PIZZAHUTS SQL ANALYSIS

By Tapas Parida

HELLO!

My name is Tapas.
In this project, I used SQL to analyze pizza sales data. I performed queries to calculate revenue, find top-selling pizzas, analyze trends, and more

STEPS

-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 categor 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 ordere per day.

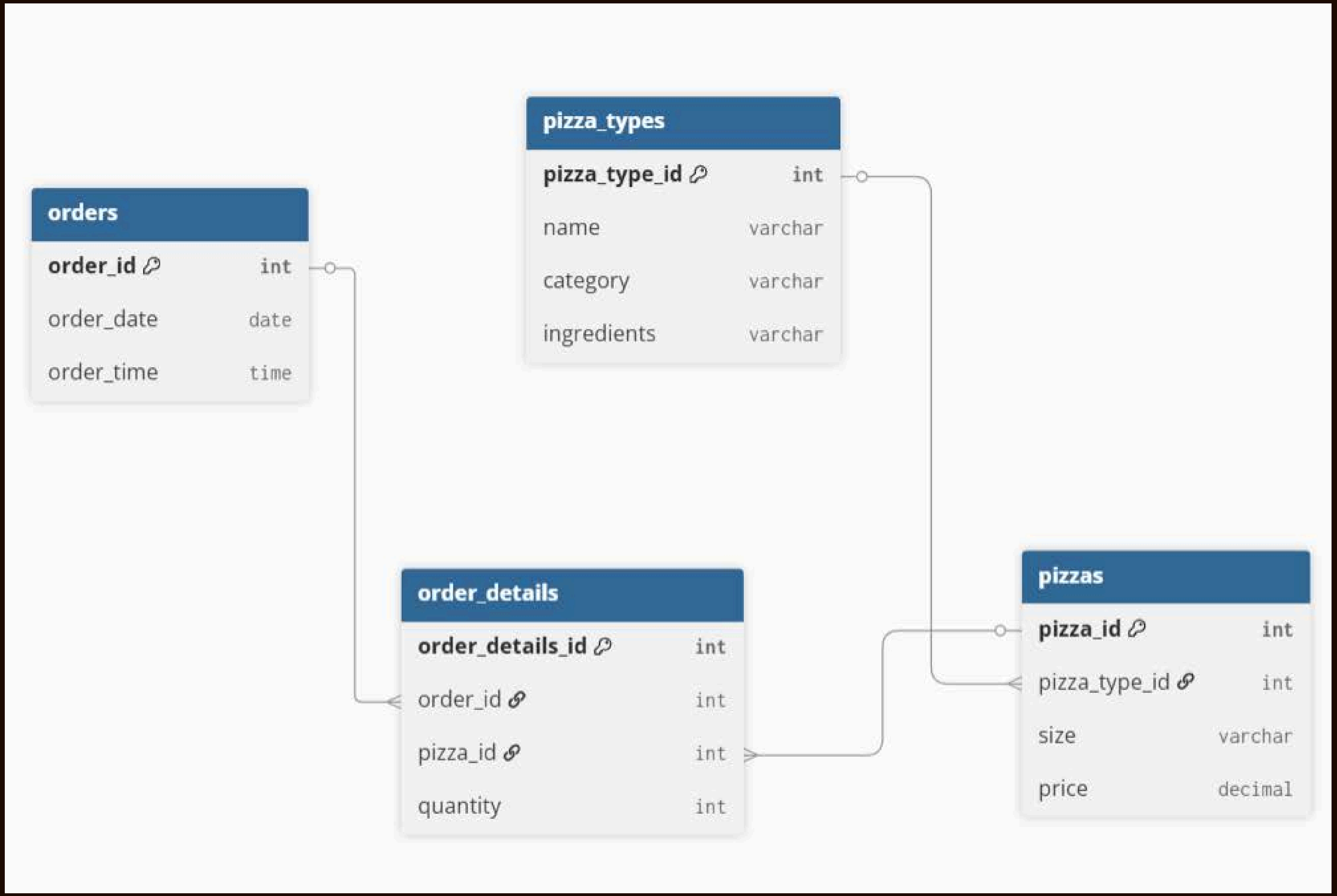
-Determine the top 3 most ordered pizza types based on revenue.

-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 pizz category.

SCHEMA



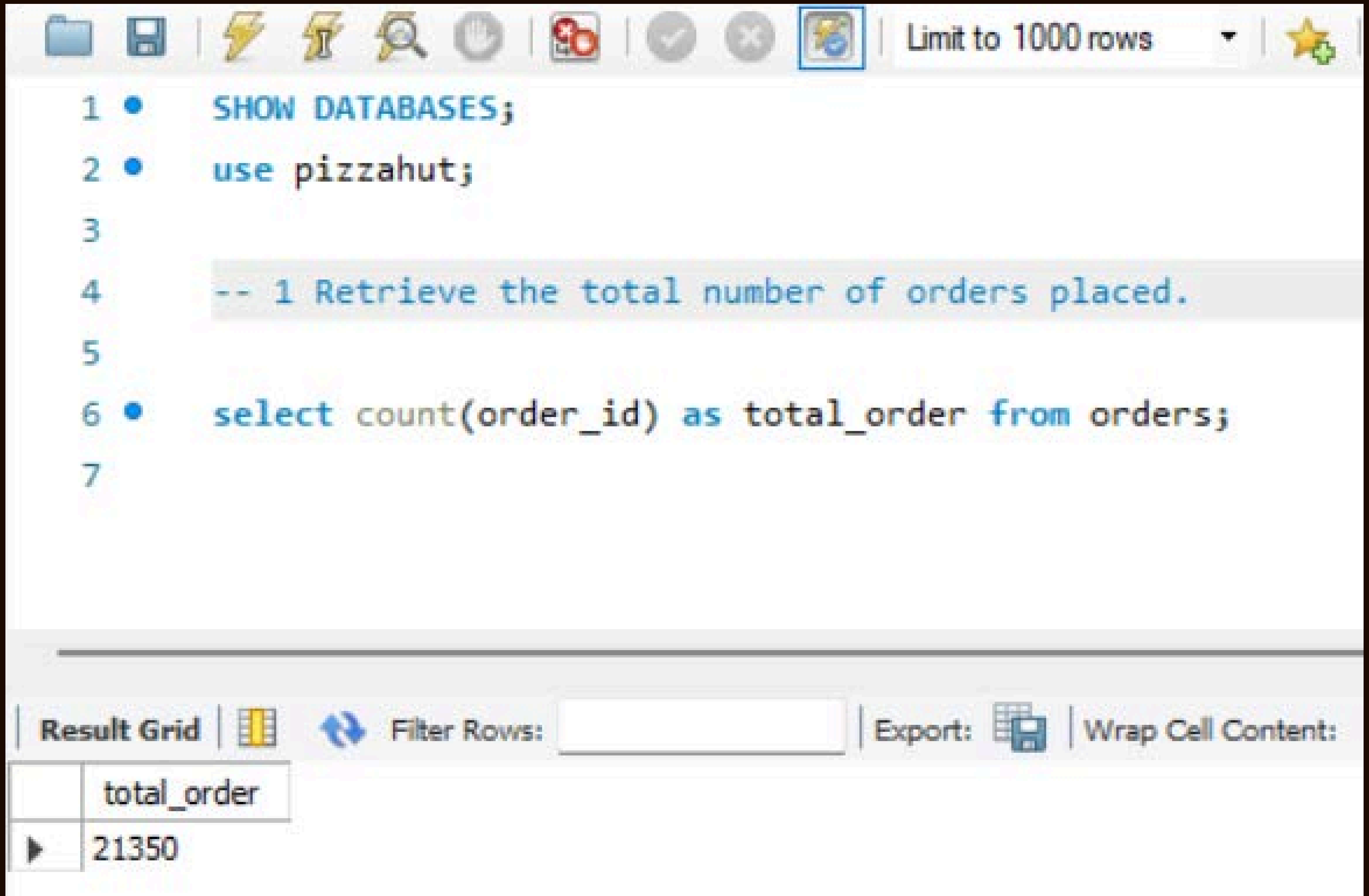
orders	
order_id	int
order_date	date
order_time	time

