SQL PROJECT ON PIZZA SALES

INTRODUCTION

My Self Roni Sarkar, an enthusiastic data analyst with a passion for uncovering insights through data. Today I successfully completed a first project on SQL, analyzing pizza sales data. This achievement showcases a strong foundation in data analysis and a dedication to continuous learning.

.

BRIEF SUMMARY

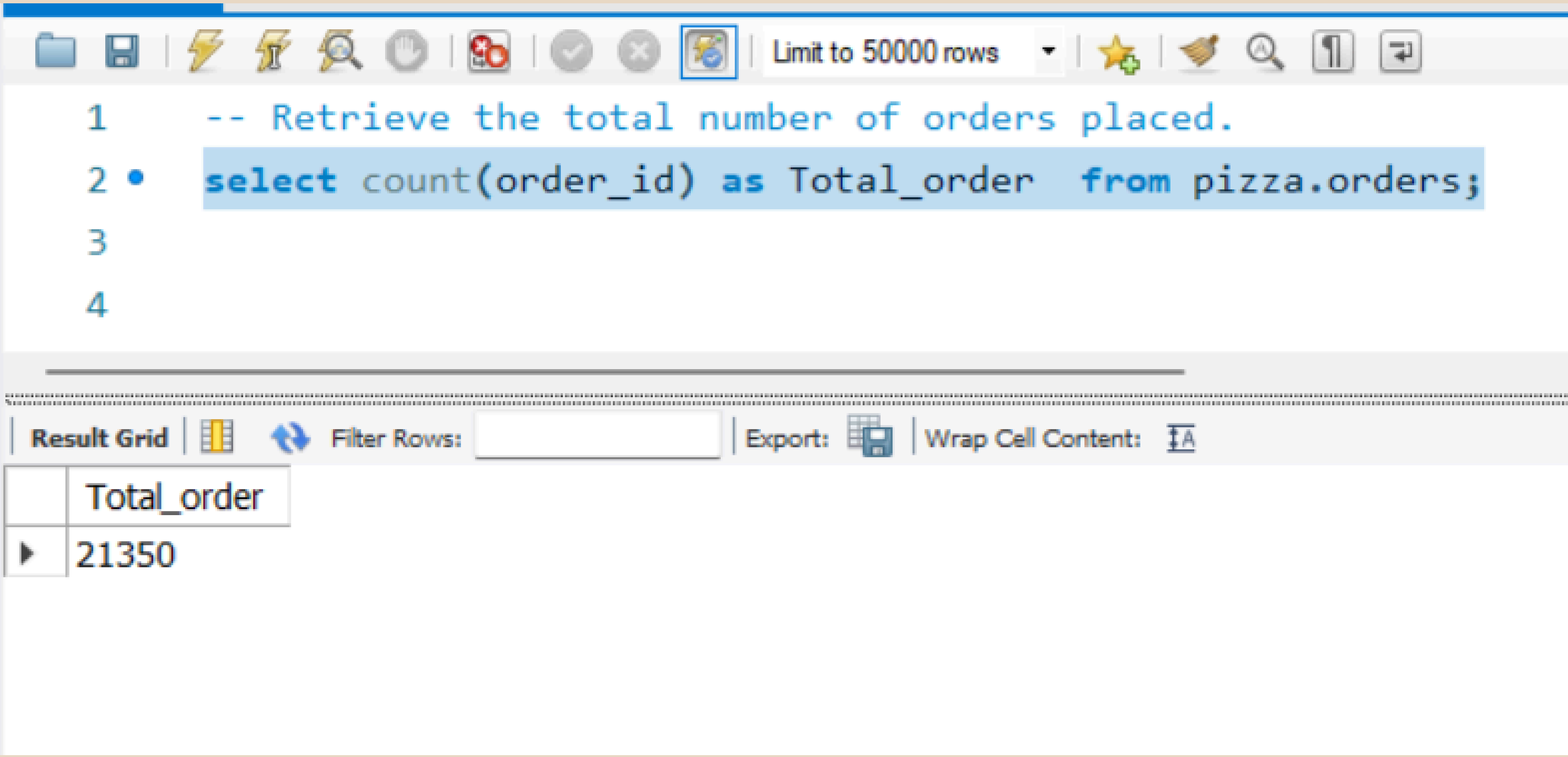
I accessed the pizza sales data from Ayushi Jain's GitHub account, an instructor at WScubetech. This comprehensive dataset, crucial for my SQL project, provided valuable insights into sales patterns. Utilizing open-source platforms like GitHub underscores the importance of community-driven resources in enhancing learning and practical application for data enthusiasts.

PROJECT QUESTION

- 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.

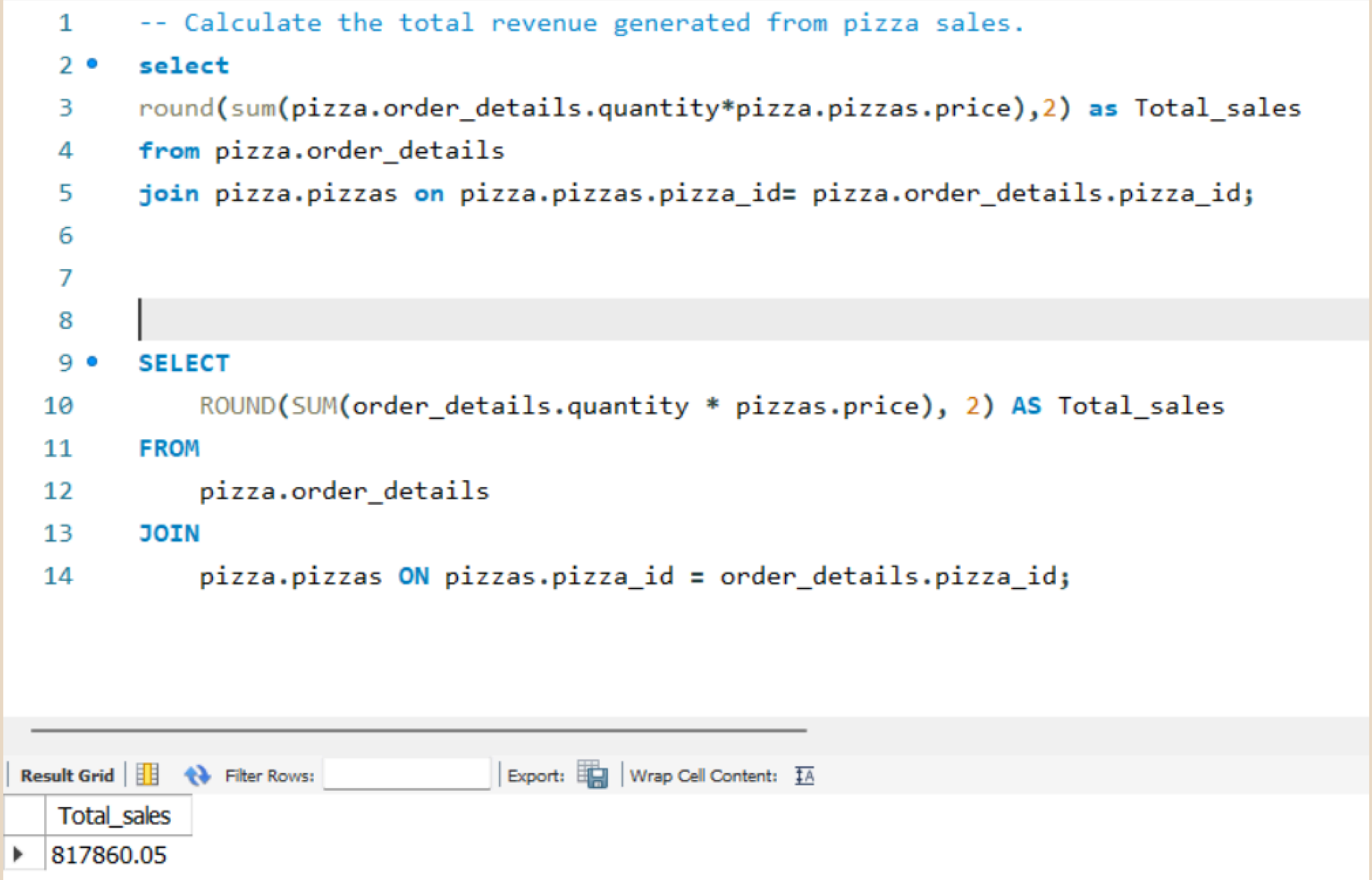
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per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

