SQL PROJECT ON PIZZA SALES

SQL PROJECT FOR DATA ANALYSIS

PRESENTED BY:
Ashwini Gaikwad

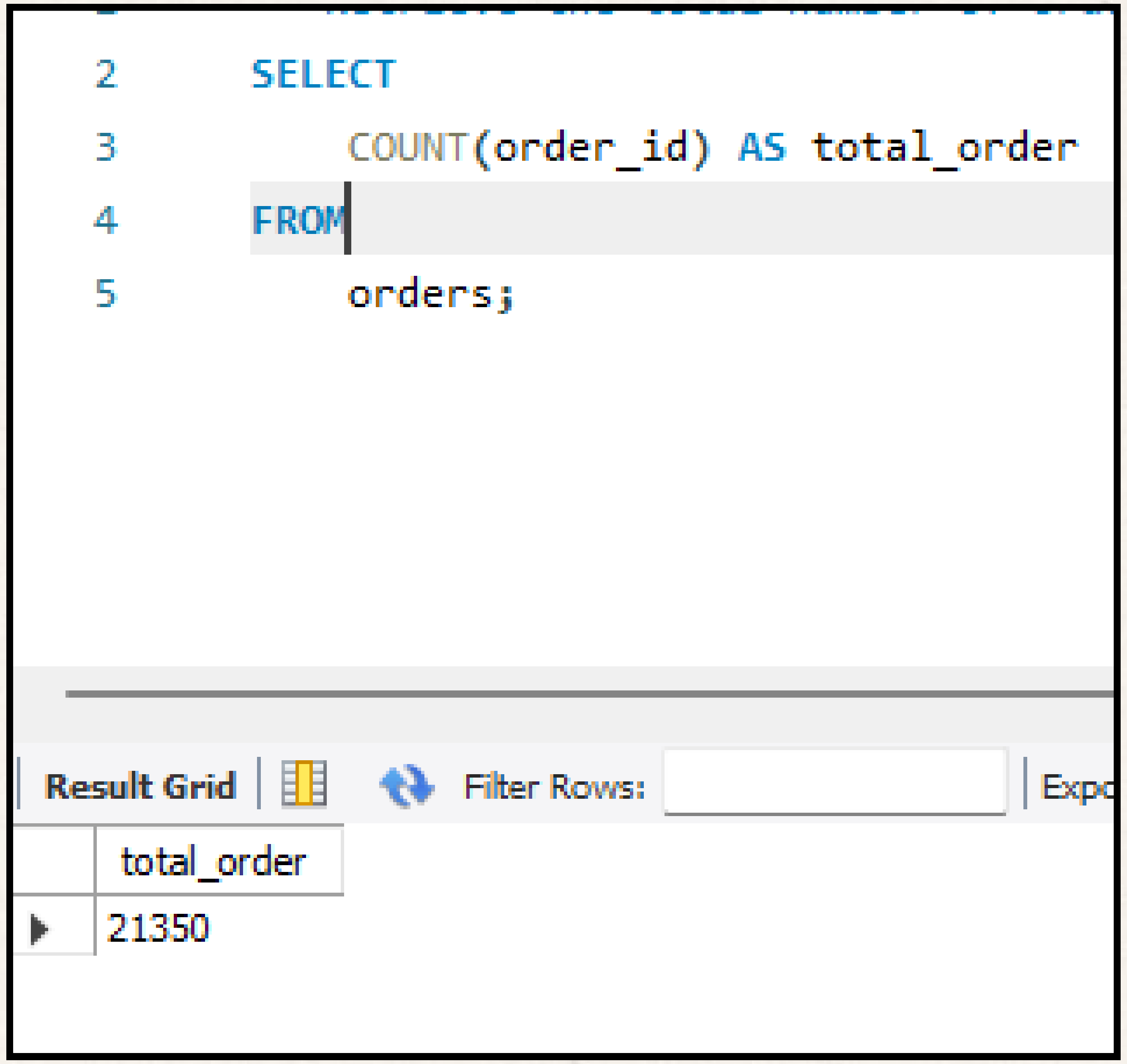
Hello!

I have utilize SQL queries using MySQL to solve questions that were related to Pizza Sales data.

Question

Retrieve the total number of orders placed.

SQL Query



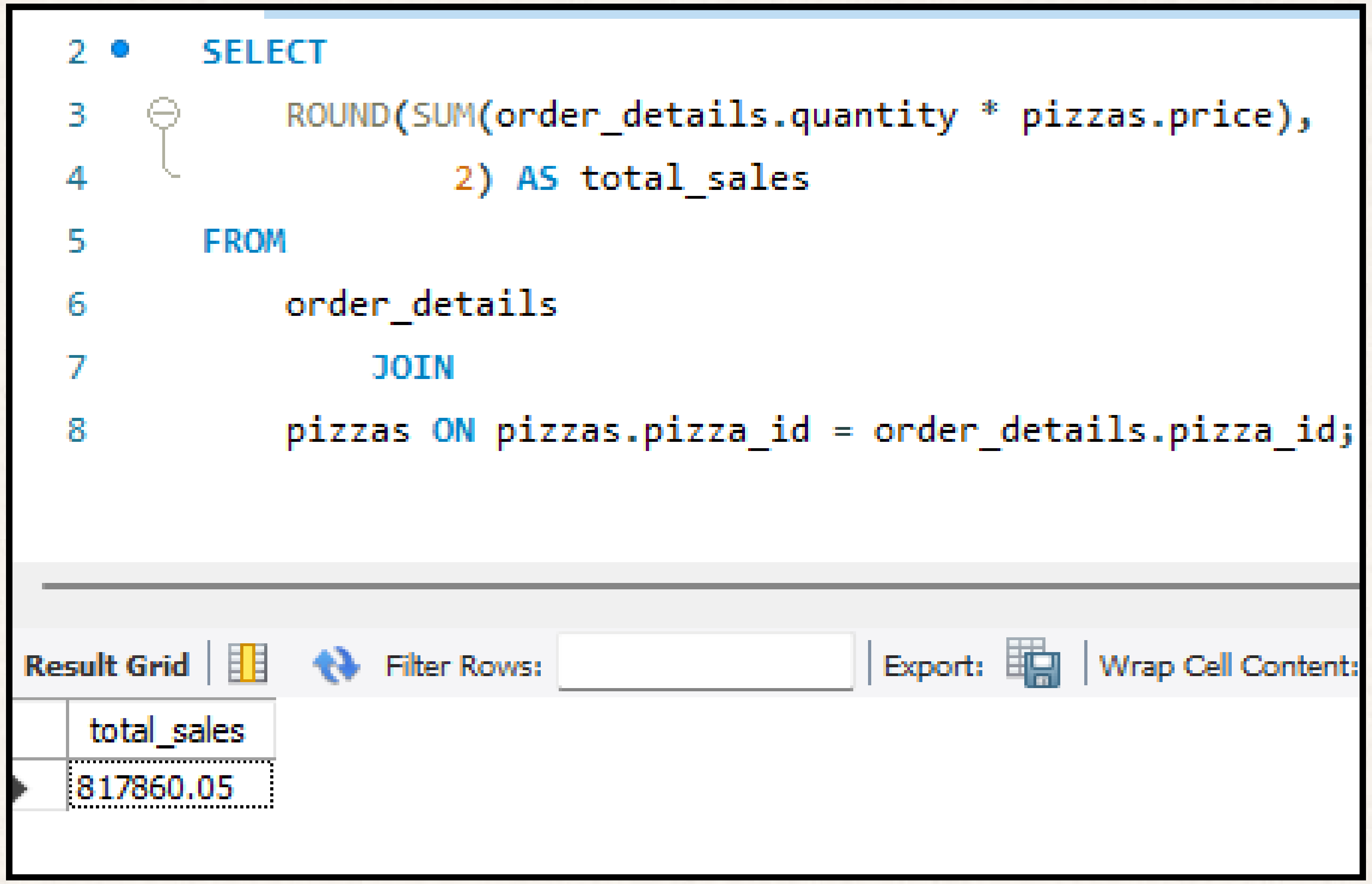
```
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```