pizza_types	
pizza_type_id	int
name	varchar
category	varchar
ingredients	varchar

order_details	
order_details_id	int
order_id	int
pizza_id	int
quantity	int

pizzas	
pizza_id	int
pizza_type_id	int
size	varchar
price	decimal

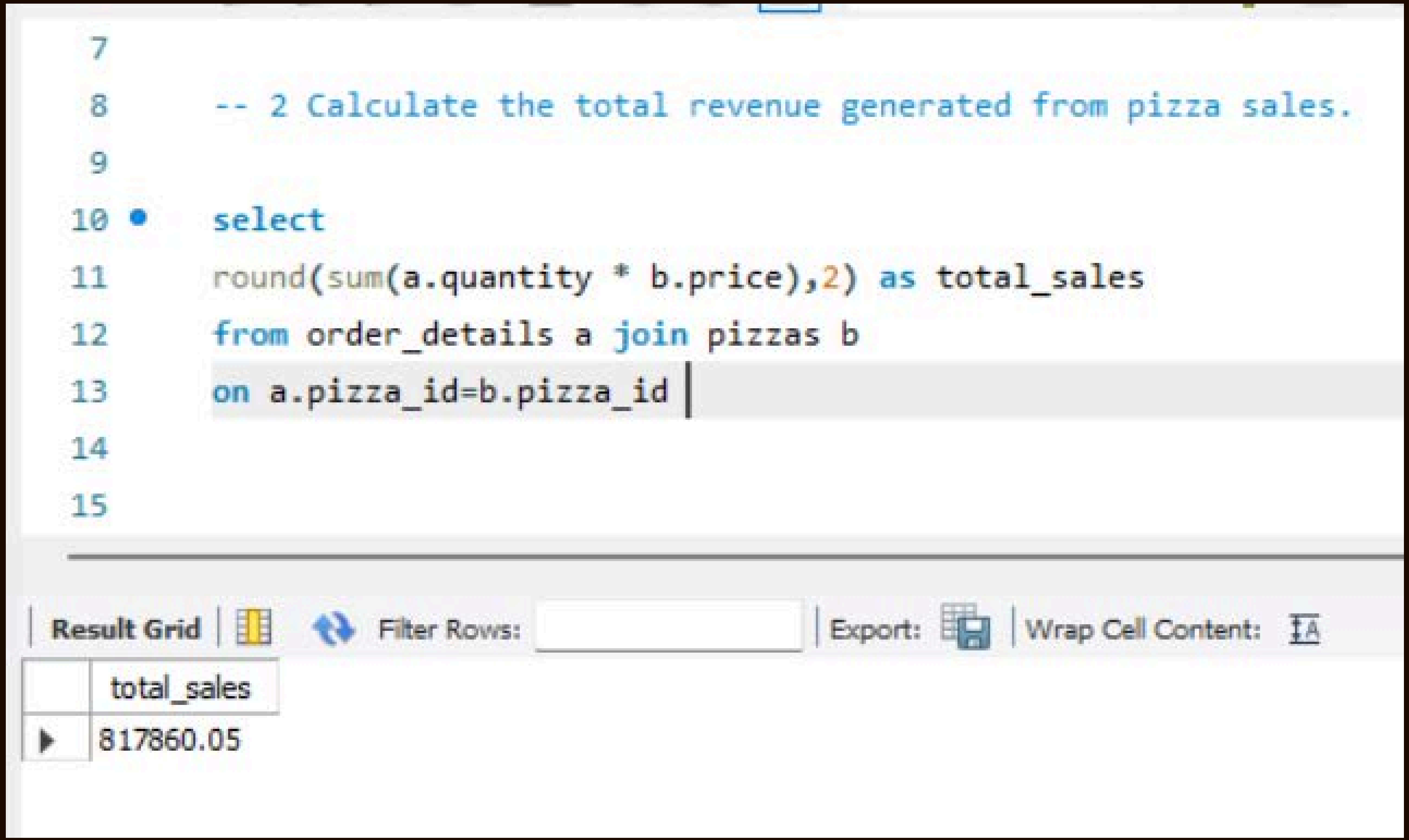
Retrieve the total number of orders placed.



```sql
SHOW DATABASES;
use pizzahut;

-- 1 Retrieve the total number of orders placed.

select count(order_id) as total_order from orders;