- 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.



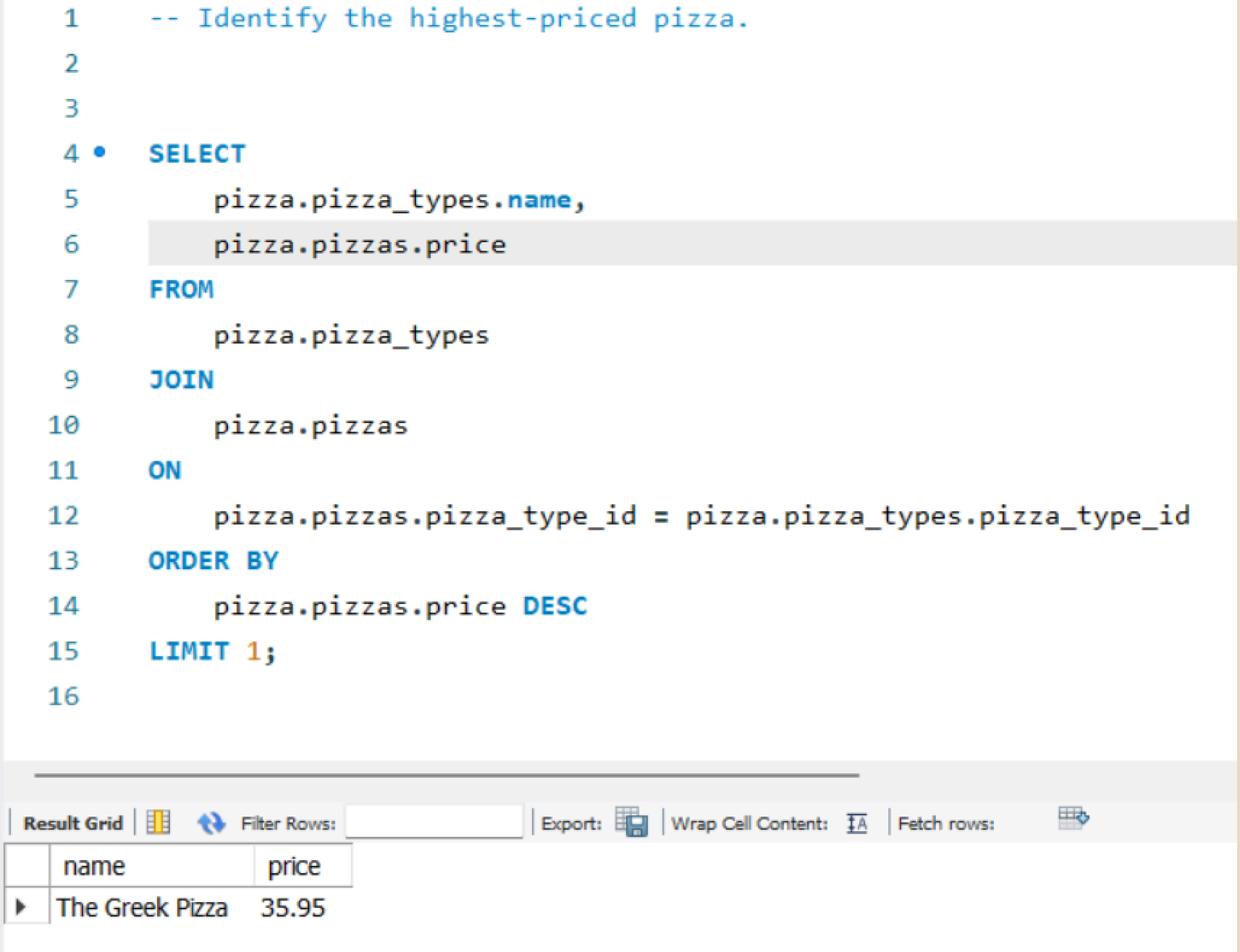
```sql
-- Retrieve the total number of orders placed.
select count(order_id) as Total_order  from pizza.orders;
```

```
Total_order
21350
```



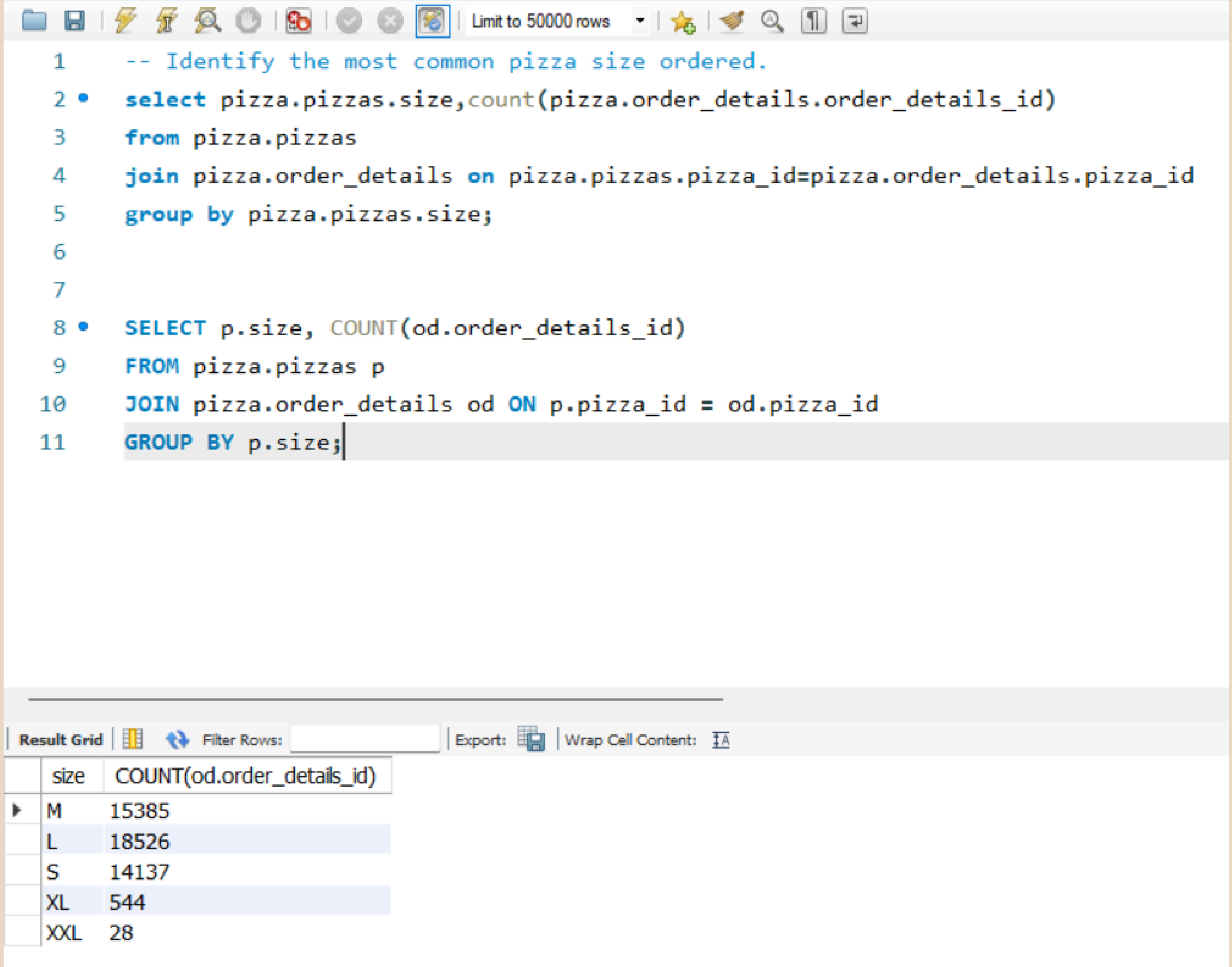
```sql
-- Calculate the total revenue generated from pizza sales.
select
round(sum(pizza.order_details.quantity*pizza.pizzas.price),2) as Total_sales
from pizza.order_details
join pizza.pizzas on pizza.pizzas.pizza_id= pizza.order_details.pizza_id;



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

