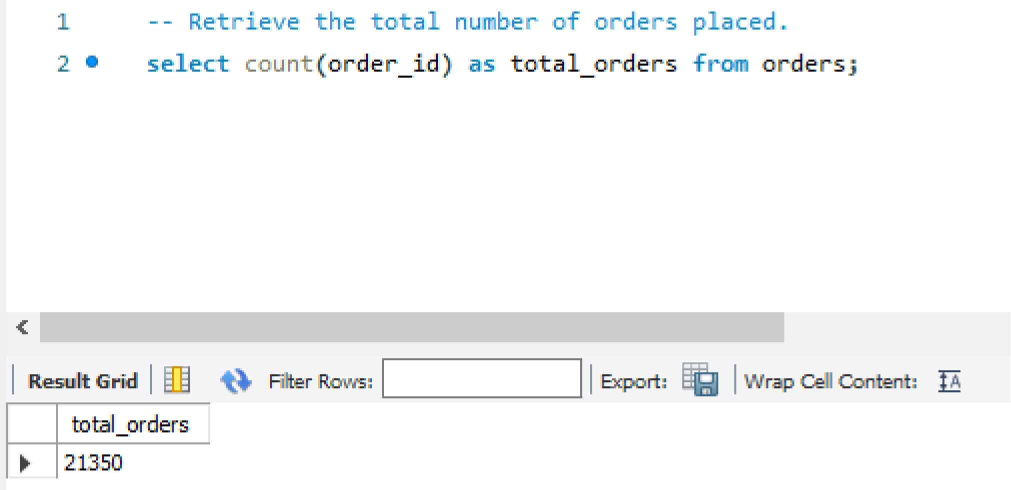
Pizza Sales SQL Queries

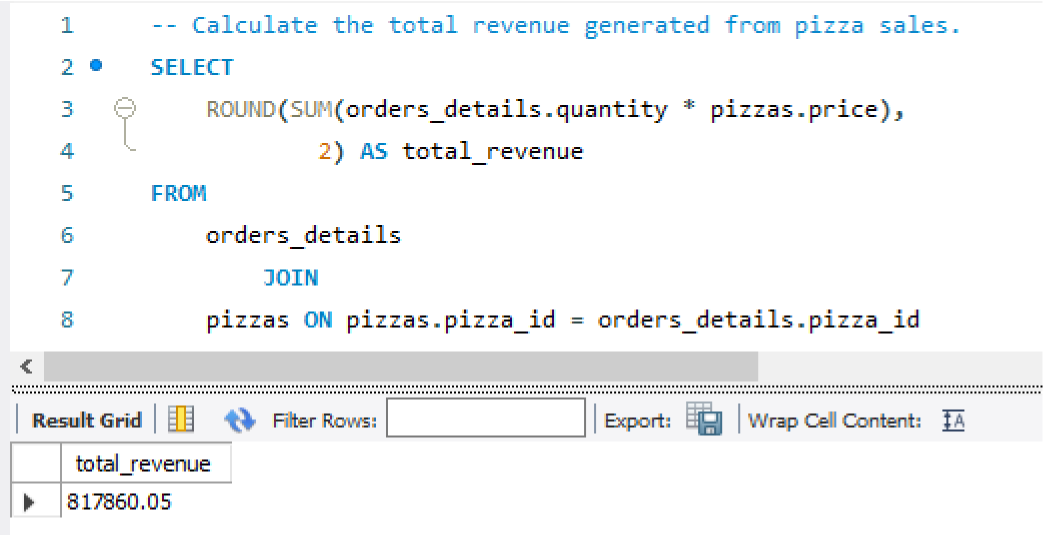
1. Retrieve the total number of orders placed.



