Home About Contact

PIZZA SALES DATA-ANALYSIS USING SQL

BY - SAGNIK DHAR

Home About Contact

PROJECT INTRODUCTION

- Analyzed a pizza sales dataset using SQL to extract actionable insights.
- Focused on evaluating revenue across different pizza types, sizes, and categories.
- Used SQL joins, aggregation functions, and filtering to explore the data.
- Findings will help support decisions in marketing, menu design, and operations.

DATASET OVERVIEW

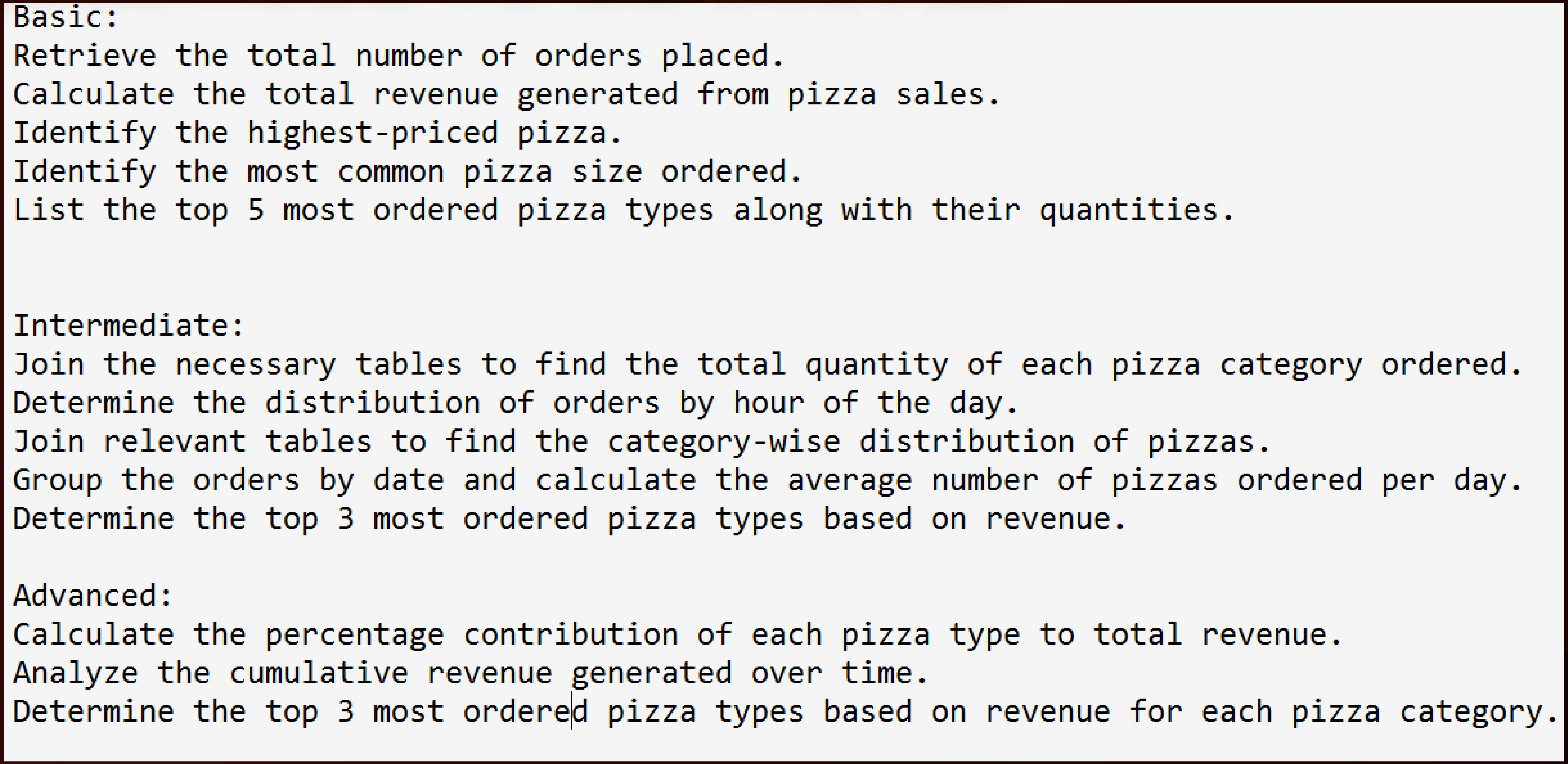
Home About Contact

Tables used in the project :

Table Name		Description
• orders	⟶	orders_id, order_time, order_date
• orders_details	⟶	orders_id, order_details_id, pizza_id, quantity
• pizzas	⟶	pizza_id, pizza_type_id, size, prize
• pizza_types	⟶	pizza_type_id, category, name , ingrediants

Home About Contact

QUESTIONS TO BE SOLVED



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.

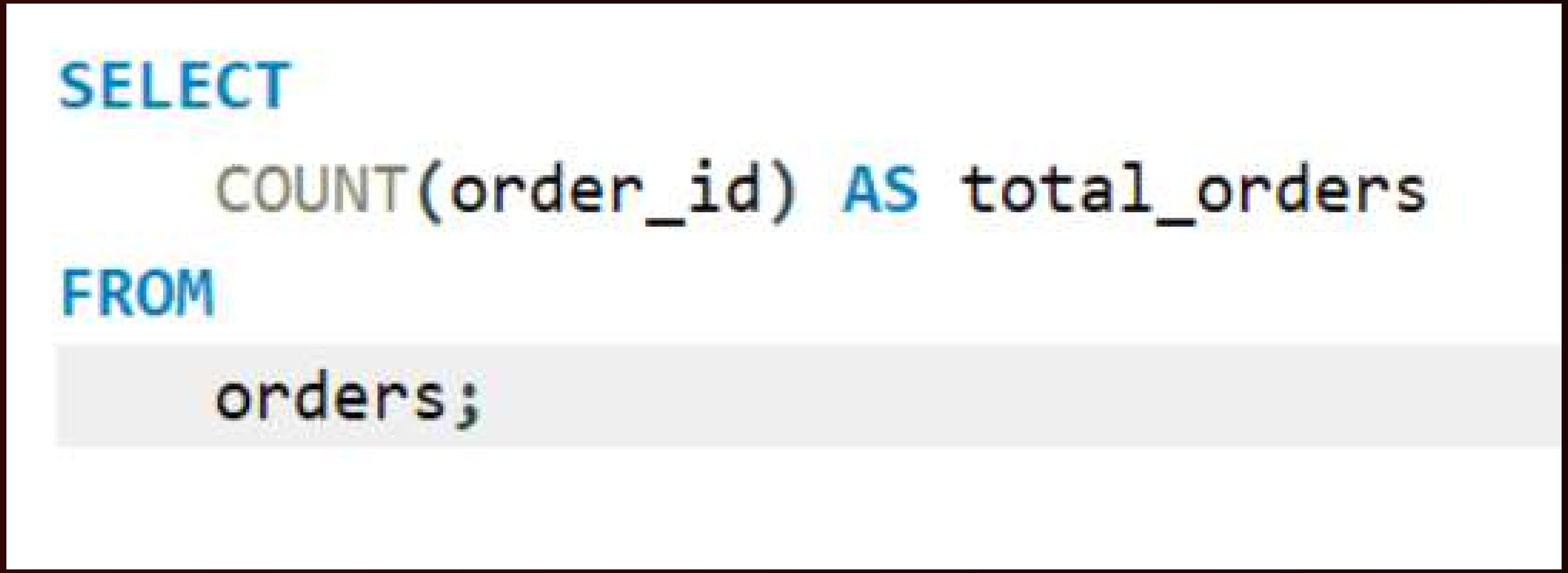
Intermediate:
Join the necessary tables to find the total quantity of each pizza category ordered.
Determine the distribution of orders by hour of the day.
Join relevant tables to find the category-wise distribution of pizzas.
Group the orders by date and calculate the average number of pizzas ordered 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.