```
total_order
21350
```

Question

Calculate the total revenue generated from pizza sales.

SQL Query



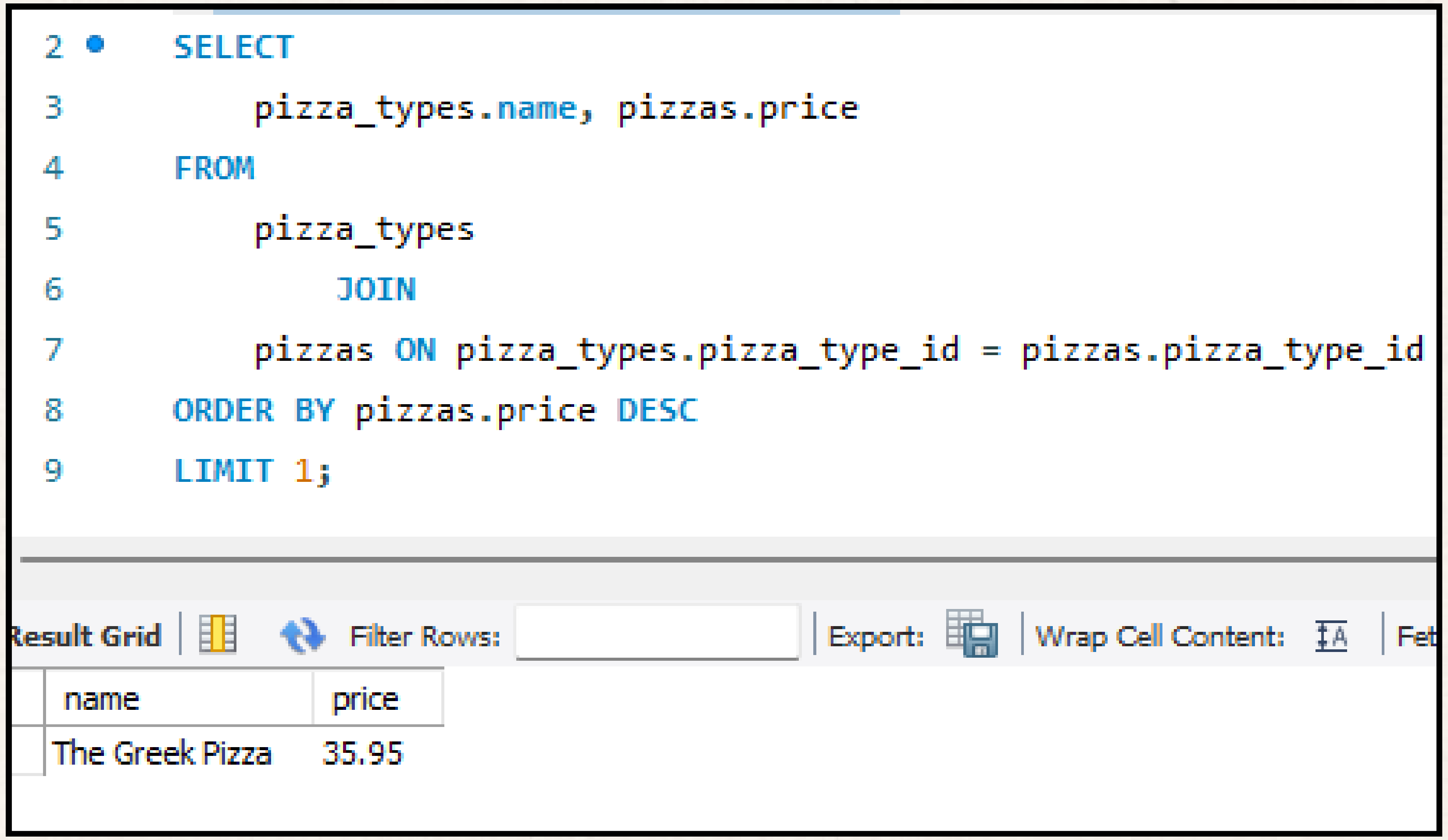
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