```

```
total_order
21350
```

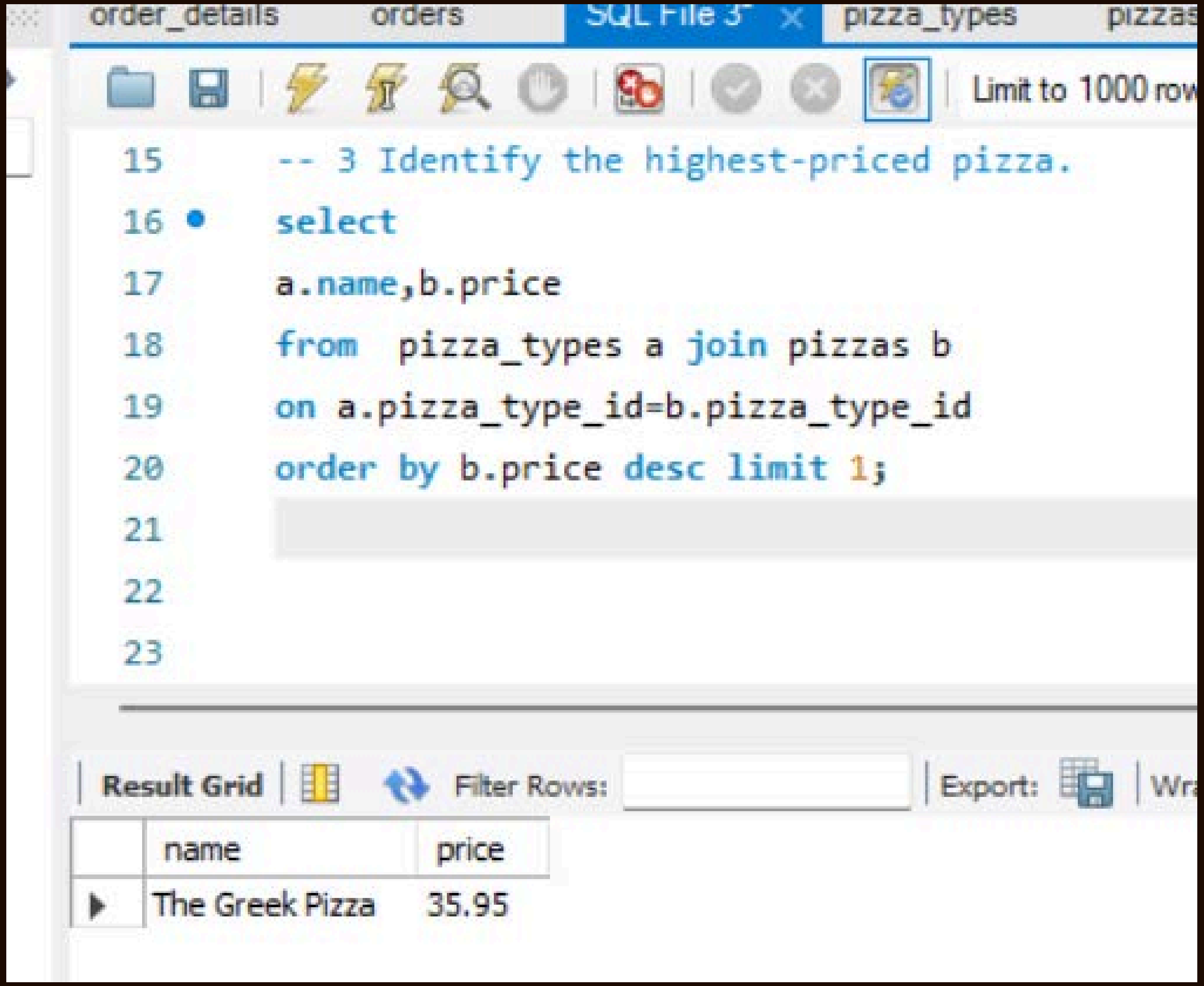
Calculate the total revenue generated from pizza sales.



```sql
-- 2 Calculate the total revenue generated from pizza sales.

select
round(sum(a.quantity * b.price),2) as total_sales
from order_details a join pizzas b
on a.pizza_id=b.pizza_id
```

```
total_sales
817860.05
```

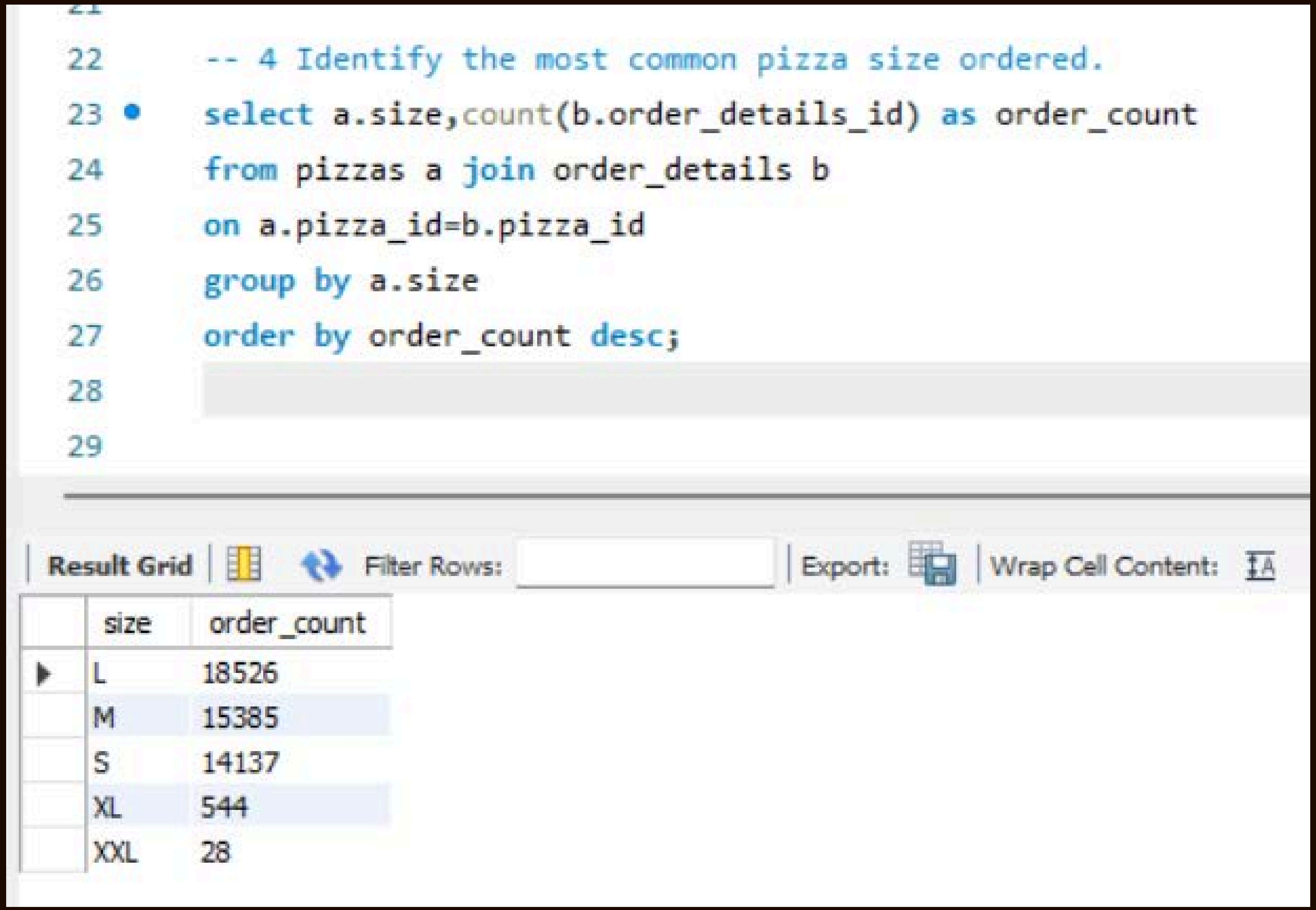
Identify the highest-priced pizza.



```sql
-- 3 Identify the highest-priced pizza.
select
a.name,b.price
from  pizza_types a join pizzas b
on a.pizza_type_id=b.pizza_type_id
order by b.price desc limit 1;
```

