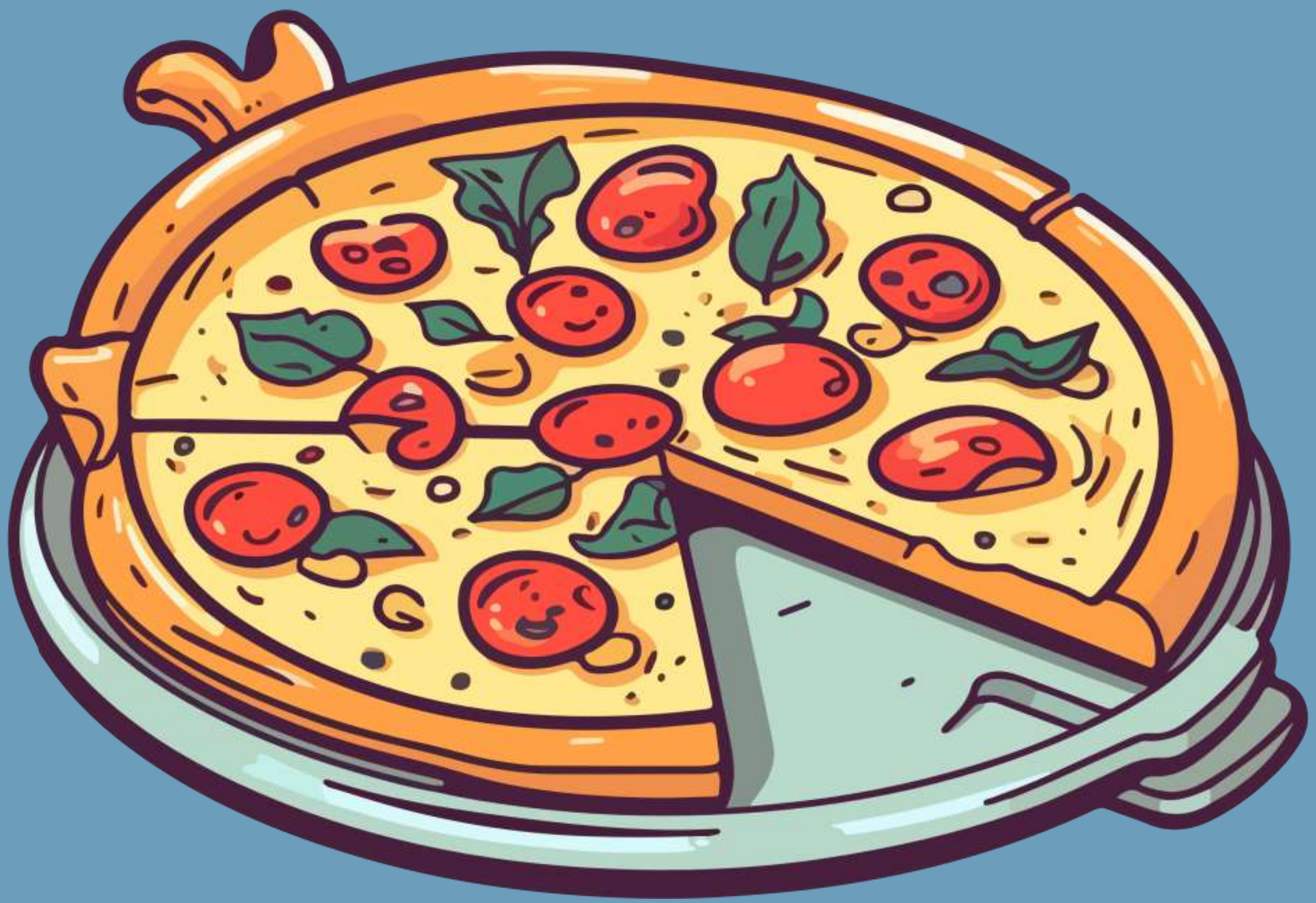
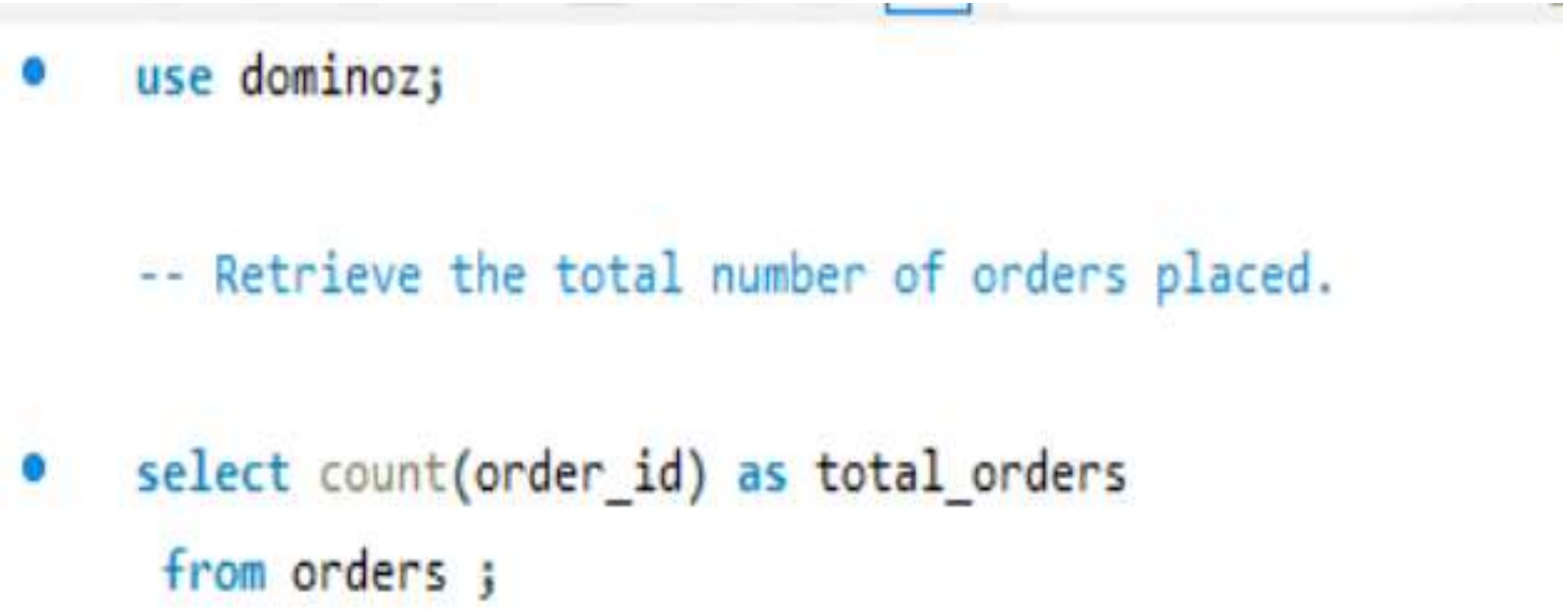


PIZZA SALES REPORT

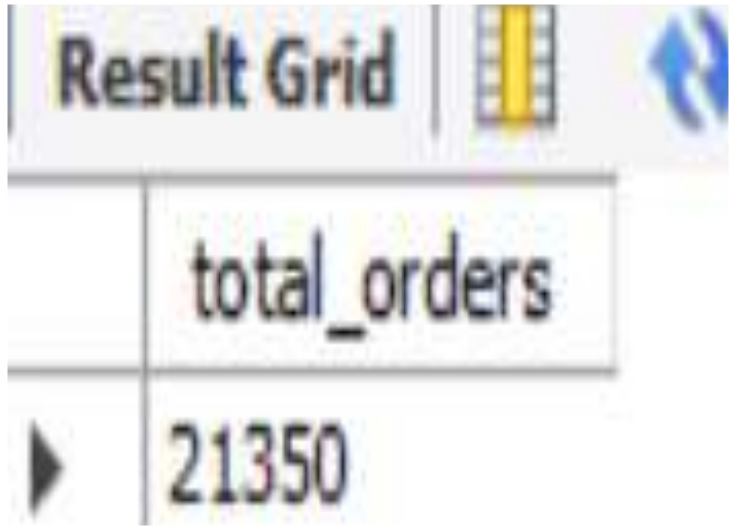
Dataset

- Orders
- order_details
- pizzas
- pizza_type

Retrieve the total number of orders placed.