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

total_orders
21350

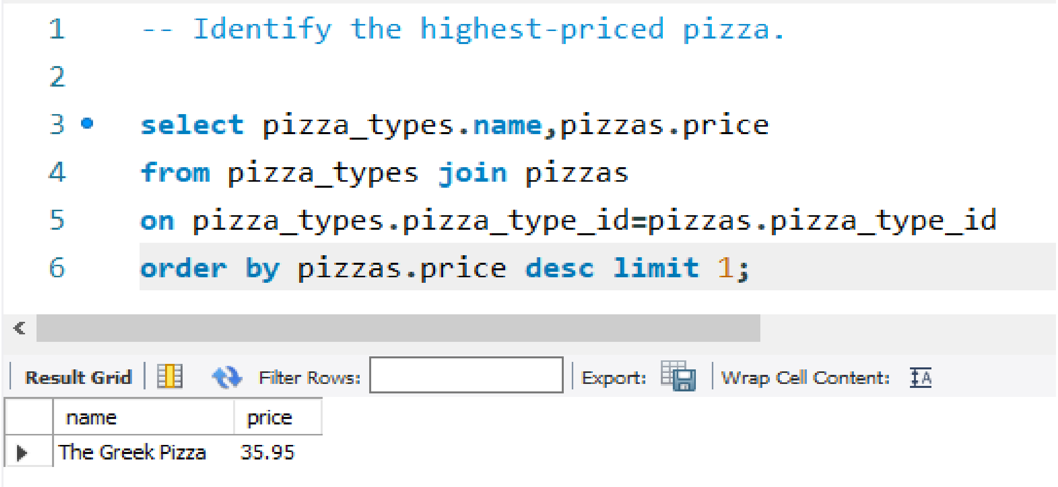
2. Calculate the total revenue generated from pizza sales.



```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

total_revenue
817860.05

3. Identify the highest-priced pizza.

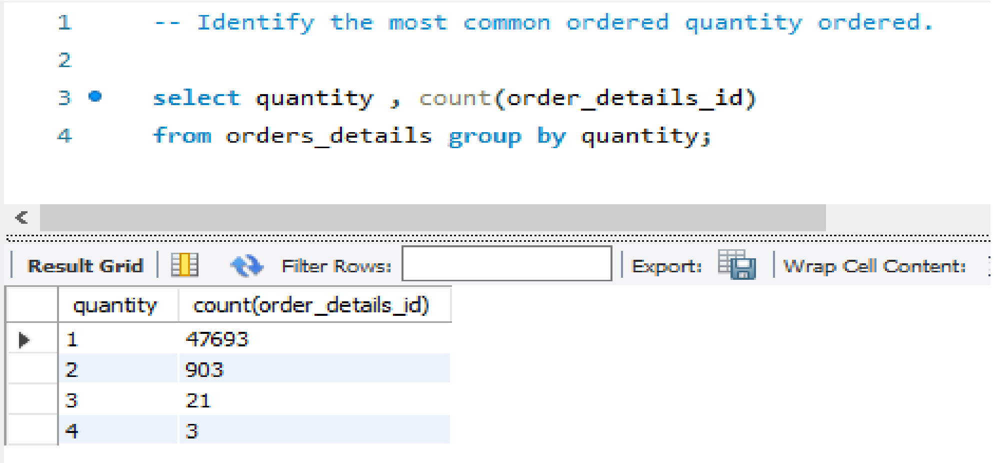


```sql
-- Identify the highest-priced pizza.

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 ordered quantity ordered.

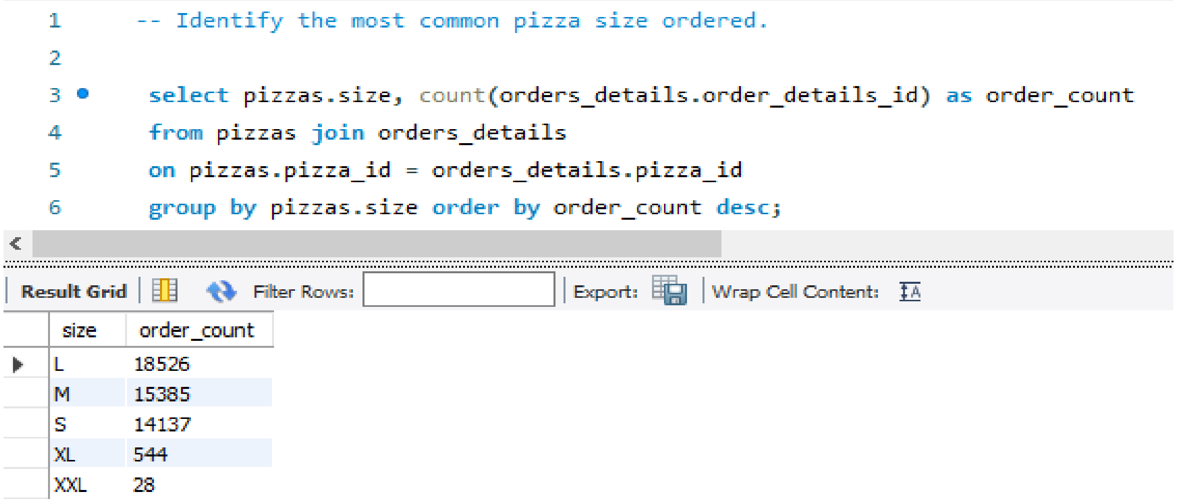


```sql
-- Identify the most common ordered quantity ordered.

select quantity , count(order_details_id)
from orders_details group by quantity;
```