```
total_sales
817860.05
```

Question

Identify the highest-priced pizza.

SQL Query



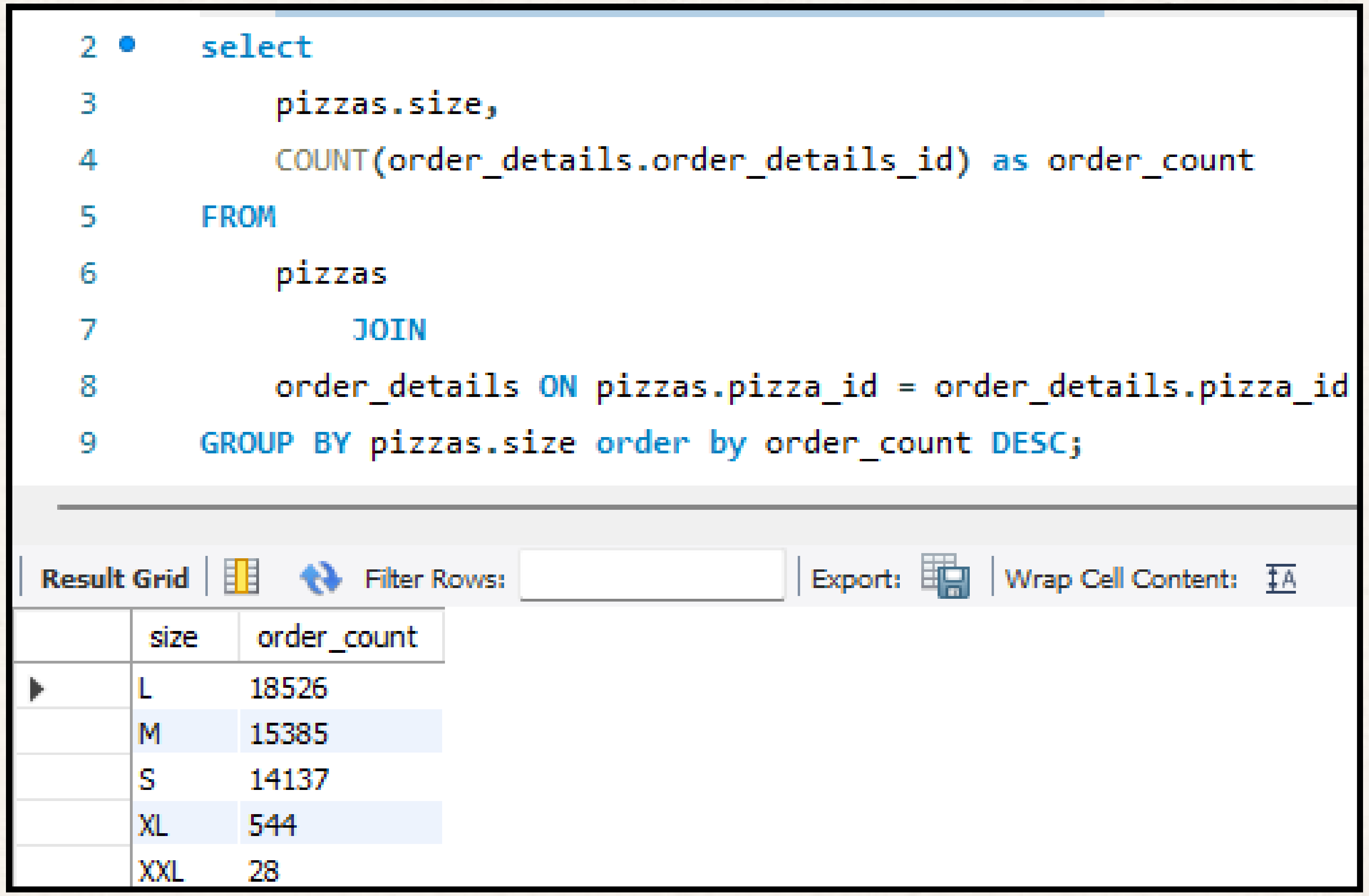
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95

Question

Identify the most common pizza size ordered.

SQL Query



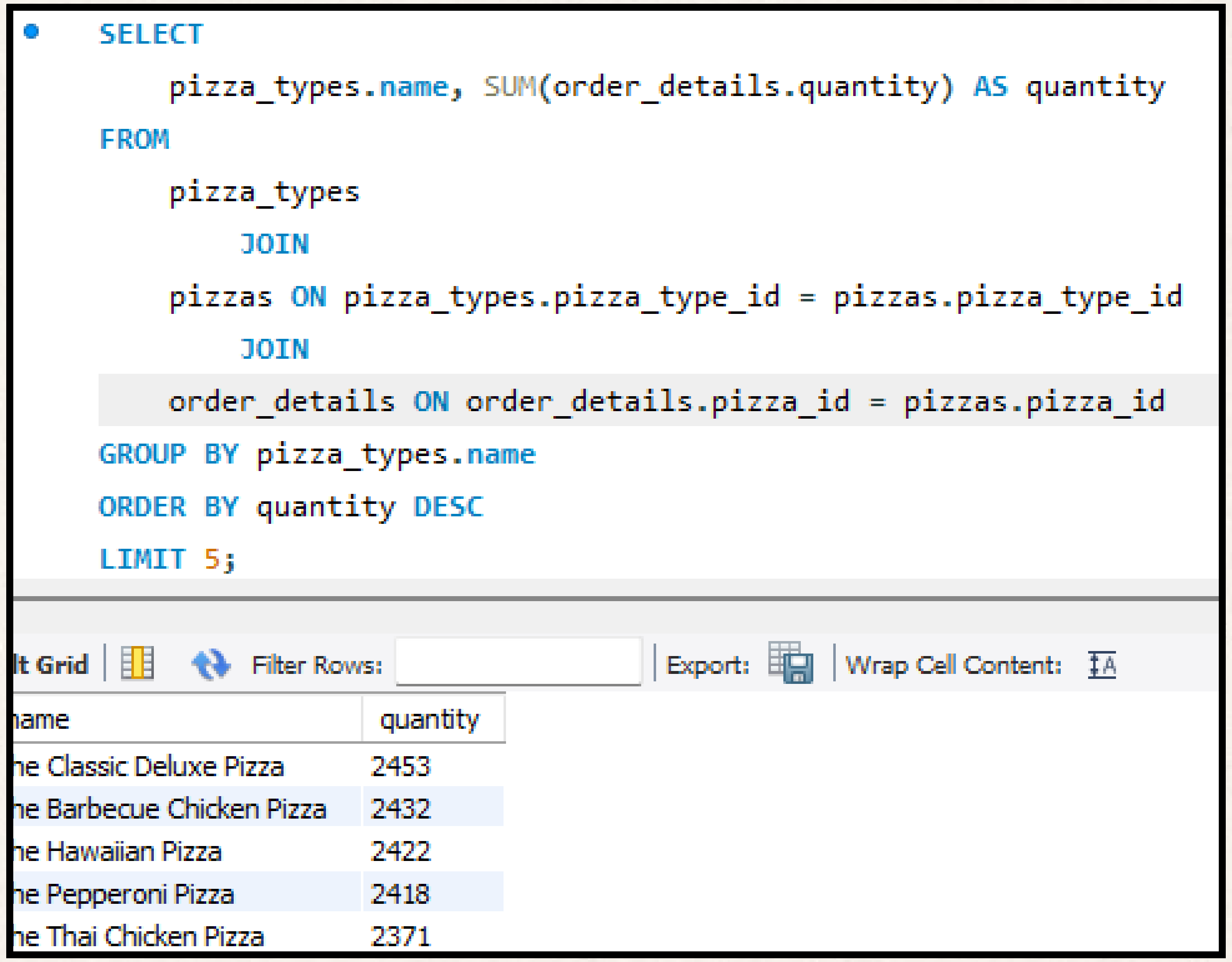
```
select
    pizzas.size,
    COUNT(order_details.order_details_id) as order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size order by order_count DESC;
