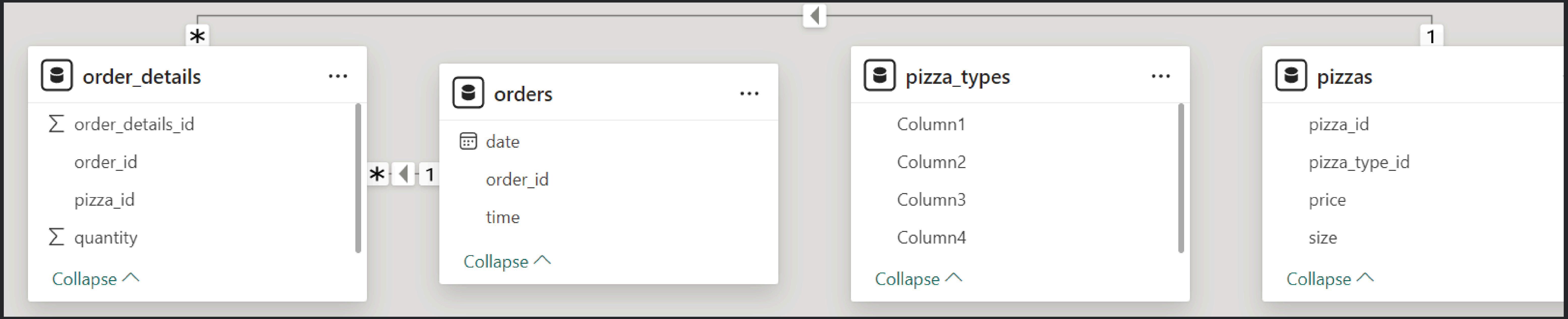
# SQL Project on Pizza Sales

# I am Nakul Sehrawat and in this project I have utilized SQL queries to answer questions on pizza sales

in  this project in have taken datasets of pizza sales and answered some questions related to it like which pizza category had maximum sale and many other.

# Database



**order_details**
- Σ order_details_id
- order_id
- pizza_id
- Σ quantity

Collapse ^

**orders**
- date
- order_id
- time

Collapse ^

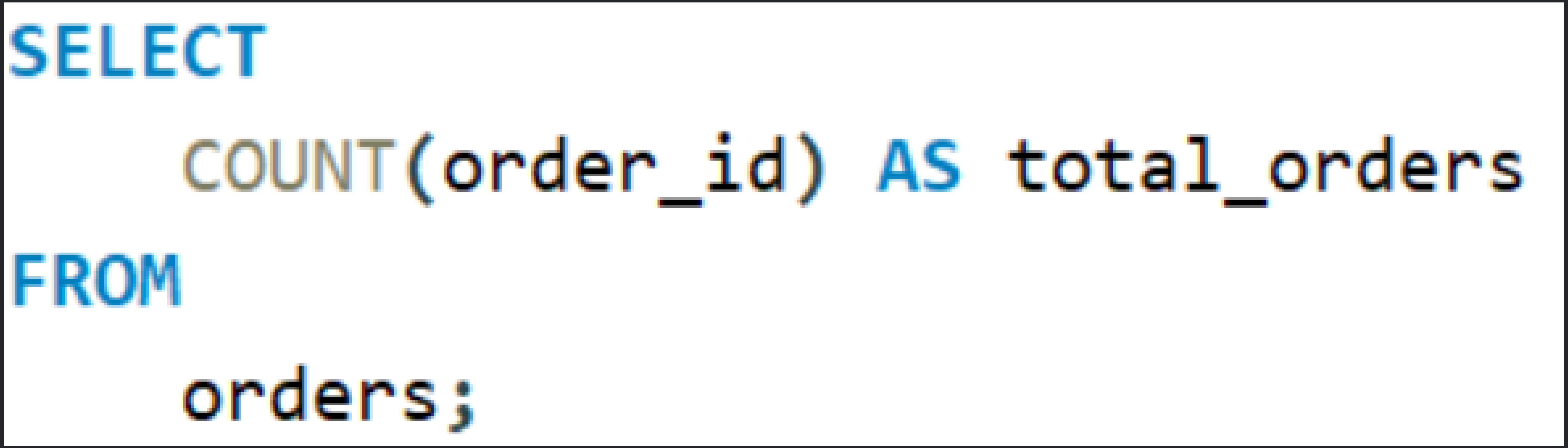
**pizza_types**
- Column1
- Column2
- Column3
- Column4