name	price
The Greek Pizza	35.95

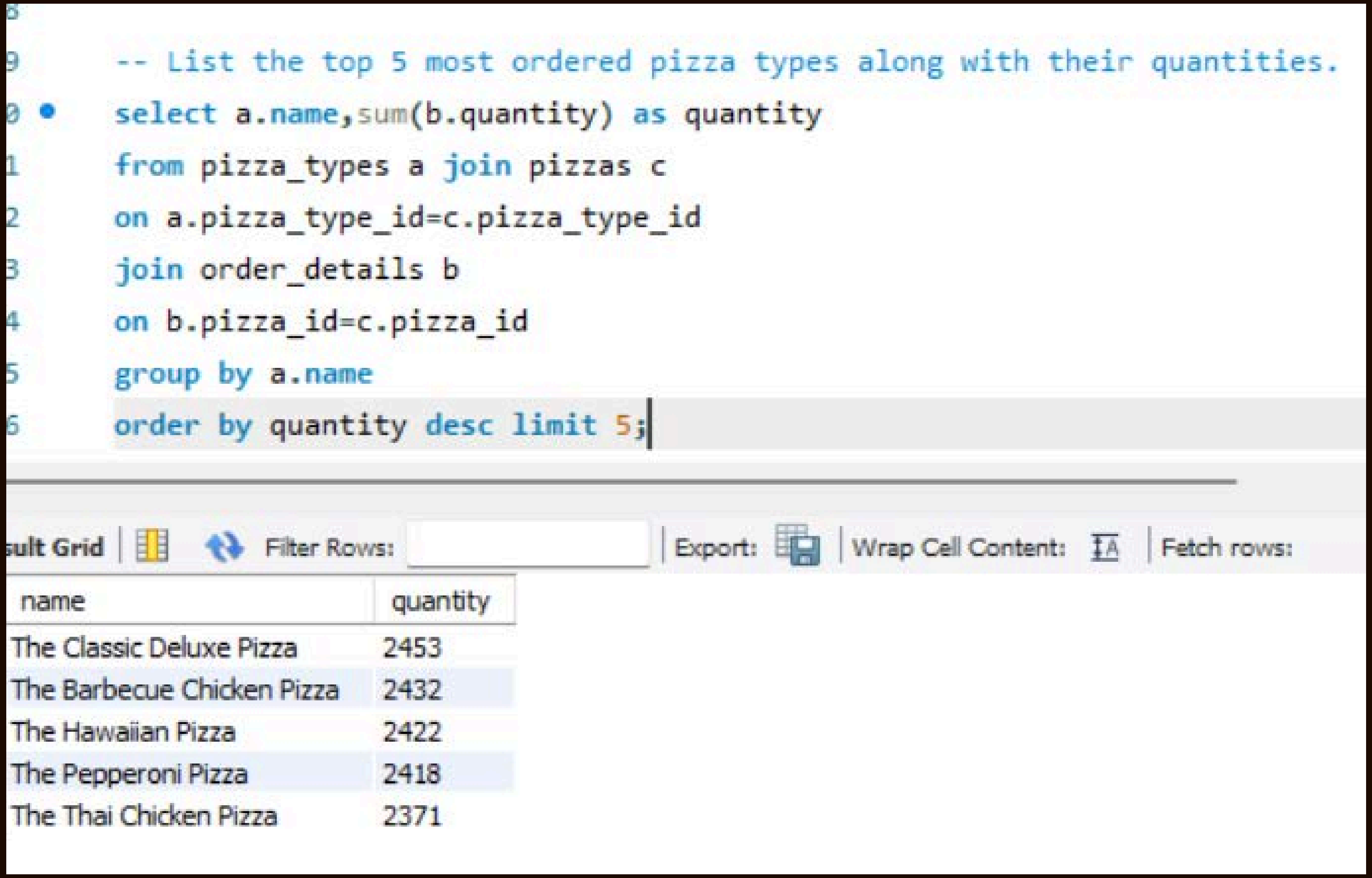
Identify the most common pizza size ordered.



```
-- 4 Identify the most common pizza size ordered.
select a.size,count(b.order_details_id) as order_count
from pizzas a join order_details b
on a.pizza_id=b.pizza_id
group by a.size
order by order_count desc;
```

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

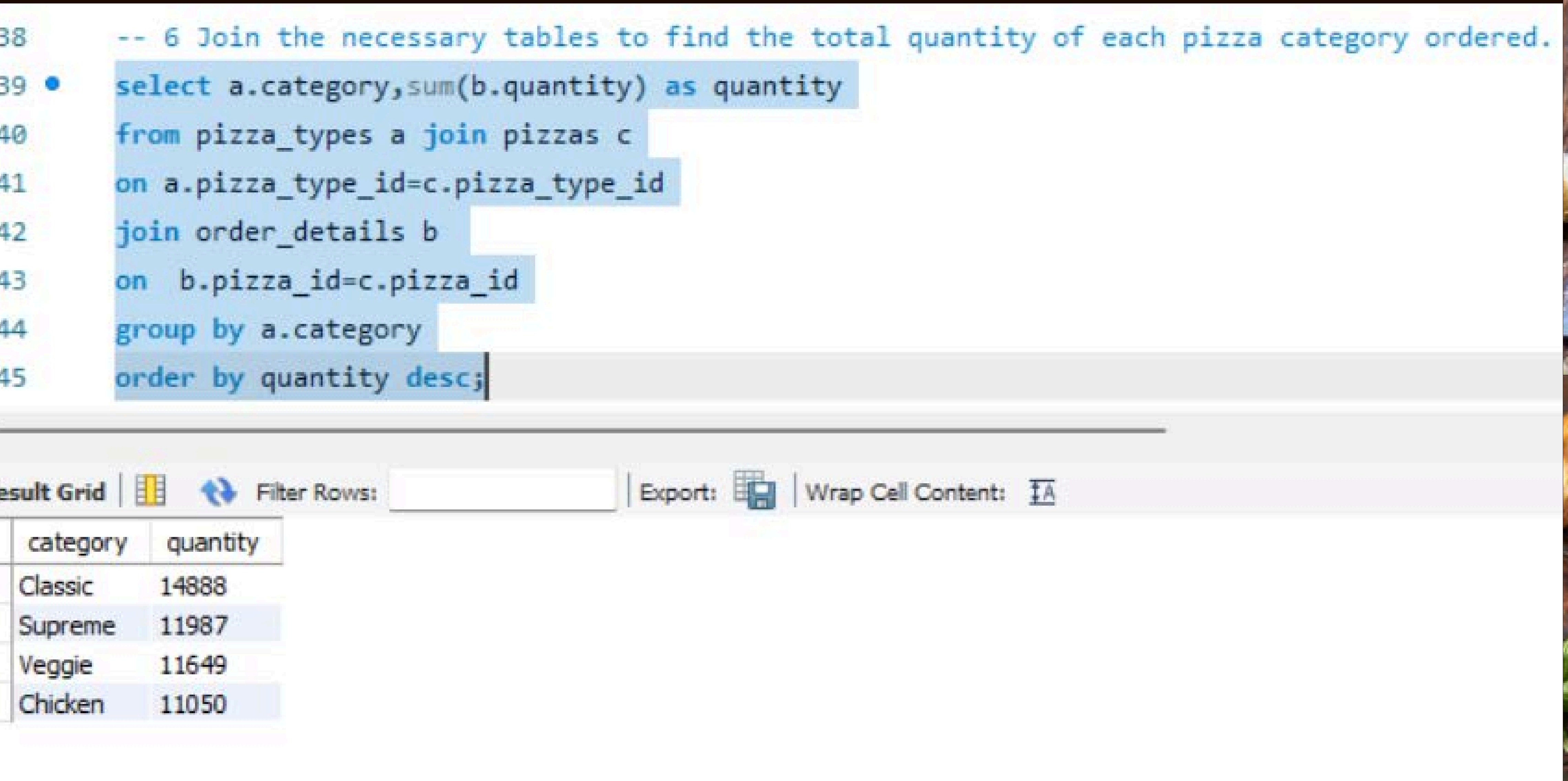
List the top 5 most ordered pizza types along with their quantities.