quantity	count(order_details_id)
1	47693
2	903
3	21
4	3

5. Identify the most common pizza size ordered.

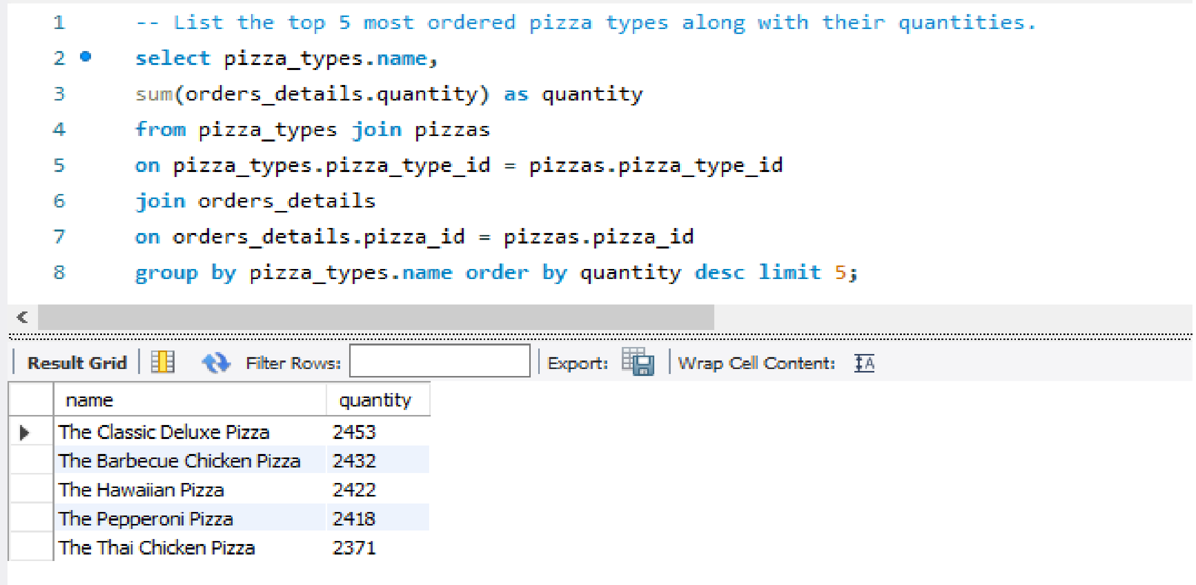


```sql
-- Identify the most common pizza size ordered.

select pizzas.size, count(orders_details.order_details_id) as order_count
from pizzas join orders_details
on pizzas.pizza_id = orders_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

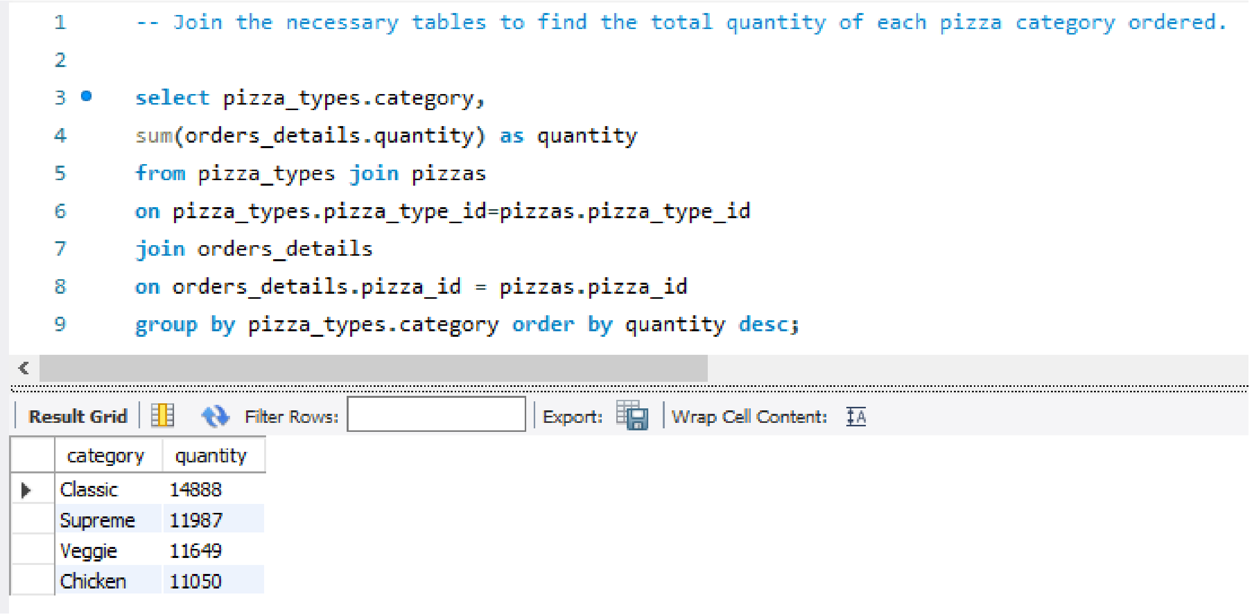
6. List the top 5 most ordered pizza types along with their quantities.



