza	30161.75	Classic
The Spicy Italian Pizza	34831.25	Supreme
The Italian Supreme Pizza	33476.75	Supreme
The Sicilian Pizza	30940.5	Supreme