Collapse ^

**pizzas**
- pizza_id
- pizza_type_id
- price
- size

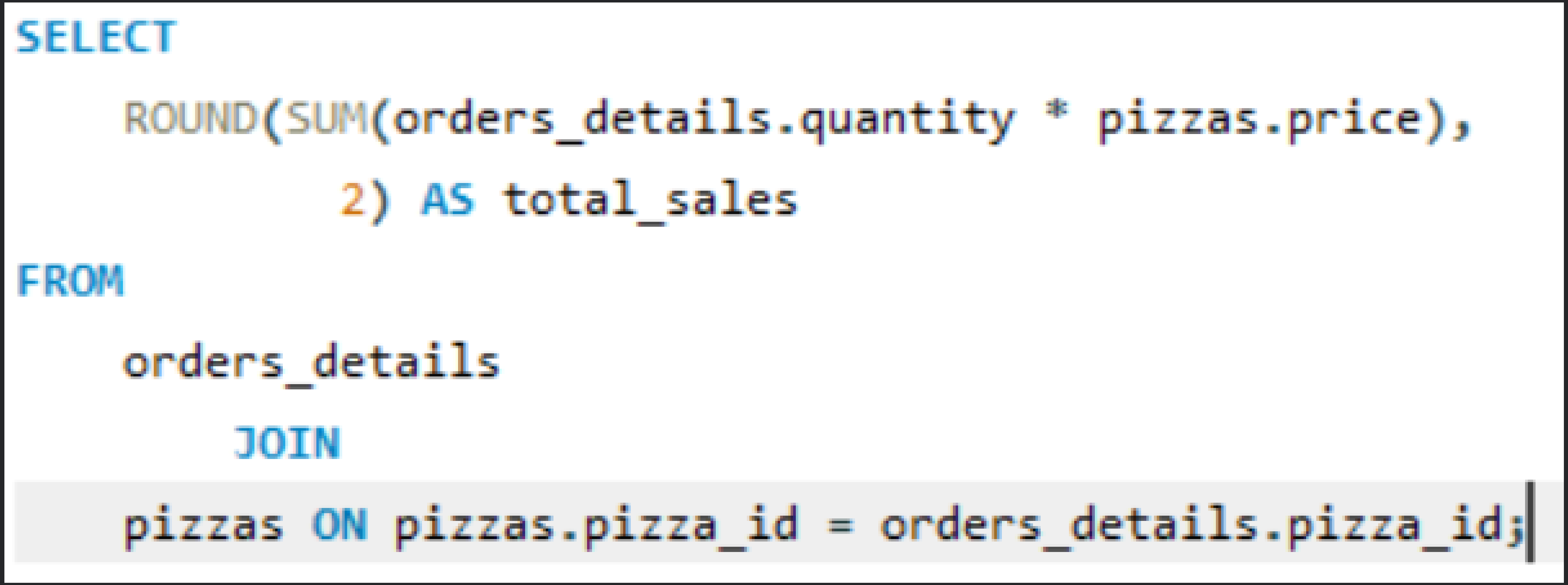
Collapse ^

# Retrieve the total number of order placed



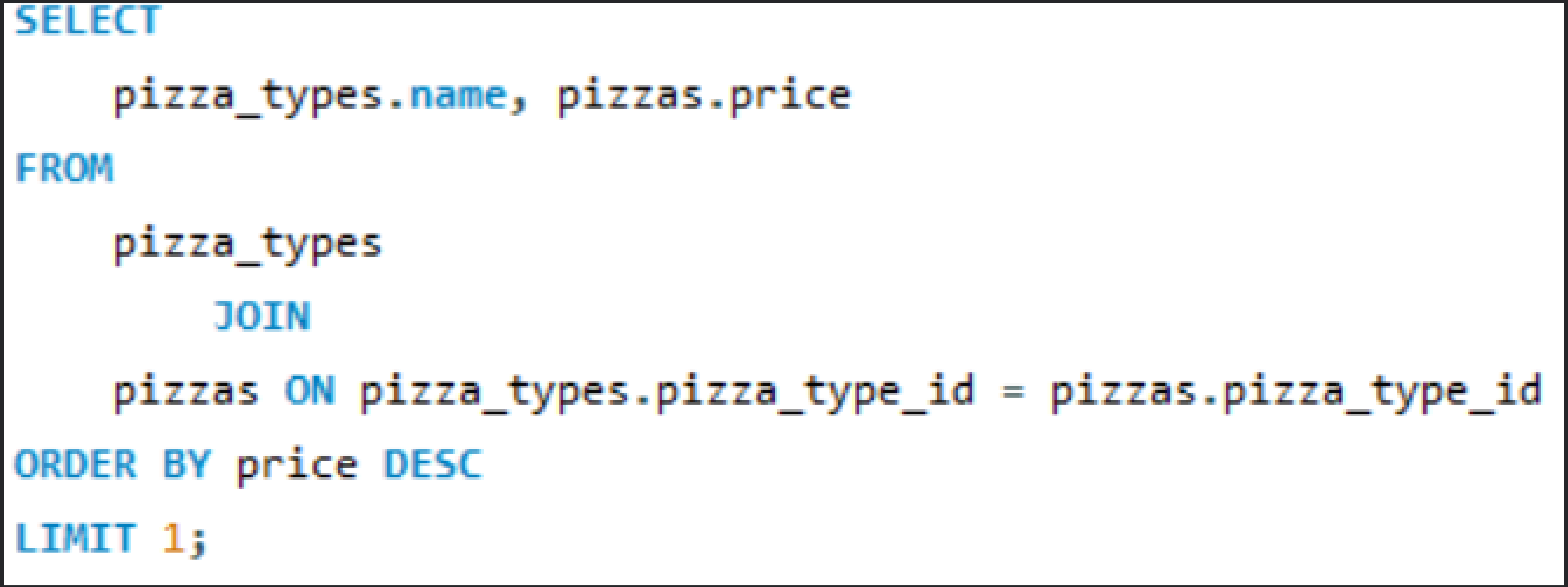
```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Calculate total revenue generated from pizza sales



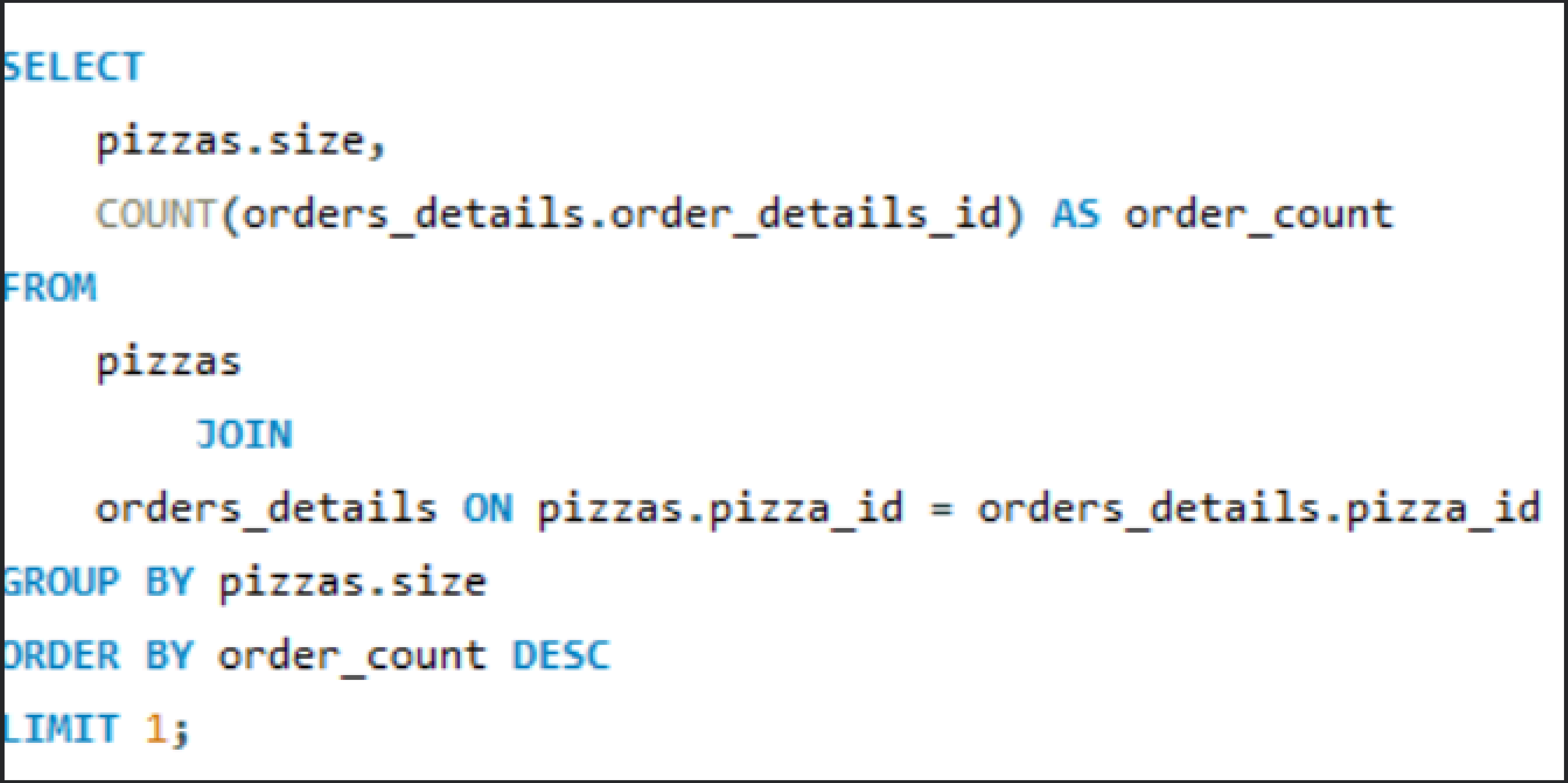
```sql
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