```sql
-- List the top 5 most ordered pizza types along with their quantities.
select pizza_types.name,
sum(orders_details.quantity) 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.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

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

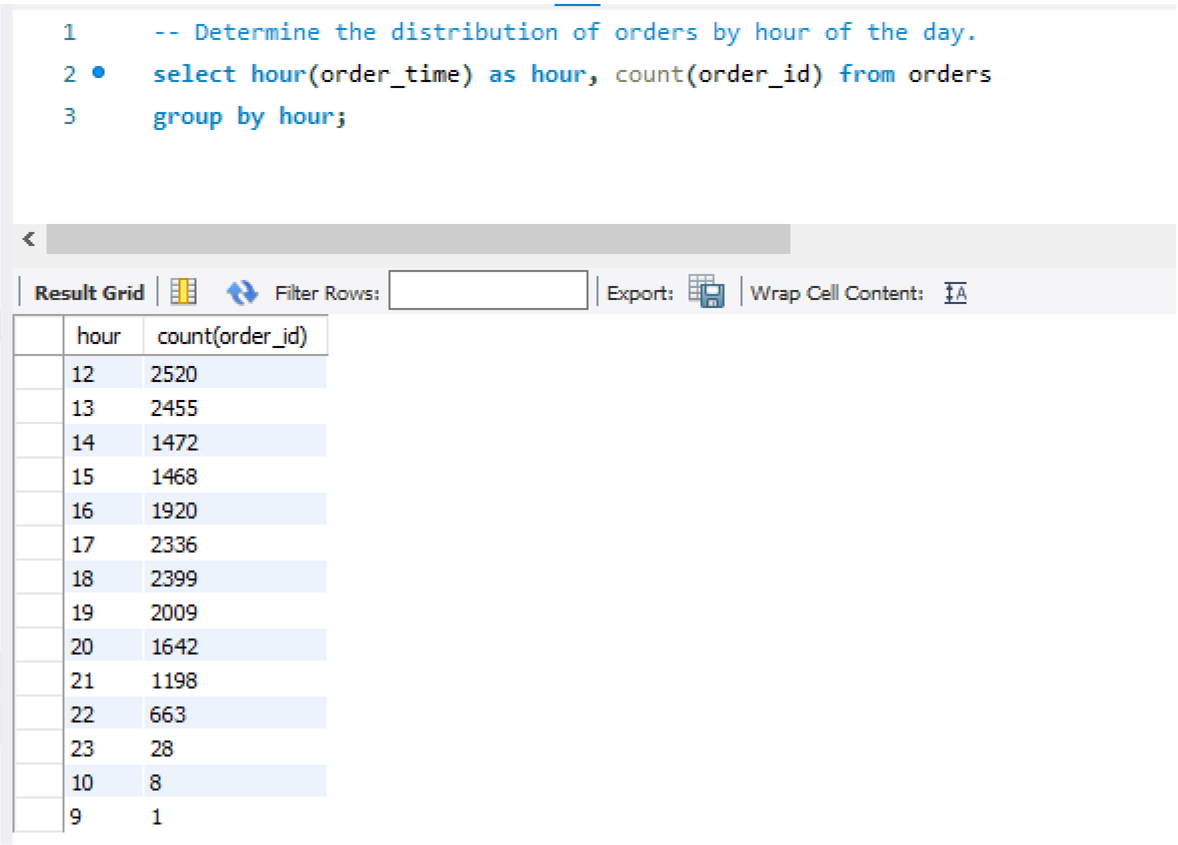


```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

select pizza_types.category,