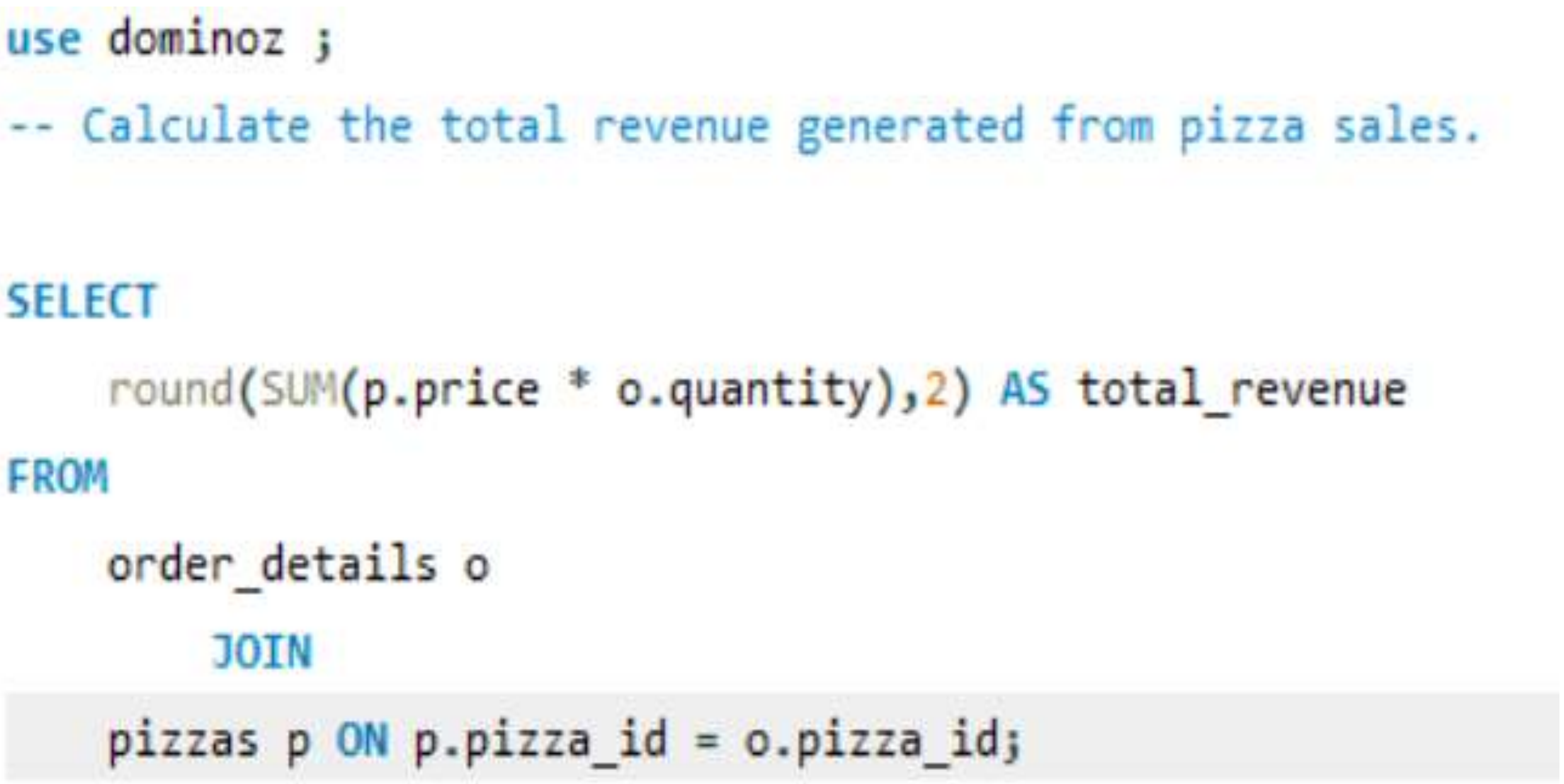


```
use dominoz;

-- Retrieve the total number of orders placed.

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

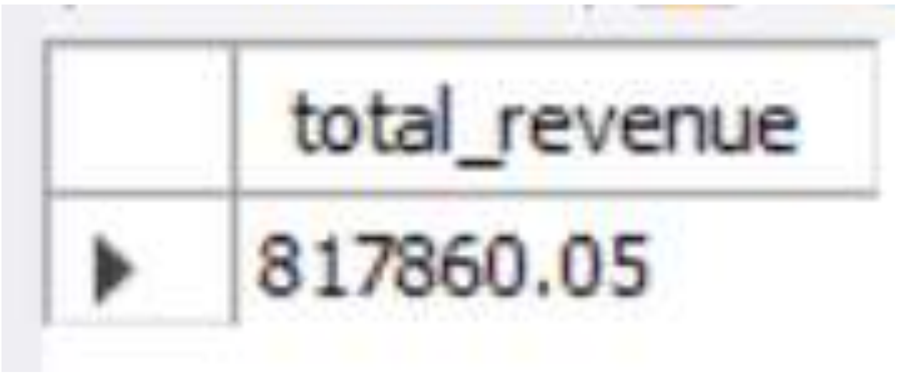


```
total_orders
21350
```

Calculate the total revenue generated from pizza sales.

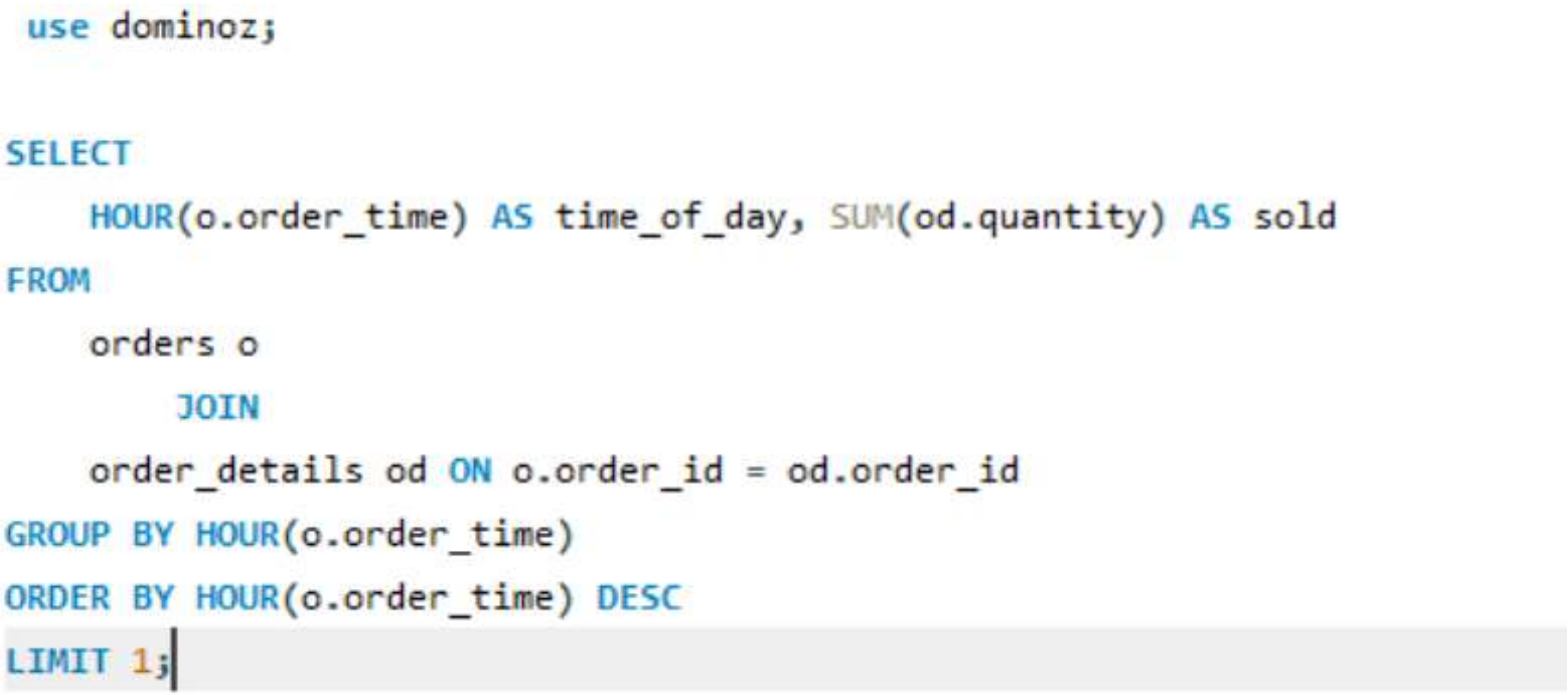


```
use dominoz ;
-- Calculate the total revenue generated from pizza sales.