# Identify the highest price pizza



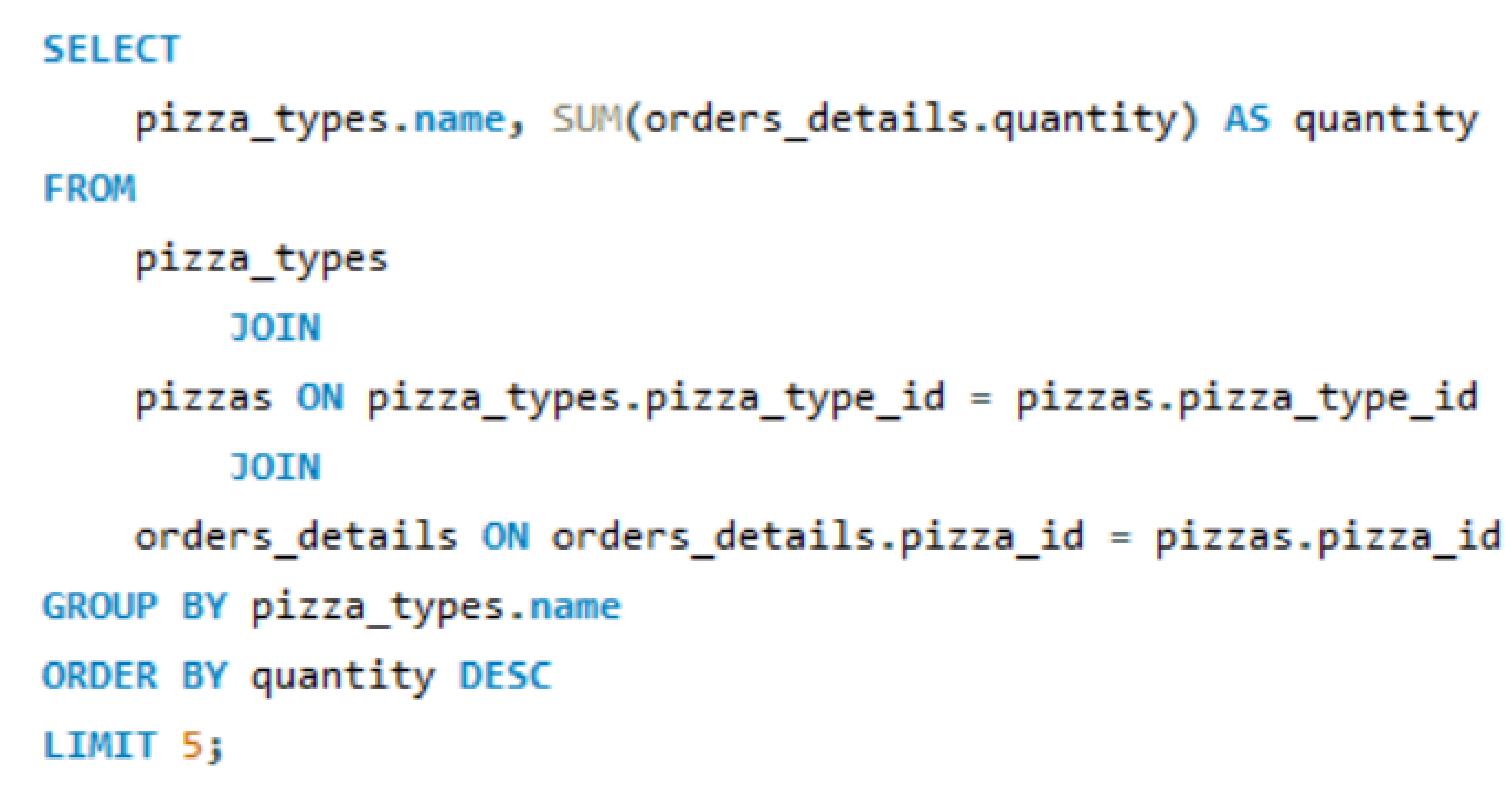
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Identify the most common pizza size ordered