sum(orders_details.quantity) 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 desc;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

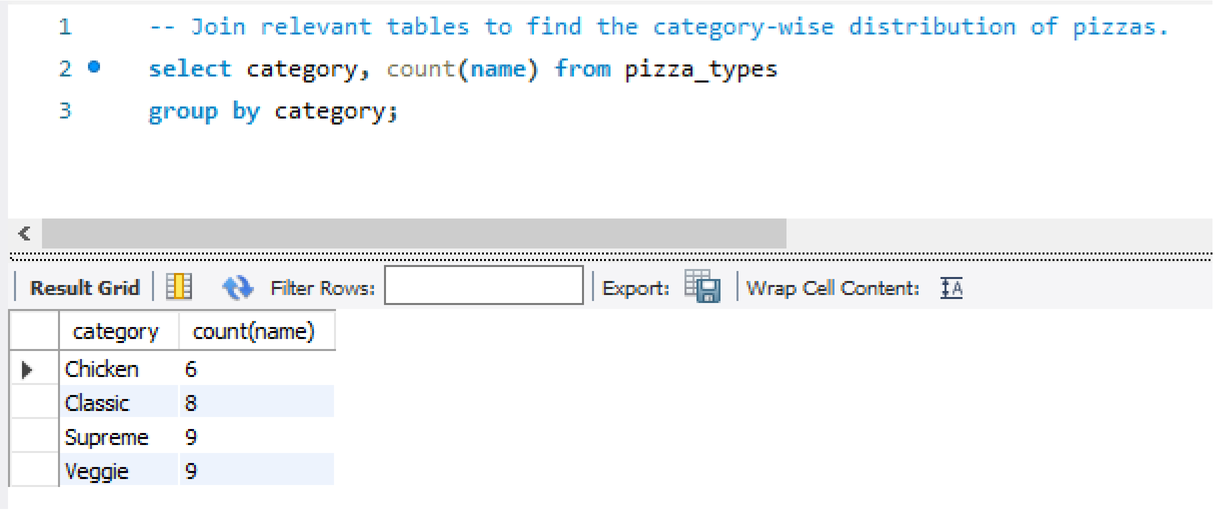
8. Determine the distribution of orders by hour of the day.



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

hour	count(order_id)
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

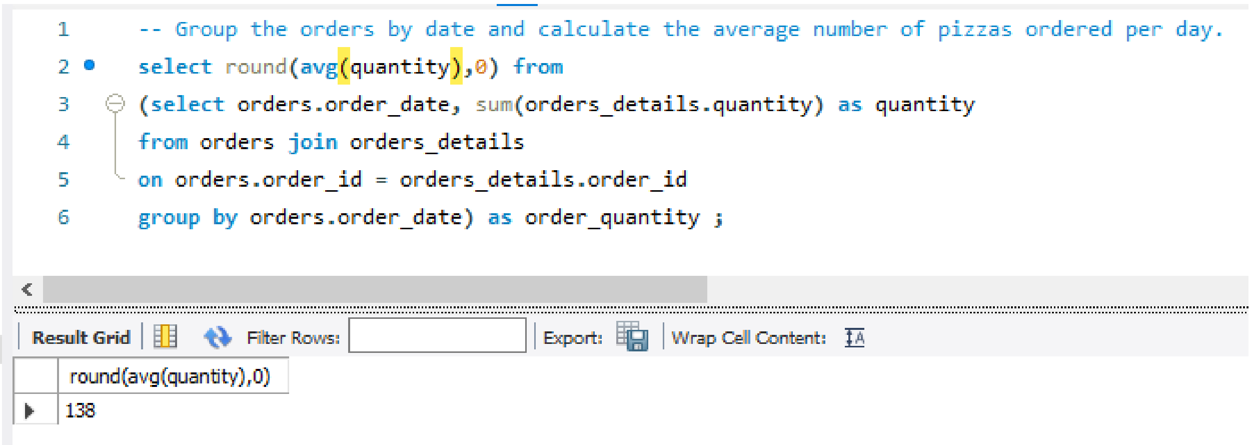
9. Join relevant tables to find the category-wise distribution of pizzas.