Retrieve the total number of orders placed

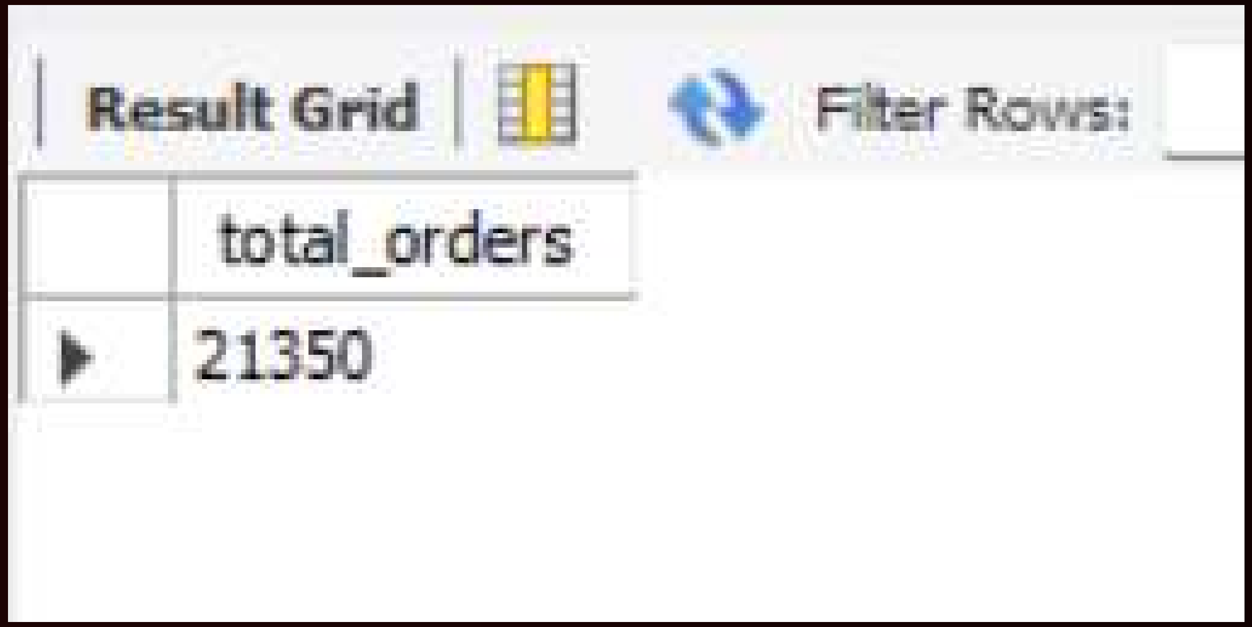
Contact

INPUT



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

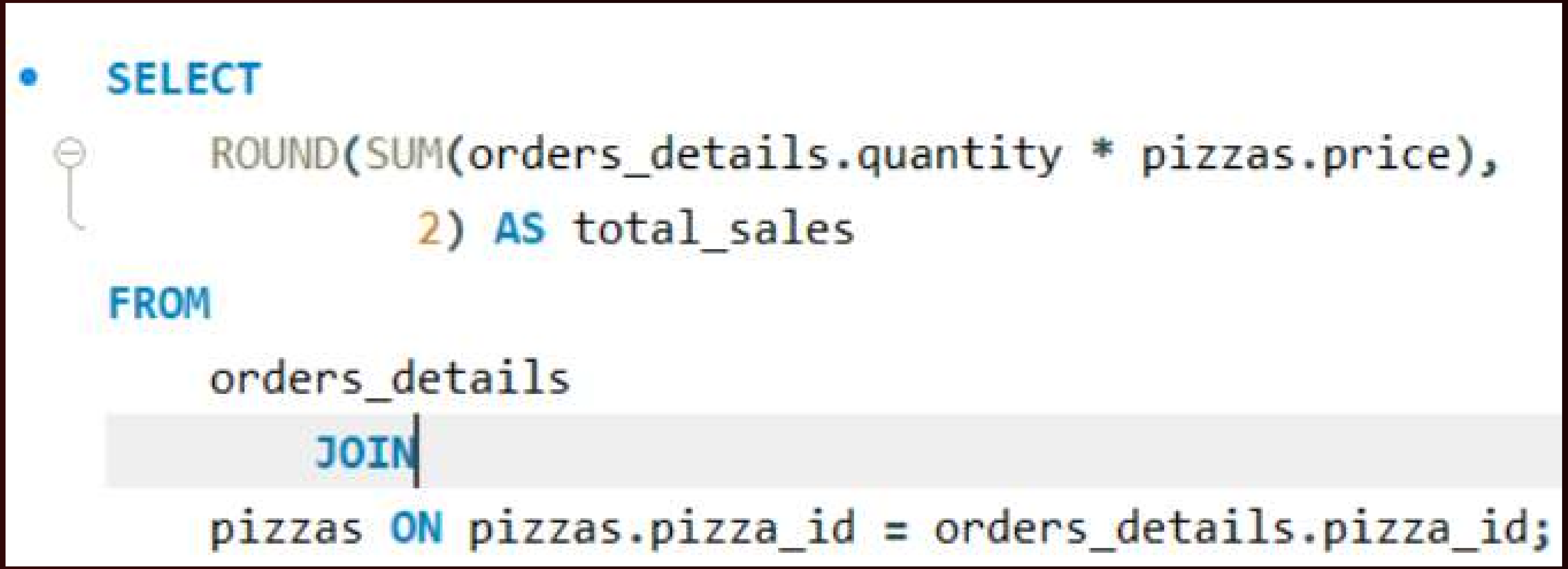
OUTPUT



```
total_orders
21350
```

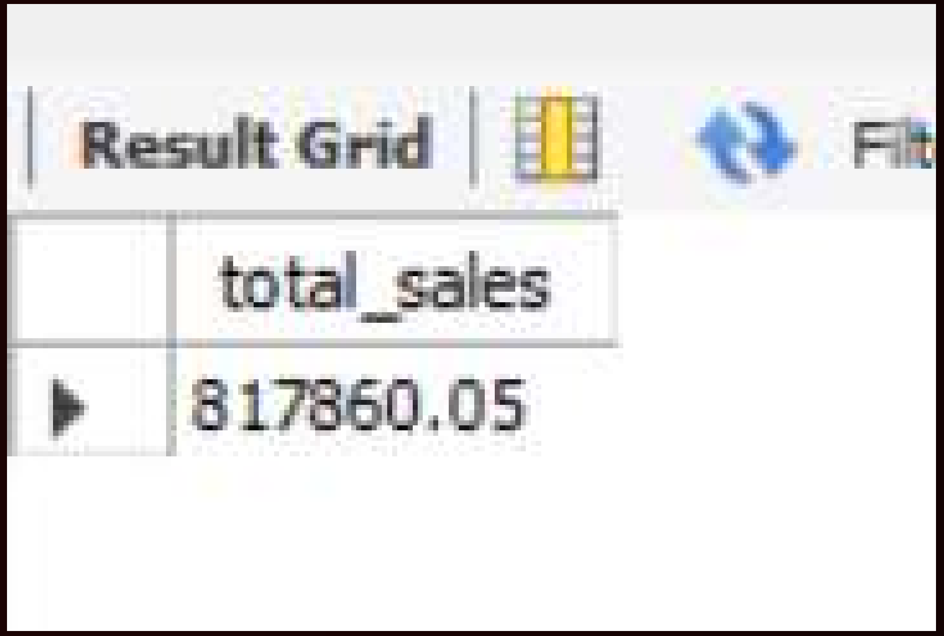
Calculate the total revenue generated from pizza sales.

INPUT



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

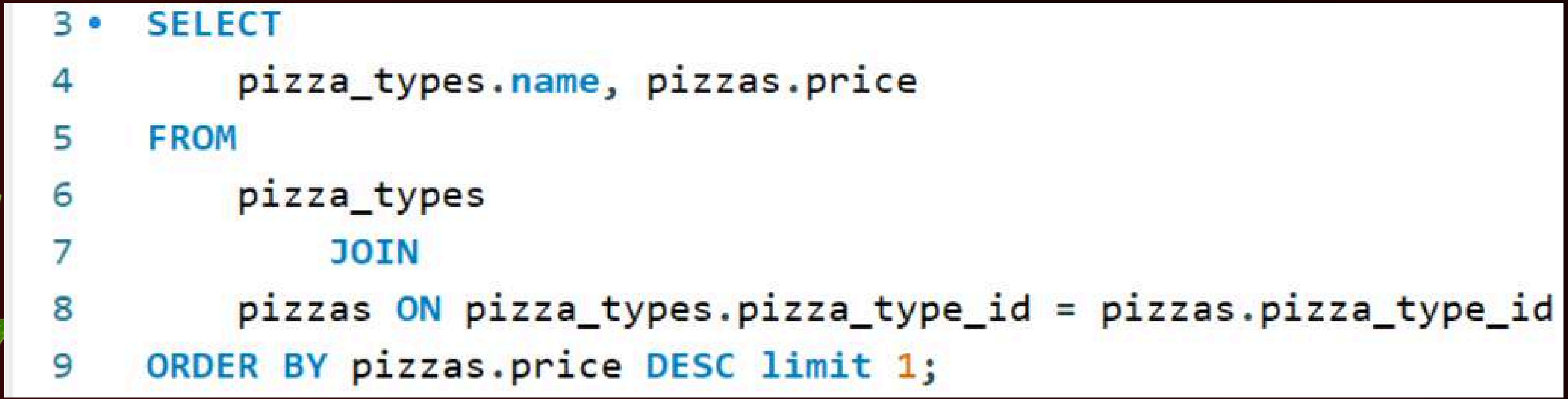
OUTPUT



```
total_sales
817860.05
```

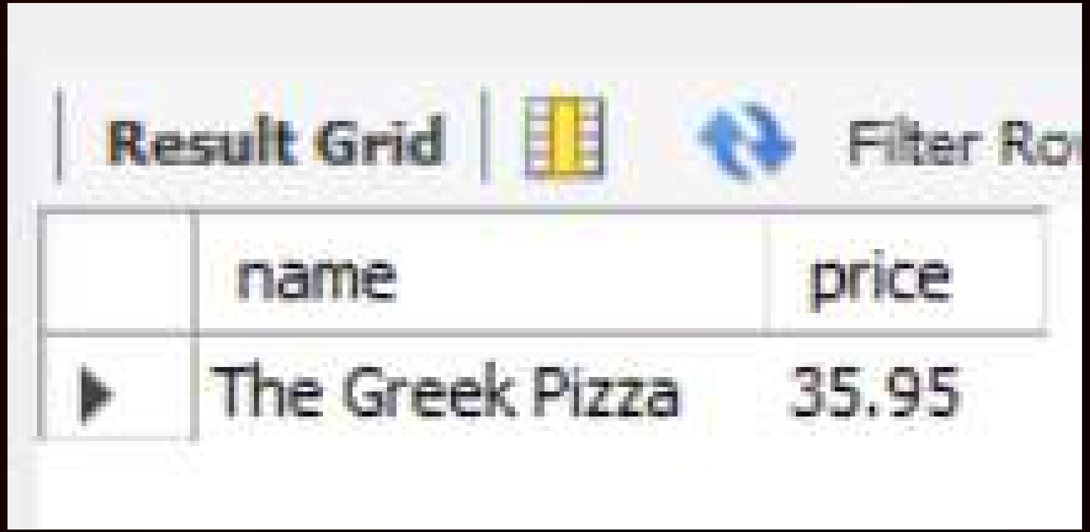
Identify the highest-priced pizza.