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

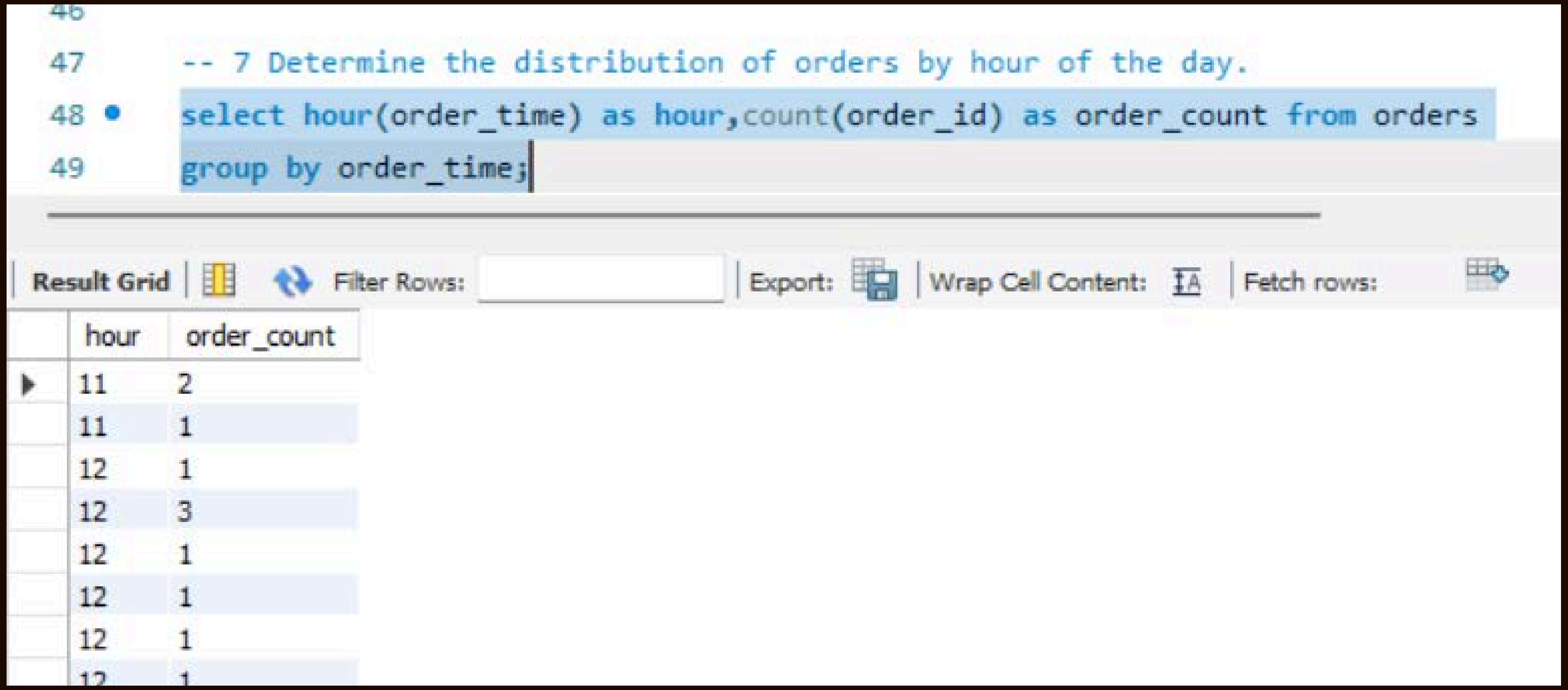
Join the necessary tables to find the total quantity of each pizza category ordered.



```sql
-- 6 Join the necessary tables to find the total quantity of each pizza category ordered.
select a.category,sum(b.quantity) as quantity
from pizza_types a join pizzas c
on a.pizza_type_id=c.pizza_type_id
join order_details b
on  b.pizza_id=c.pizza_id
group by a.category
order by quantity desc;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

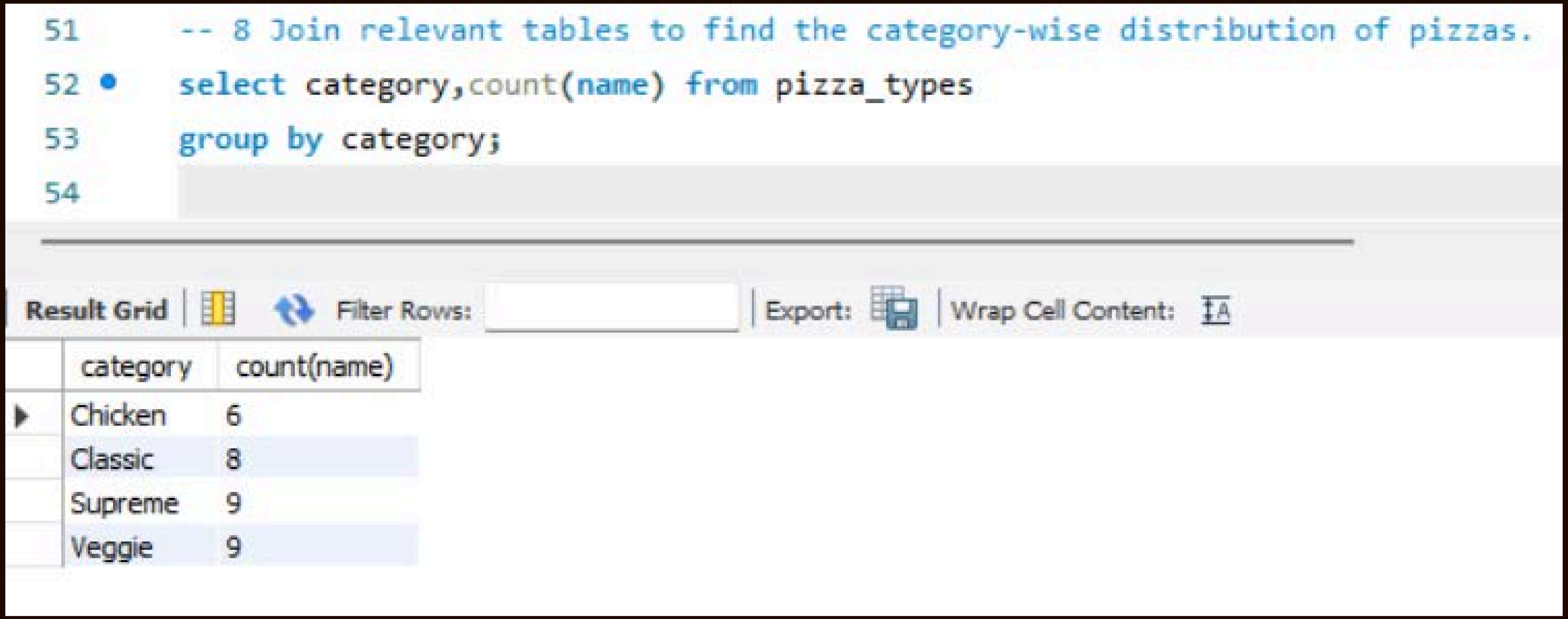
Determine the distribution of orders by hour of the day.