```sql
-- 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

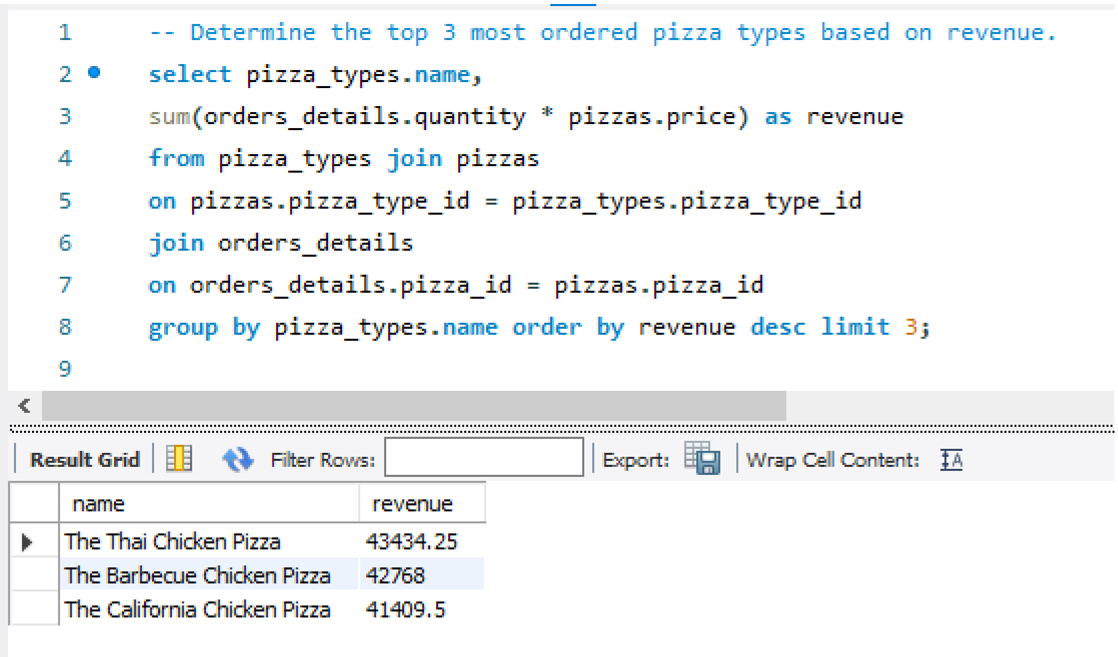
10. Group the orders by date and calculate the average number of pizzas ordered per day.



