Where Every Slice is a Taste of Perfection

PIZZA SALES MYSQL PROJECT

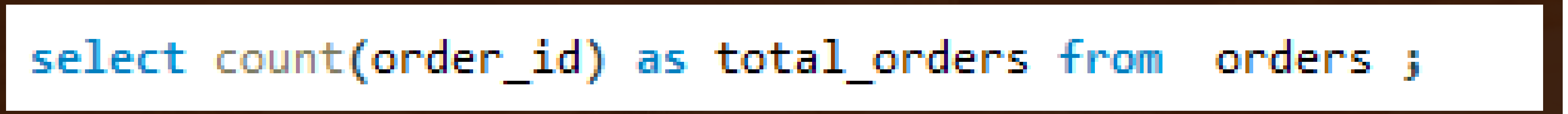
Start Your Slide

INTRODUCTION

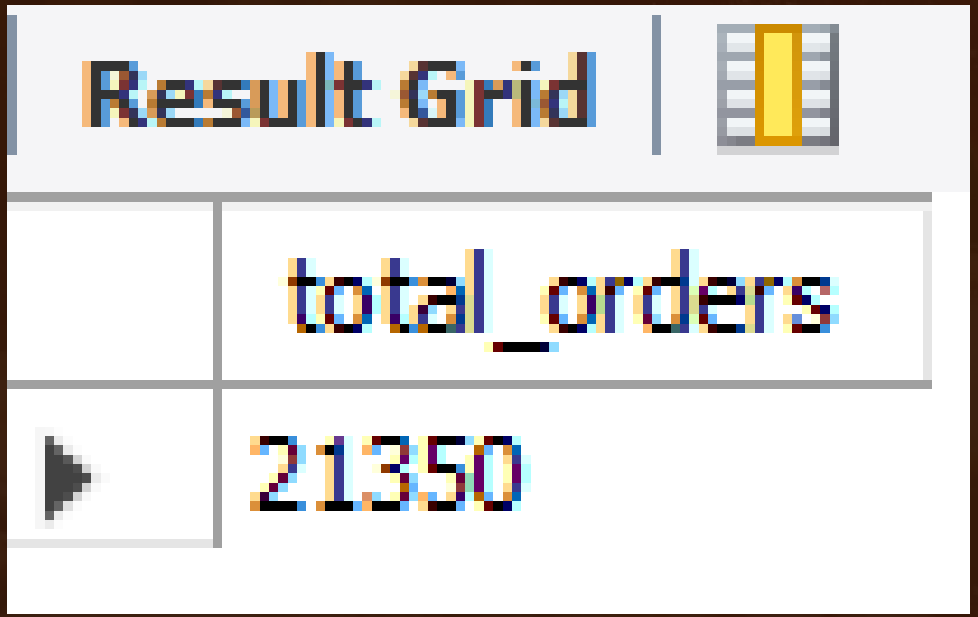
This presentation showcases SQL queries used for analyzing pizza sales.
Queries cover basic, intermediate, and advanced analysis.
Insights include revenue, top-selling pizzas, order trends, and category-wise analysis.

SALES ANALYSIS

Query 1: Retrieve the total number of orders placed

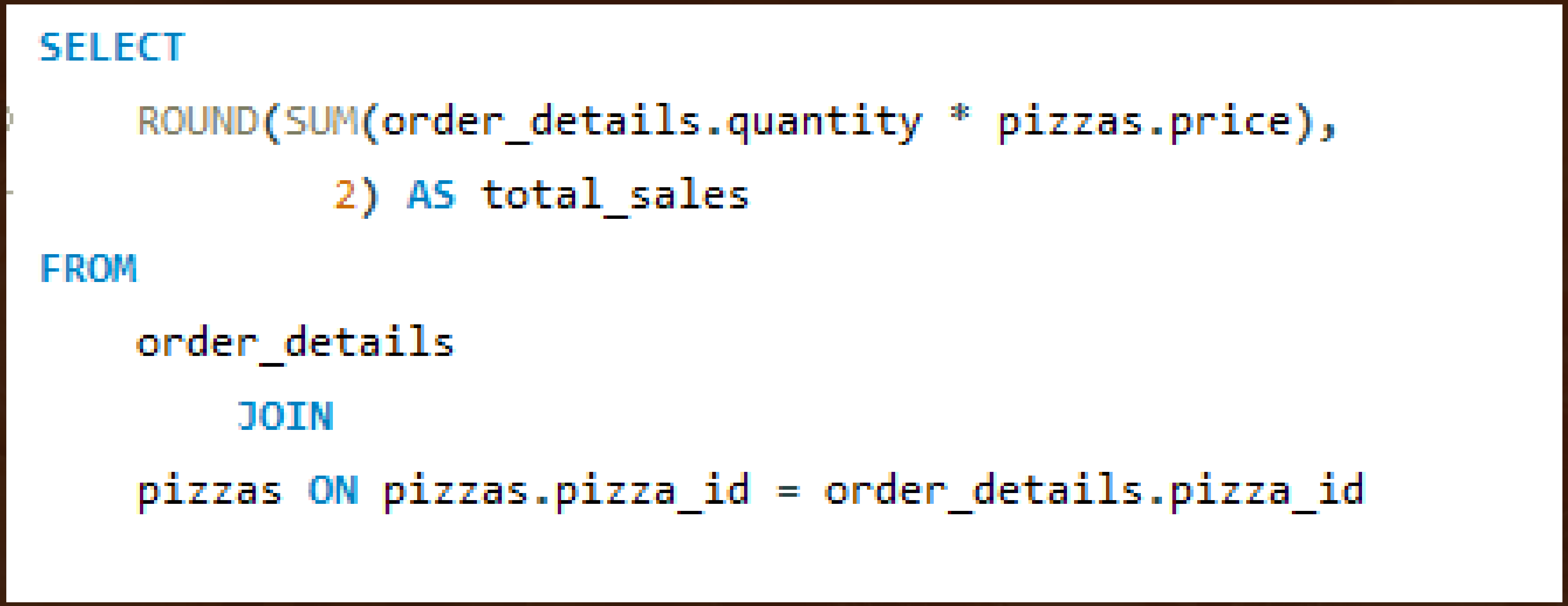


```sql
select count(order_id) as total_orders from  orders ;
```