```
Total_sales
817860.05
```



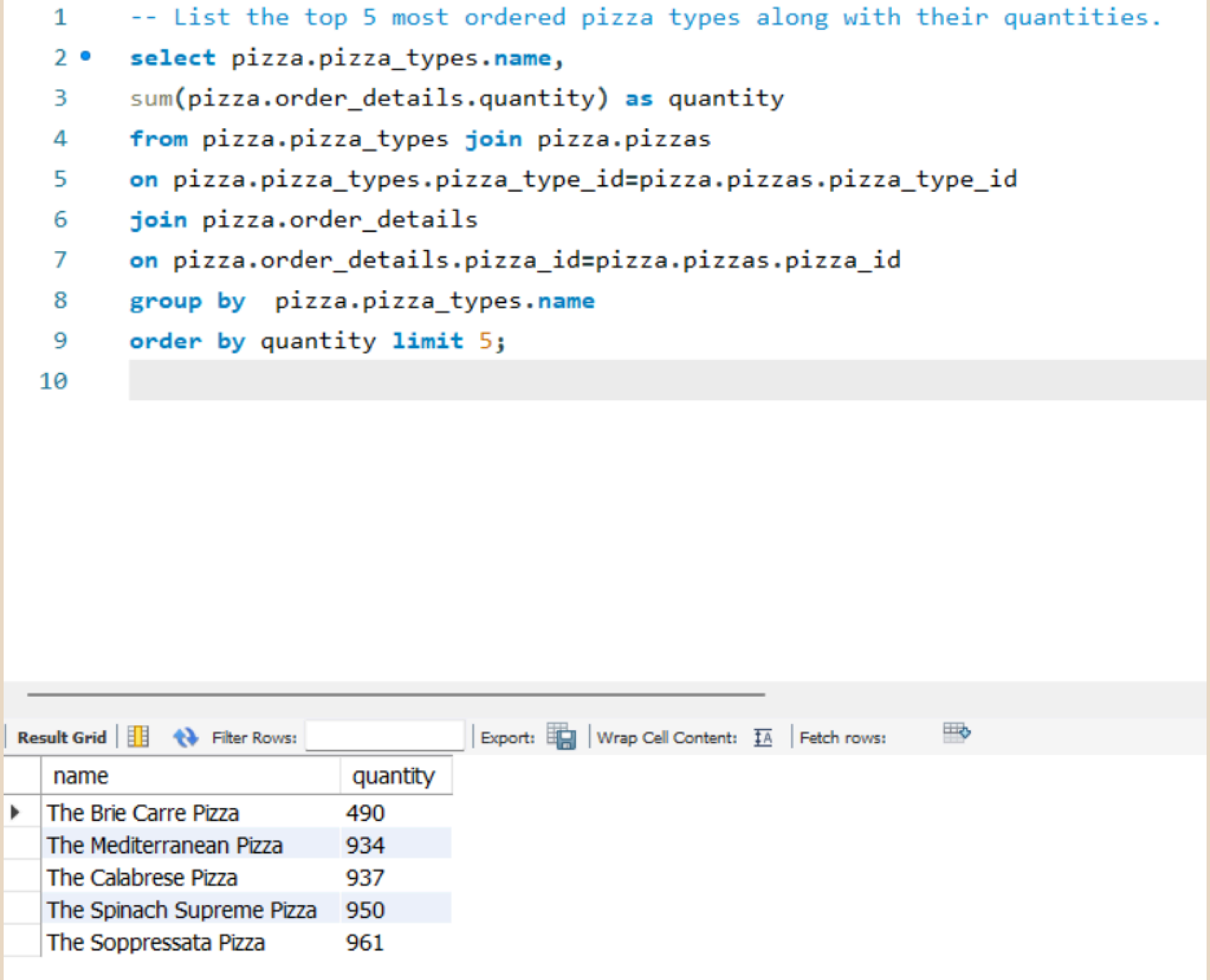
```sql
-- Identify the highest-priced pizza.