INPUT



```
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;
```

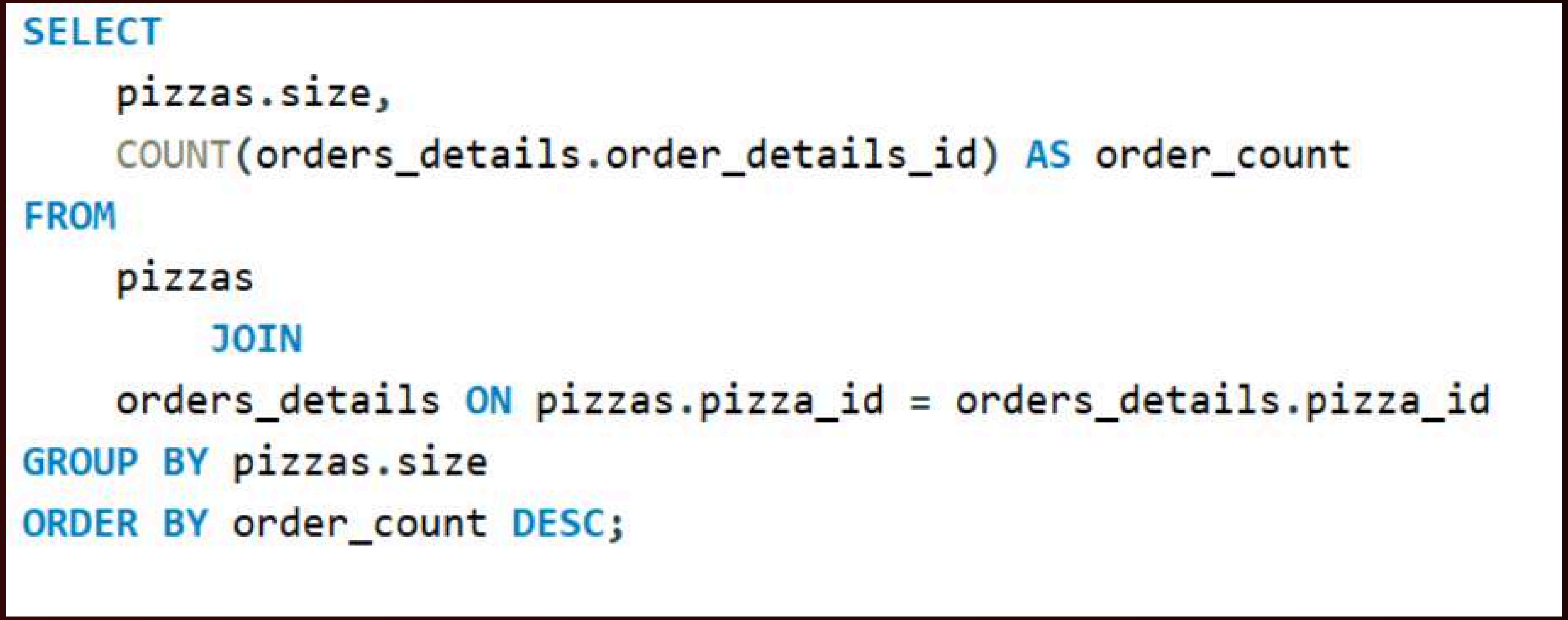
OUTPUT



```
name             price
The Greek Pizza  35.95
```

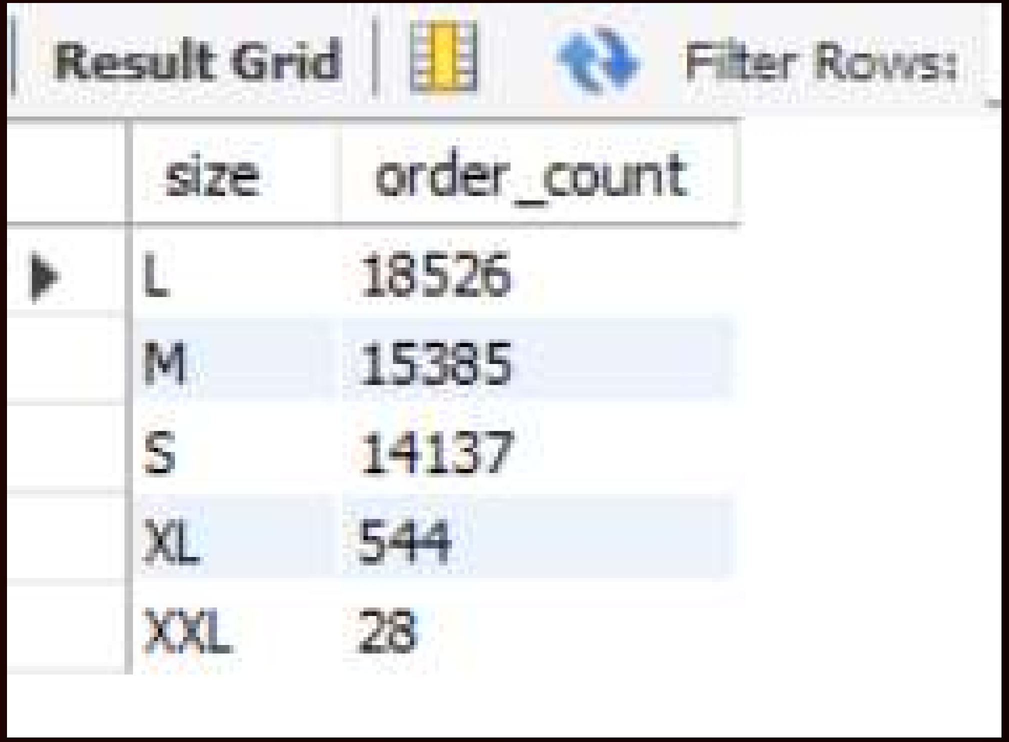
Identify the most common pizza size ordered.

INPUT



```sql
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;
```

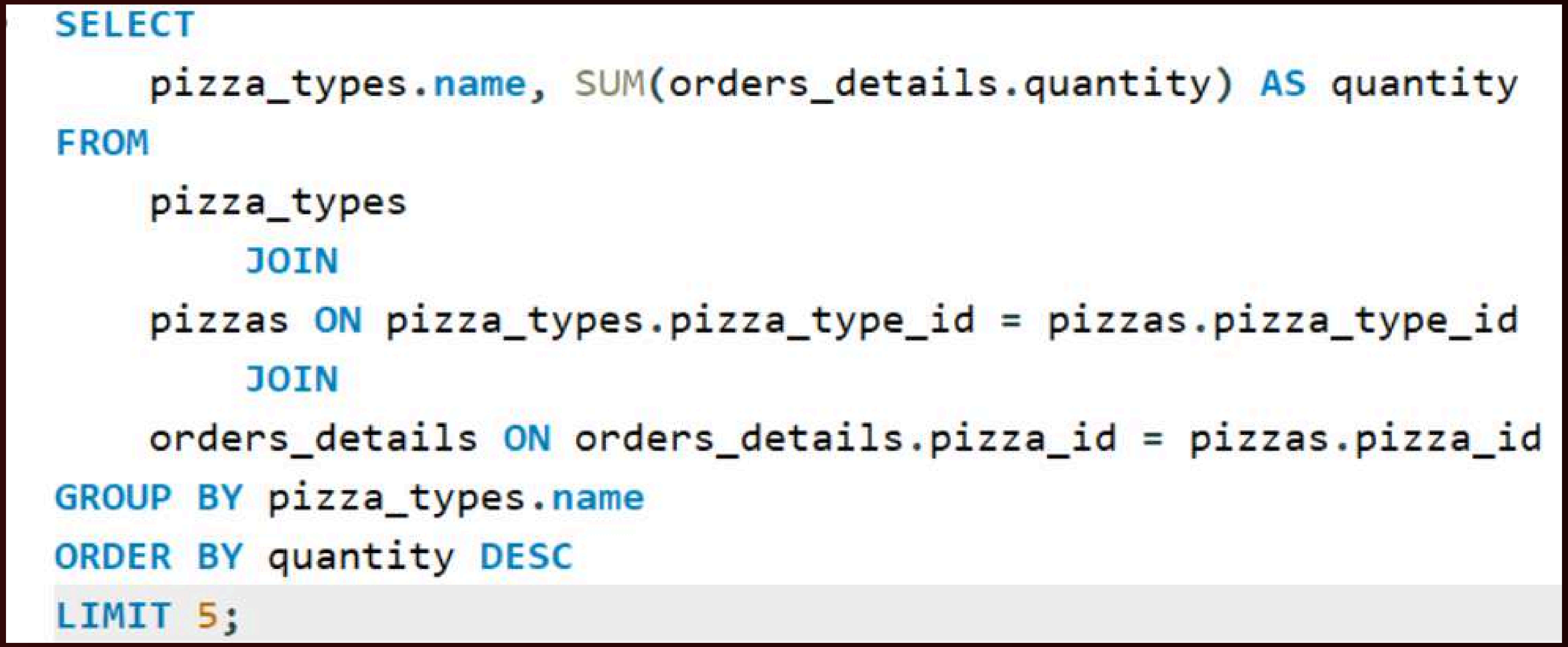
OUTPUT



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

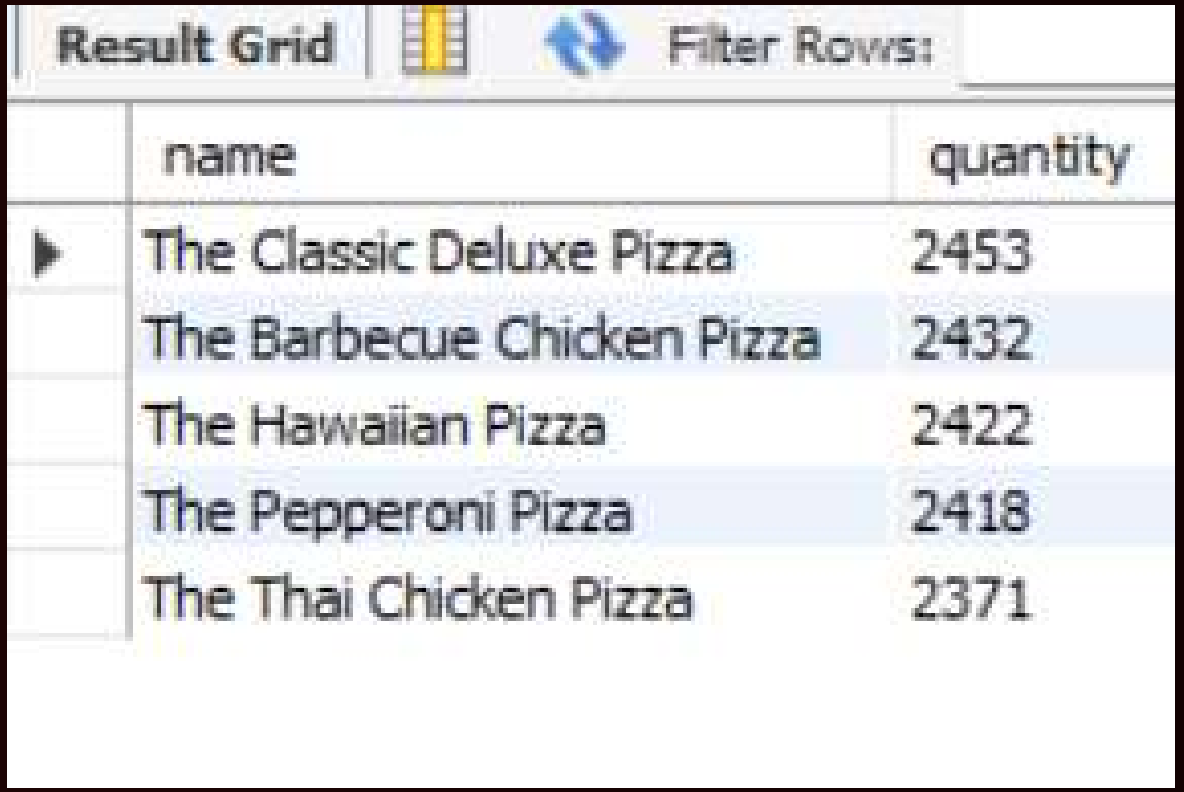
List the top 5 most ordered pizza types along with their quantities.