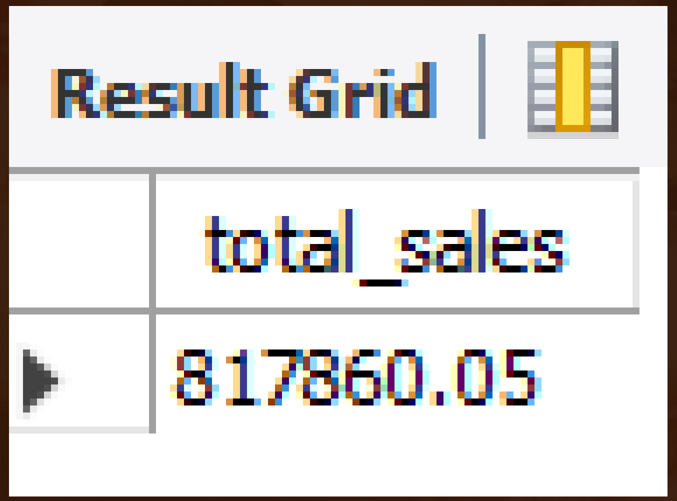
```
total_orders
21350
```

Query 2: Calculate the total revenue generated from pizza sales



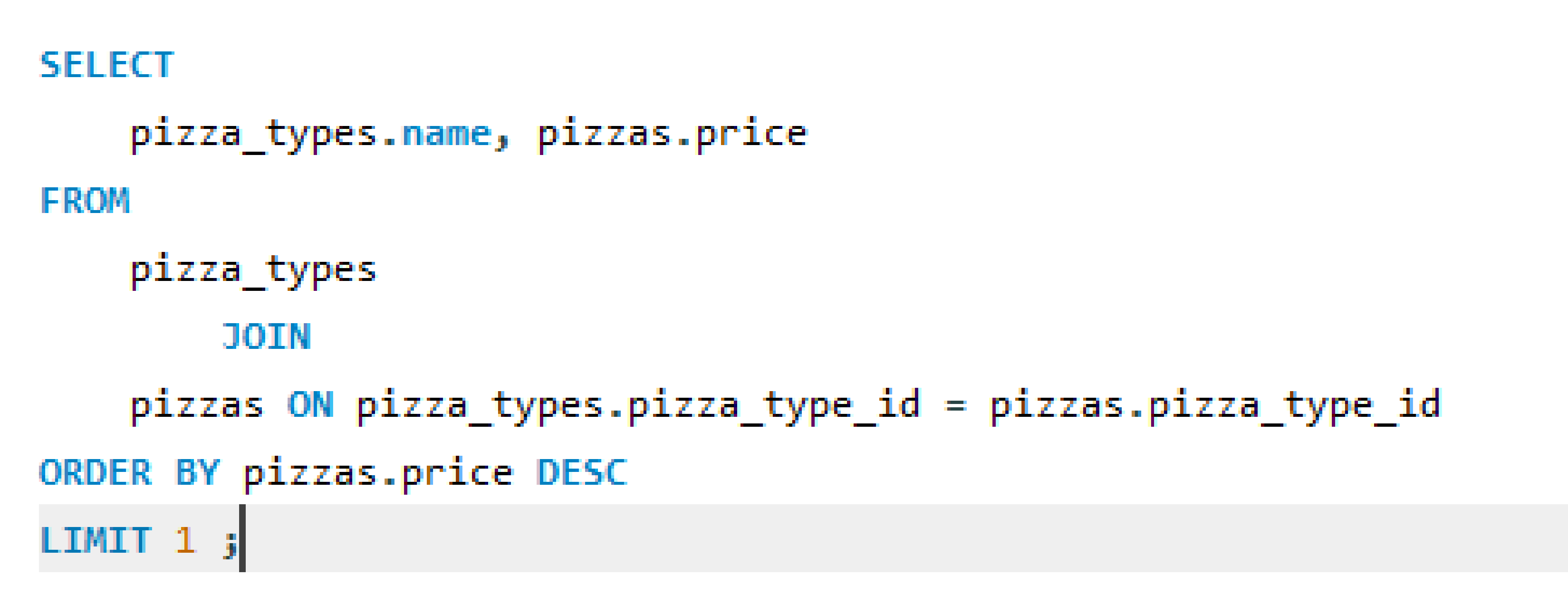
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