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

```

name	price
The Greek Pizza	35.95



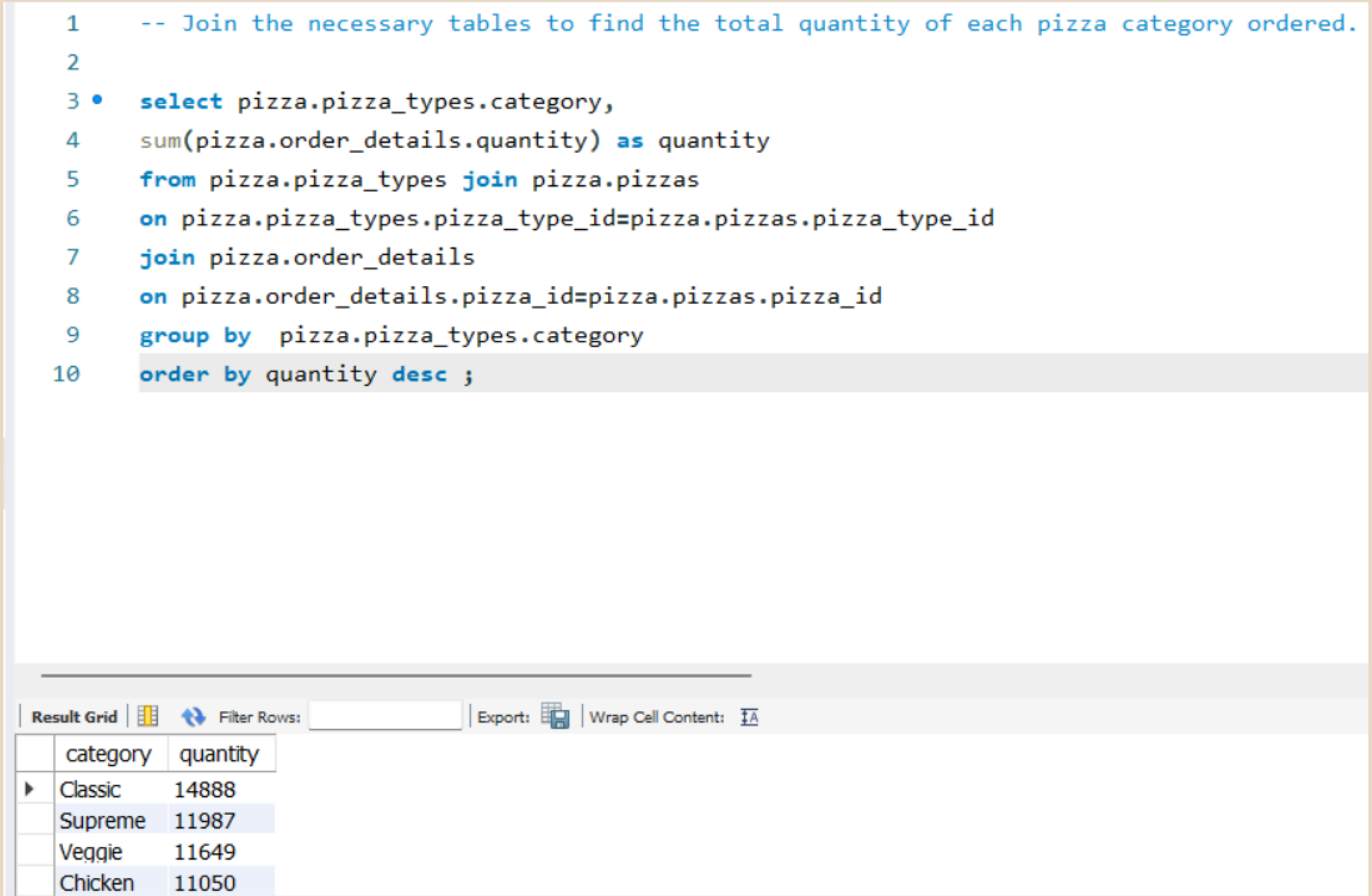
```sql
-- Identify the most common pizza size ordered.
select pizza.pizzas.size,count(pizza.order_details.order_details_id)
from pizza.pizzas
join pizza.order_details on pizza.pizzas.pizza_id=pizza.order_details.pizza_id
group by pizza.pizzas.size;


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

size	COUNT(od.order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28



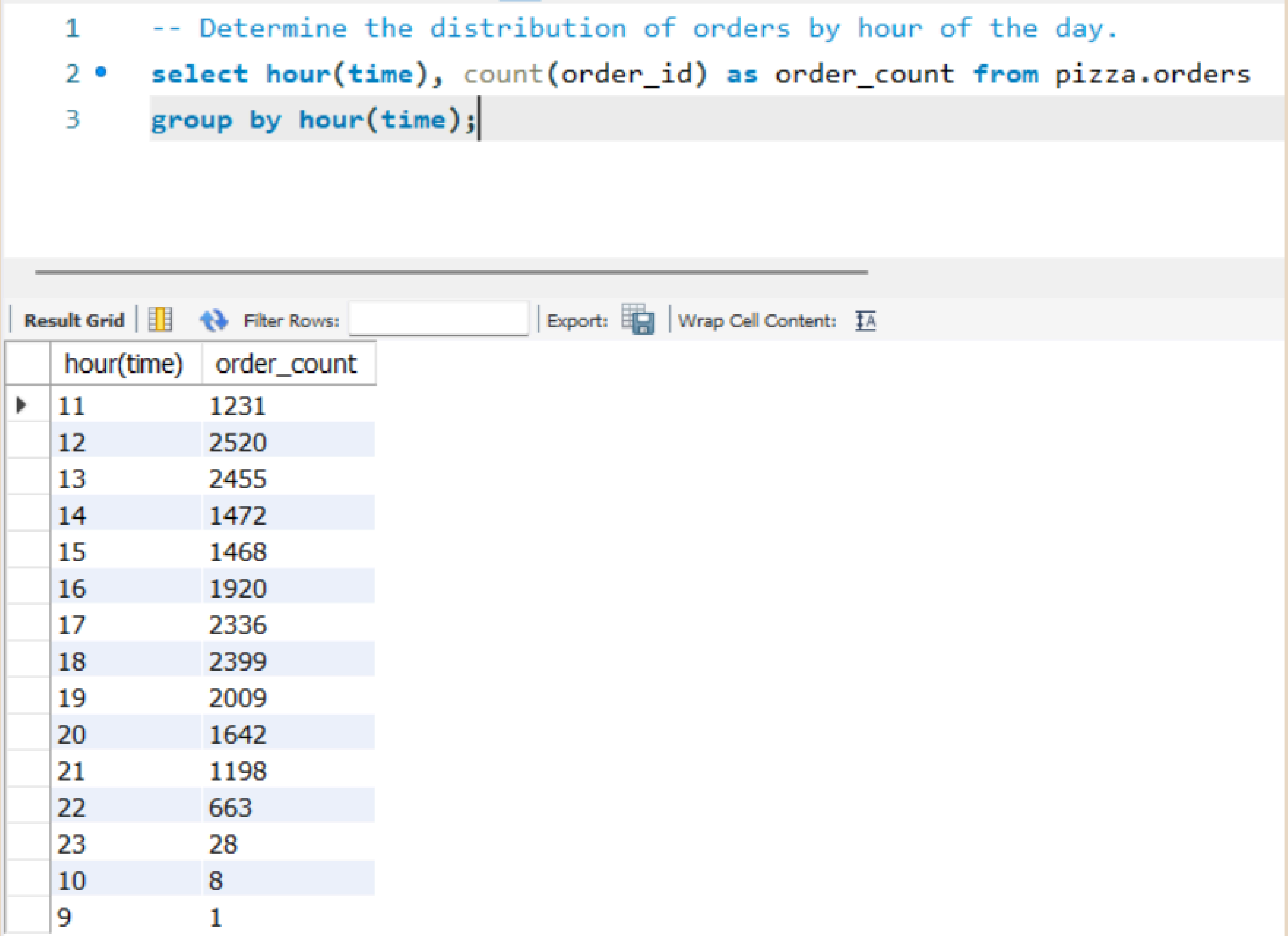
```sql
-- List the top 5 most ordered pizza types along with their quantities.
select pizza.pizza_types.name,
sum(pizza.order_details.quantity) as quantity
from pizza.pizza_types join pizza.pizzas
on pizza.pizza_types.pizza_type_id=pizza.pizzas.pizza_type_id
join pizza.order_details
on pizza.order_details.pizza_id=pizza.pizzas.pizza_id
group by  pizza.pizza_types.name
order by quantity limit 5;
```

name	quantity
The Brie Carre Pizza	490
The Mediterranean Pizza	934
The Calabrese Pizza	937
The Spinach Supreme Pizza	950
The Soppressata Pizza	961