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

round(avg(quantity),0)
138

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

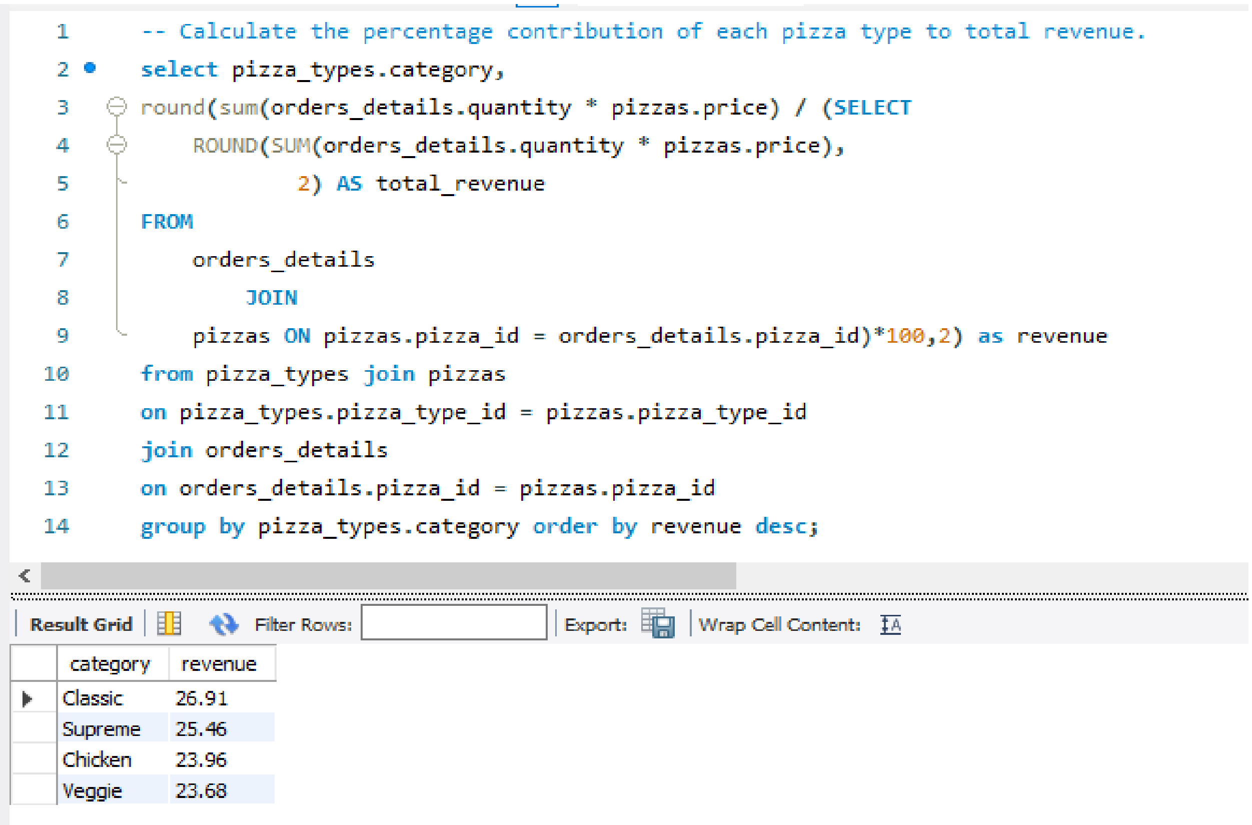


```sql
-- Determine the top 3 most ordered pizza types based on revenue.
select pizza_types.name,
sum(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group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

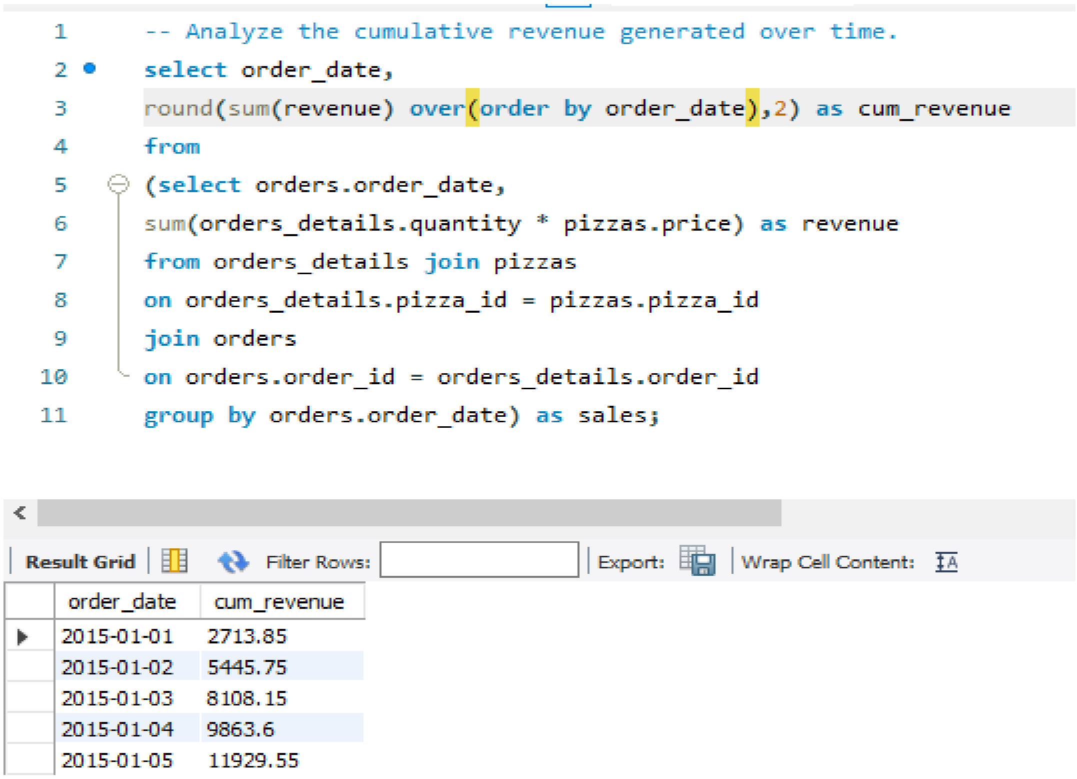
12. Calculate the percentage contribution of each pizza type to total revenue.