ORDER NOW

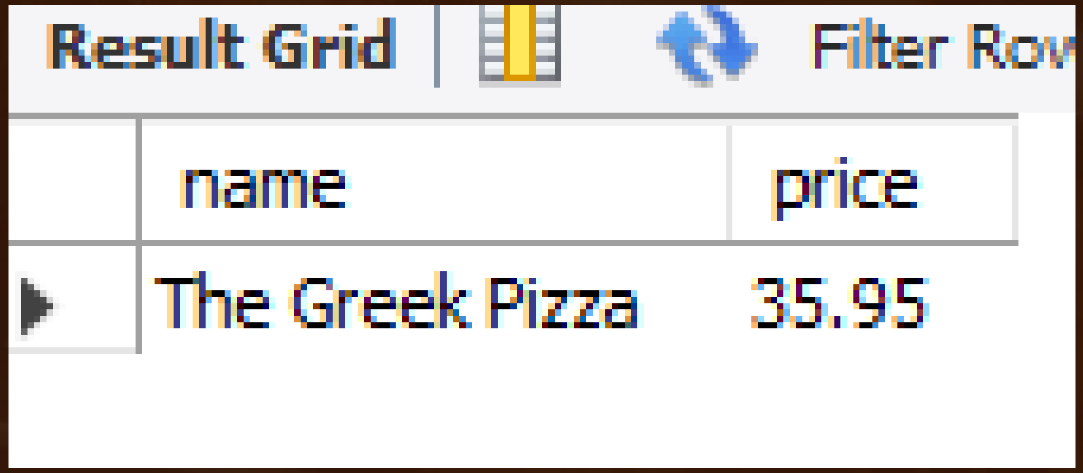


total_sales
817860.05

Querry 3 : Identify the highest price

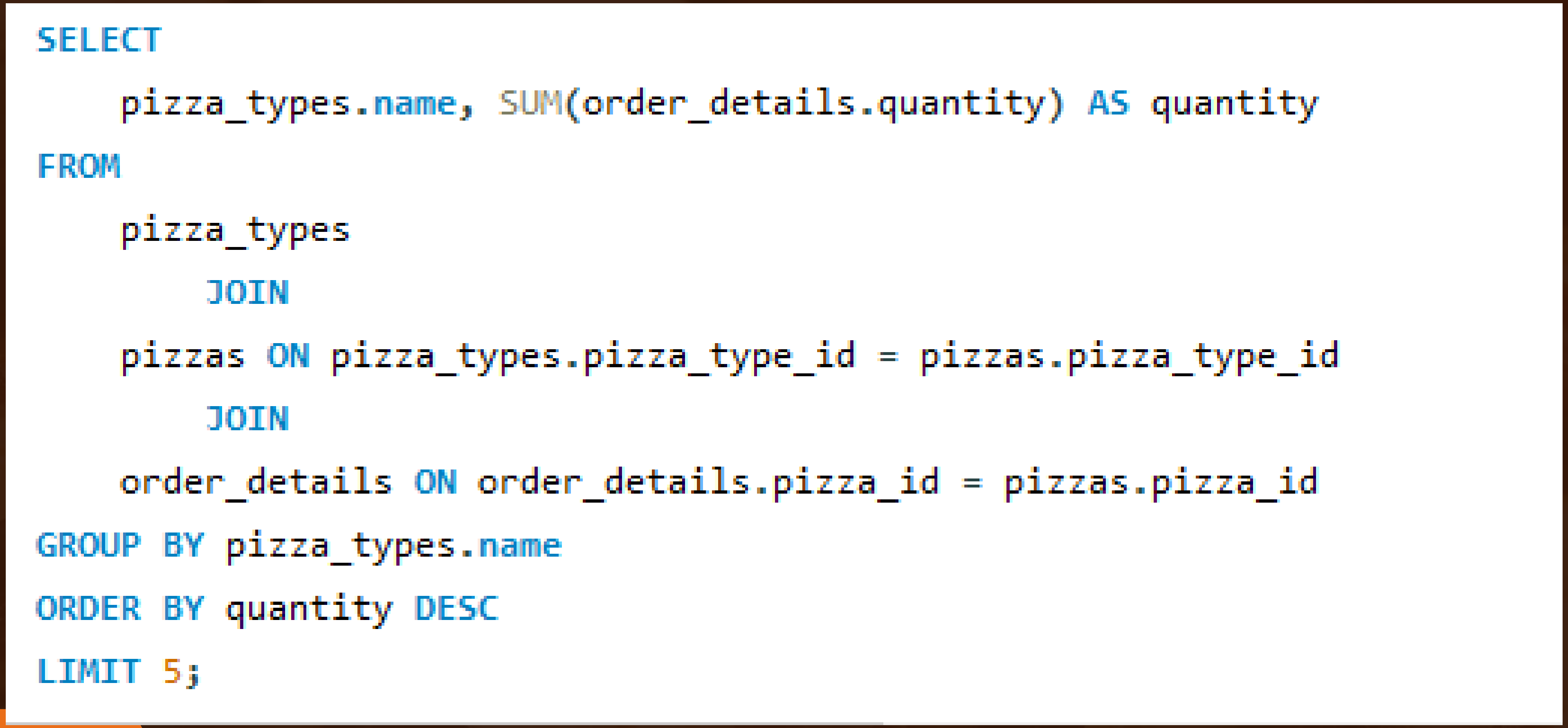


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1 ;
```

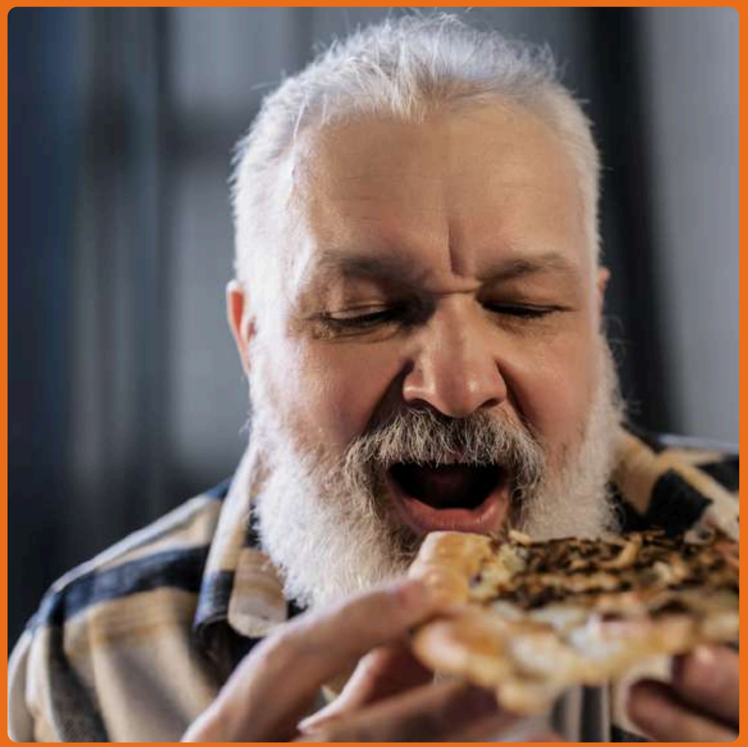


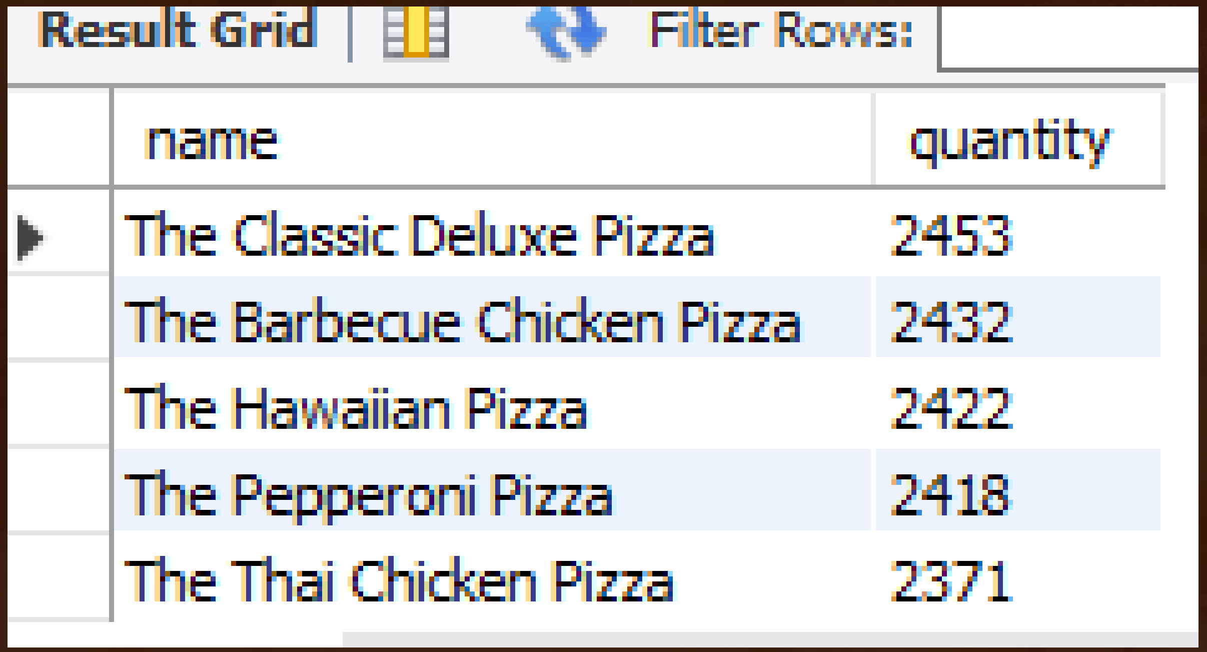
name	price
The Greek Pizza	35.95

Querry:4 Ilist the top 5 most ordered pizza types allong with their quantities



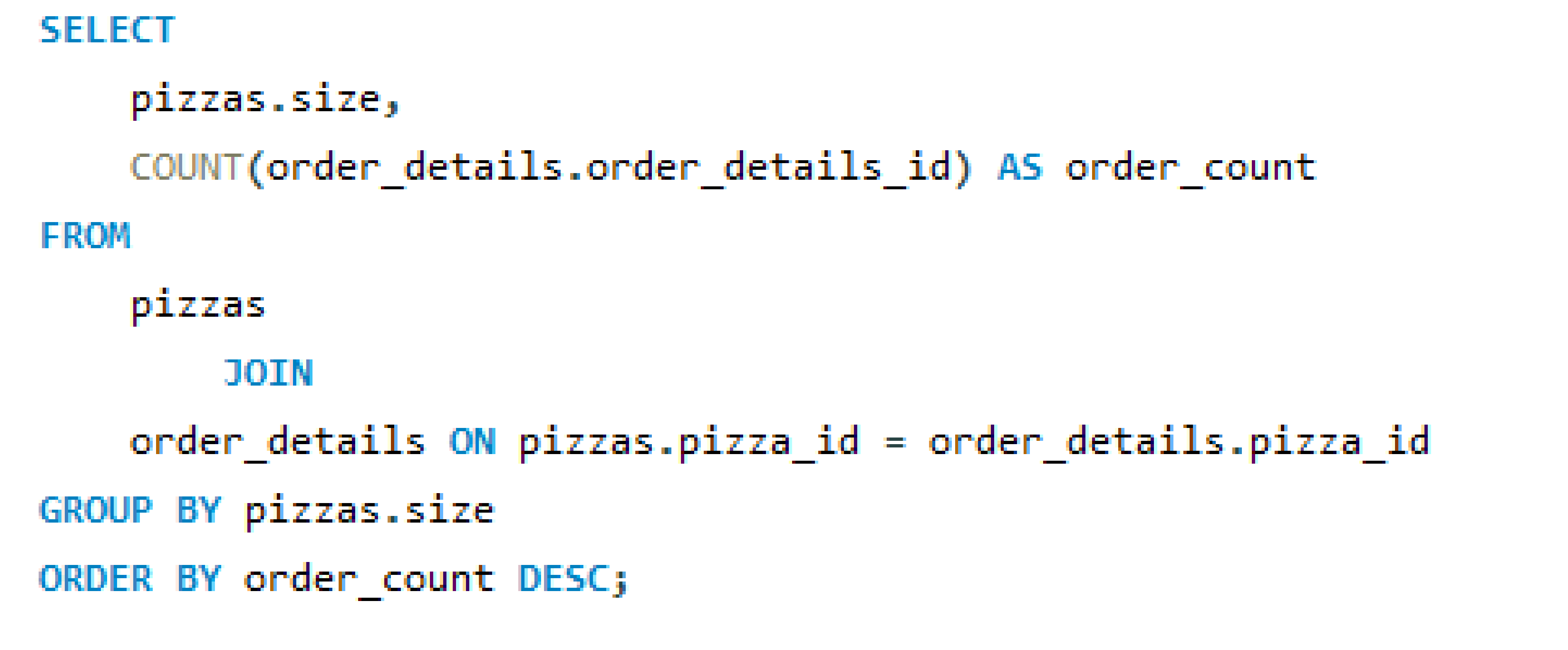
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;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



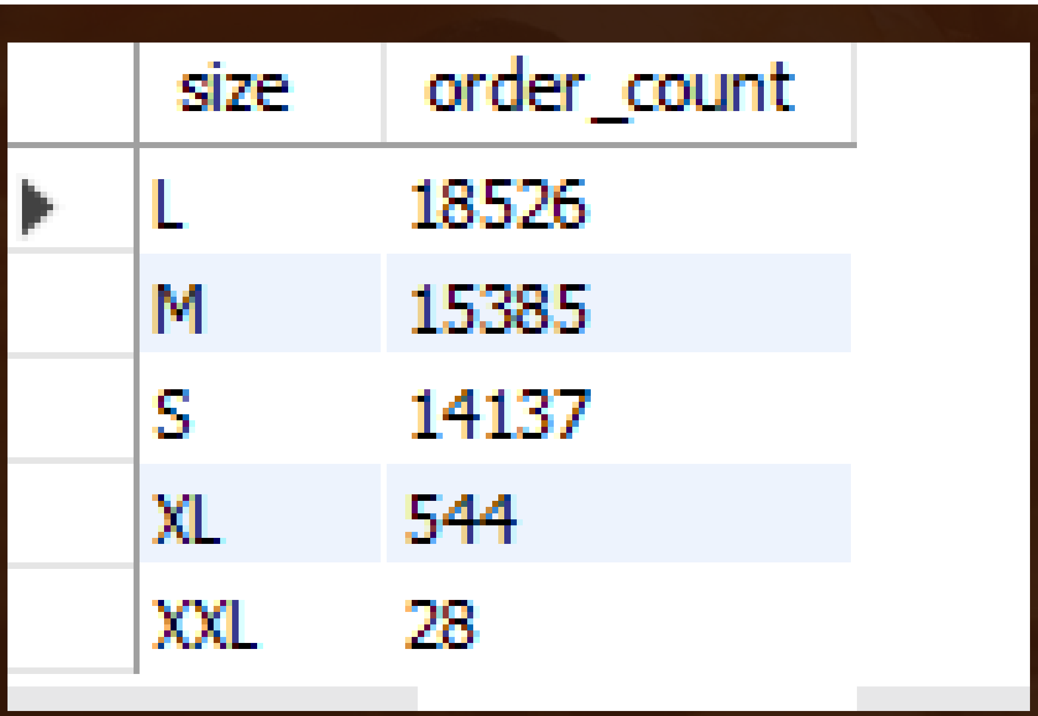


name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Querry 5: Identify the most common pizza size ordered.

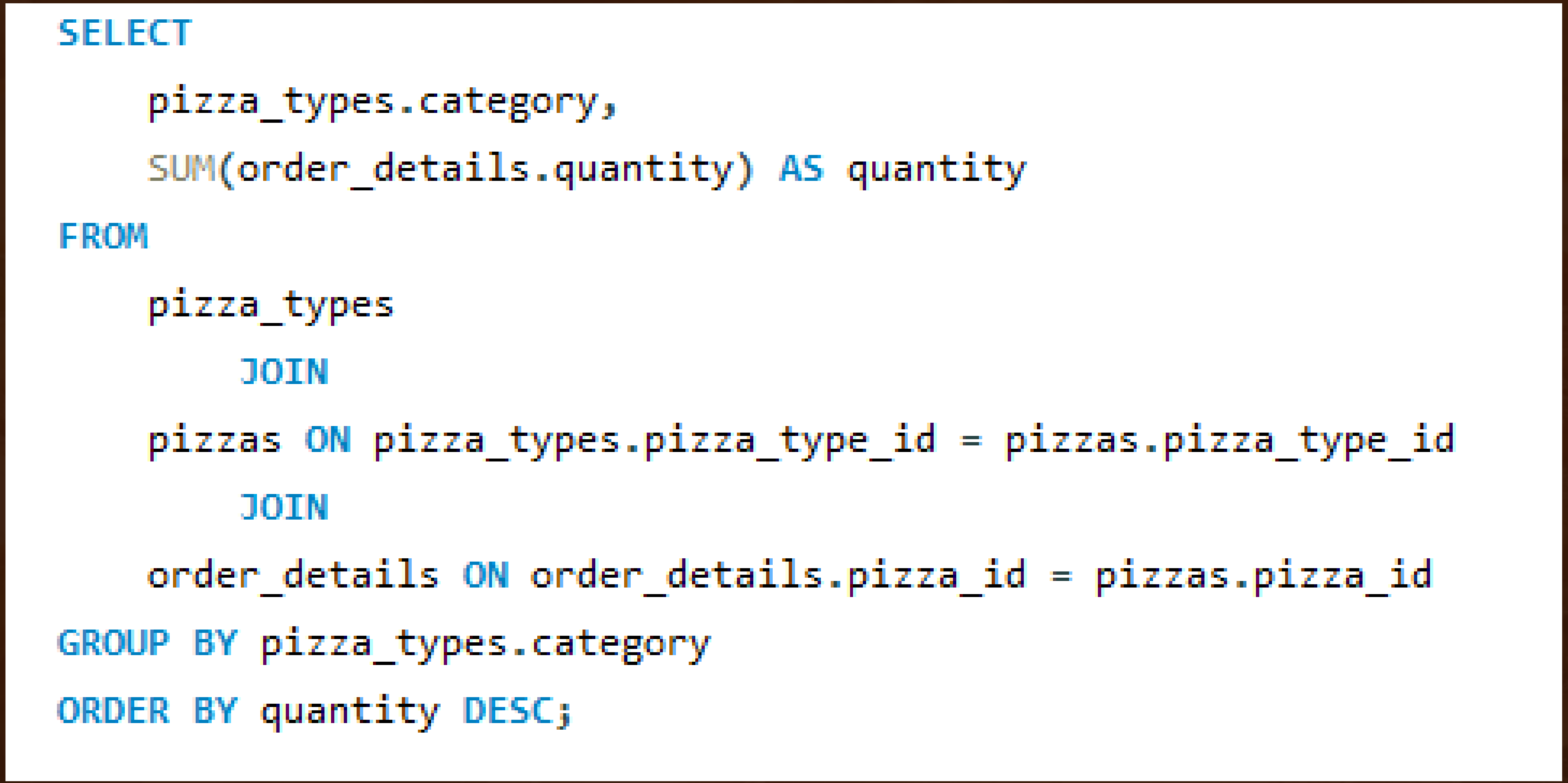