```sql
-- 7 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 order_time;
```

hour	order_count
11	2
11	1
12	1
12	3
12	1
12	1
12	1
12	1

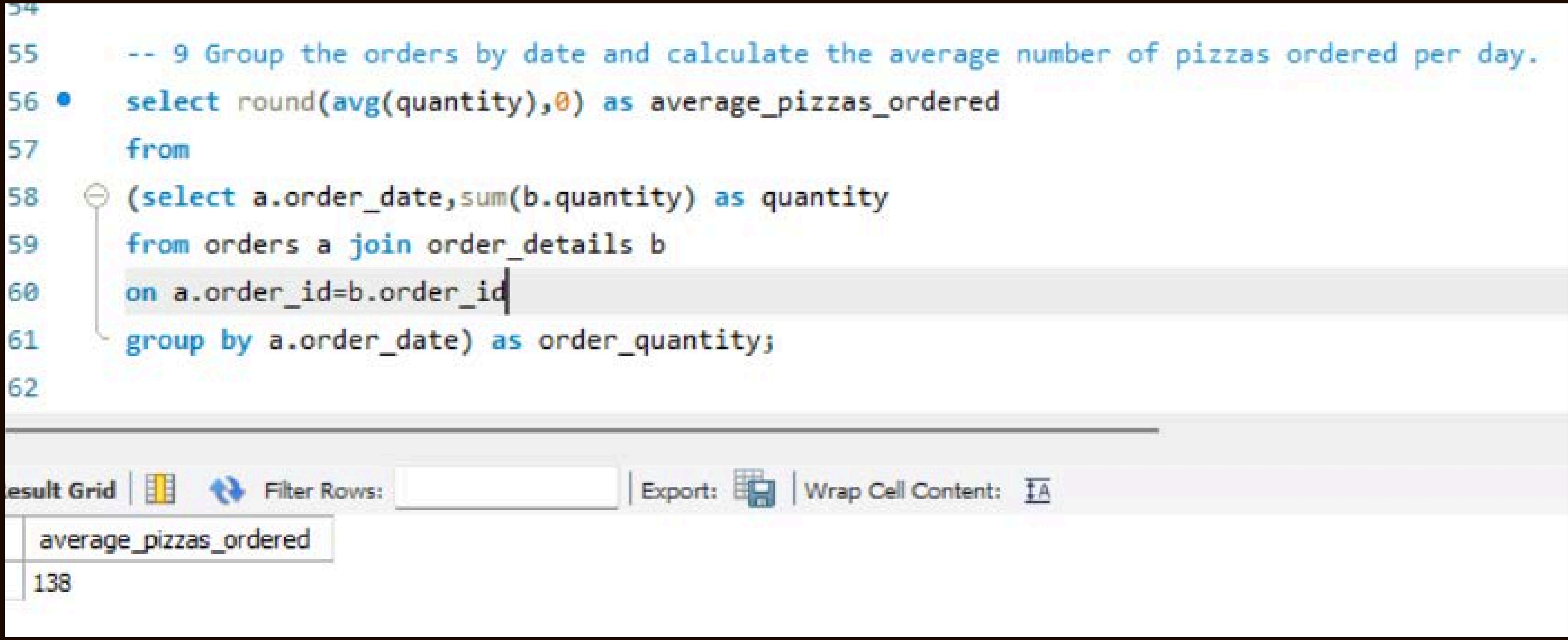
Join relevant tables to find the category-wise distribution of pizzas.



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

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

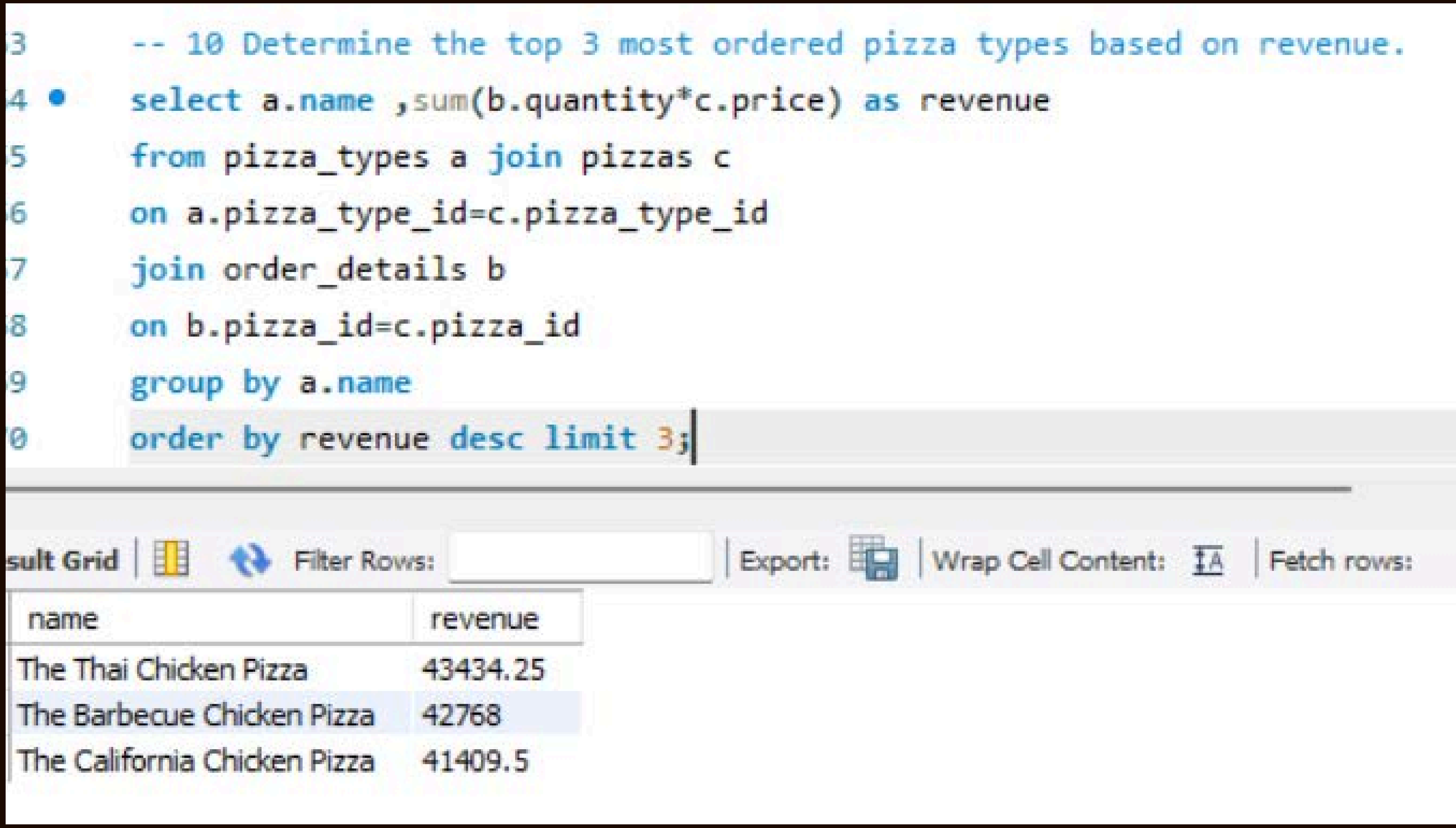
Group the orders by date and calculate the average number of pizzas ordered per day.



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