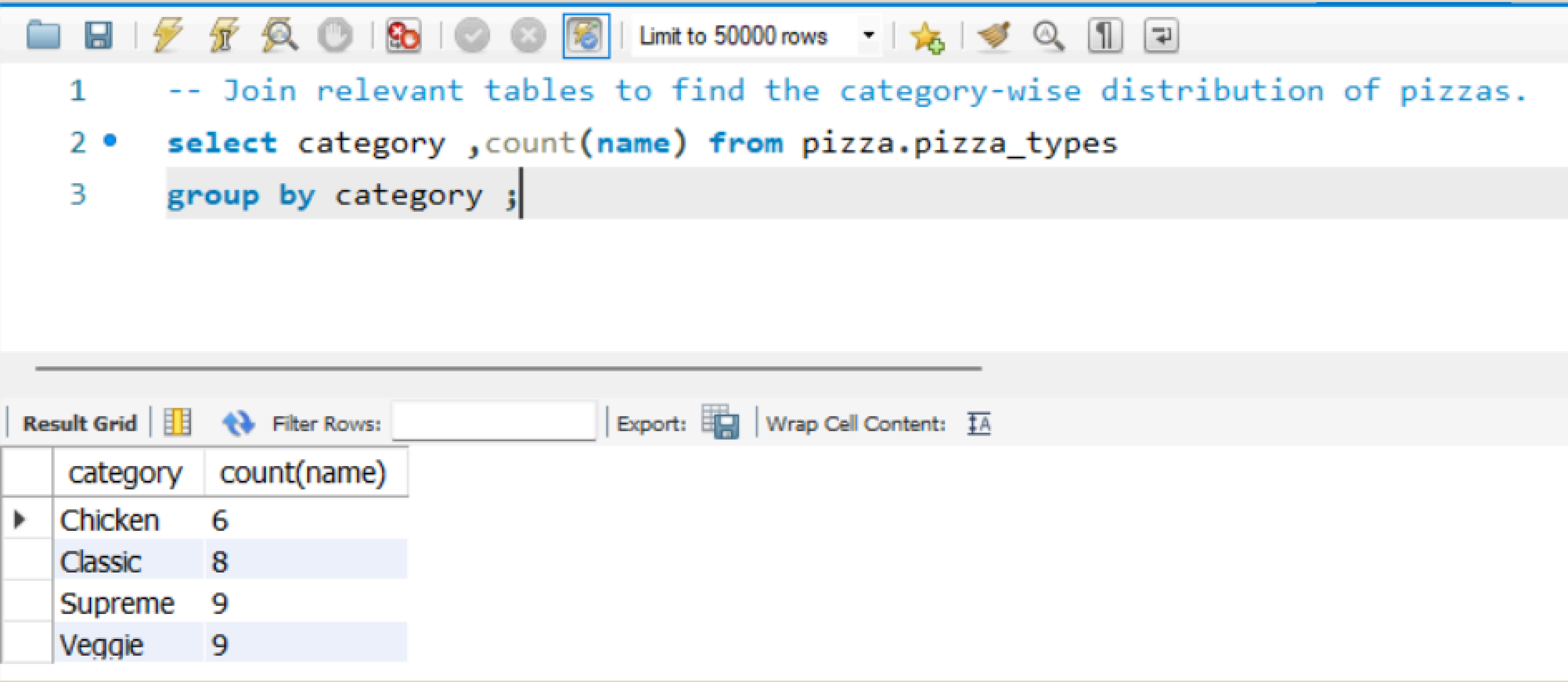
```
-- Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza.pizza_types.category,
sum(pizza.order_details.quantity) as quantity
from pizza.pizza_types join pizza.pizzas
on pizza.pizza_types.pizza_type_id=pizza.pizzas.pizza_type_id
join pizza.order_details
on pizza.order_details.pizza_id=pizza.pizzas.pizza_id
group by  pizza.pizza_types.category
order by quantity desc ;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



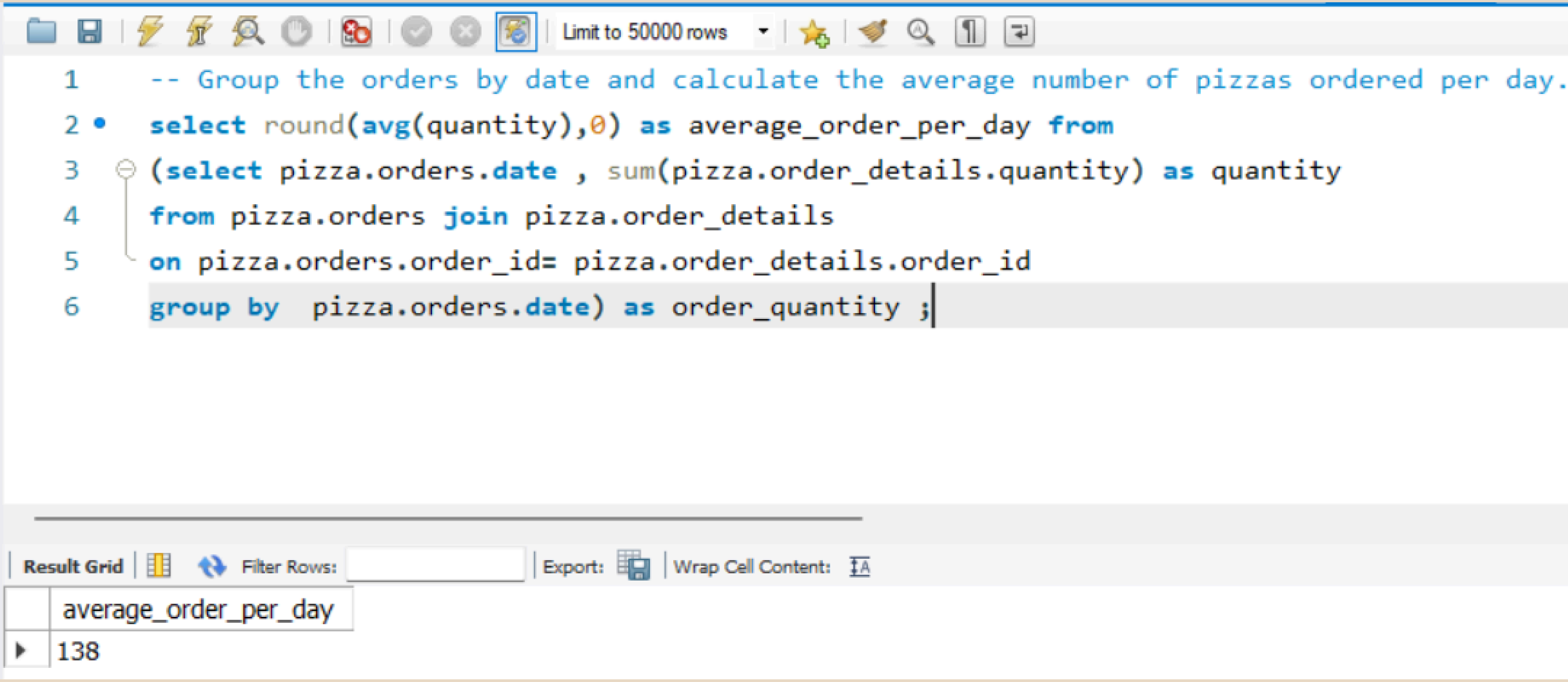
```sql
-- Determine the distribution of orders by hour of the day.
select hour(time), count(order_id) as order_count from pizza.orders
group by hour(time);
```

hour(time)	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