INPUT



```
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;
```

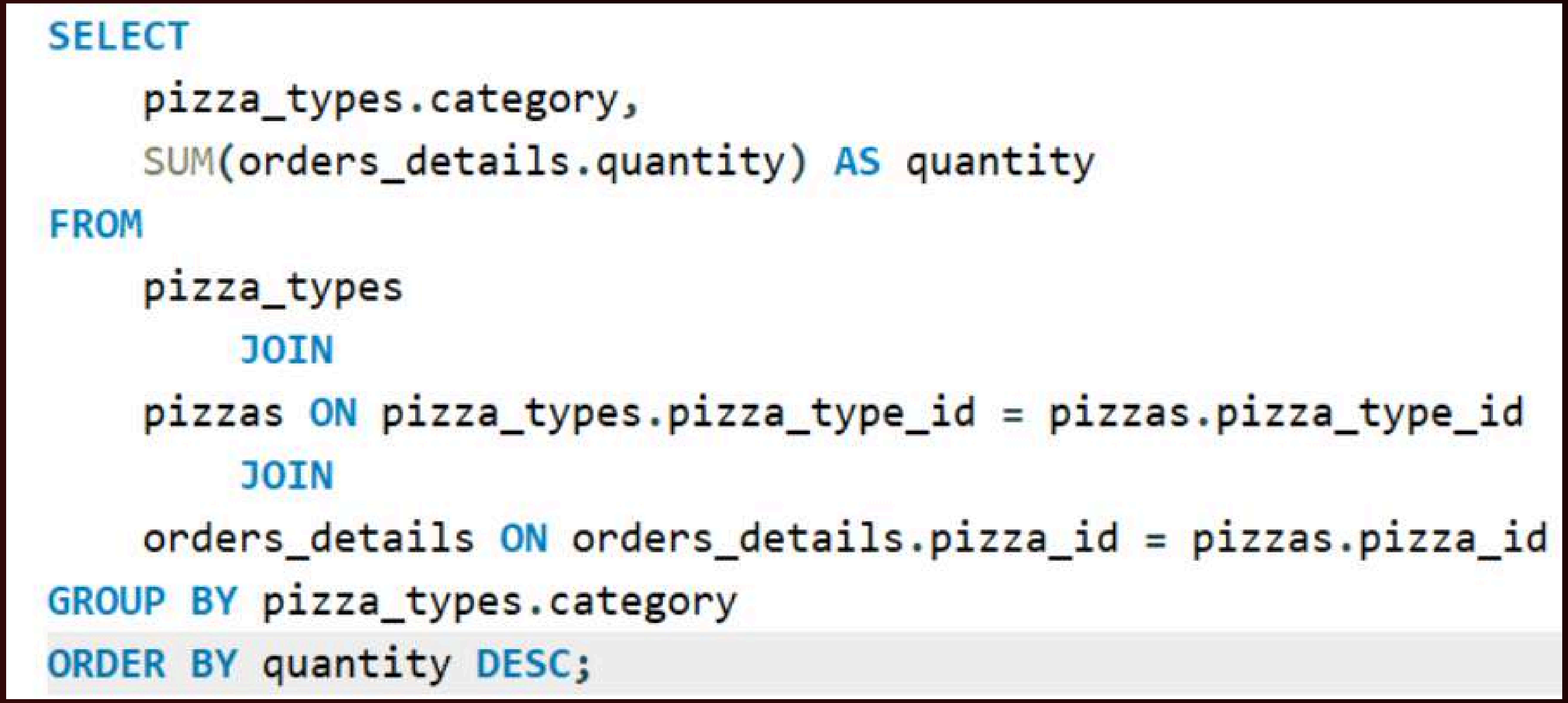
OUTPUT



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

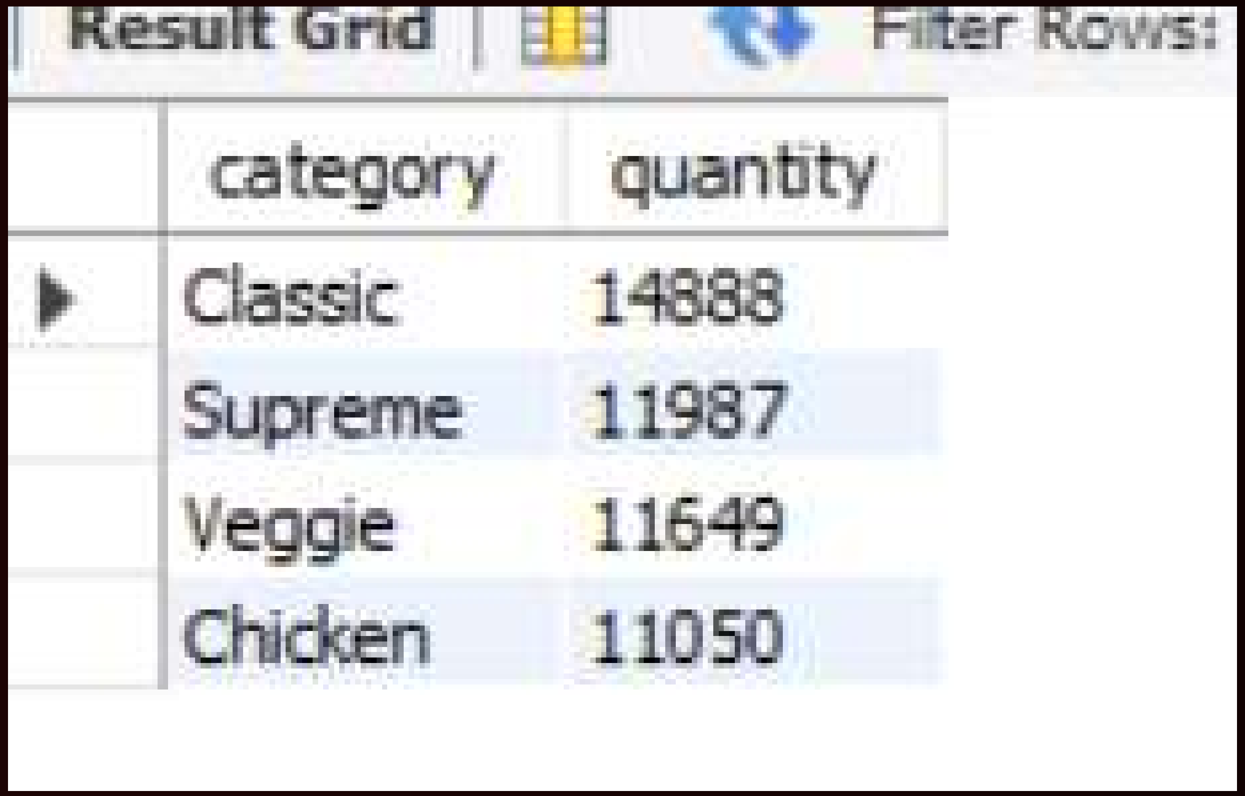
Join the necessary tables to find the total quantity of each pizza category ordered.