```
average_pizzas_ordered
138
```

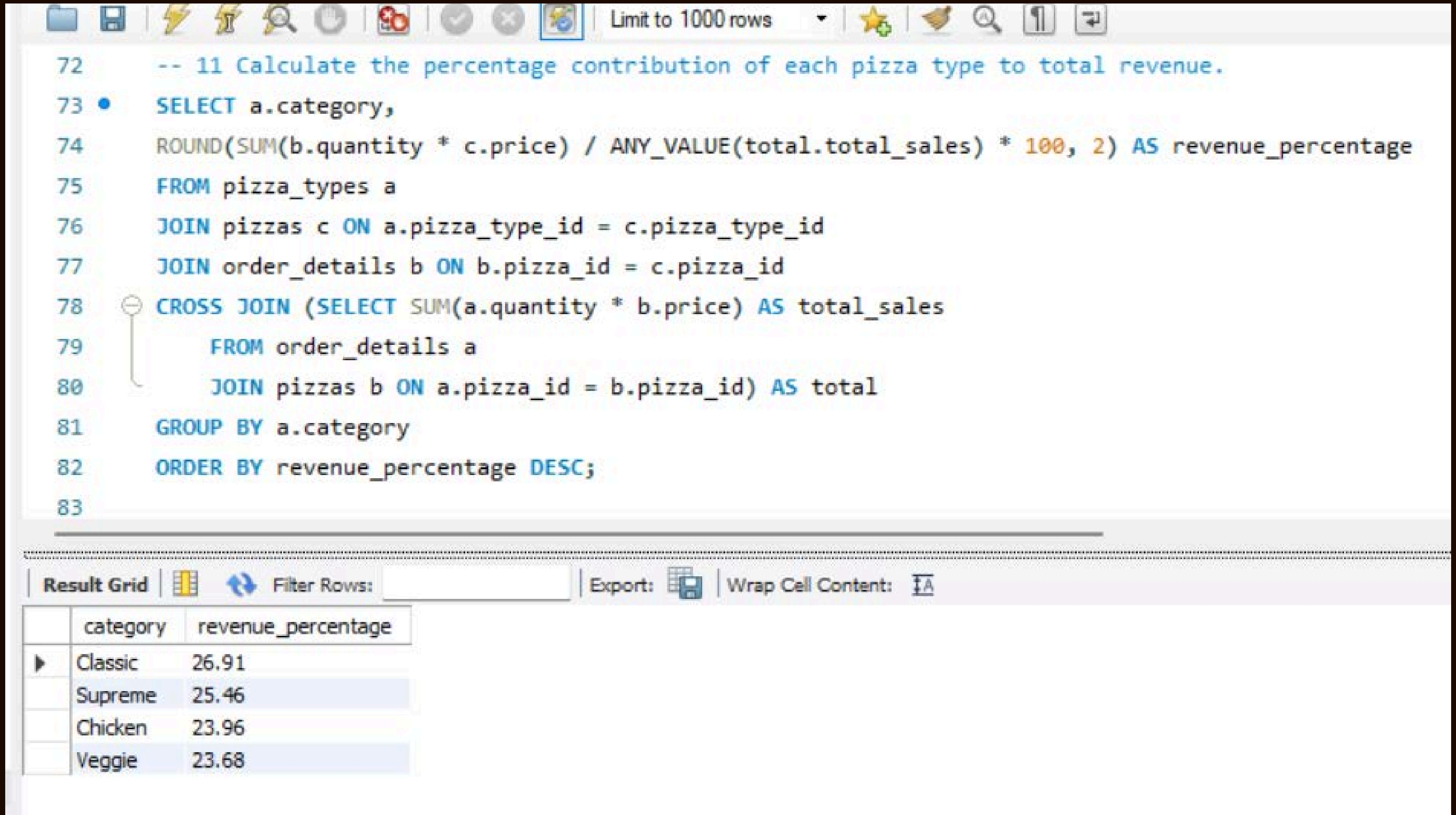
Determine the top 3 most ordered pizza types based on revenue.



```
-- 10 Determine the top 3 most ordered pizza types based on revenue.
select a.name ,sum(b.quantity*c.price) as revenue
from pizza_types a join pizzas c
on a.pizza_type_id=c.pizza_type_id
join order_details b
on b.pizza_id=c.pizza_id
group by a.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
```

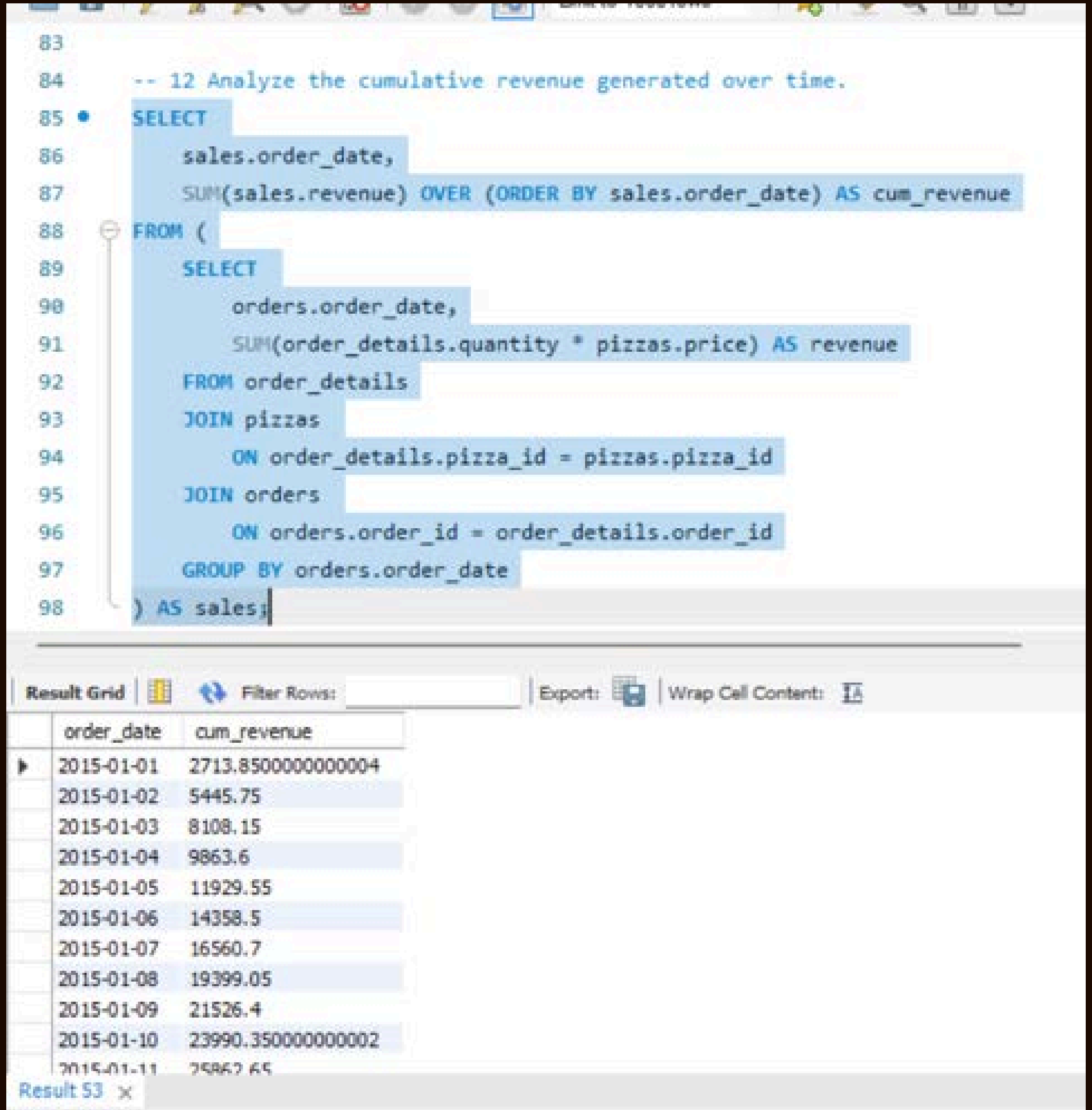
Calculate the percentage contribution of each pizza type to total revenue.