```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
select category ,count(name) from pizza.pizza_types
group by category ;
```

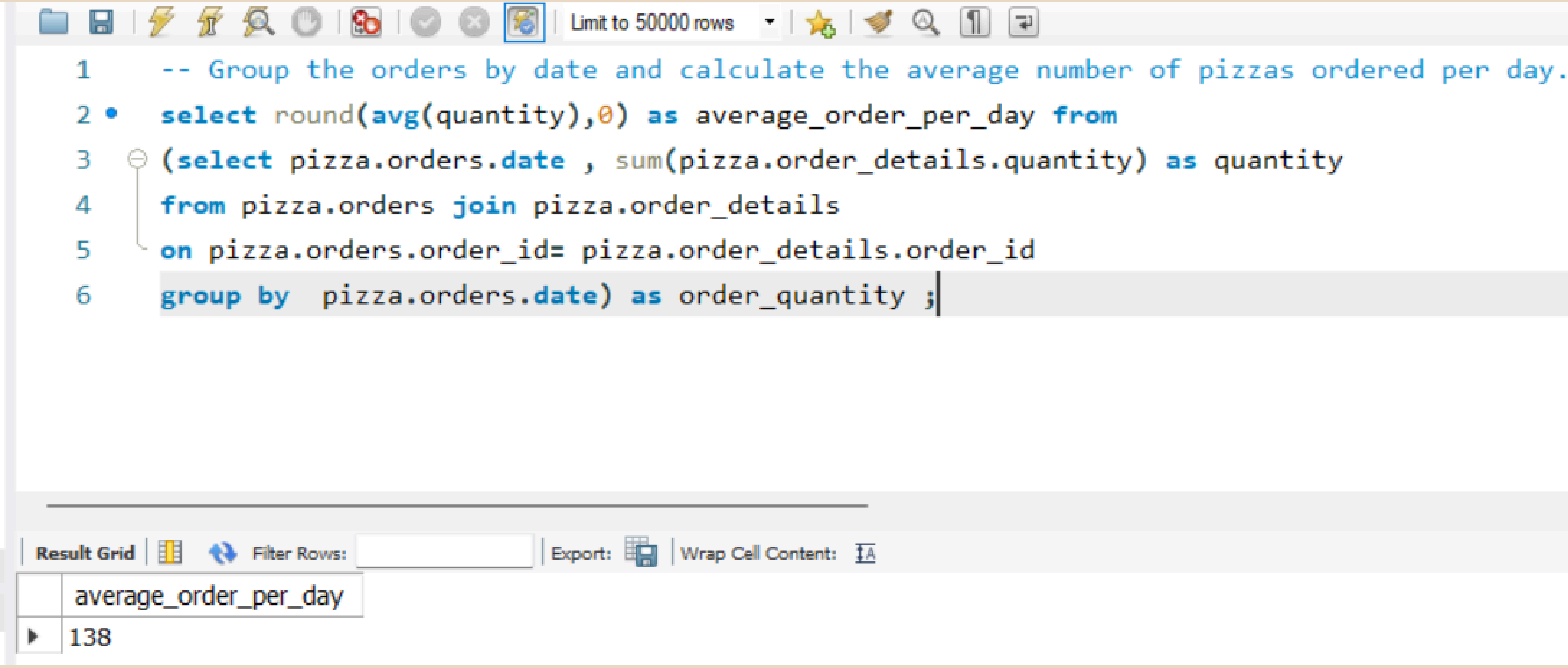
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



Limit to 50000 rows

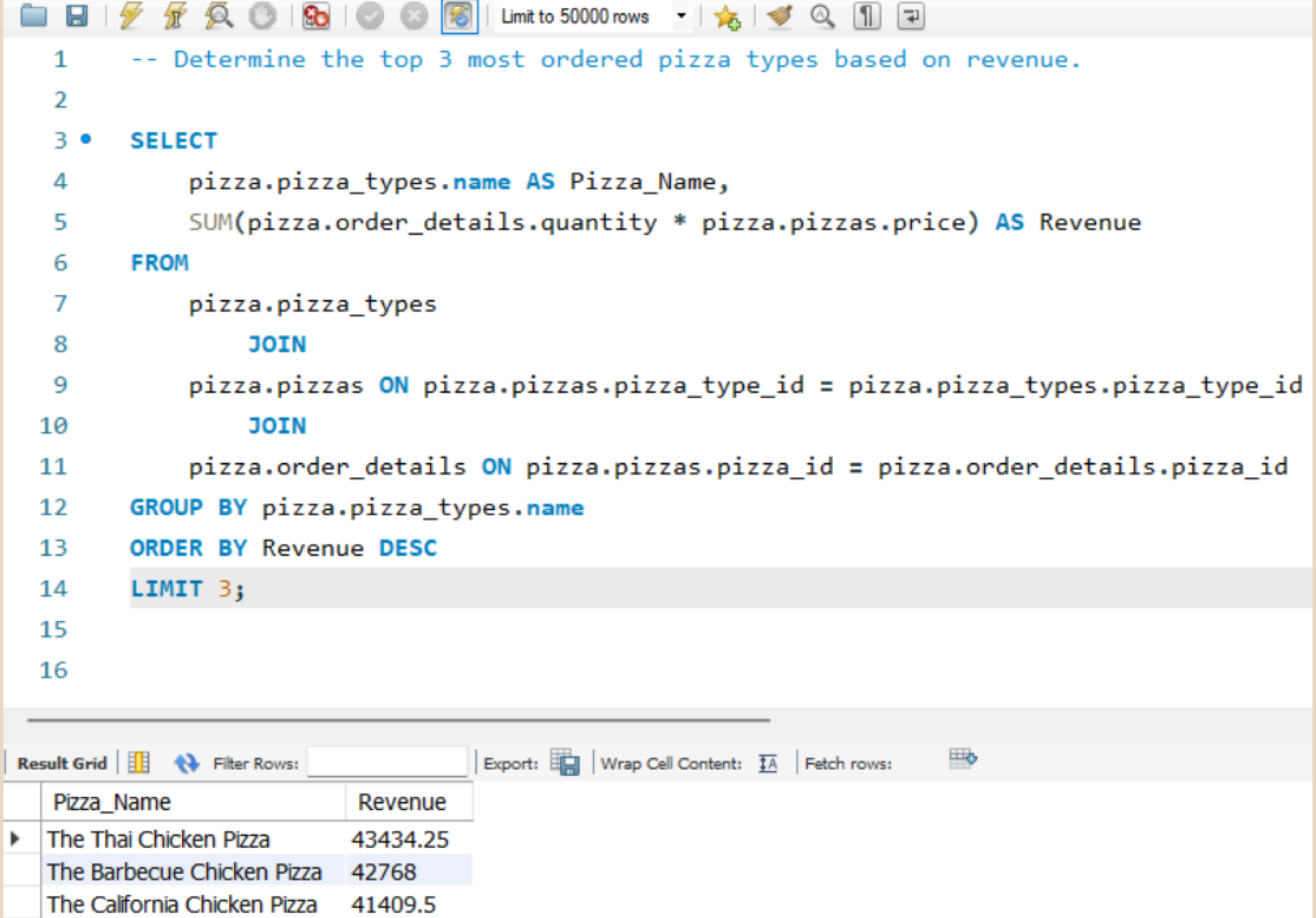
```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
select round(avg(quantity),0) as average_order_per_day from
(select pizza.orders.date , sum(pizza.order_details.quantity) as quantity
from pizza.orders join pizza.order_details
on pizza.orders.order_id= pizza.order_details.order_id
group by  pizza.orders.date) as order_quantity ;
```