INPUT



```
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;
```

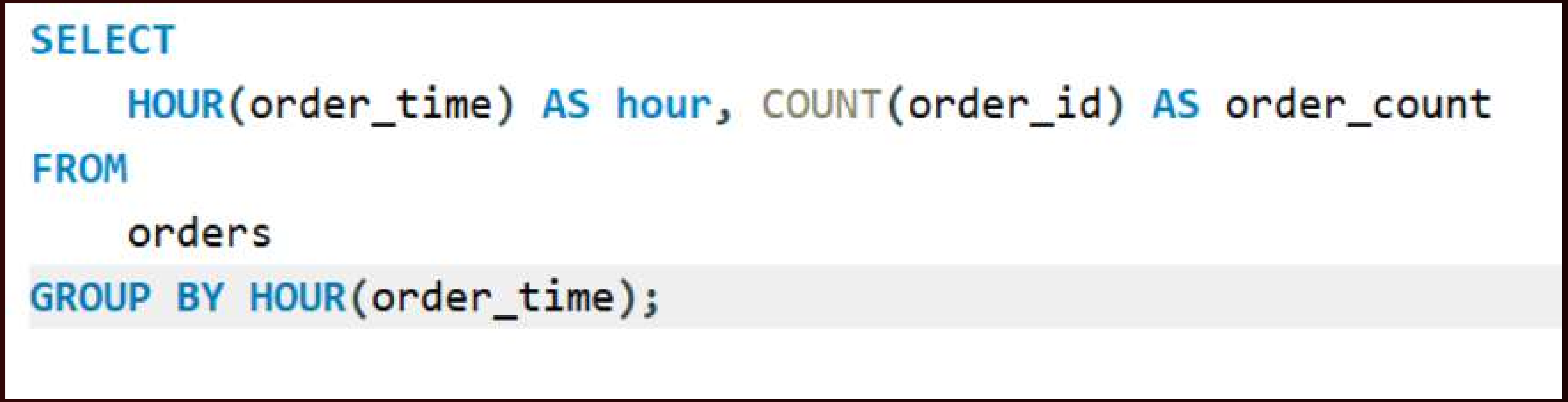
OUTPUT



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

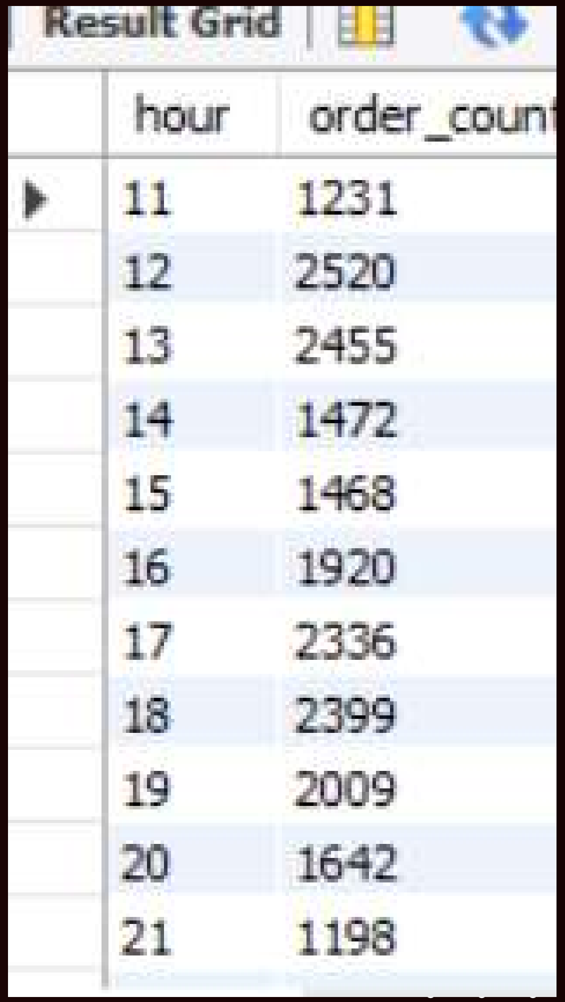
Determine the distribution of orders by hour of the day.

INPUT



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

OUTPUT

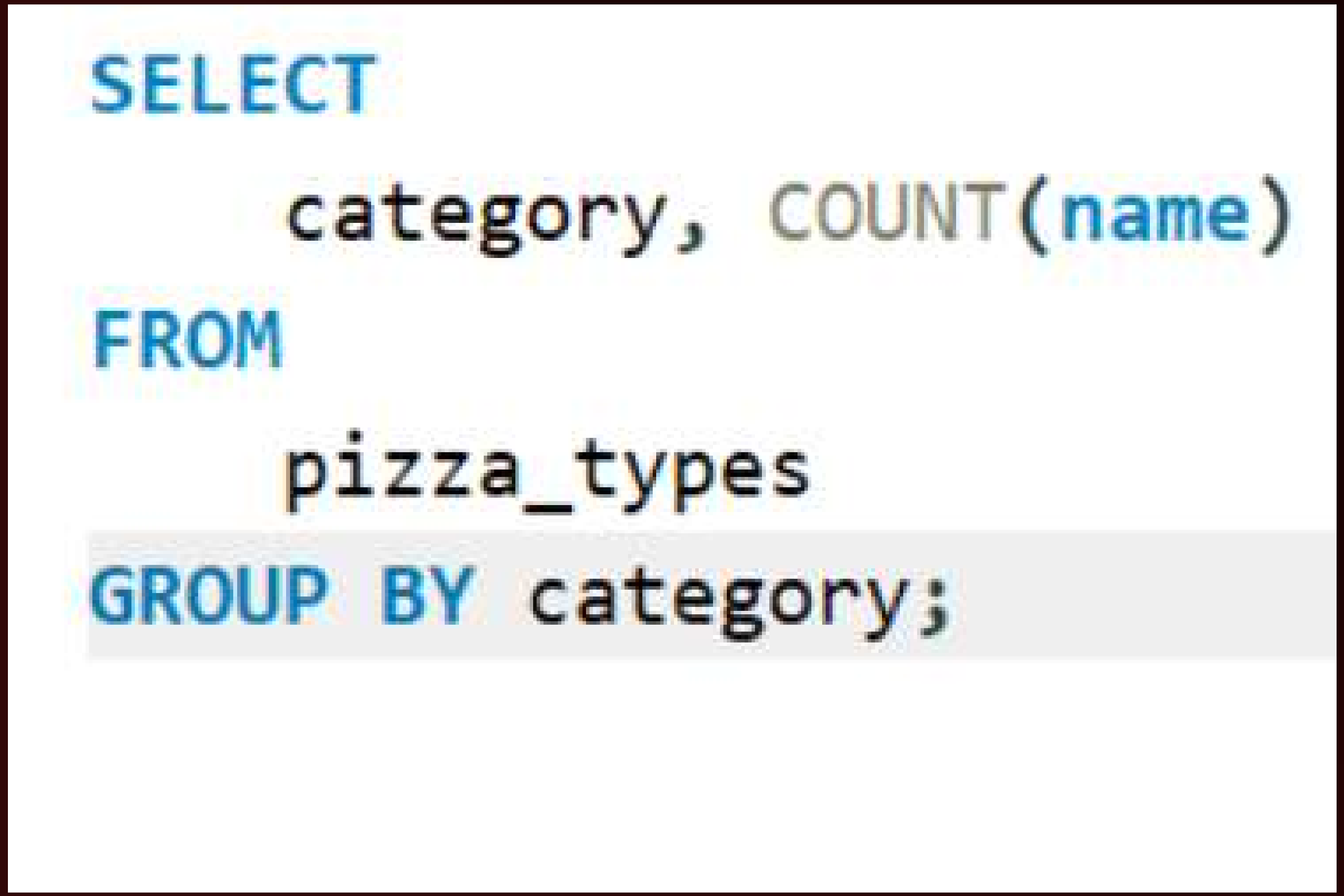


Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

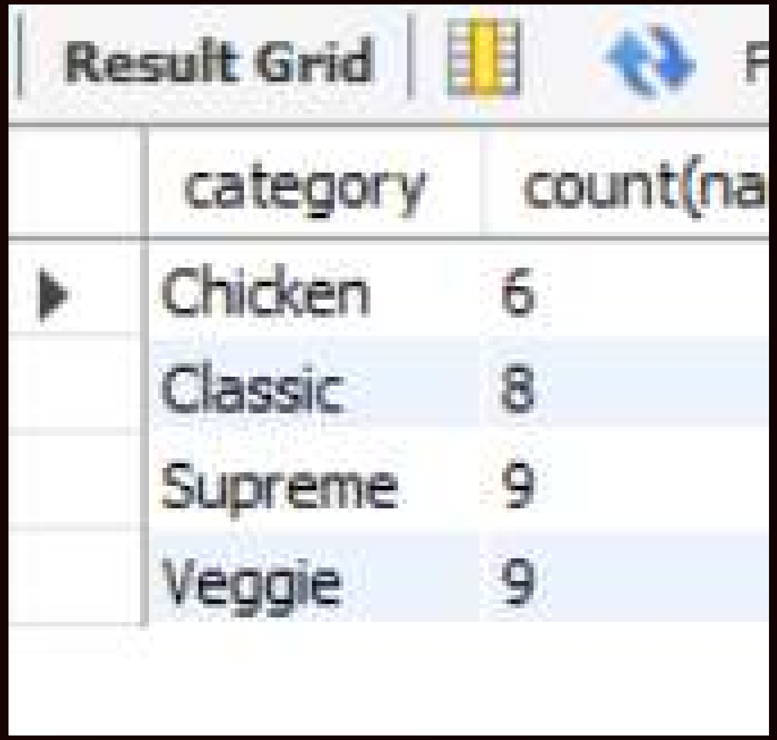
Join relevant tables to find the category-wise distribution of pizzas.

INPUT



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

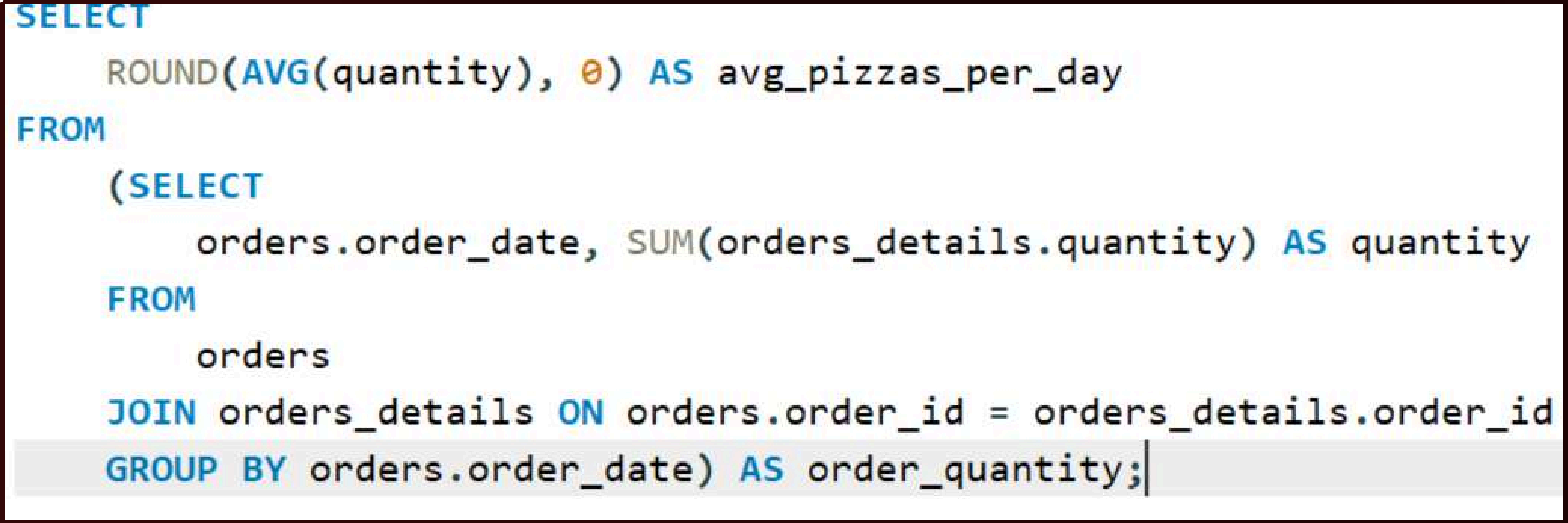
OUTPUT



```
category  count(na
Chicken   6
Classic   8
Supreme   9
Veggie    9
```