```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



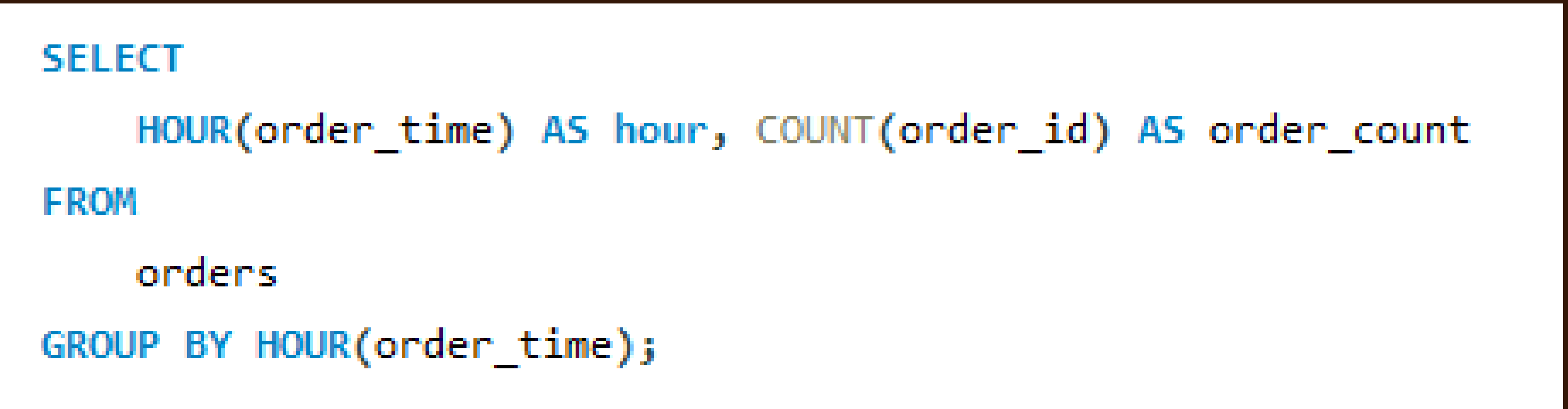
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Querry 6 : join the necessary tables to find the total quantity of each pizza category ordered.