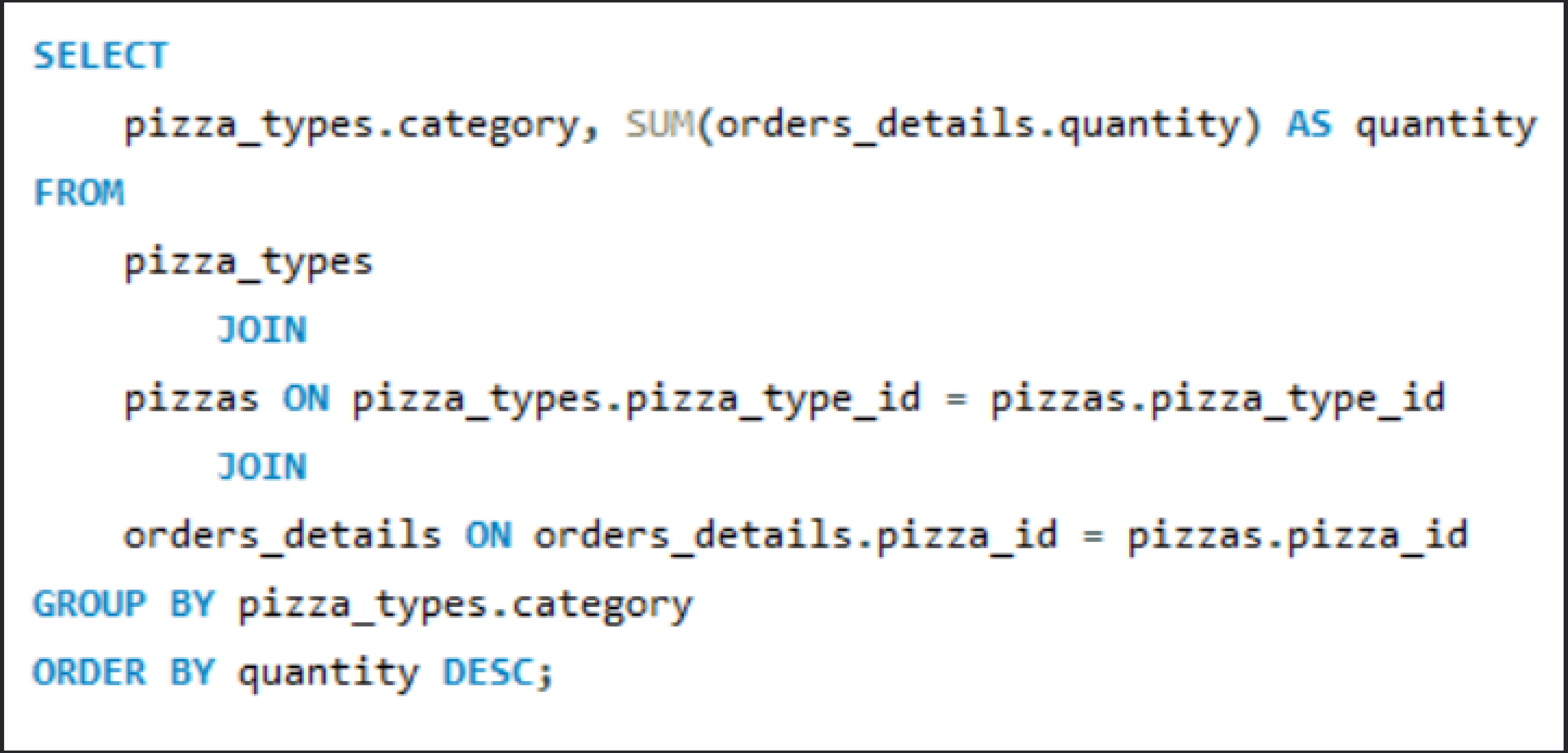
```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

## List the top 5 most ordered pizza types with their quantities



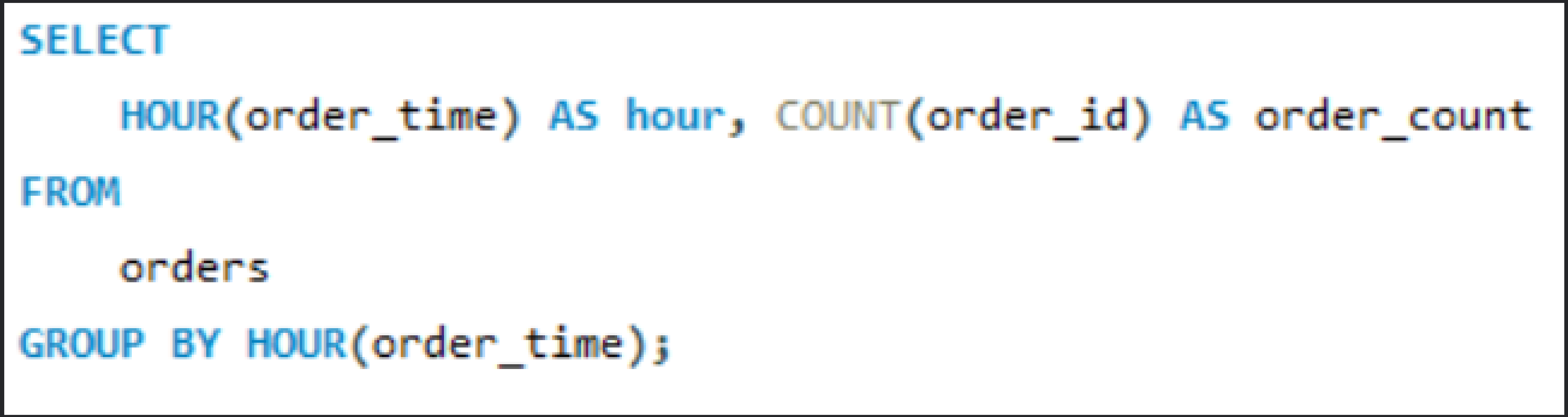
```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