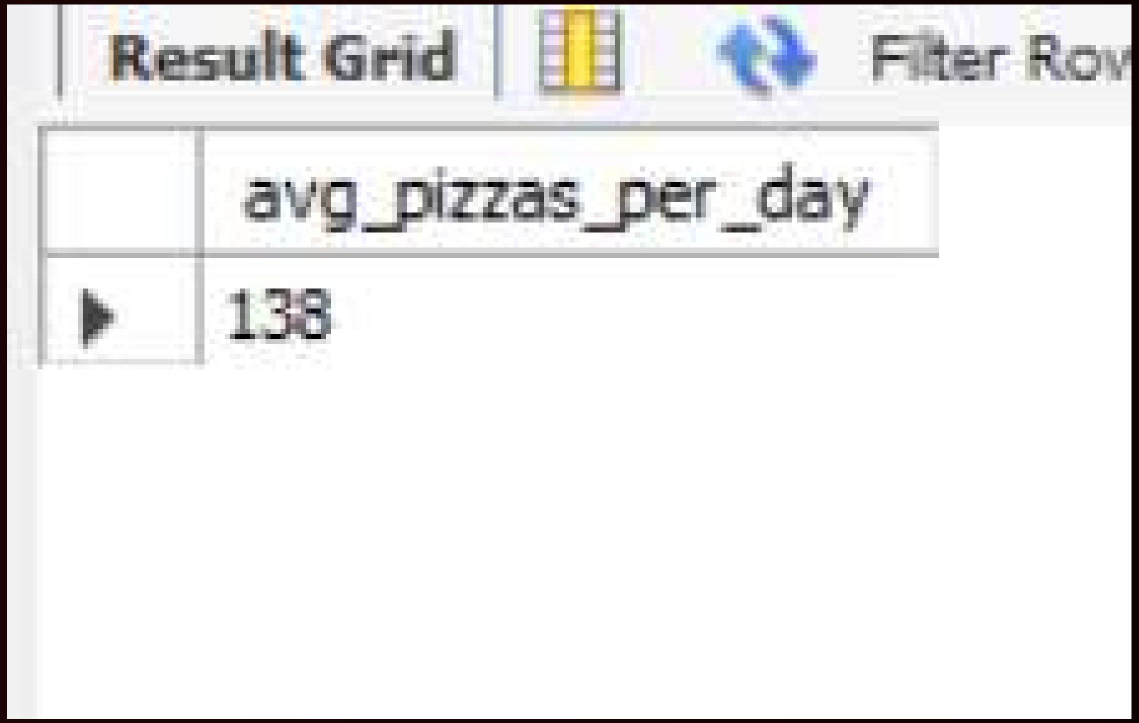
Group the orders by date and calculate the average number of pizzas ordered per day.

INPUT



```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzas_per_day
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;
```

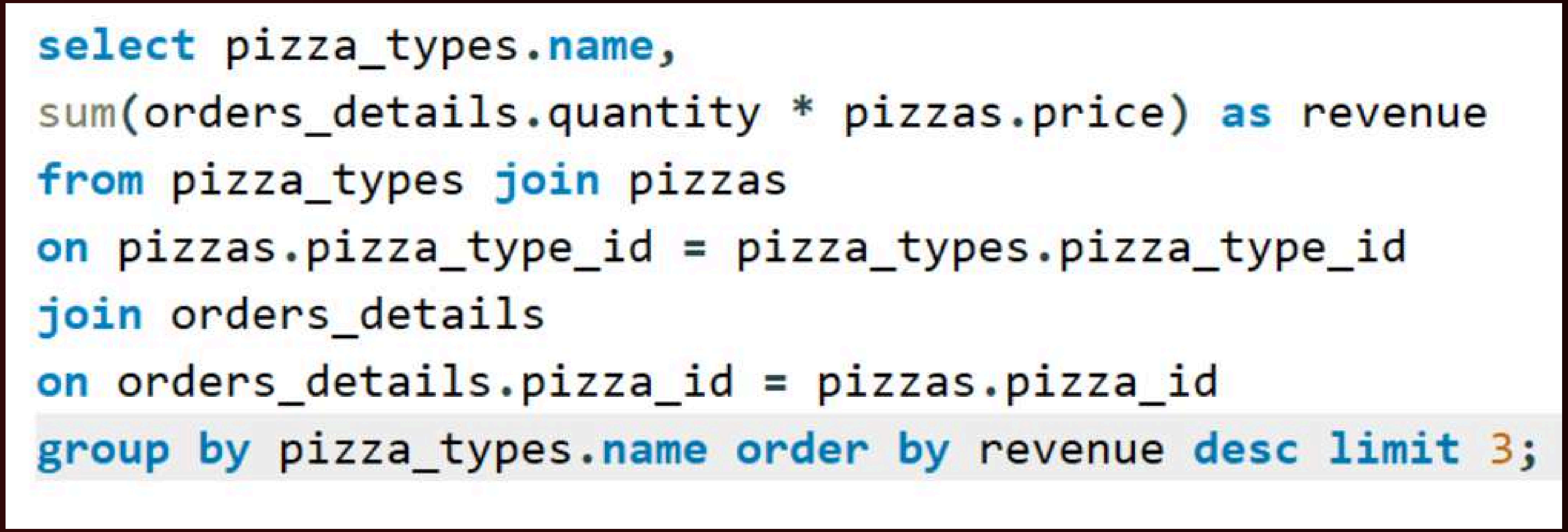
OUTPUT



```
avg_pizzas_per_day
138
```

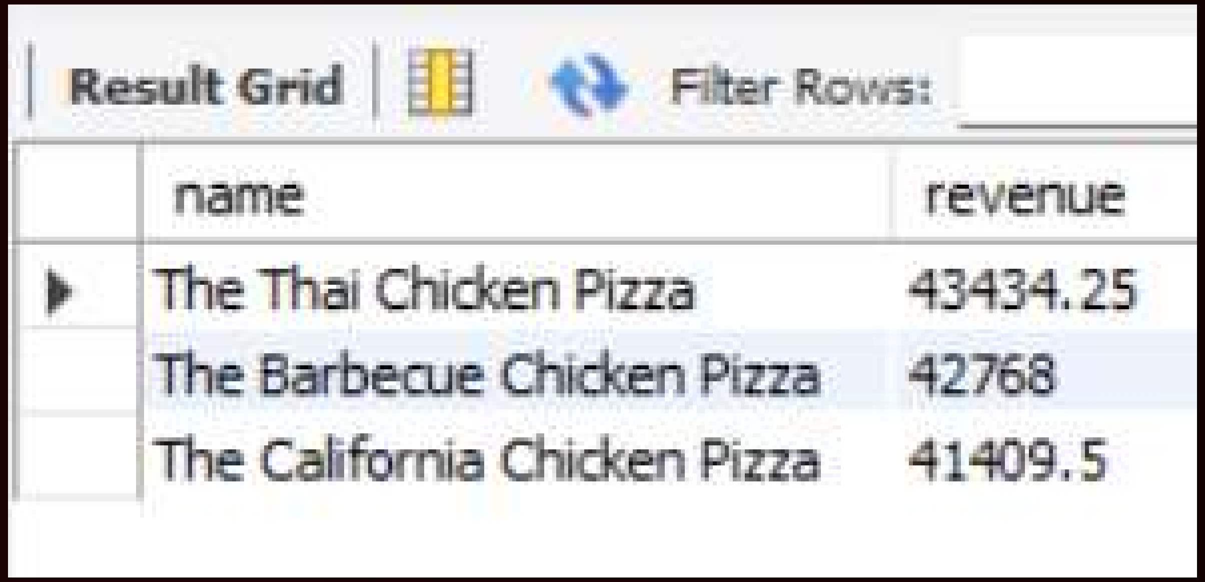
Determine the top 3 most ordered pizza types based on revenue.