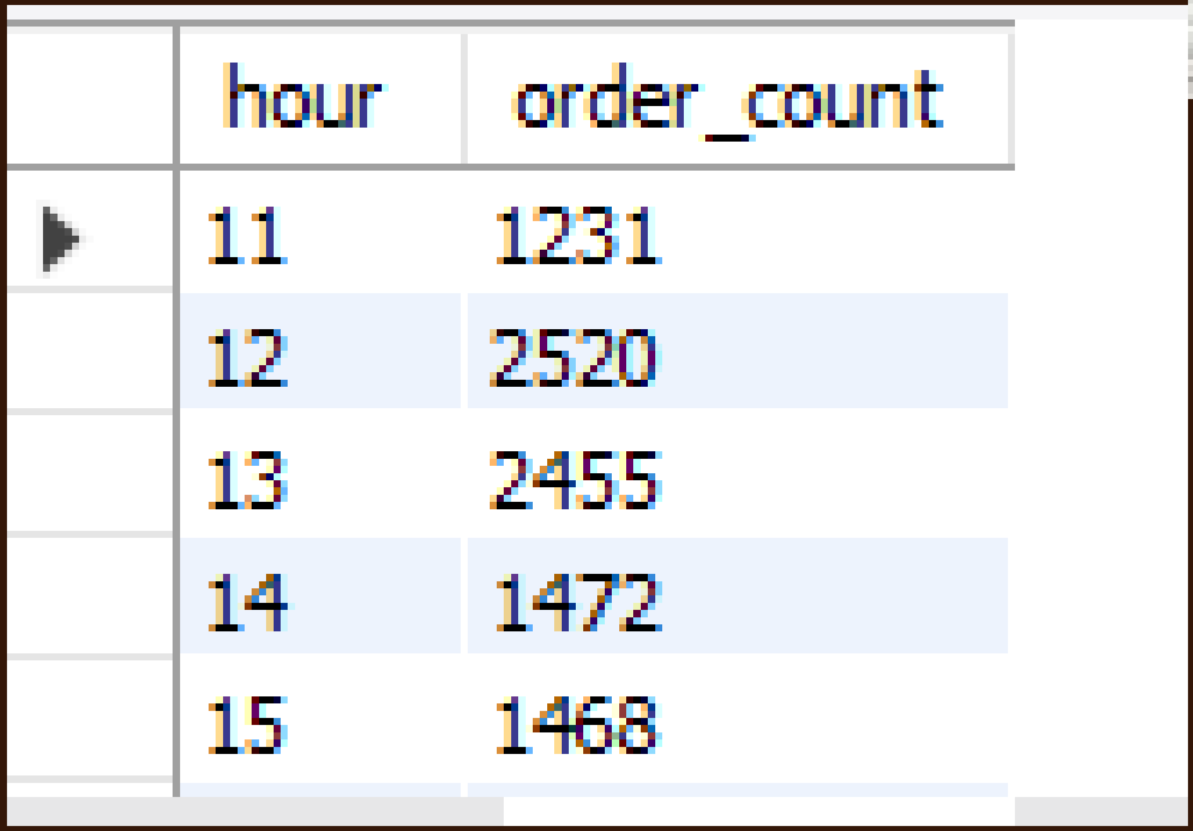


```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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

Querry 7 : Determine the distribution of orders by hour of the day

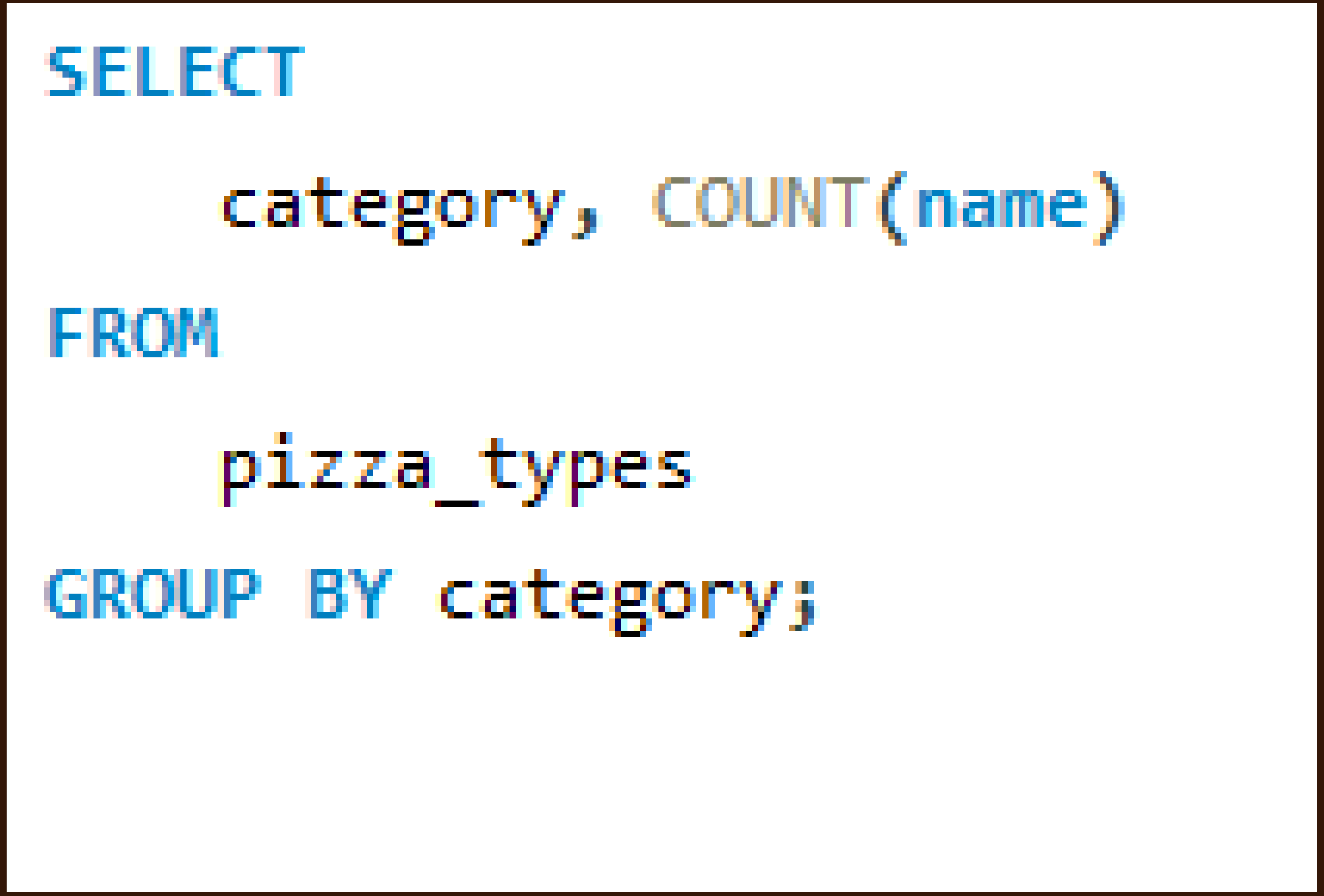


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

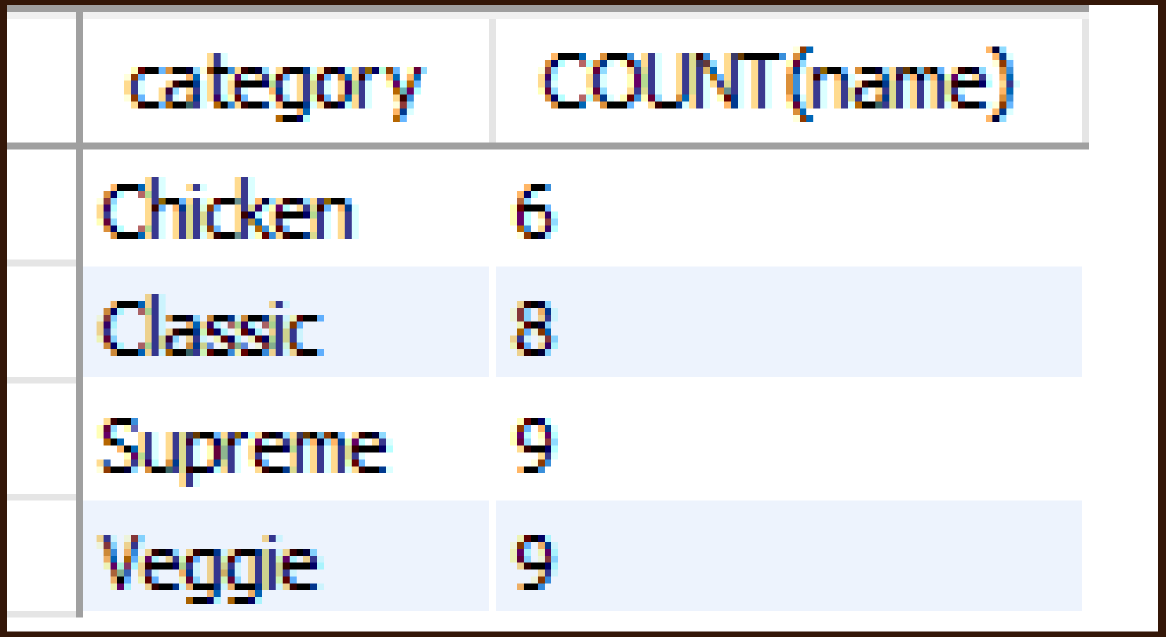


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

Querry 8: join relevent tables to find the category wise distribution of pizzas

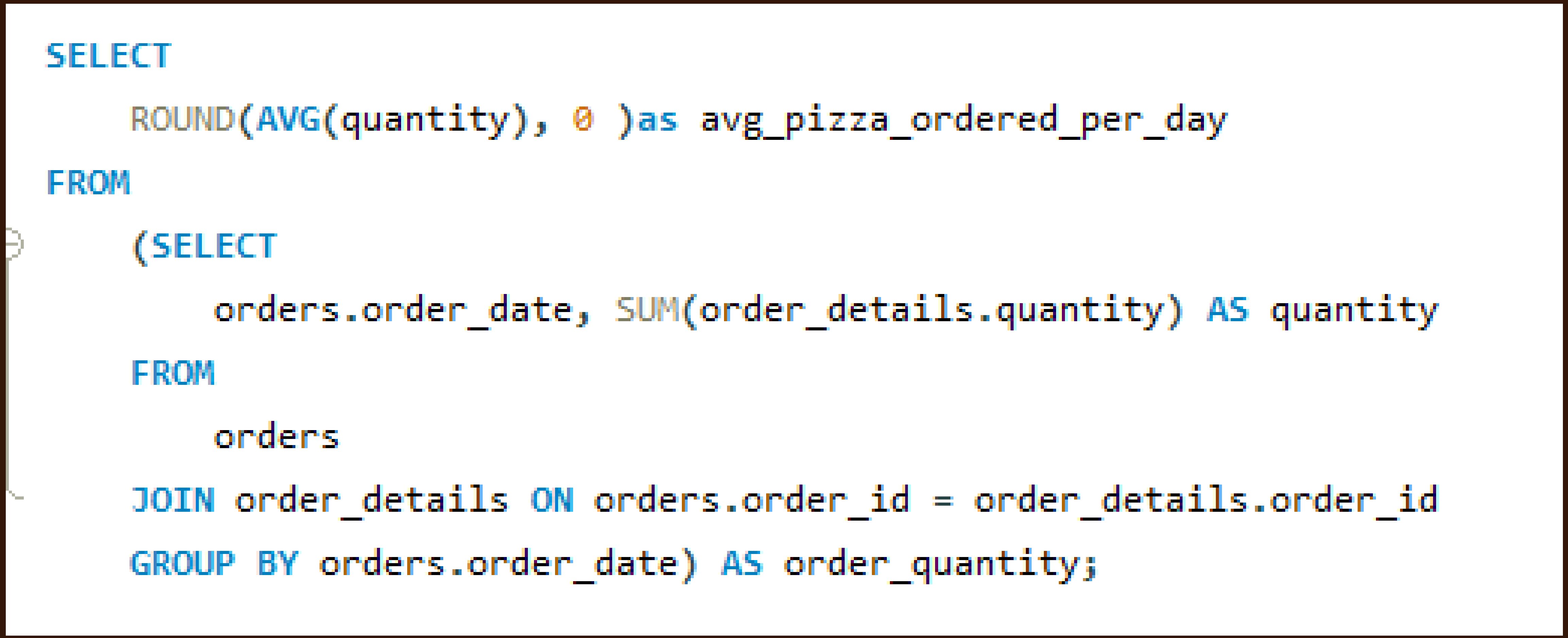