SELECT
    round(SUM(p.price * o.quantity),2) AS total_revenue
FROM
    order_details o
        JOIN
    pizzas p ON p.pizza_id = o.pizza_id;
```

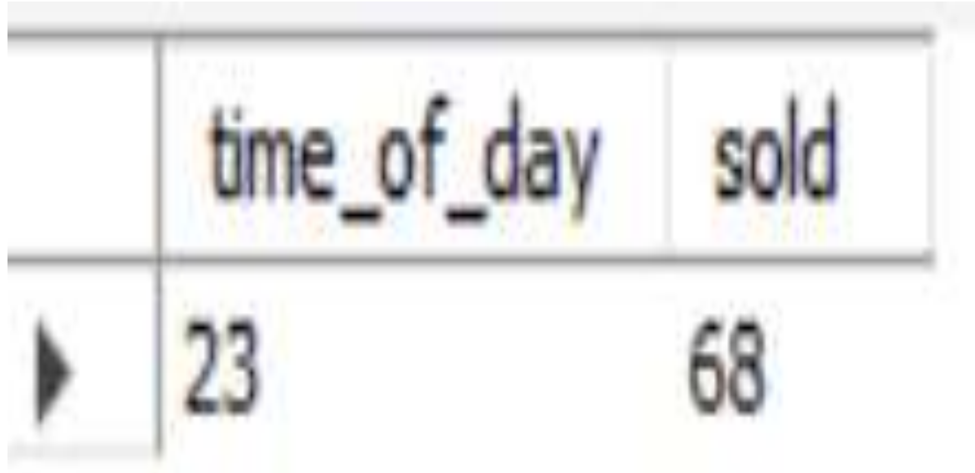


	total_revenue
▶	817860.05

Peak time of the day

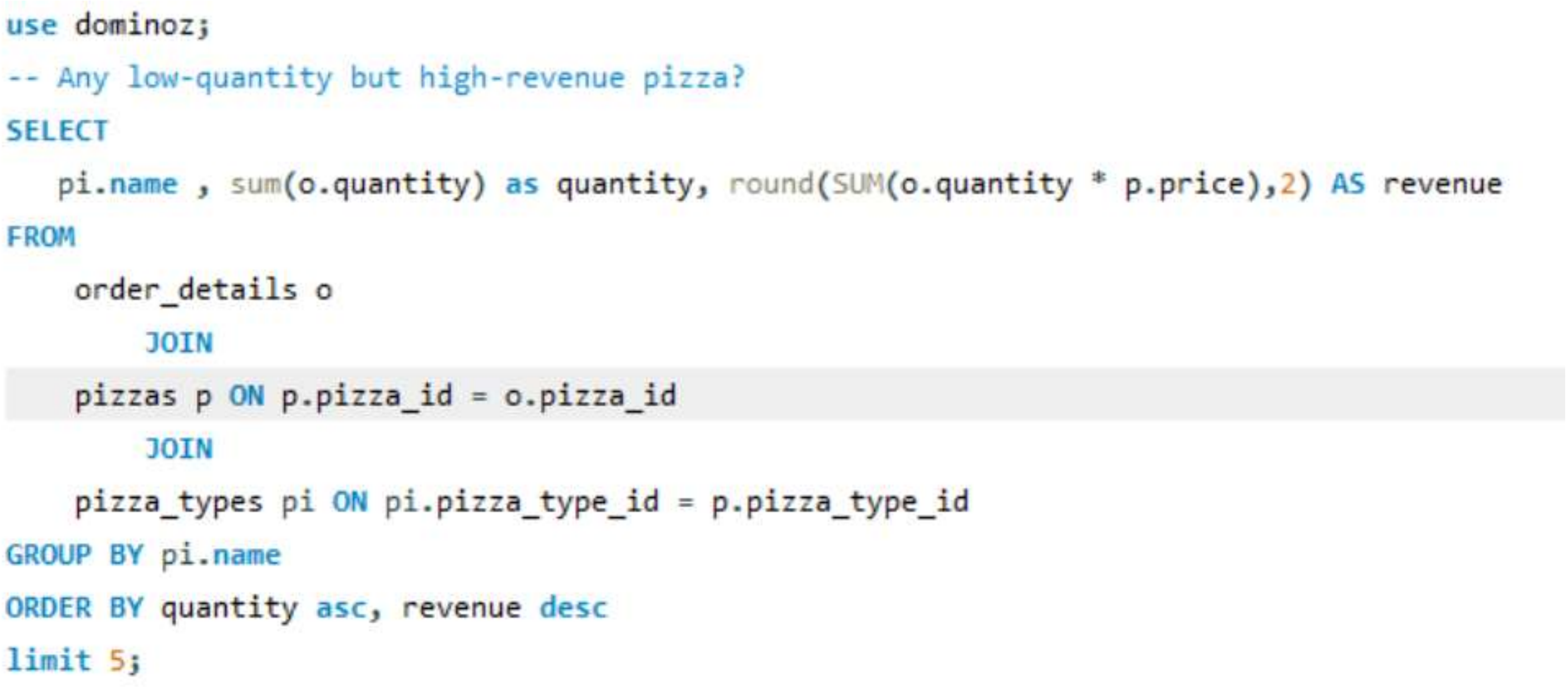


```
 use dominoz;

SELECT
    HOUR(o.order_time) AS time_of_day, SUM(od.quantity) AS sold
FROM
    orders o
        JOIN
    order_details od ON o.order_id = od.order_id
GROUP BY HOUR(o.order_time)
ORDER BY HOUR(o.order_time) DESC
LIMIT 1;
```

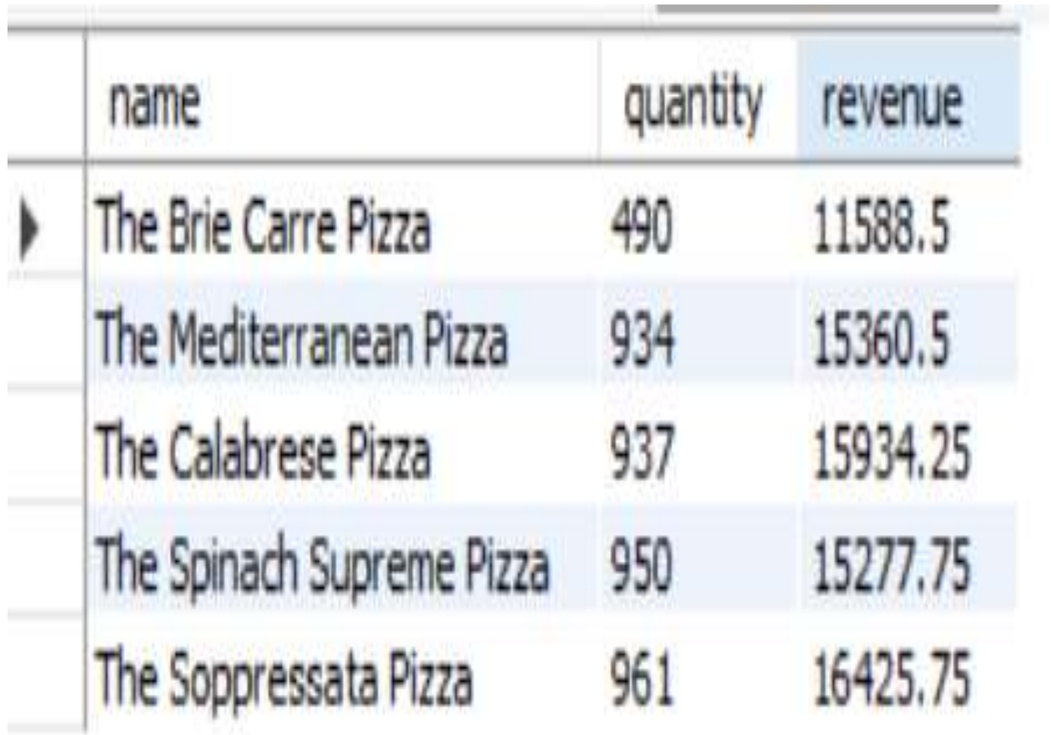


time_of_day	sold
23	68

Any low-quantity but high-revenue pizza

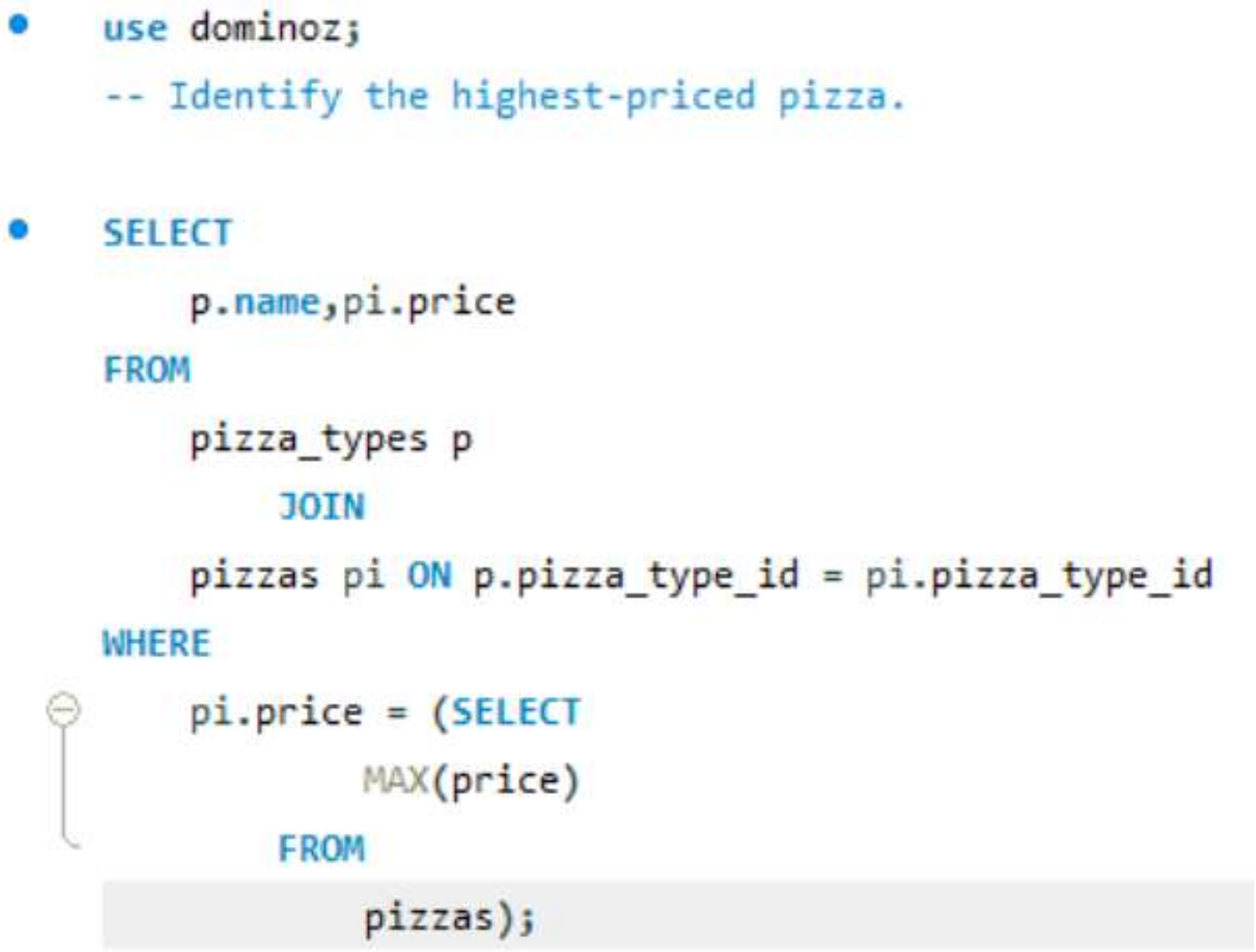