INPUT



```
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;
```

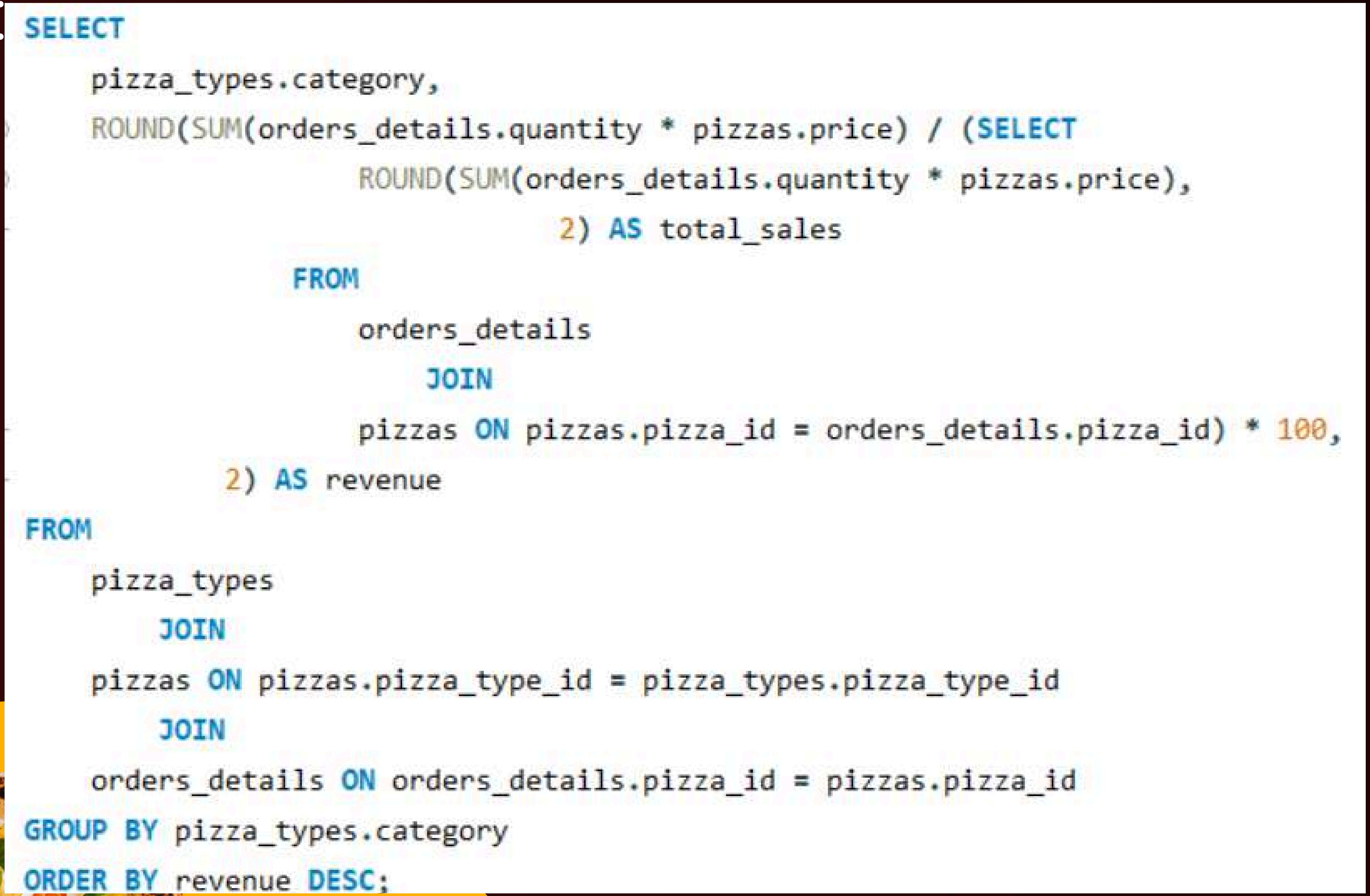
OUTPUT



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

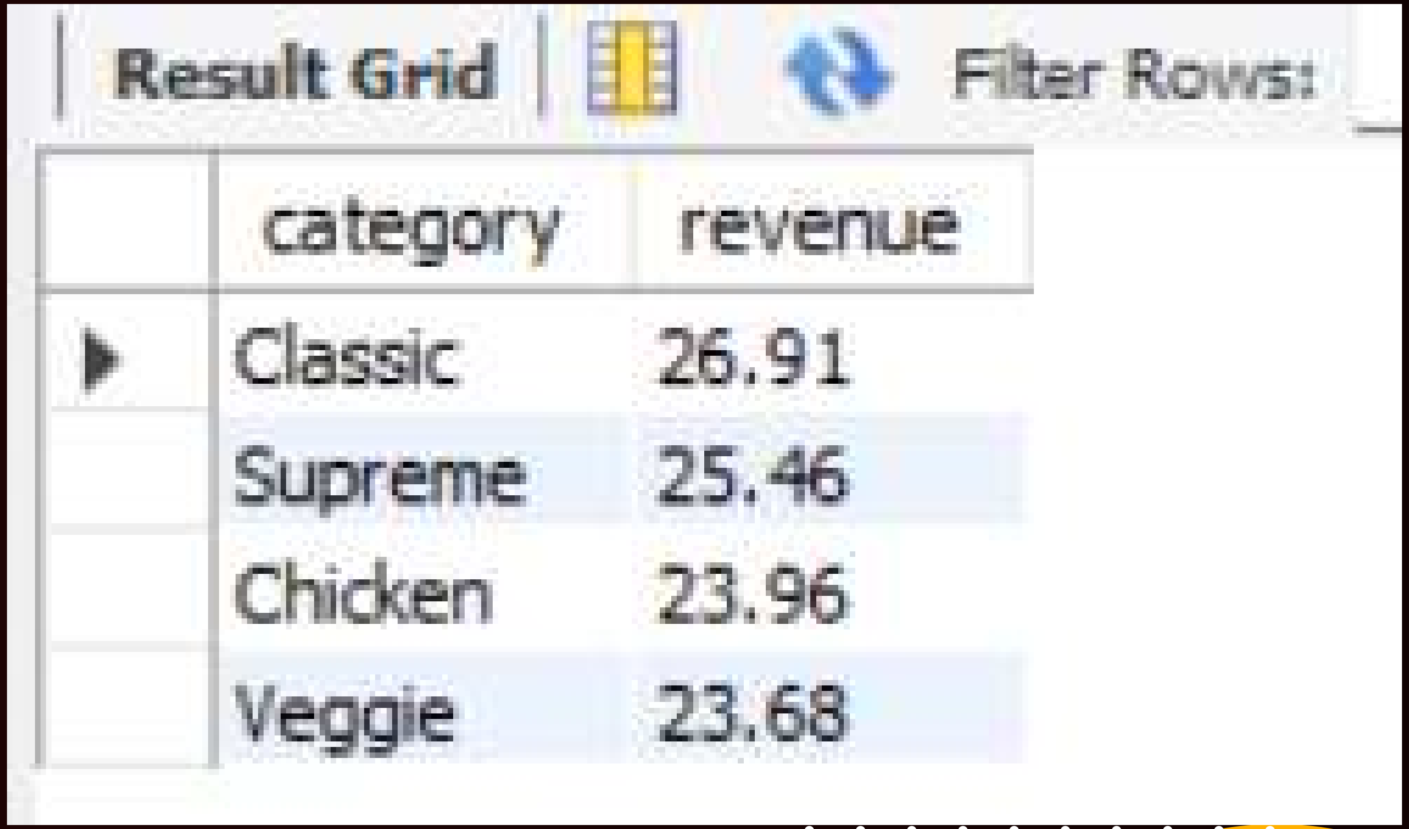
Calculate the percentage contribution of each pizza type to total revenue.

INPUT



```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(orders_details.quantity * pizzas.price),
                            2) AS total_sales
            FROM
                orders_details
                    JOIN
                pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
        2) 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.category
ORDER BY revenue DESC;
```