```sql
-- 11 Calculate the percentage contribution of each pizza type to total revenue.
SELECT a.category,
ROUND(SUM(b.quantity * c.price) / ANY_VALUE(total.total_sales) * 100, 2) AS revenue_percentage
FROM pizza_types a
JOIN pizzas c ON a.pizza_type_id = c.pizza_type_id
JOIN order_details b ON b.pizza_id = c.pizza_id
CROSS JOIN (SELECT SUM(a.quantity * b.price) AS total_sales
    FROM order_details a
    JOIN pizzas b ON a.pizza_id = b.pizza_id) AS total
GROUP BY a.category
ORDER BY revenue_percentage DESC;
```

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

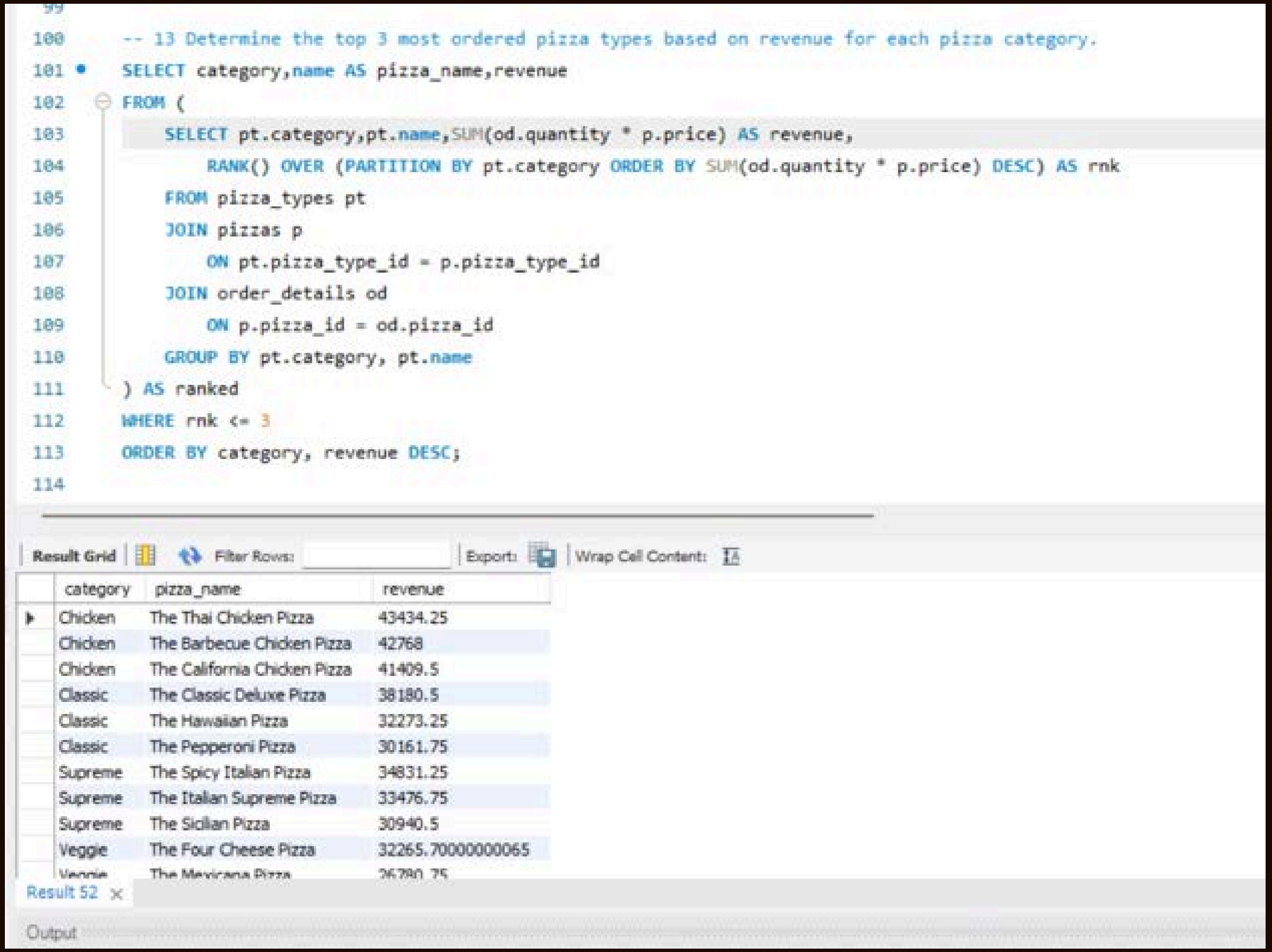
Analyze the cumulative revenue generated over time.



```sql
-- 12 Analyze the cumulative revenue generated over time.
SELECT
    sales.order_date,
    SUM(sales.revenue) OVER (ORDER BY sales.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
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

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



```sql
-- 13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT category,name AS pizza_name,revenue
FROM (
    SELECT pt.category,pt.name,SUM(od.quantity * p.price) AS revenue,
        RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS rnk
    FROM pizza_types pt
    JOIN pizzas p
        ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od
        ON p.pizza_id = od.pizza_id
    GROUP BY pt.category, pt.name
) AS ranked
WHERE rnk <= 3
ORDER BY category, revenue DESC;
```

category	pizza_name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75

THANK YOU

tapasparida7077@gmail.com