```

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

Question

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

SQL Query



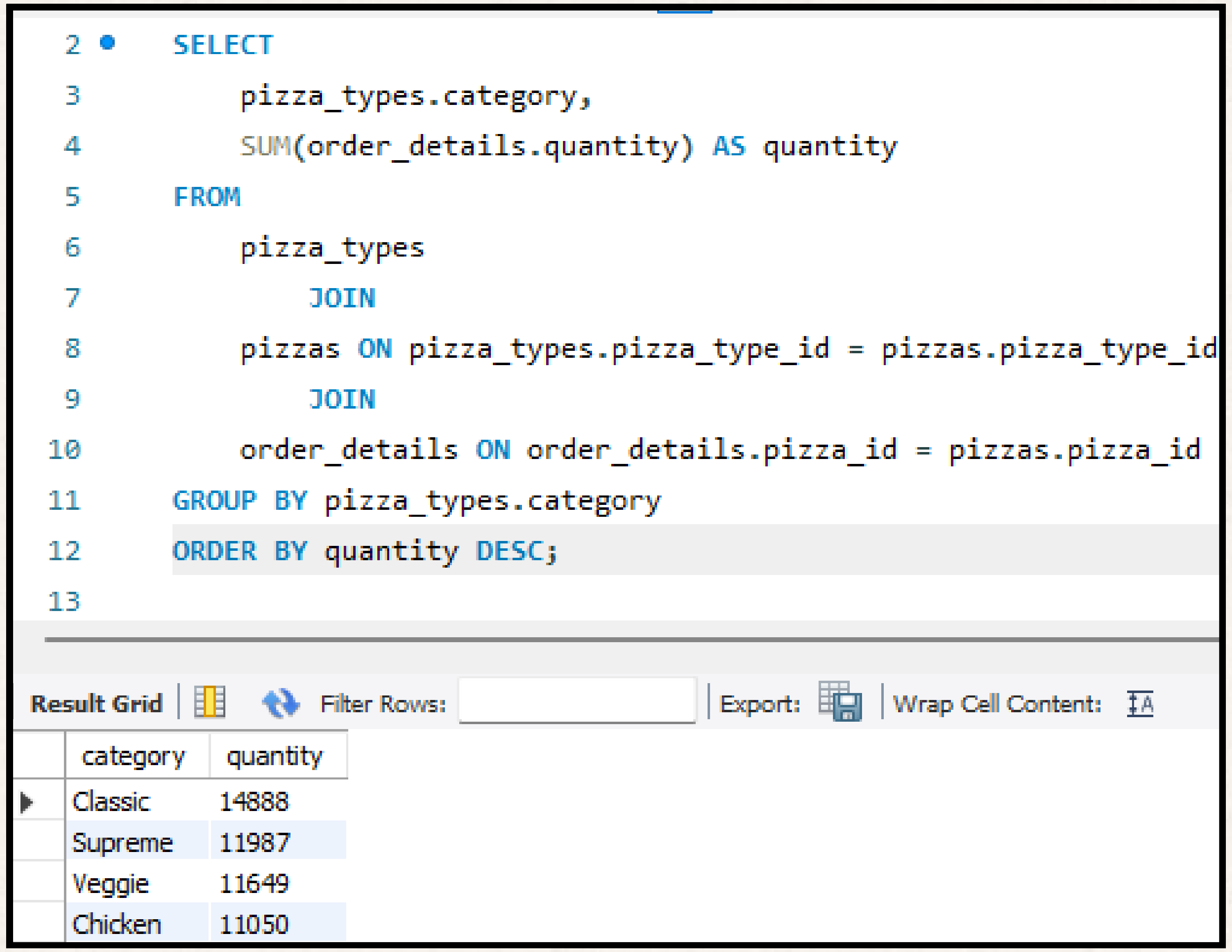
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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

ame	quantity
he Classic Deluxe Pizza	2453
he Barbecue Chicken Pizza	2432