```
use dominoz;
-- Any low-quantity but high-revenue pizza?
SELECT
   pi.name , sum(o.quantity) as quantity, round(SUM(o.quantity * p.price),2) AS revenue
FROM
    order_details o
        JOIN
    pizzas p ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pi ON pi.pizza_type_id = p.pizza_type_id
GROUP BY pi.name
ORDER BY quantity asc, revenue desc
limit 5;
```

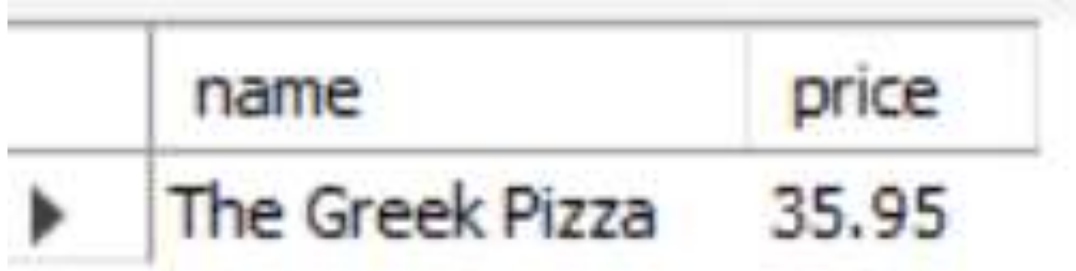


name	quantity	revenue
The Brie Carre Pizza	490	11588.5
The Mediterranean Pizza	934	15360.5
The Calabrese Pizza	937	15934.25
The Spinach Supreme Pizza	950	15277.75
The Soppressata Pizza	961	16425.75

Identify the highest-priced pizza.

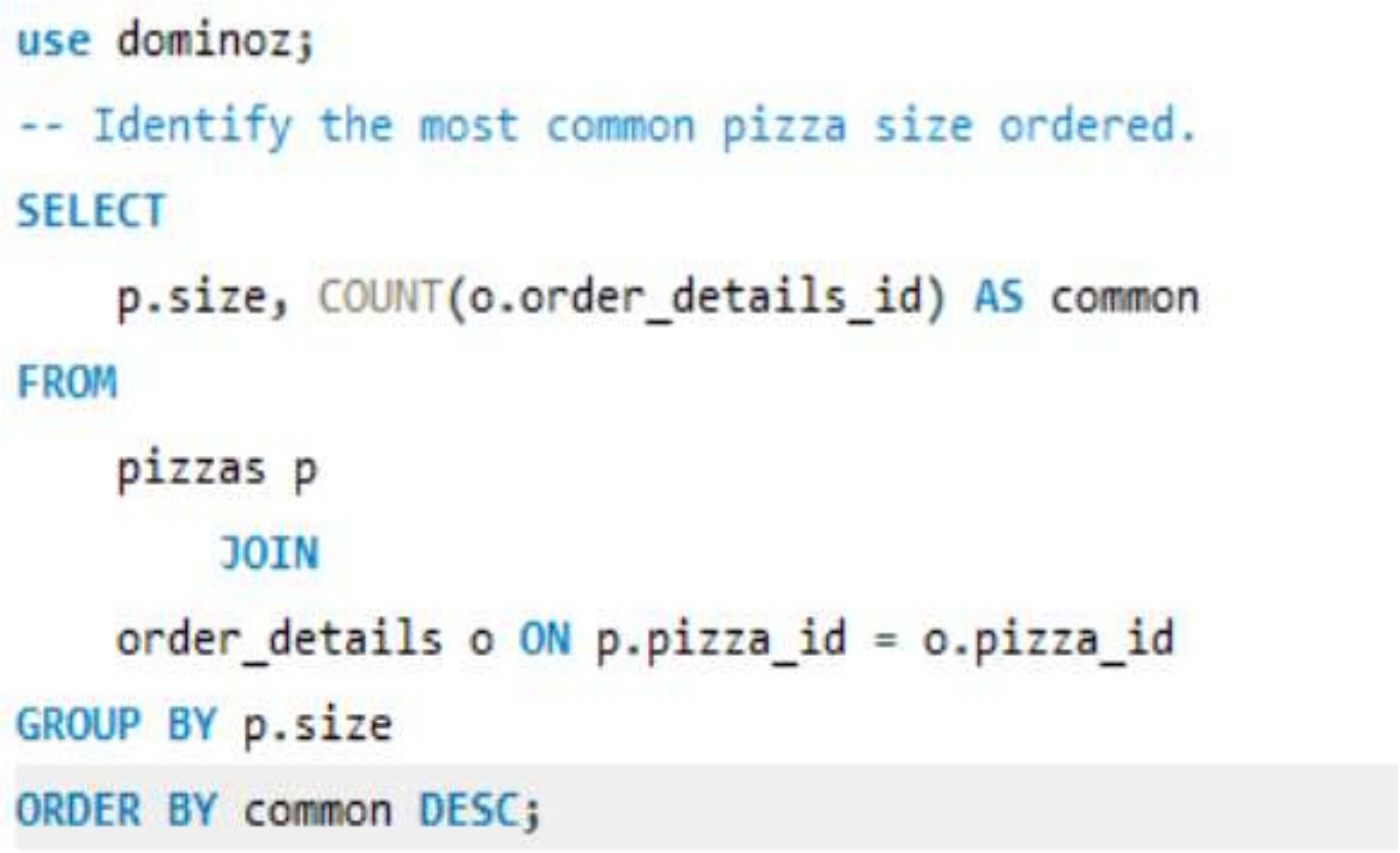


```
use dominoz;
-- Identify the highest-priced pizza.

SELECT
    p.name,pi.price
FROM
    pizza_types p
        JOIN
    pizzas pi ON p.pizza_type_id = pi.pizza_type_id
WHERE
    pi.price = (SELECT
            MAX(price)
        FROM
            pizzas);
```

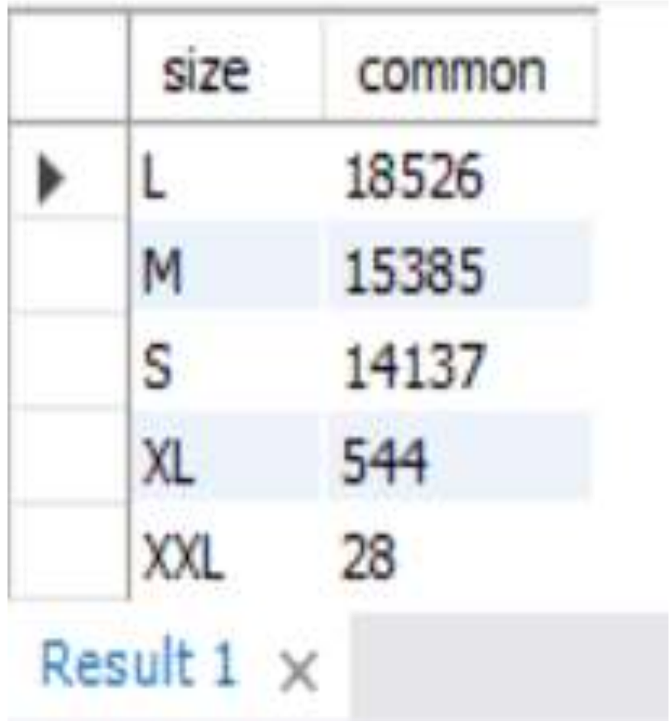


name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

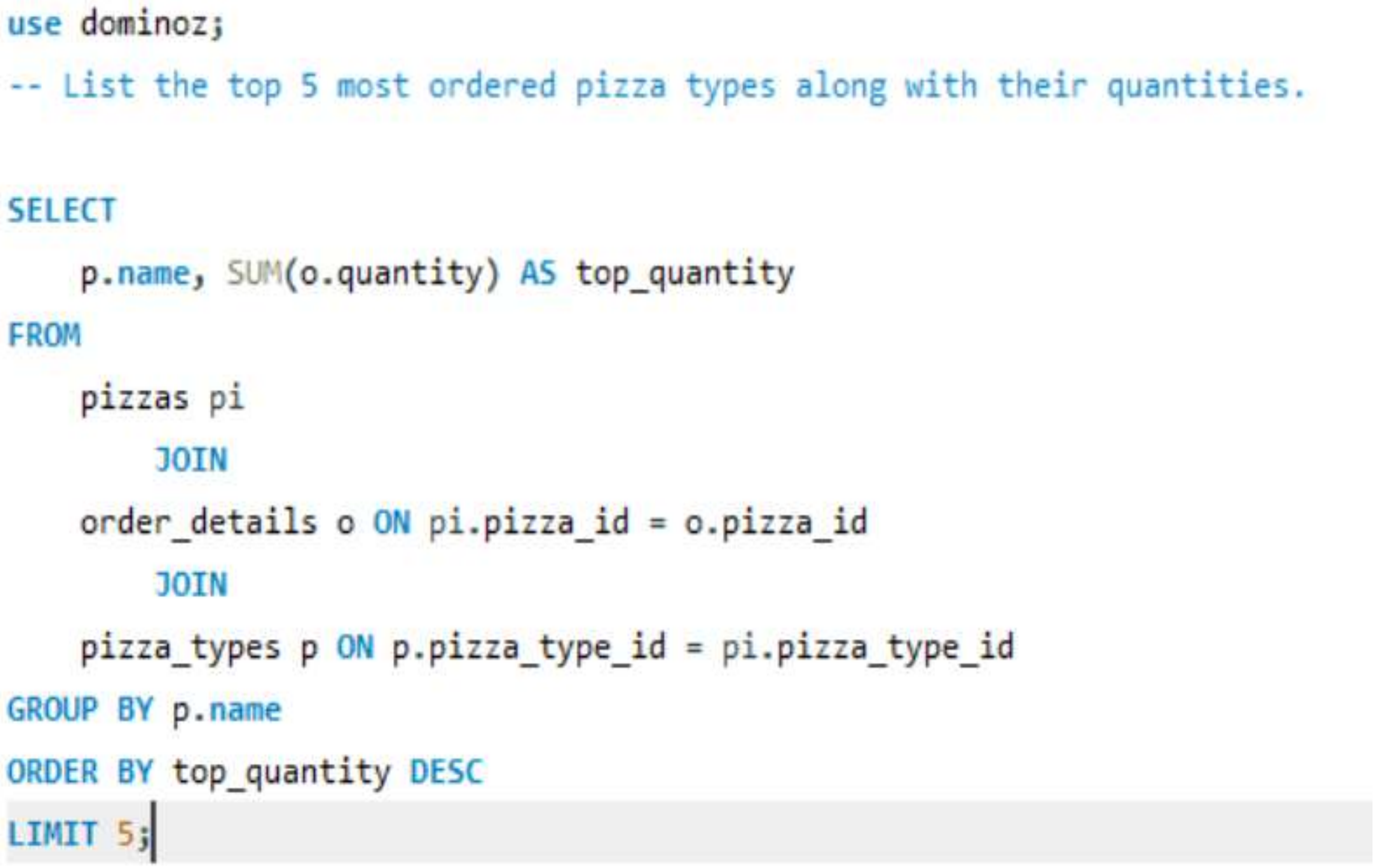