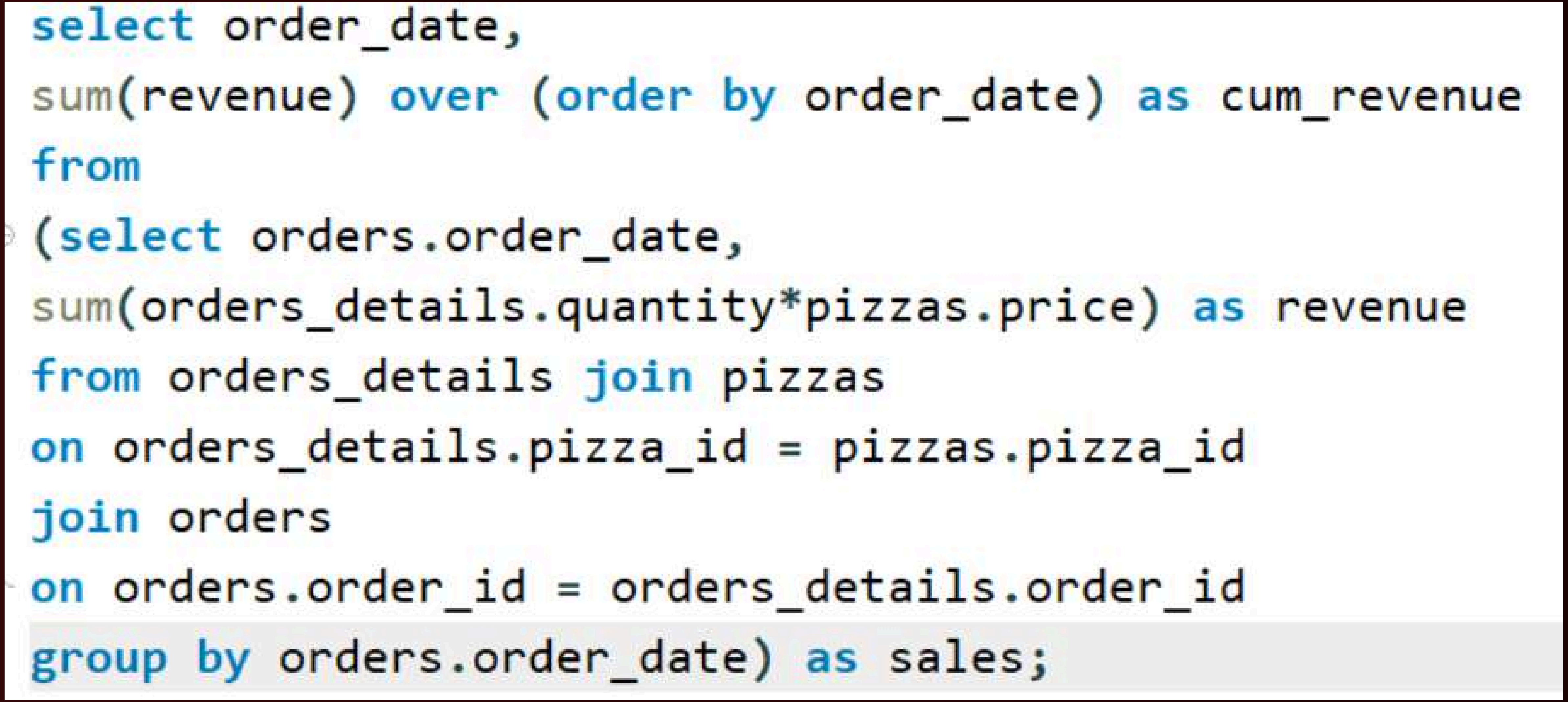
OUTPUT



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

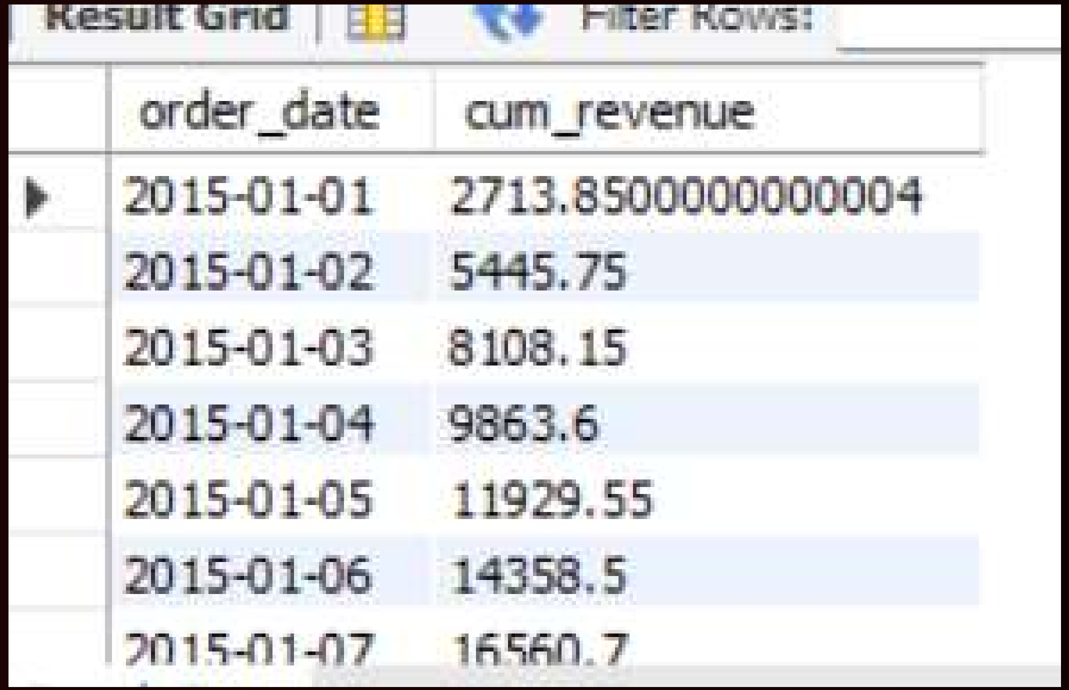
Analyze the cumulative revenue generated over time.

INPUT



```
select order_date,
sum(revenue) over (order by order_date) 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;
```

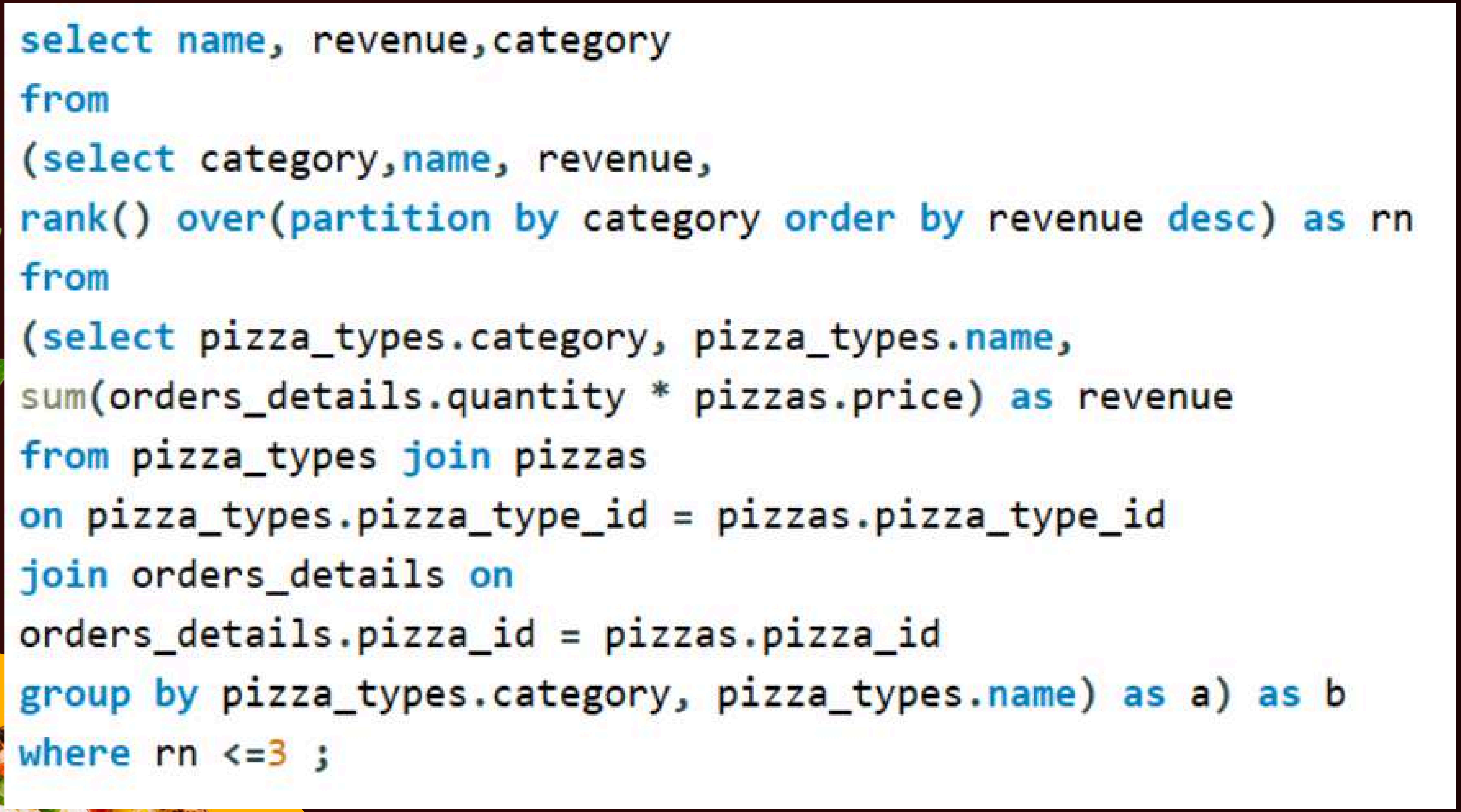
OUTPUT



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

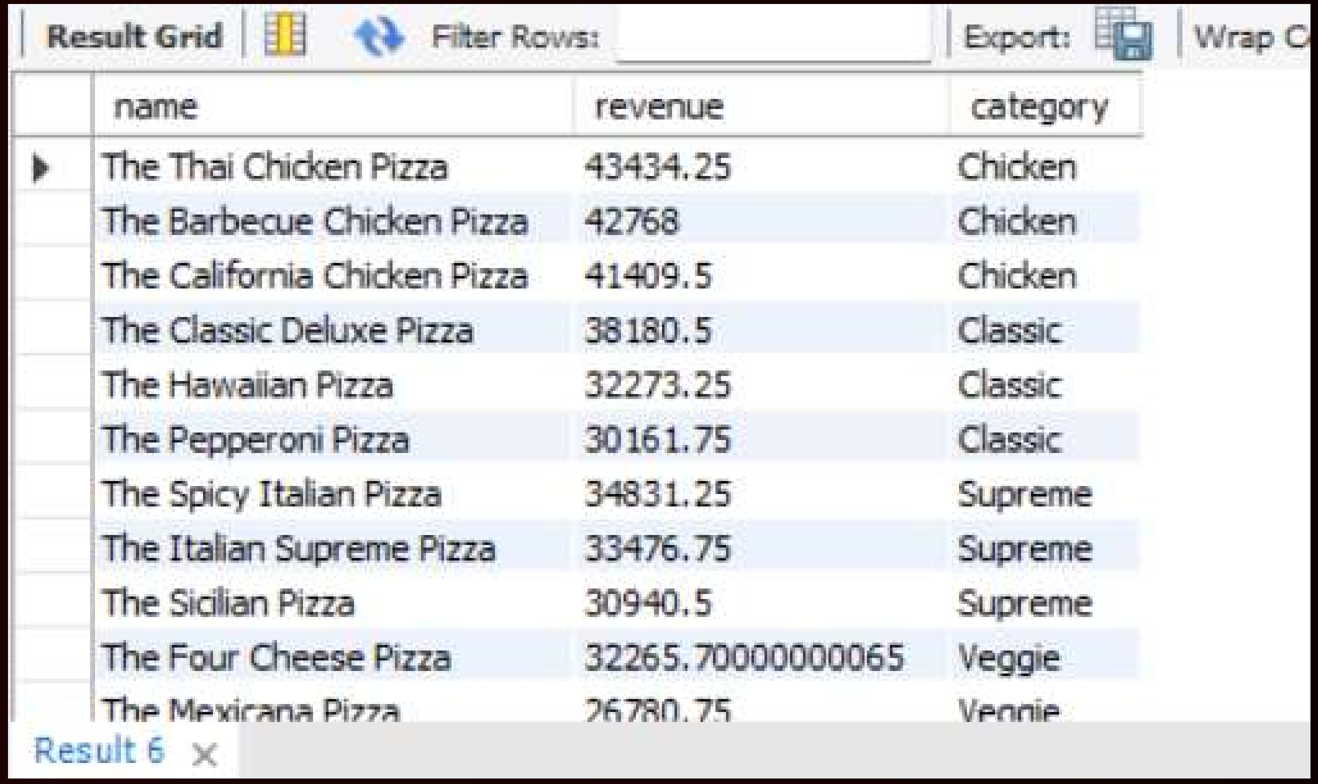
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

INPUT



```
select name, revenue,category
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 ;
```

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
The Four Cheese Pizza	32265.7000000065	Veggie
The Mexicana Pizza	26780.75	Veggie

THANK YOU

Home About Contact

Thank you for viewing my SQL-based Data Analysis project on Pizza Sales.
I appreciate your time and attention.
I'm open to any questions or feedback!

Presented by: Sagnik Dhar
Email: sagnikdhar297@gmail.com
LinkedIn: linkedin_sagnik
Github: Github_sagnik