he Hawaiian Pizza	2422
he Pepperoni Pizza	2418
he Thai Chicken Pizza	2371

Question

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

SQL Query



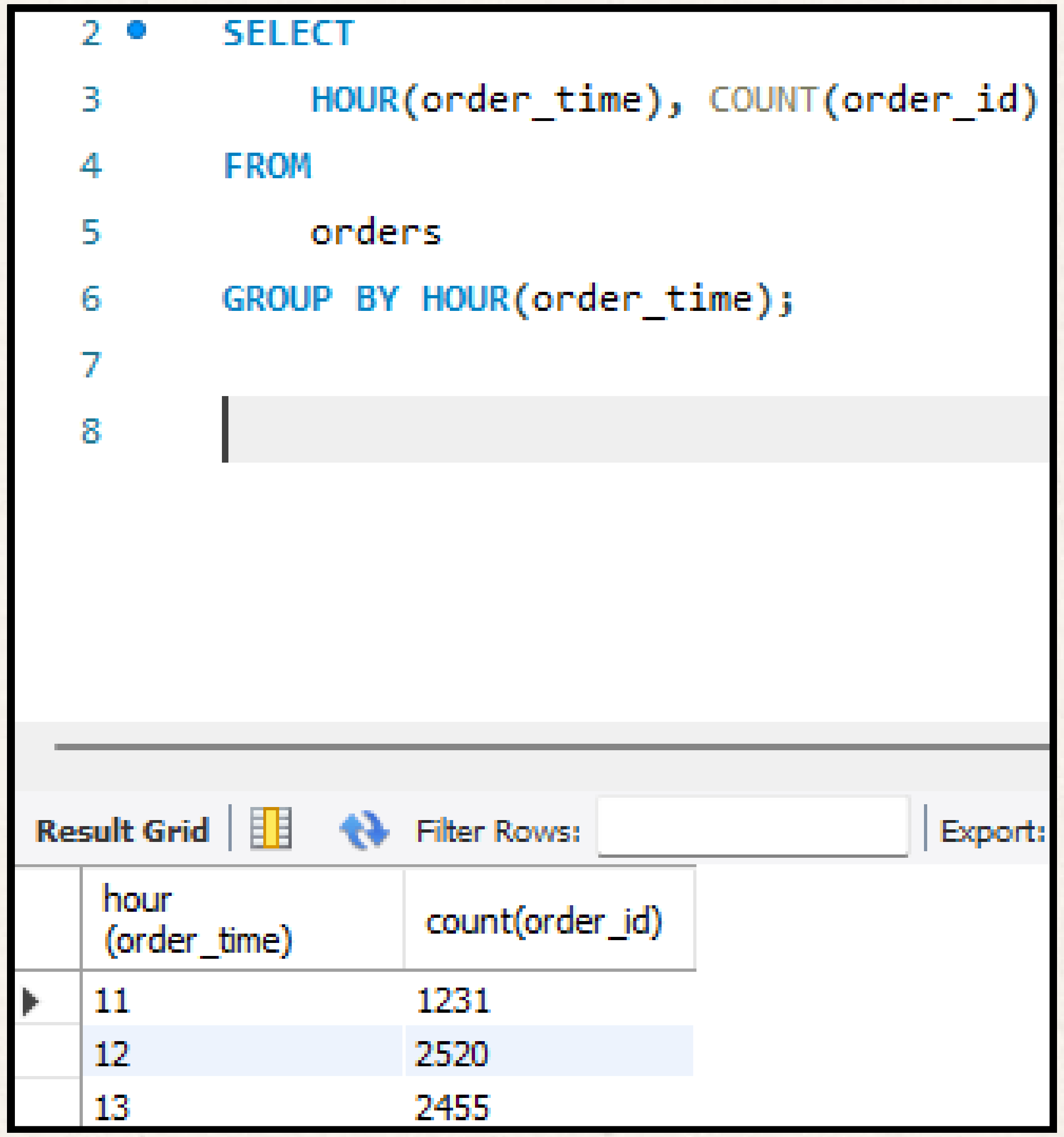
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Question

Determine the distribution of orders by hour of the day.

SQL Query



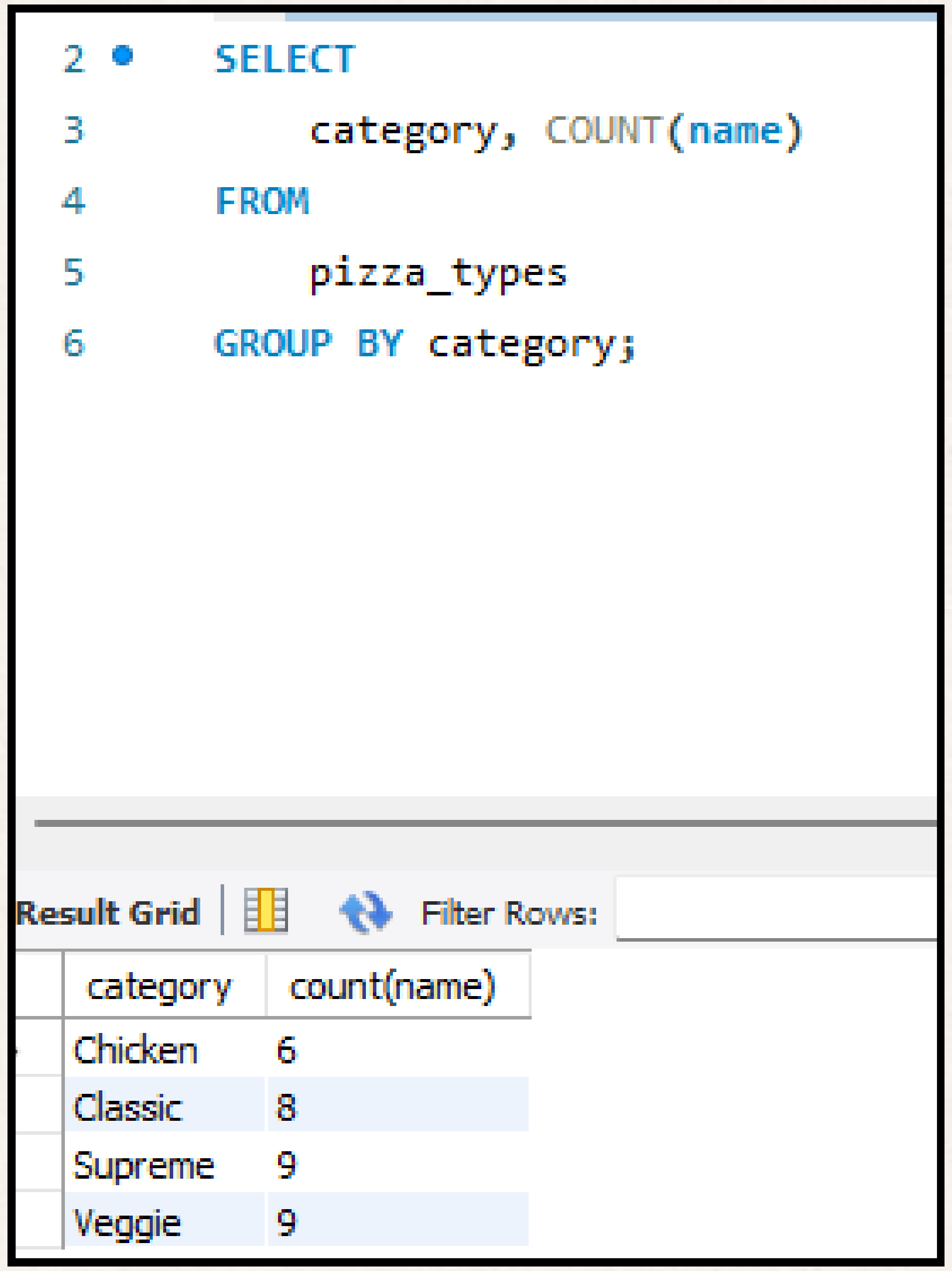
```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