```
use dominoz;
-- Identify the most common pizza size ordered.
SELECT
    p.size, COUNT(o.order_details_id) AS common
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
GROUP BY p.size
ORDER BY common DESC;
```

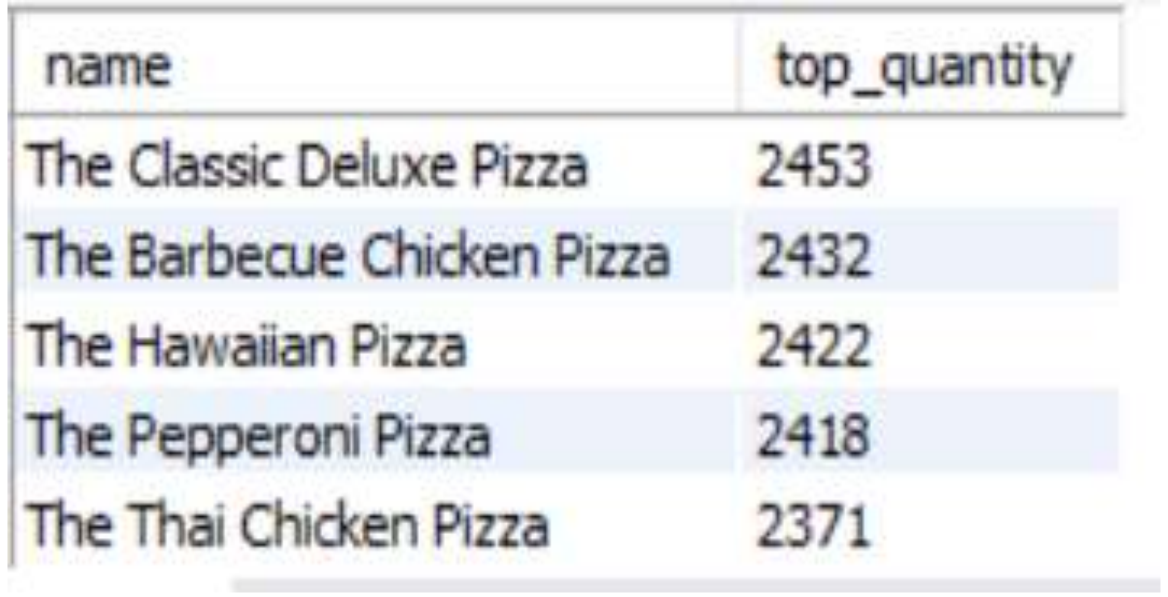


size	common
L	18526
M	15385
S	14137
XL	544
XXL	28

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

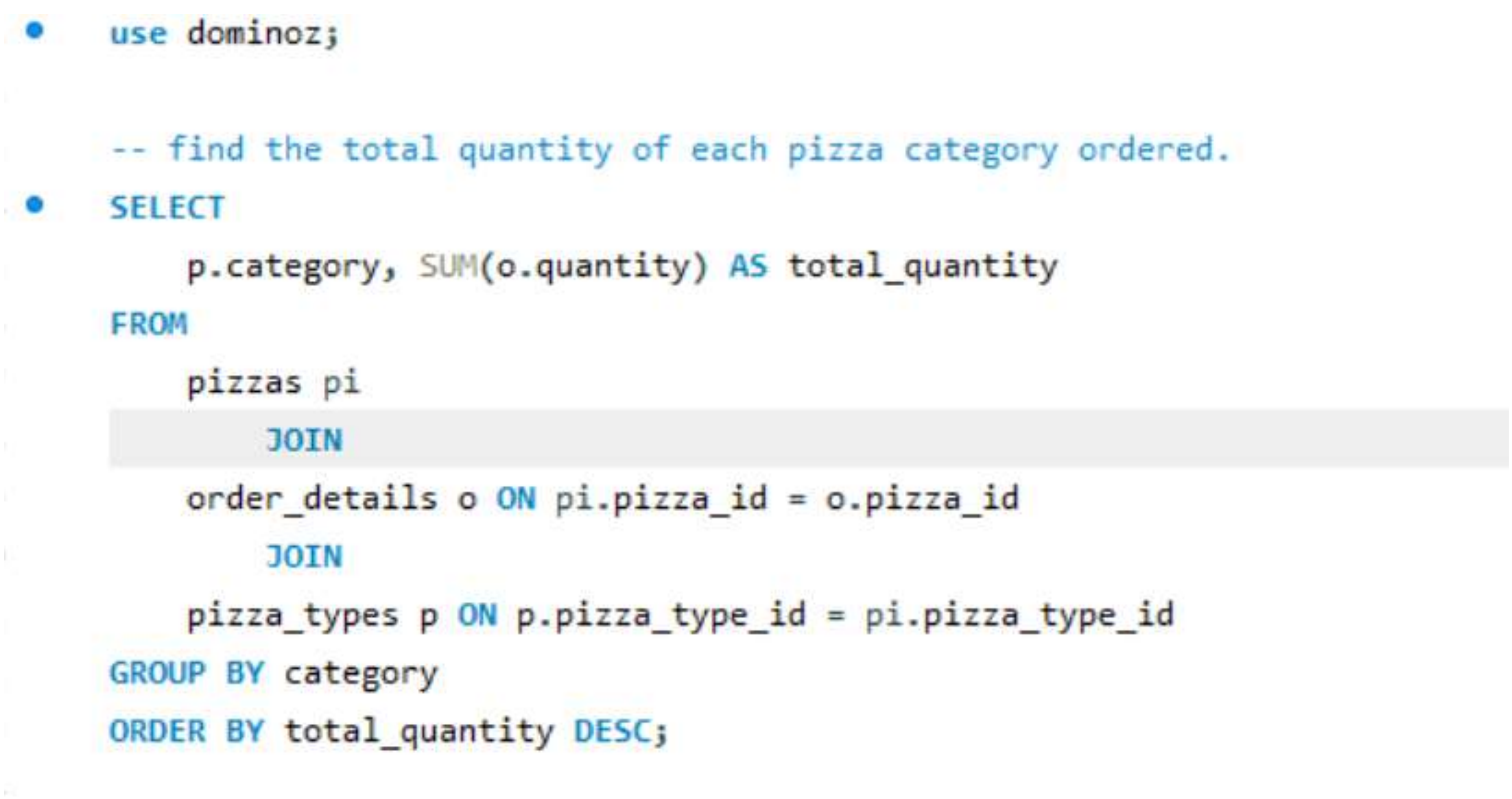


```
use dominoz;
-- List the top 5 most ordered pizza types along with their quantities.