```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

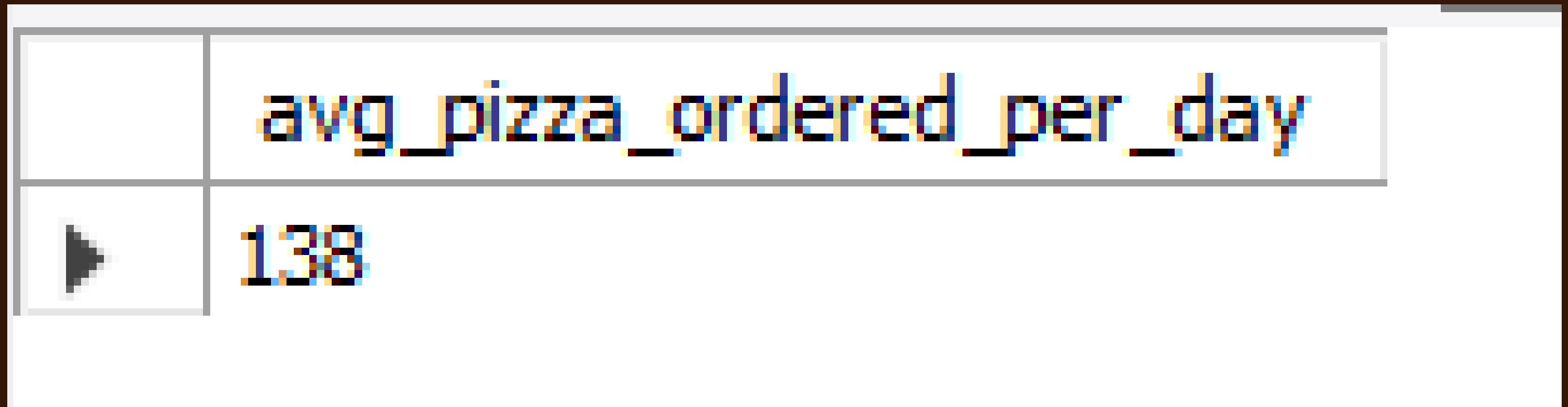


category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Querry 9: Group the orders by date and calculate the average number of pizzas ordered per day.