```
average_order_per_day
138
```



```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
select round(avg(quantity),0) as average_order_per_day from
(select pizza.orders.date , sum(pizza.order_details.quantity) as quantity
from pizza.orders join pizza.order_details
on pizza.orders.order_id= pizza.order_details.order_id
group by  pizza.orders.date) as order_quantity ;
```

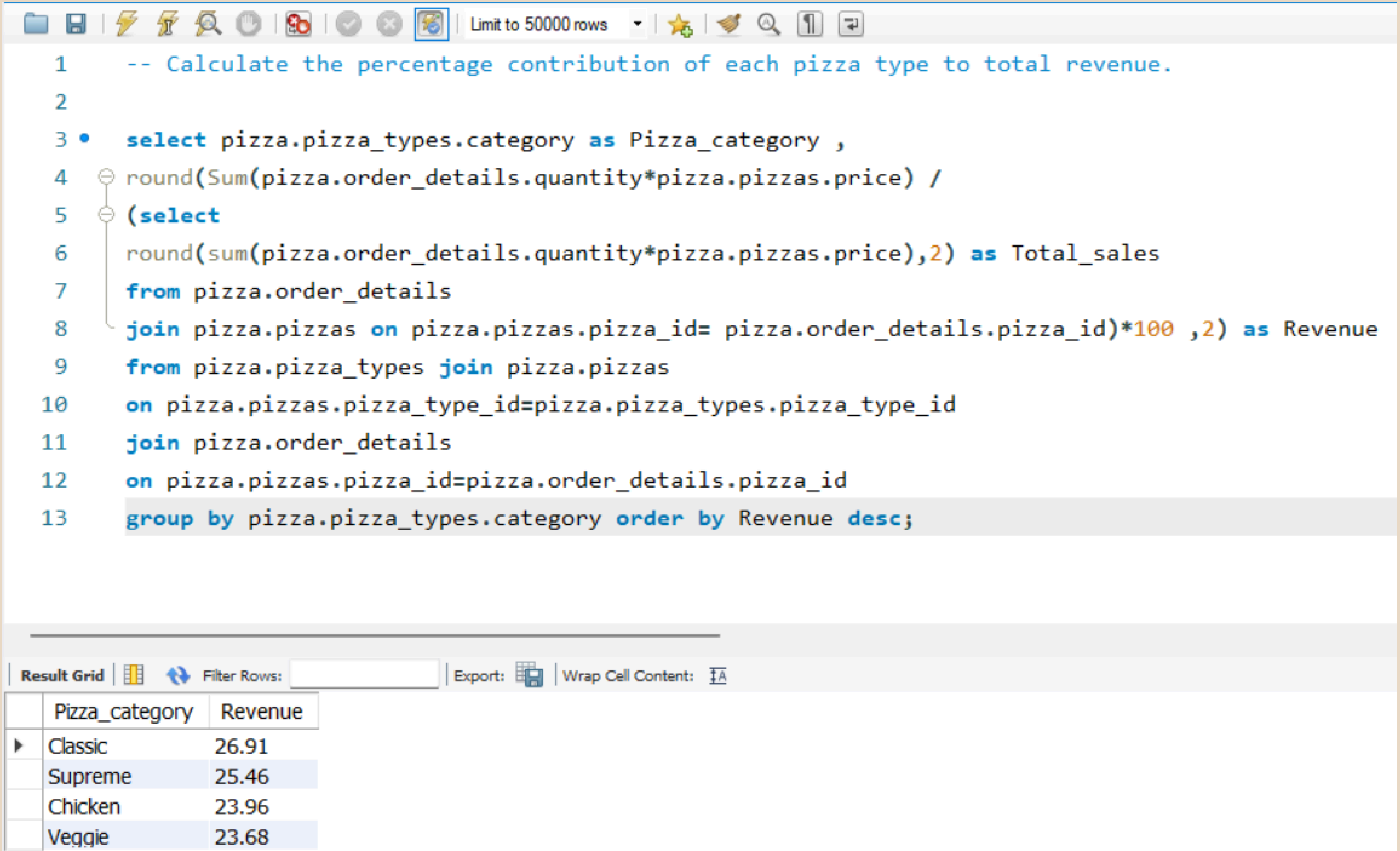
average_order_per_day
138



Limit to 50000 rows

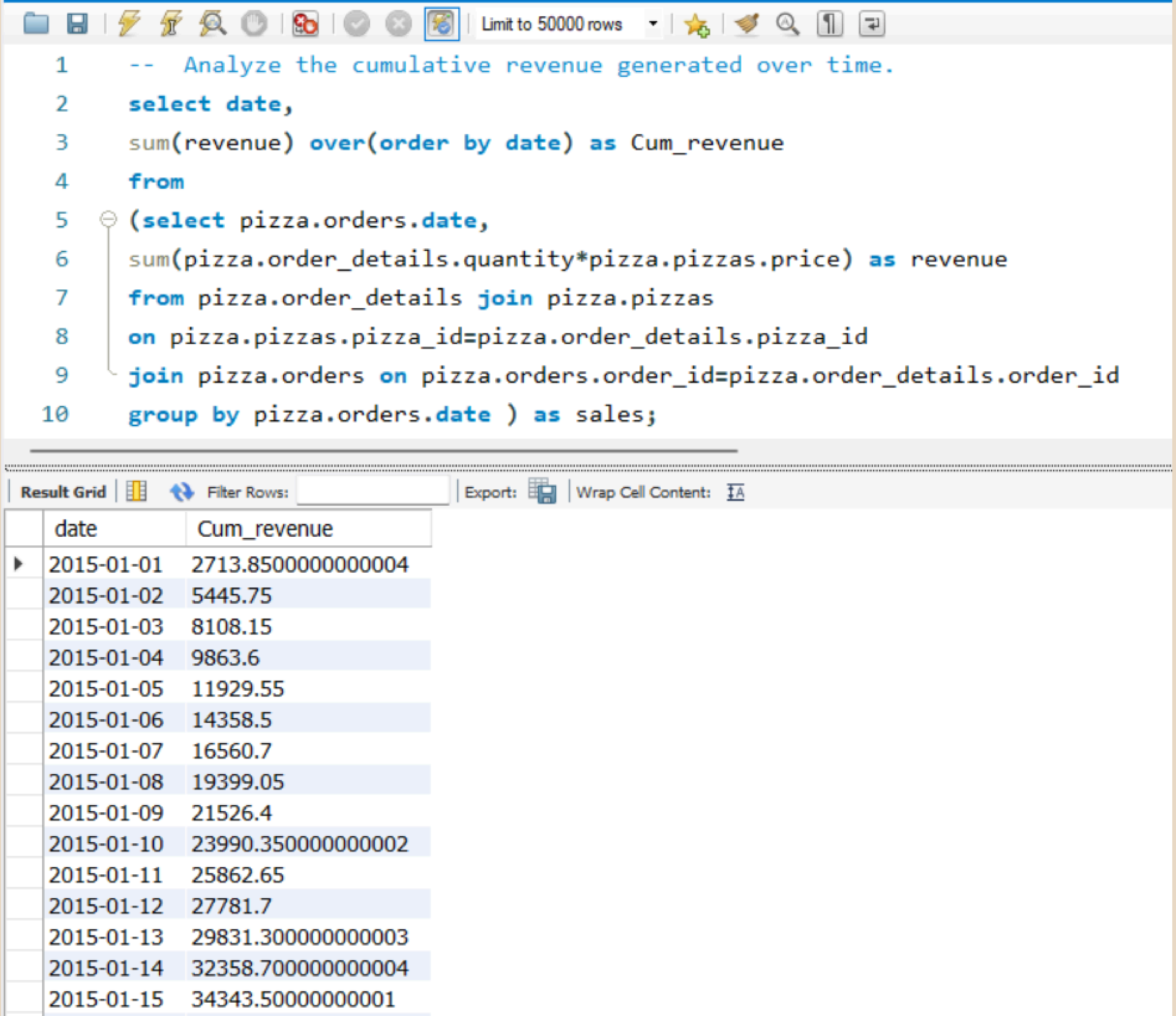
```sql
-- Determine the top 3 most ordered pizza types based on revenue.