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

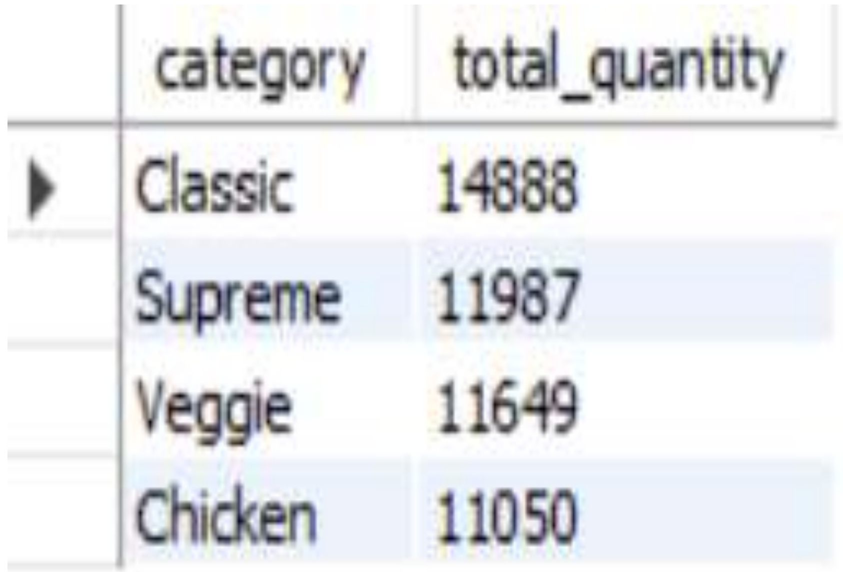


name	top_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

find the total quantity of each pizza category ordered.

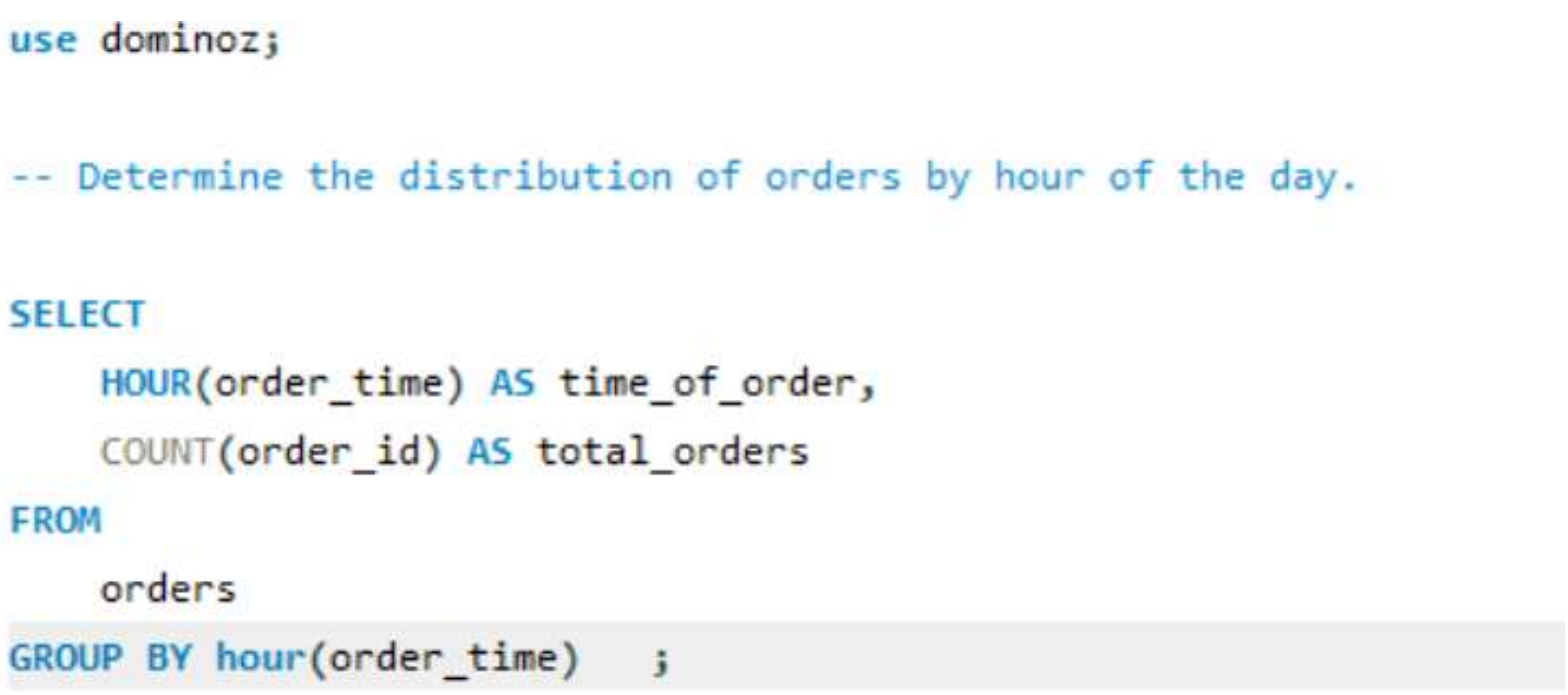


```
use dominoz;

-- find the total quantity of each pizza category ordered.
SELECT
    p.category, SUM(o.quantity) AS total_quantity
FROM
    pizzas pi
        JOIN
    order_details o ON pi.pizza_id = o.pizza_id
        JOIN
    pizza_types p ON p.pizza_type_id = pi.pizza_type_id
GROUP BY category
ORDER BY total_quantity DESC;
```

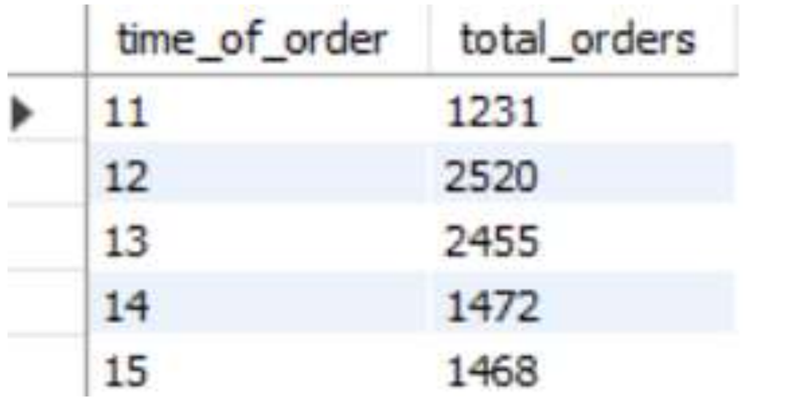


category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

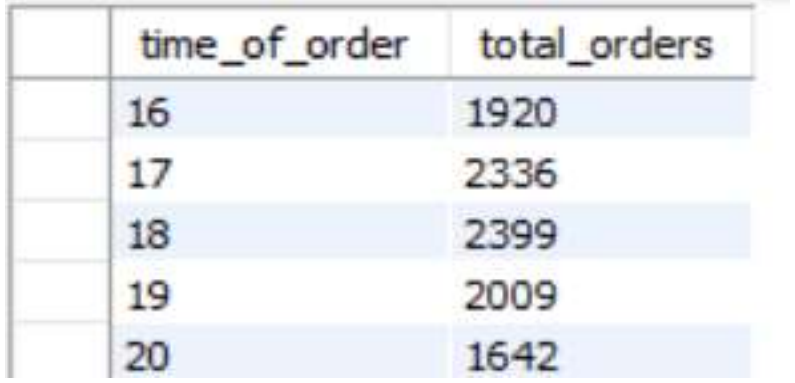
Determine the distribution of orders by hour of the day.