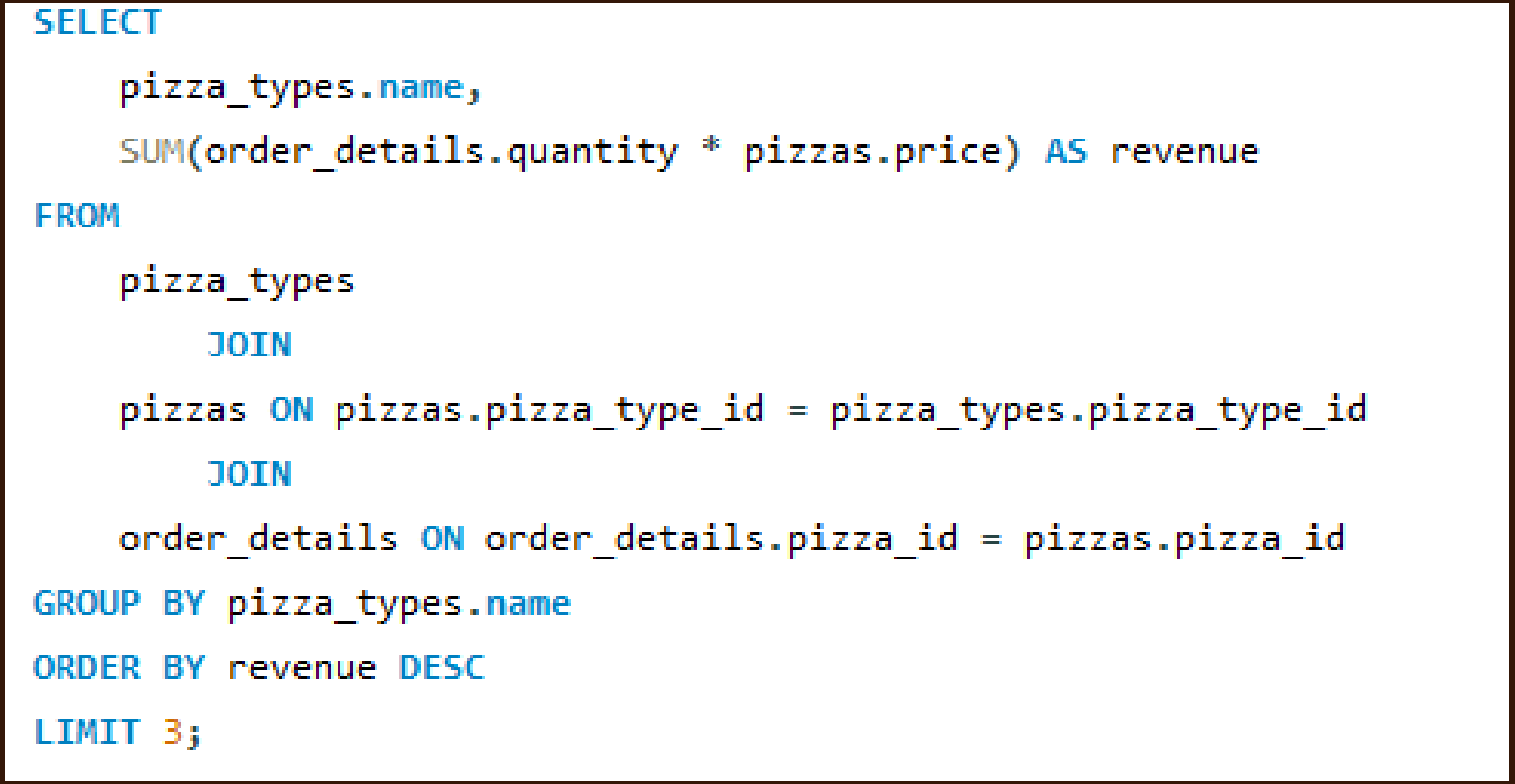


```
SELECT
    ROUND(AVG(quantity), 0 )as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

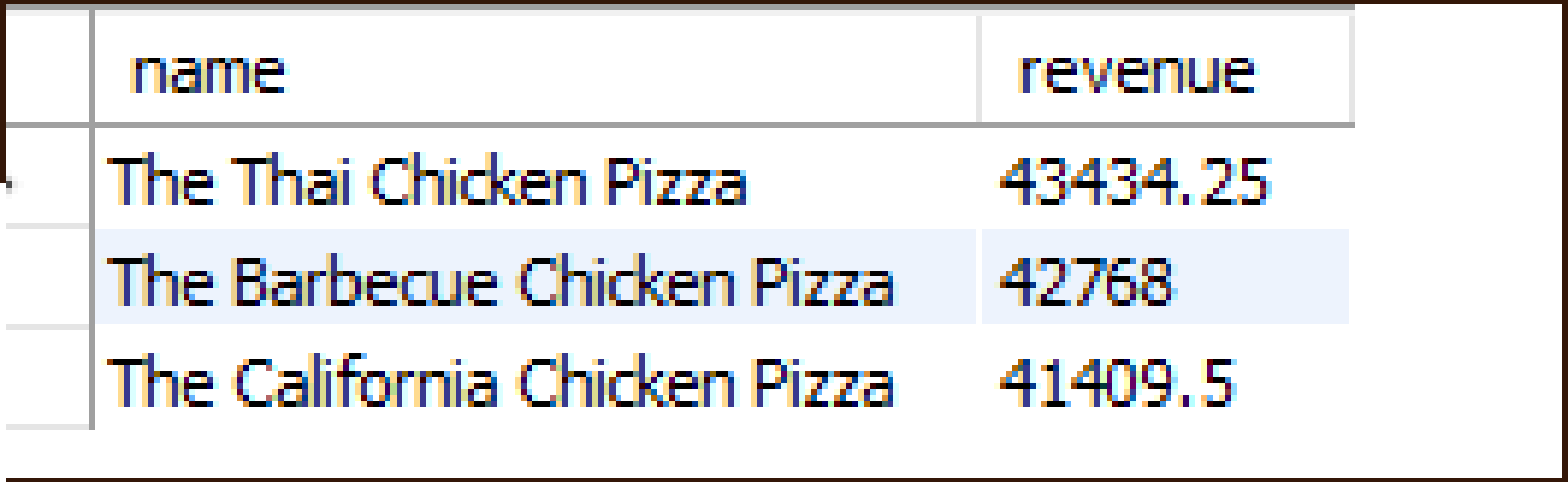


	avg_pizza_ordered_per_day
▶	138

Querry 10: Determine the top 3 most ordered pizza type based on revenue.

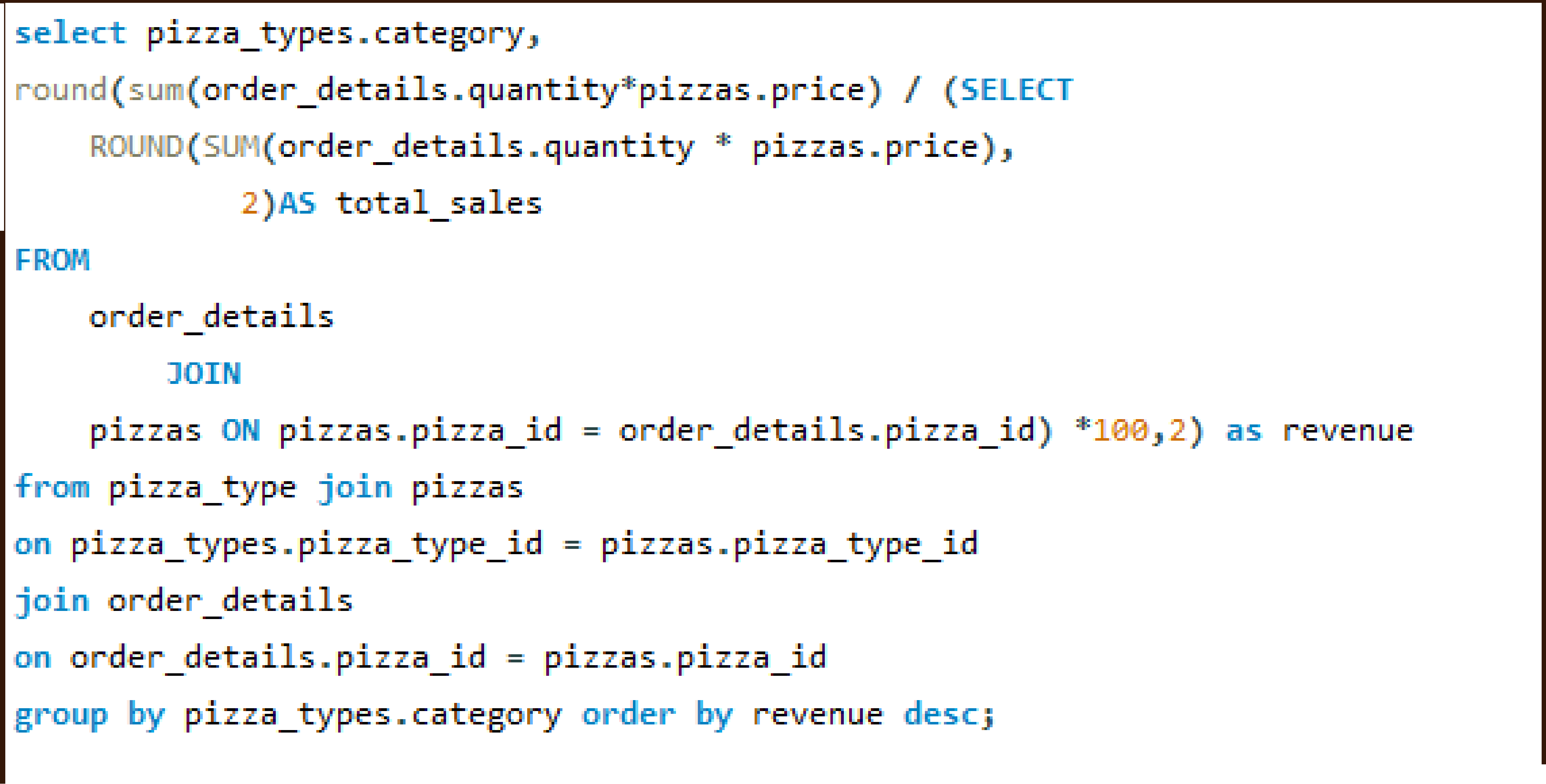


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue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



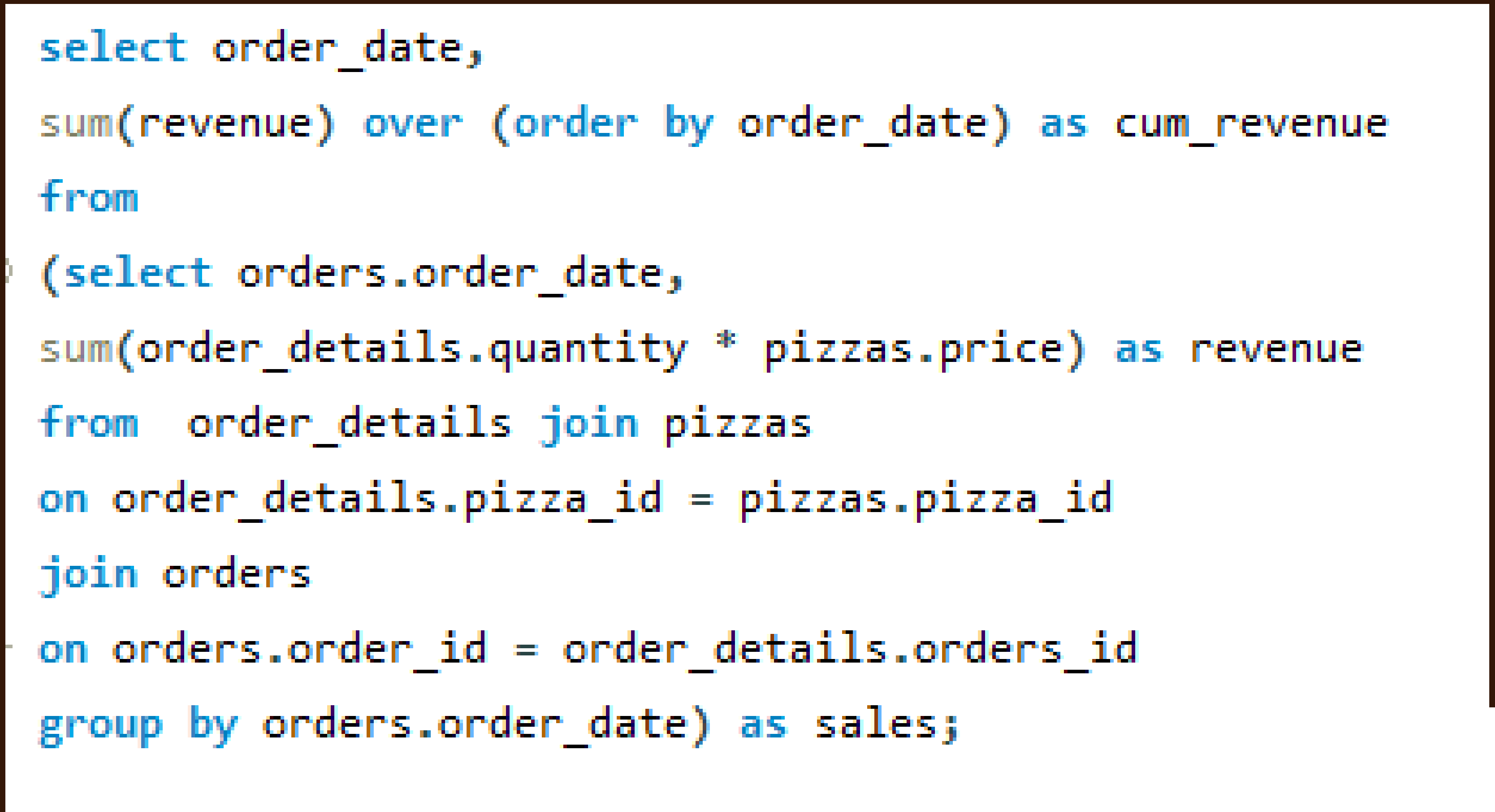
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Querry 11: Calculate the percentage contribution of each pizza type to total revenue.



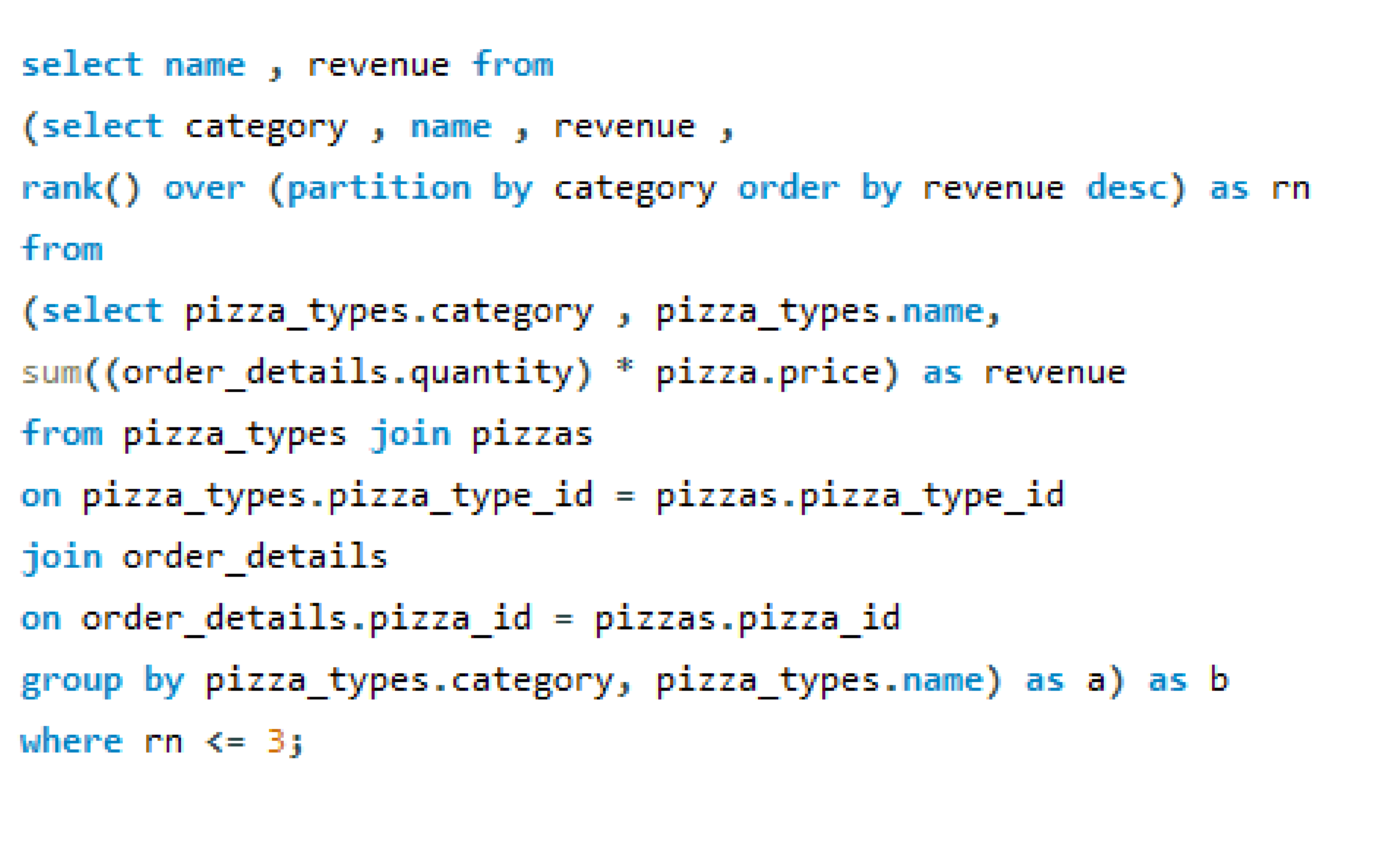
```
select pizza_types.category,