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



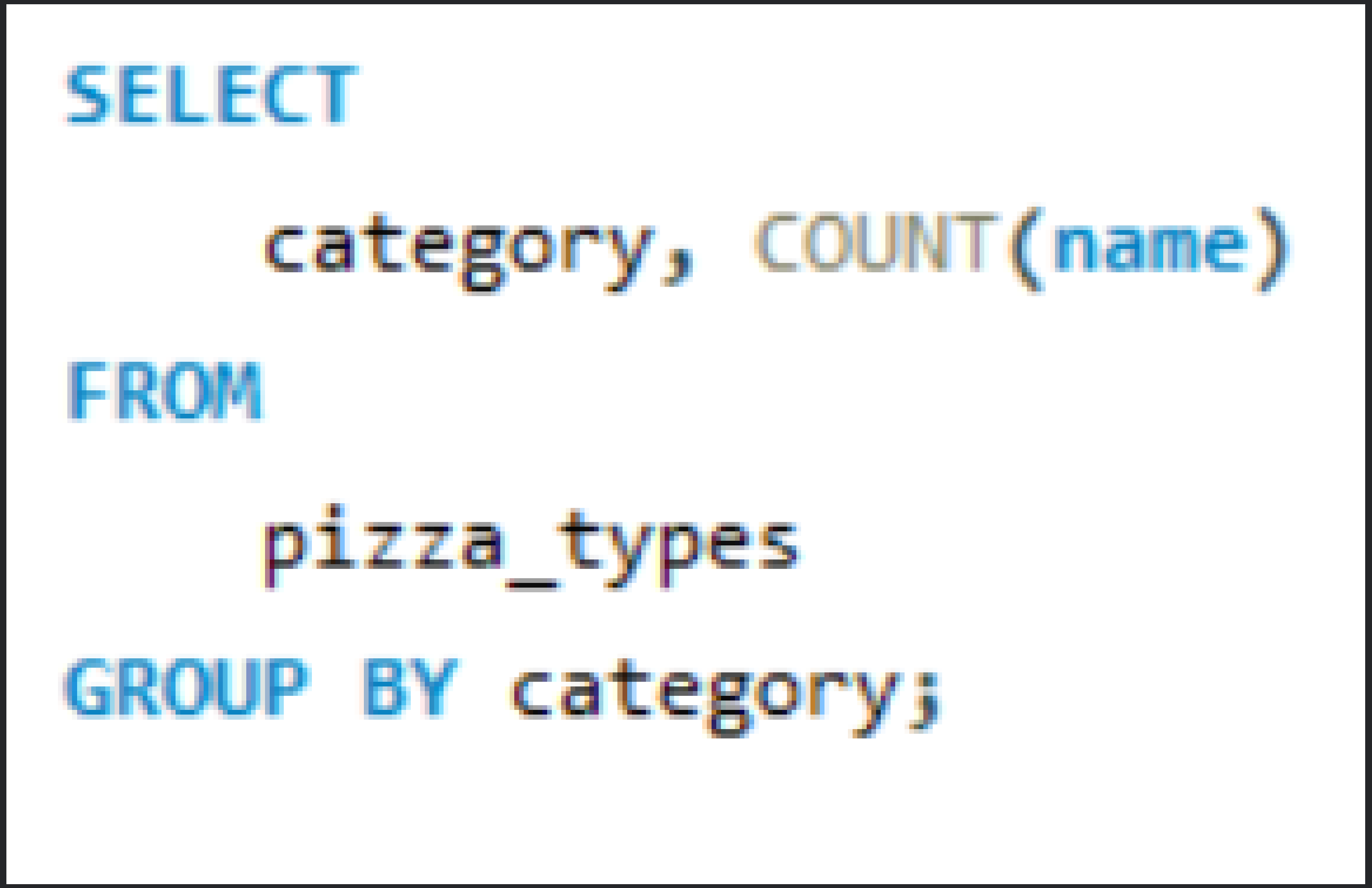
```
SELECT
    pizza_types.category, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;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

## Determine the distribution of orders by hour of the day



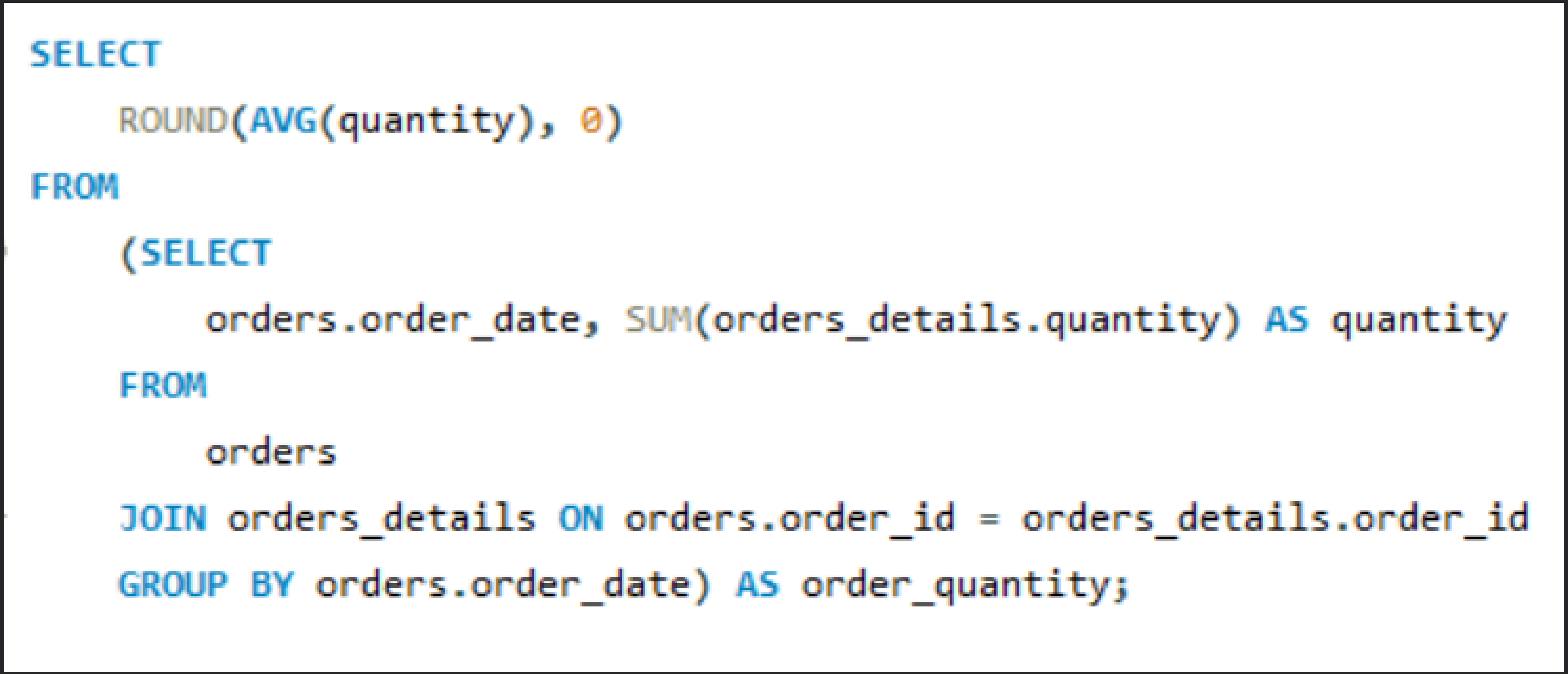
```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Join the relevant tables to find category wise distribution of pizzas