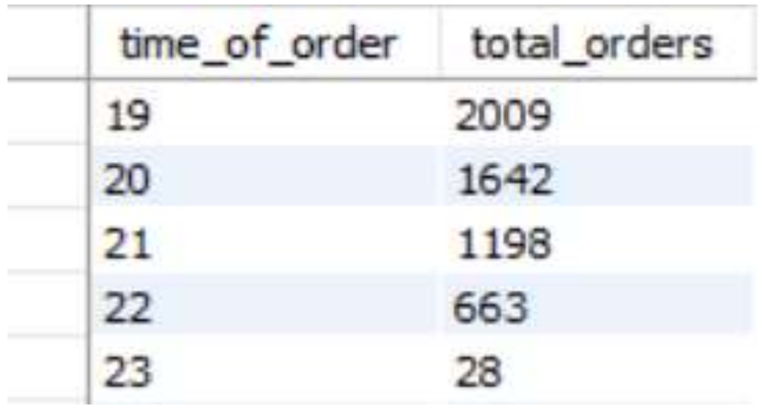
```
use dominoz;

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

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



time_of_order	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468

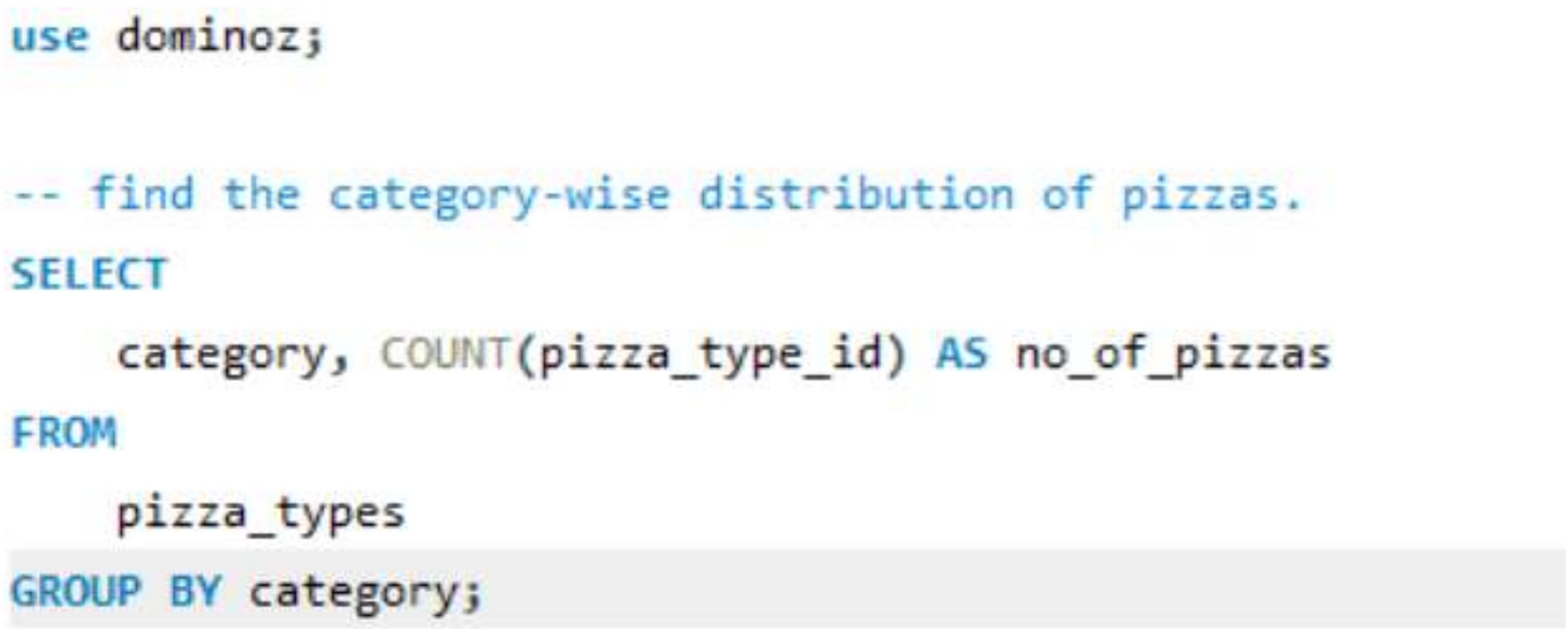


time_of_order	total_orders
16	1920
17	2336
18	2399
19	2009
20	1642

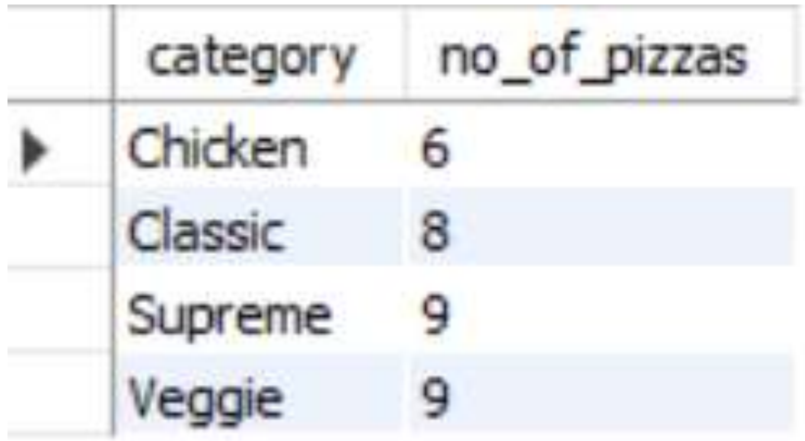


time_of_order	total_orders
19	2009
20	1642
21	1198
22	663
23	28

find the category-wise distribution of pizzas.

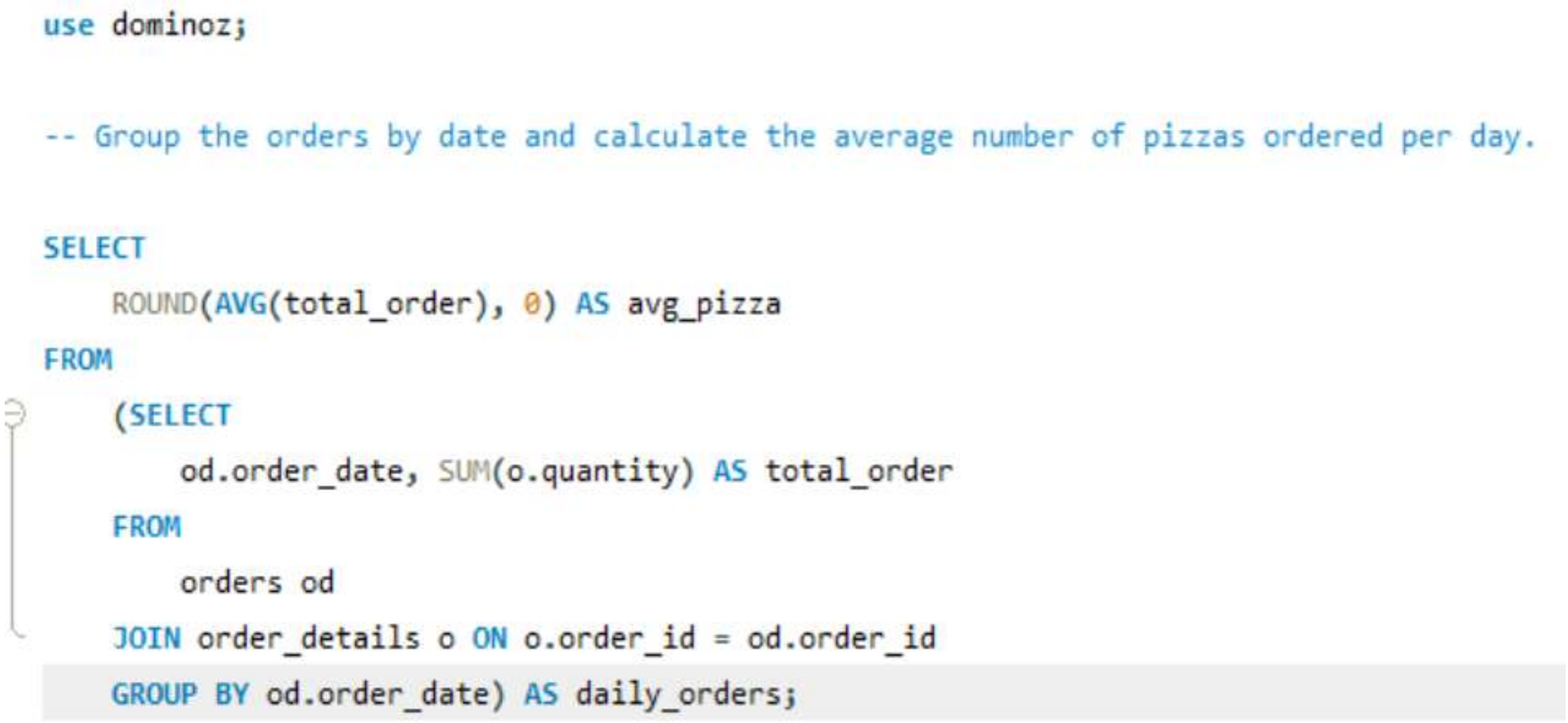


```
use dominoz;

-- find the category-wise distribution of pizzas.
SELECT
    category, COUNT(pizza_type_id) AS no_of_pizzas
FROM
    pizza_types
GROUP BY category;
```

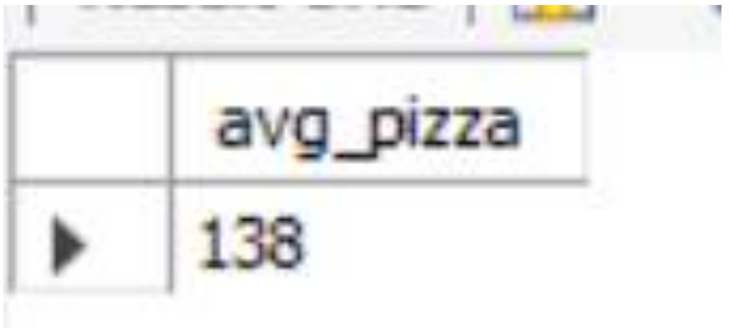


category	no_of_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

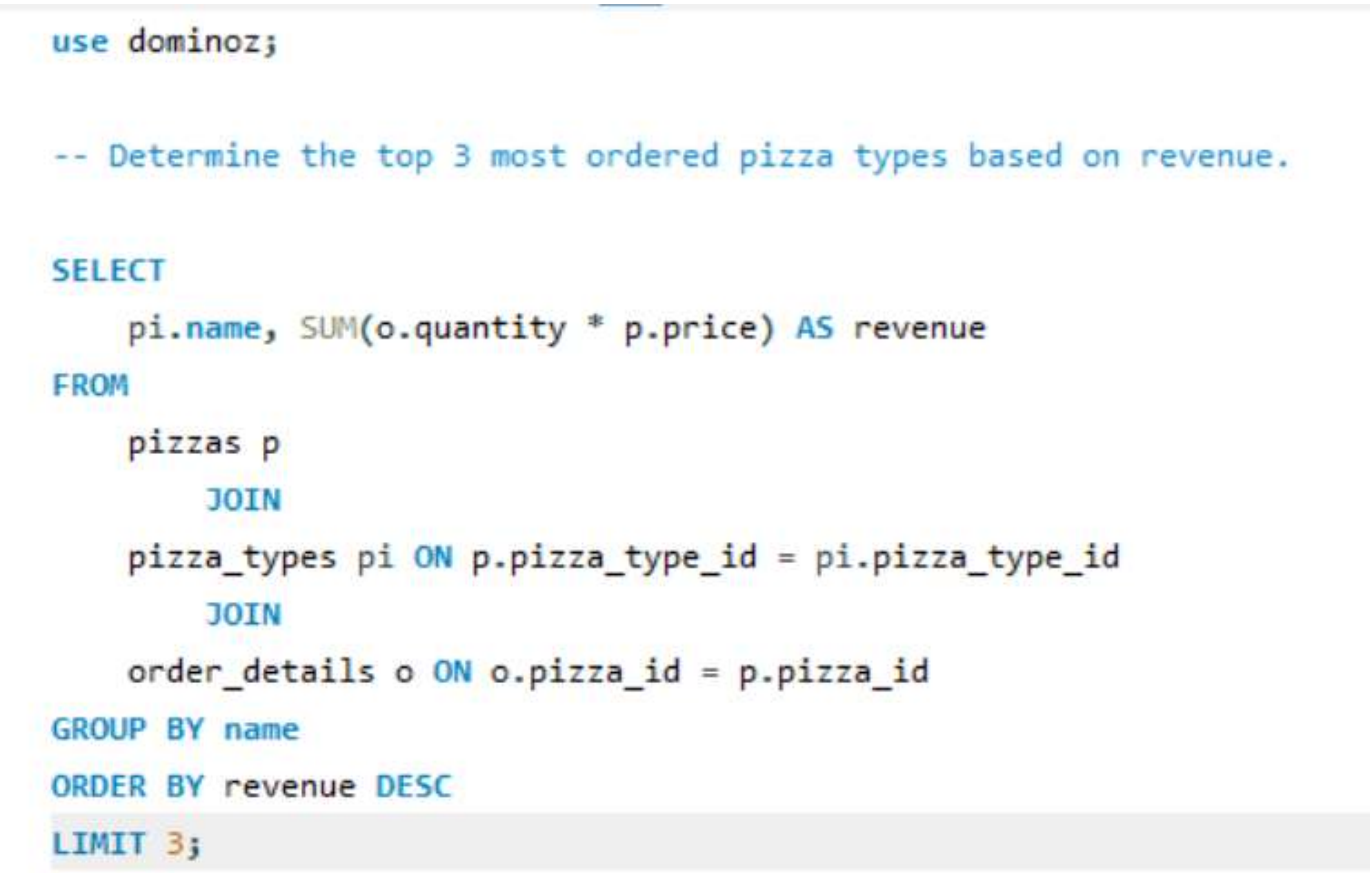