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

Pizza_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.pizza_types.category as Pizza_category ,
round(Sum(pizza.order_details.quantity*pizza.pizzas.price) /
(select
round(sum(pizza.order_details.quantity*pizza.pizzas.price),2) as Total_sales
from pizza.order_details
join pizza.pizzas on pizza.pizzas.pizza_id= pizza.order_details.pizza_id)*100 ,2) as Revenue
from pizza.pizza_types join pizza.pizzas
on pizza.pizzas.pizza_type_id=pizza.pizza_types.pizza_type_id
join pizza.order_details
on pizza.pizzas.pizza_id=pizza.order_details.pizza_id
group by pizza.pizza_types.category order by Revenue desc;
```

Pizza_category	Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

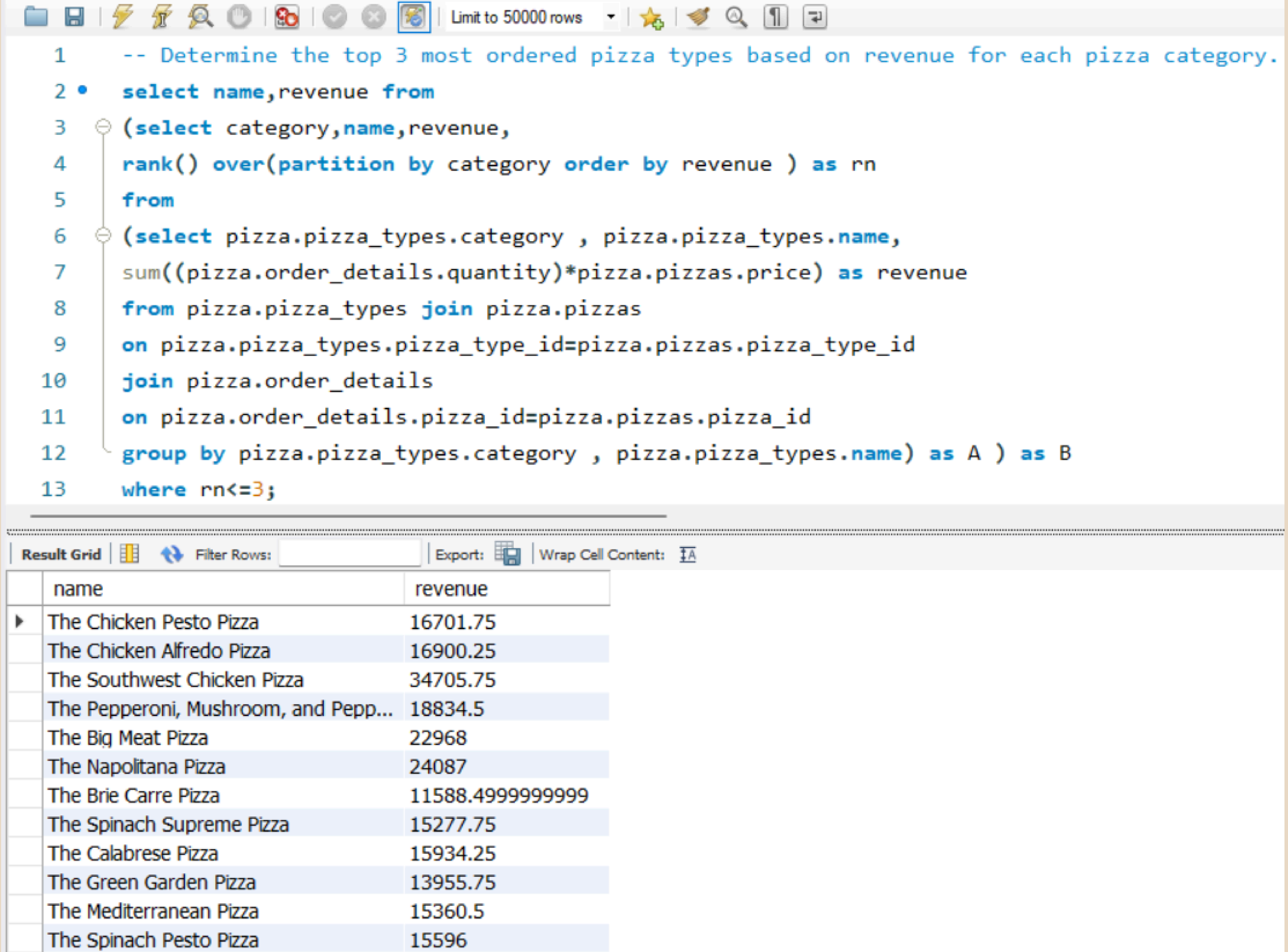


Limit to 50000 rows

```
--  Analyze the cumulative revenue generated over time.
select date,
sum(revenue) over(order by date) as Cum_revenue
from
(select pizza.orders.date,
sum(pizza.order_details.quantity*pizza.pizzas.price) as revenue
from pizza.order_details join pizza.pizzas
on pizza.pizzas.pizza_id=pizza.order_details.pizza_id
join pizza.orders on pizza.orders.order_id=pizza.order_details.order_id
group by pizza.orders.date ) as sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001



Limit to 50000 rows

```sql
-- 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 ) as rn
from
(select pizza.pizza_types.category , pizza.pizza_types.name,
sum((pizza.order_details.quantity)*pizza.pizzas.price) as revenue
from pizza.pizza_types join pizza.pizzas
on pizza.pizza_types.pizza_type_id=pizza.pizzas.pizza_type_id
join pizza.order_details
on pizza.order_details.pizza_id=pizza.pizzas.pizza_id
group by pizza.pizza_types.category , pizza.pizza_types.name) as A ) as B
where rn<=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Pepp...	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75
The Calabrese Pizza	15934.25
The Green Garden Pizza	13955.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

THANK YOU