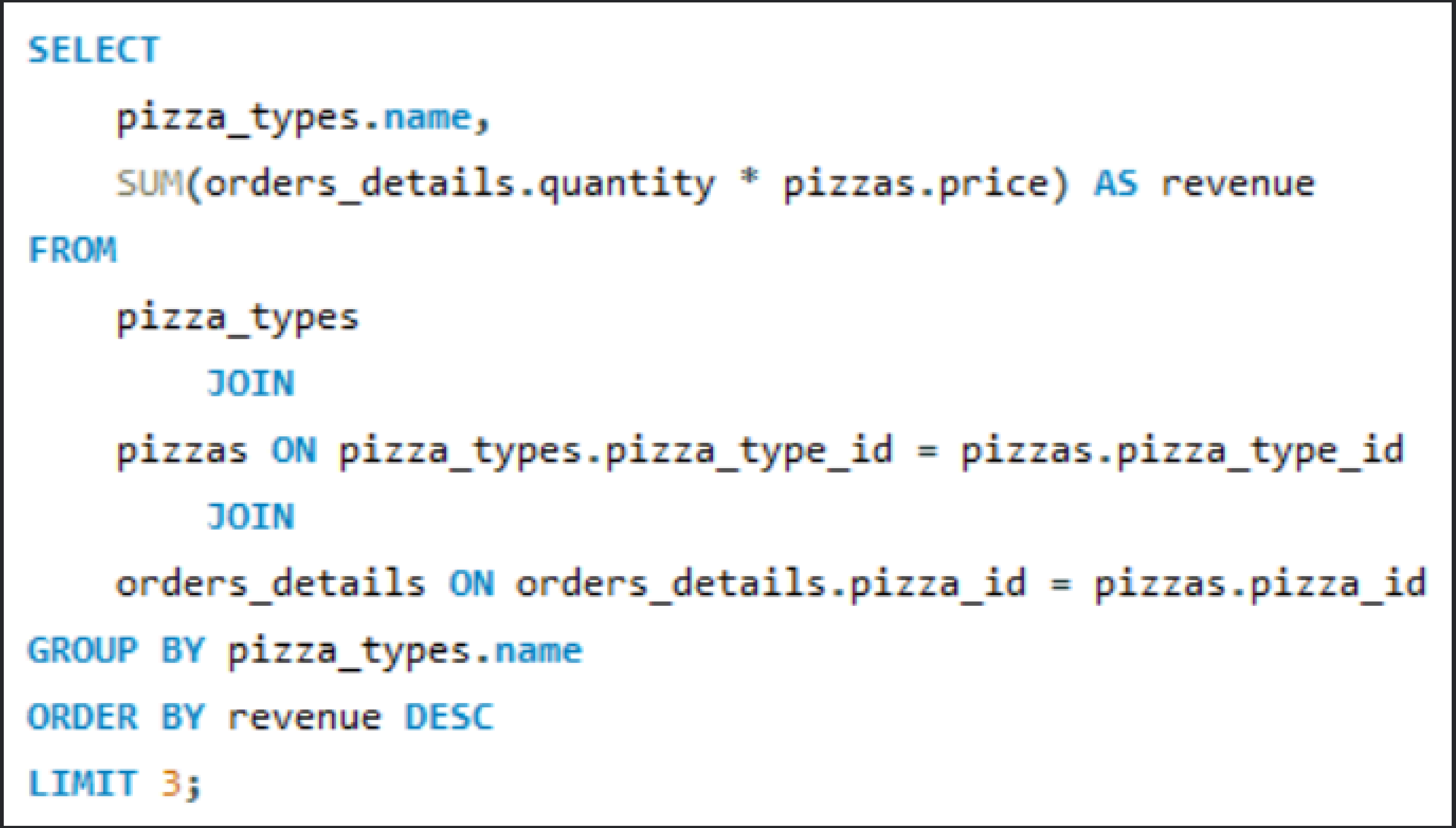
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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