round(sum(order_details.quantity*pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2)AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2) as revenue
from pizza_type join 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 revenue desc;
```

Querry 12: Analyze the cumulative revenue generated over time .

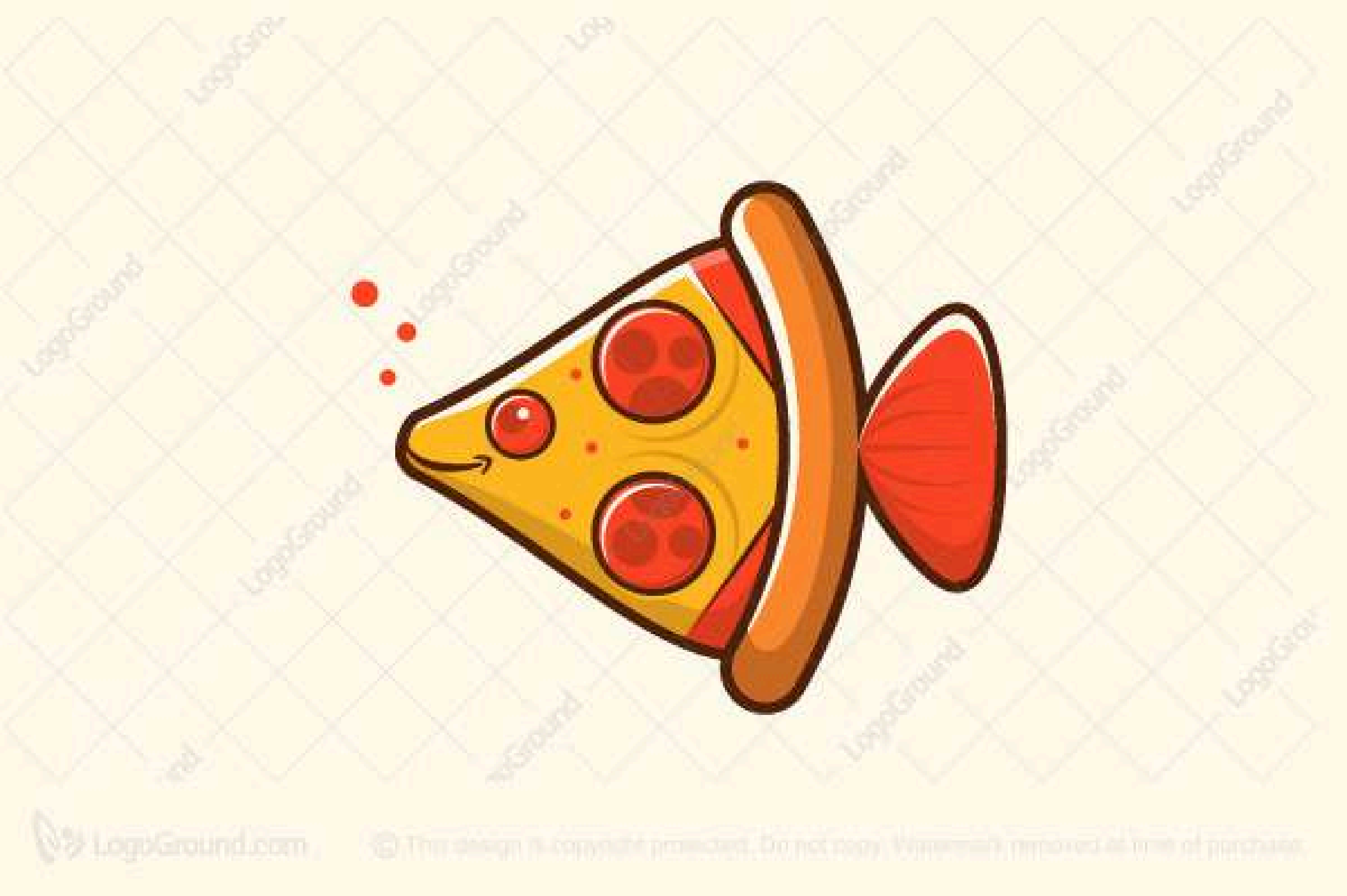


```
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from  order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.orders_id
group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

Querry 13 : Determine the top 3 most ordered piza types based on revenue for each pizza category.



```
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) * pizza.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;
```



Pizza sales Presentation

THANK YOU FOR ATTENTION

See You Next