```
-- Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category,
round(sum(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id)*100,2) 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 order by revenue desc;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

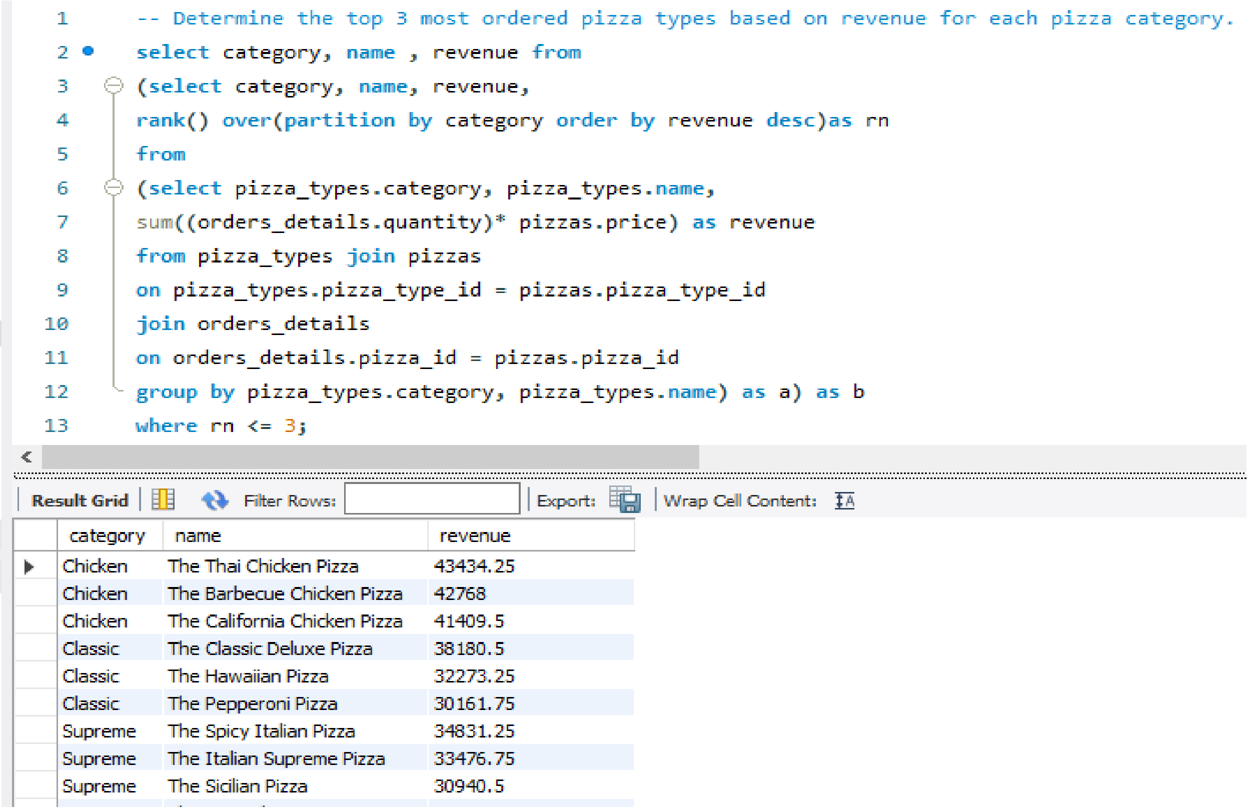
13. Analyze the cumulative revenue generated over time.



```
-- Analyze the cumulative revenue generated over time.
select order_date,
round(sum(revenue) over(order by order_date),2) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * 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	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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



```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select category, 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((orders_details.quantity)* 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;
```

category	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