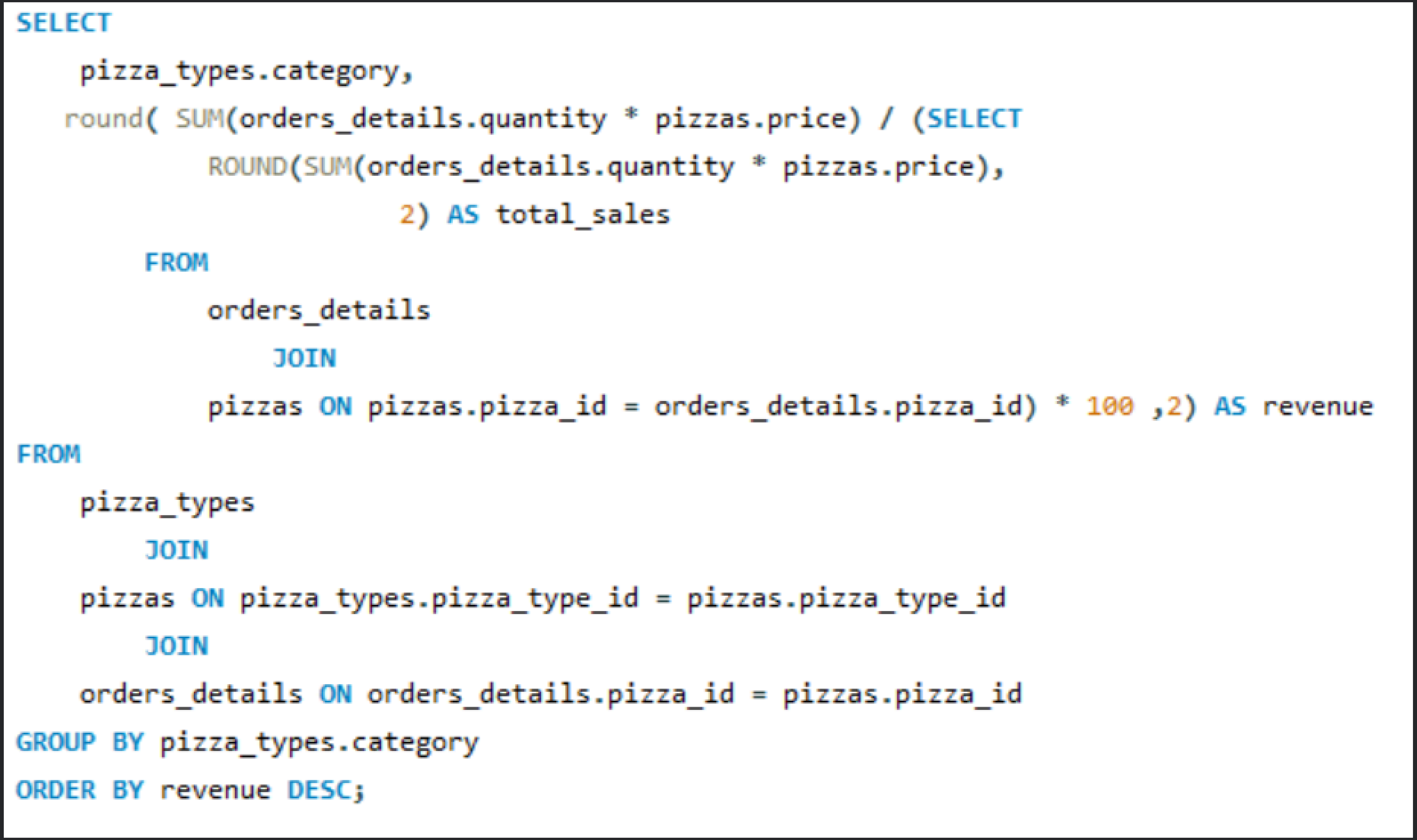
```sql
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Determine the top 3 most ordered pizza types based on revenue



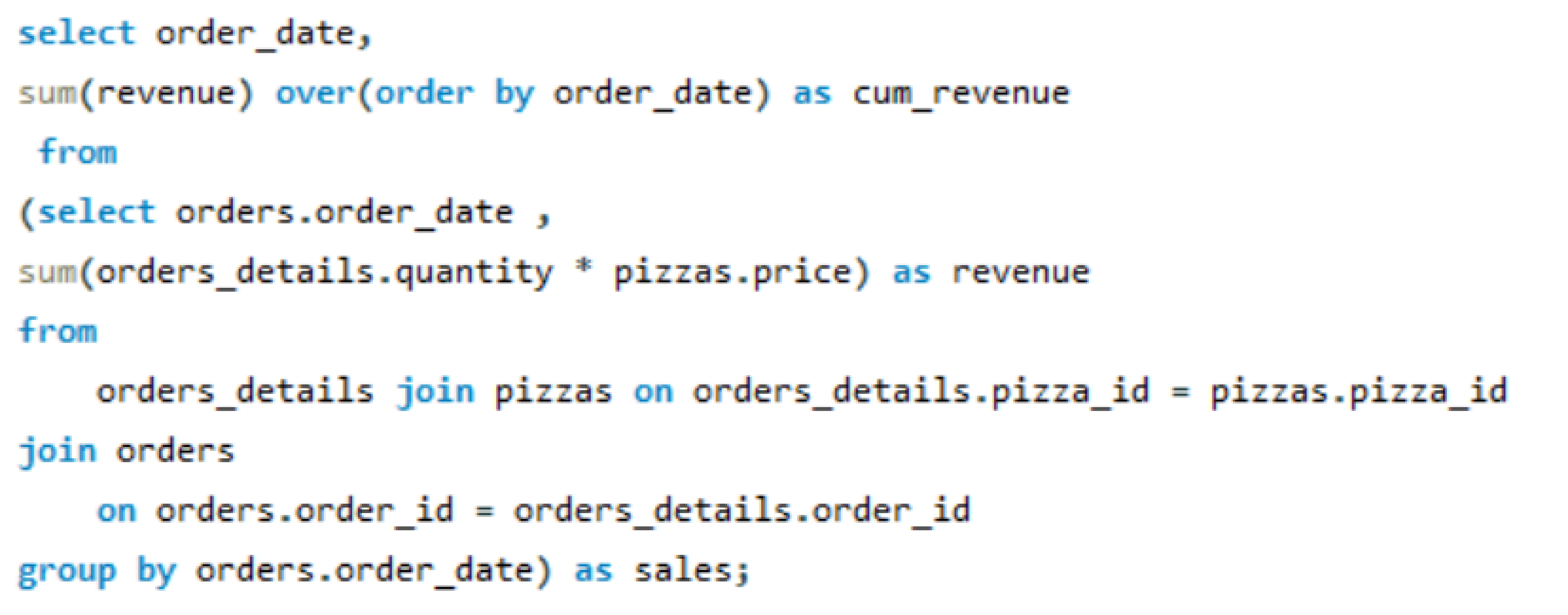
```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