hour (order_time)	count(order_id)
11	1231
12	2520
13	2455

Question

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

SQL Query



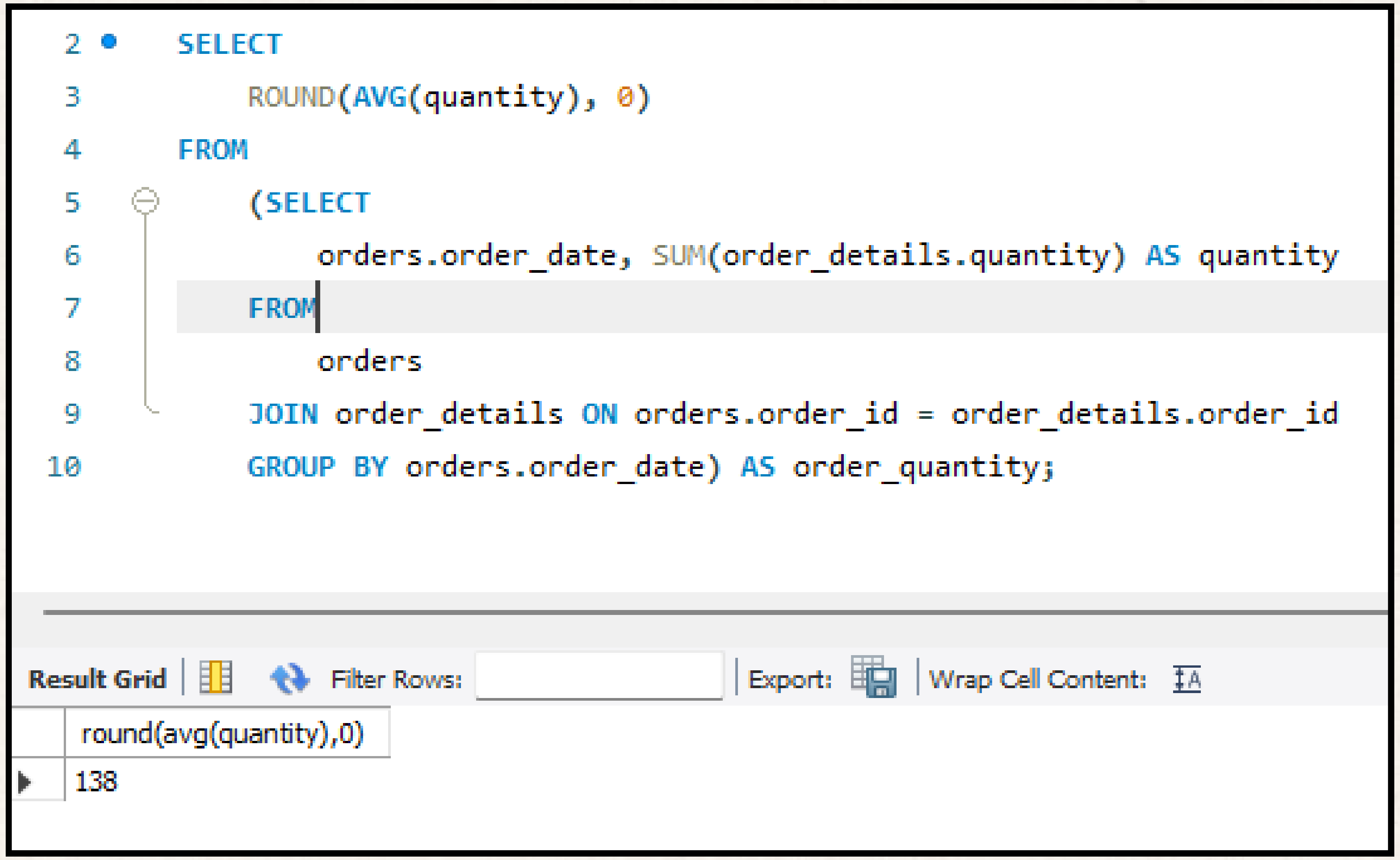
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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

Question

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

SQL Query



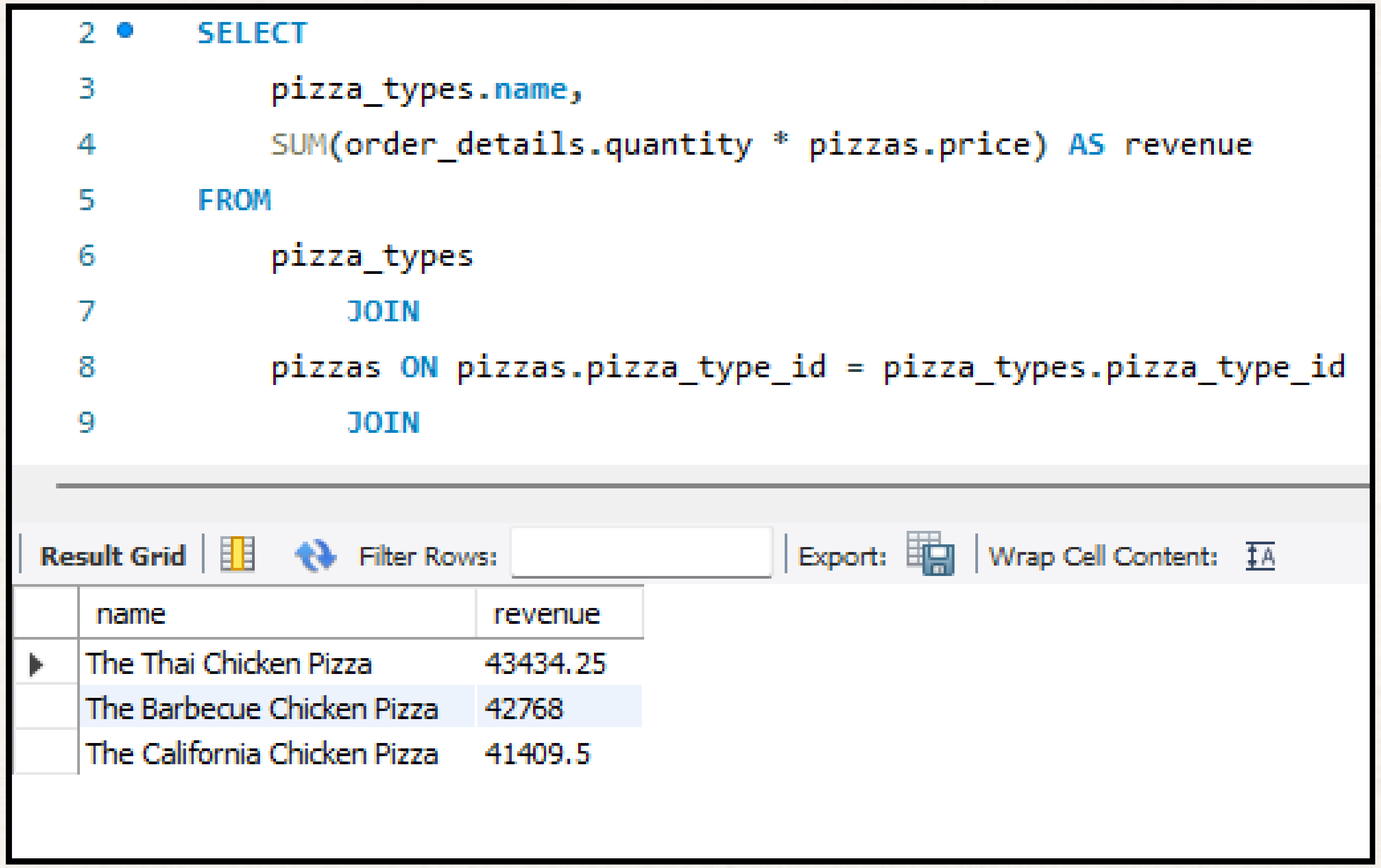
```
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