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



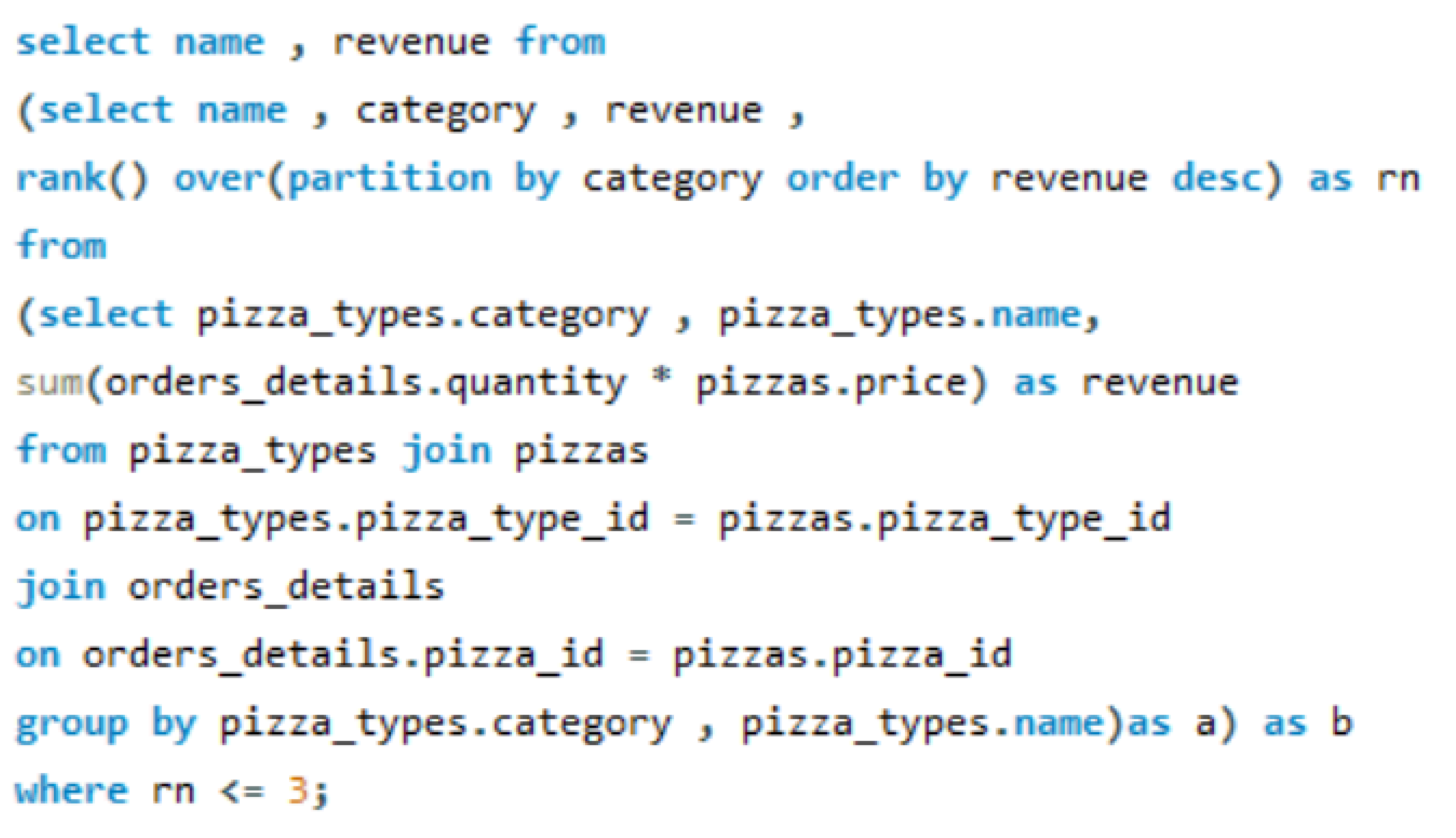
```
SELECT
    pizza_types.category,
   round( SUM(orders_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id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

## Analyse the cumulative revenue generated over time



```
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;
```

## Determine the top 3 pizza type ordered based on revenue for each pizza category



```sql
select name , revenue from
(select name , category , 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;
```

# THANK YOU!