```
round(avg(quantity),0)
138
```

Question

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

SQL Query



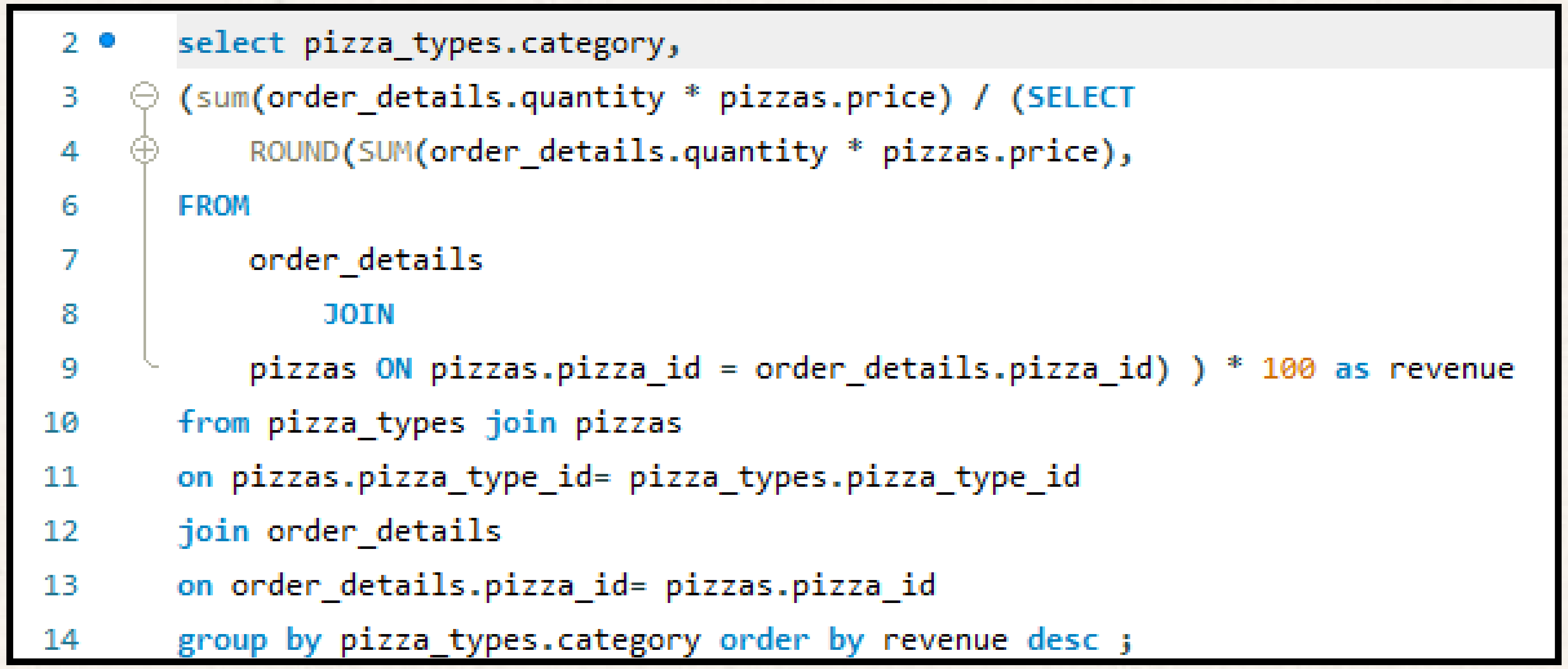
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
```

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

Question

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

SQL Query

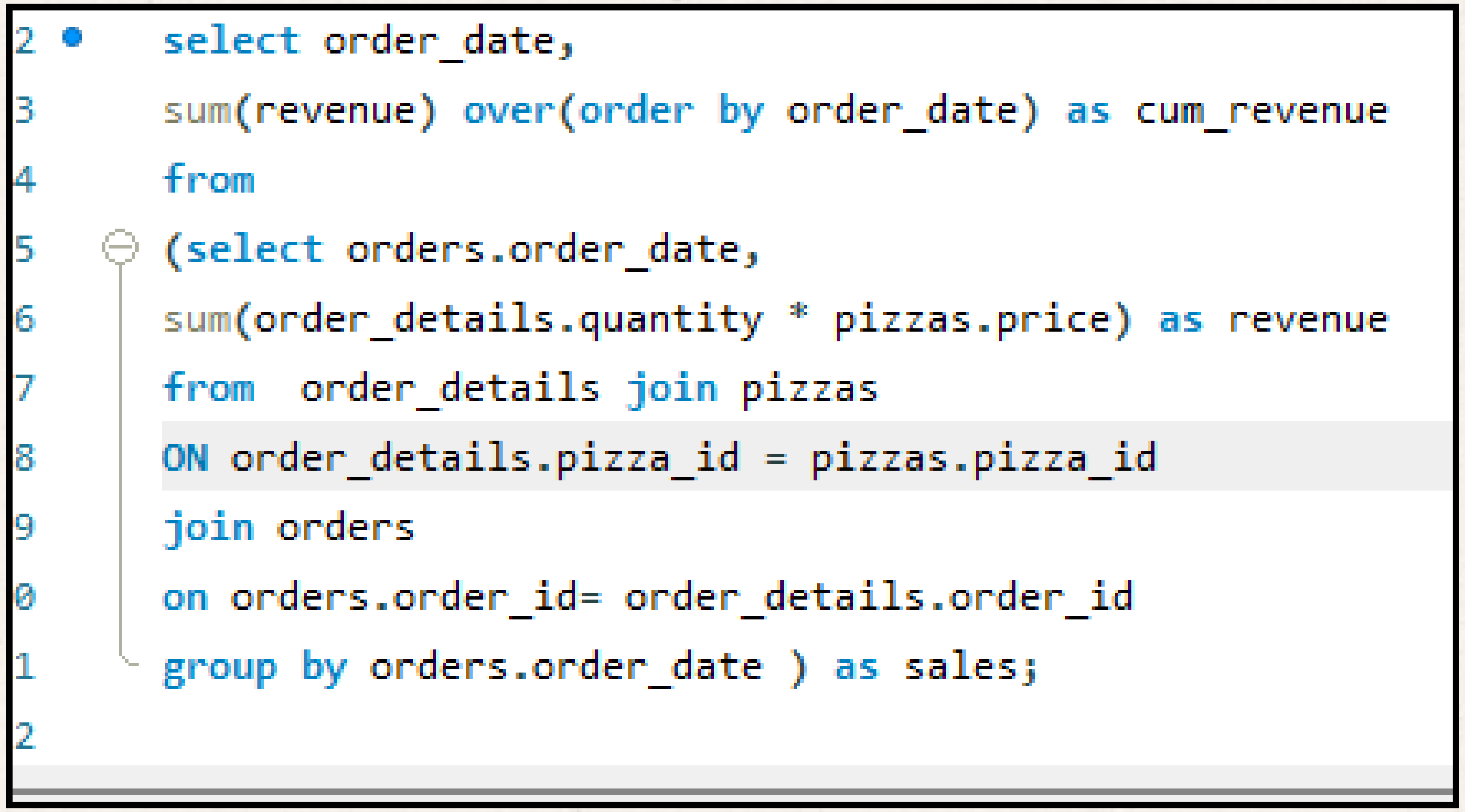


```sql
select pizza_types.category,
(sum(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) ) * 100 as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id= pizza_types.pizza_type_id
join order_details
on order_details.pizza_id= pizzas.pizza_id
group by pizza_types.category order by revenue desc ;
```

Question

Analyze the cumulative revenue generated over time.

SQL Query

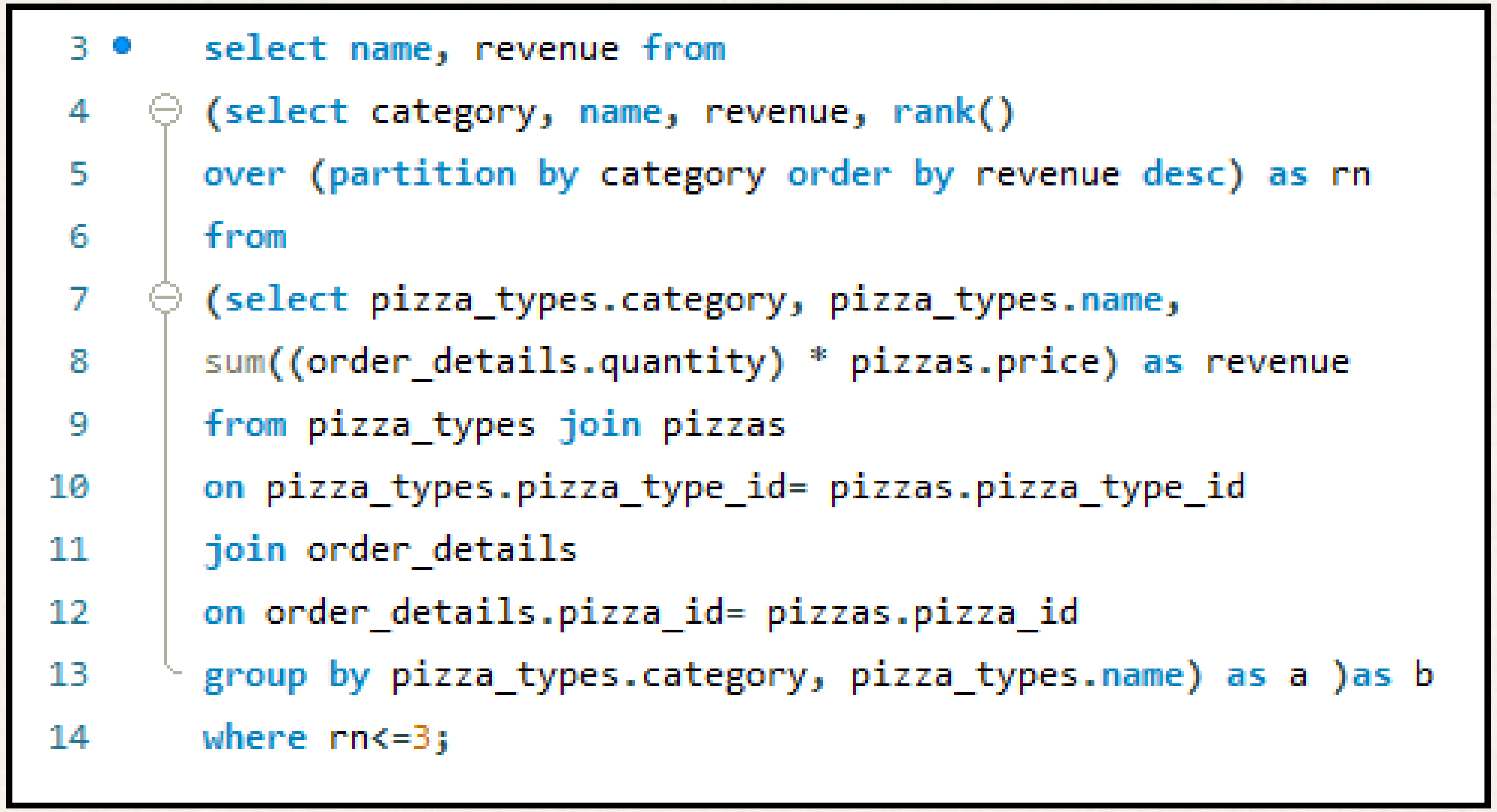


```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from  order_details join pizzas
ON order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id= order_details.order_id
group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

Question

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

SQL Query



```
select 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((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id= pizzas.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 rn<=3;
```

Thank you!

CONTACT INFO:

E-mail	gaikwad.ashwini3215@gmail.com
Linkedin	https://www.linkedin.com/in/ashwinigaikwad3215
Address	Mumbai