```
use dominoz;

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

SELECT
    ROUND(AVG(total_order), 0) AS avg_pizza
FROM
    (SELECT
        od.order_date, SUM(o.quantity) AS total_order
    FROM
        orders od
    JOIN order_details o ON o.order_id = od.order_id
    GROUP BY od.order_date) AS daily_orders;
```

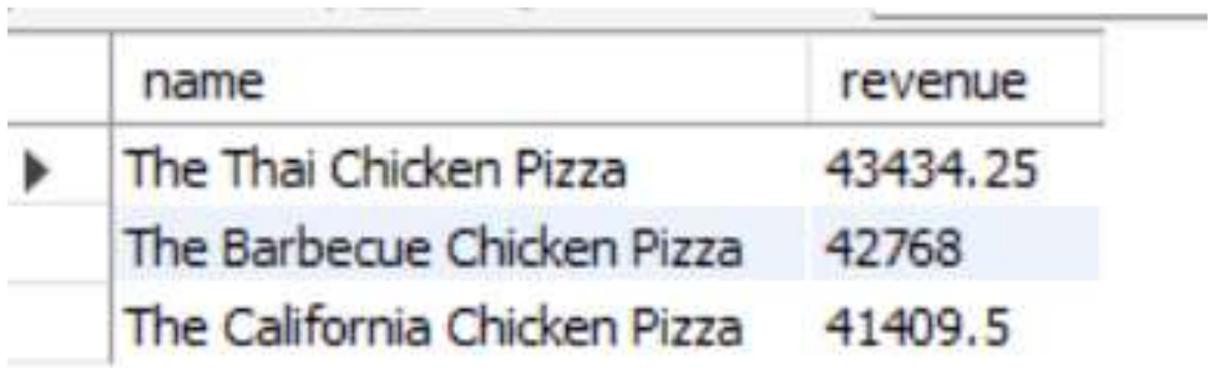


	avg_pizza
▶	138

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

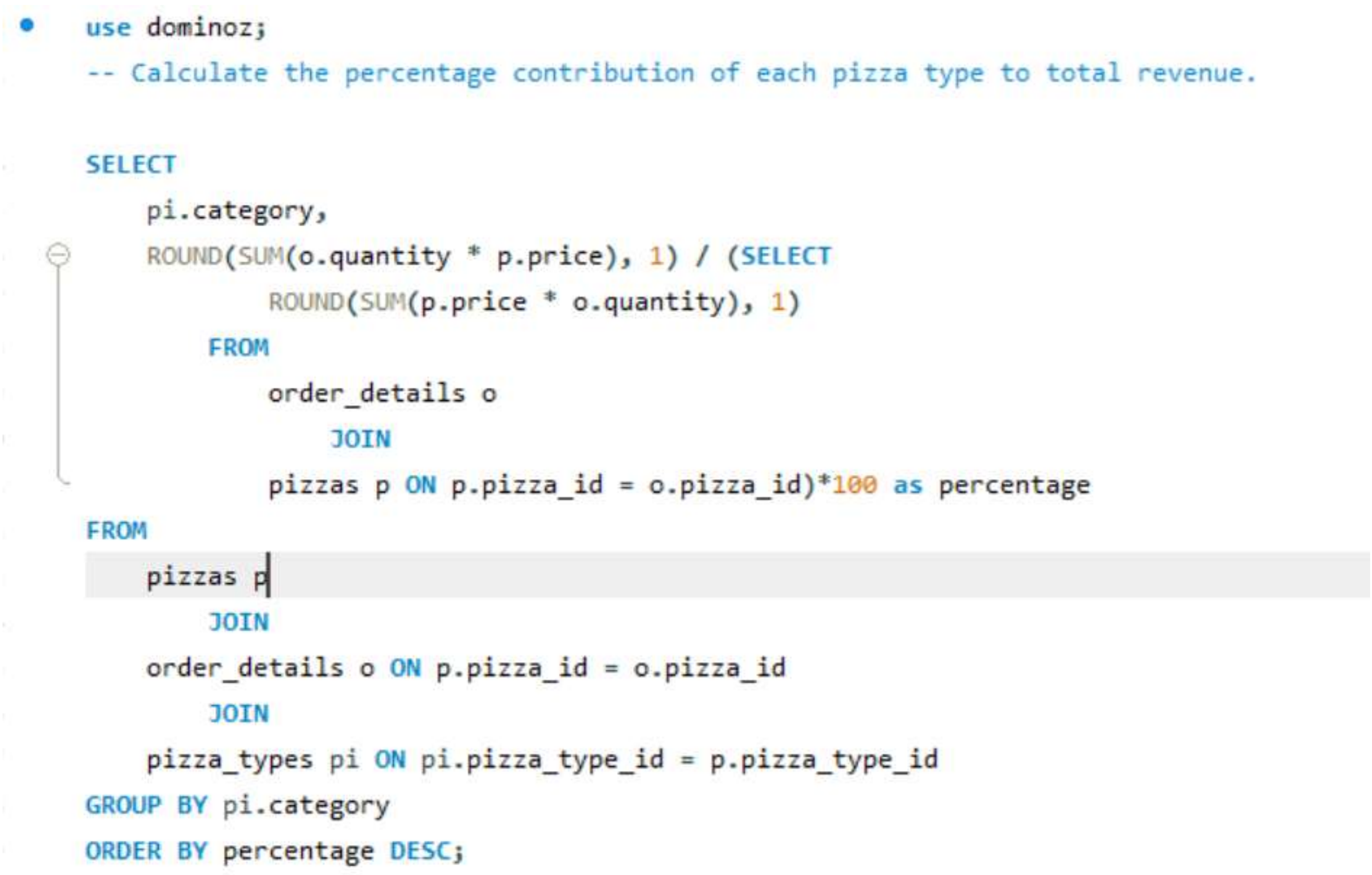


```sql
use dominoz;

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

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

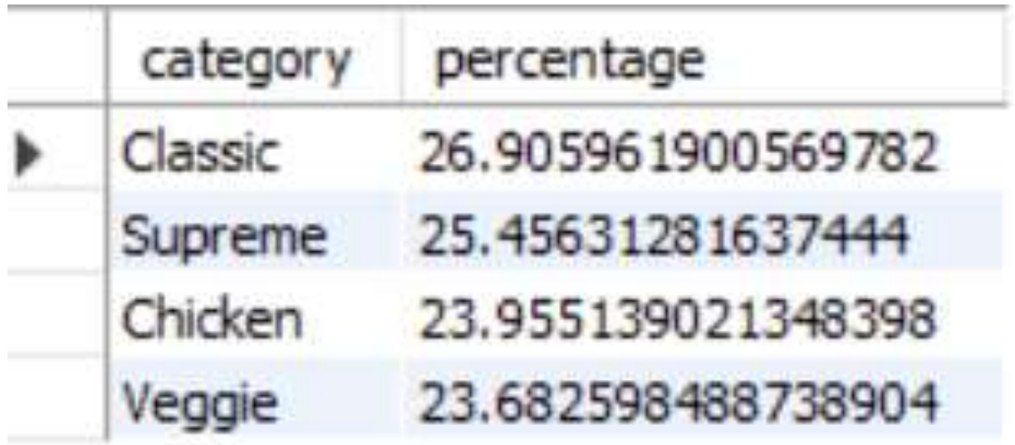


name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

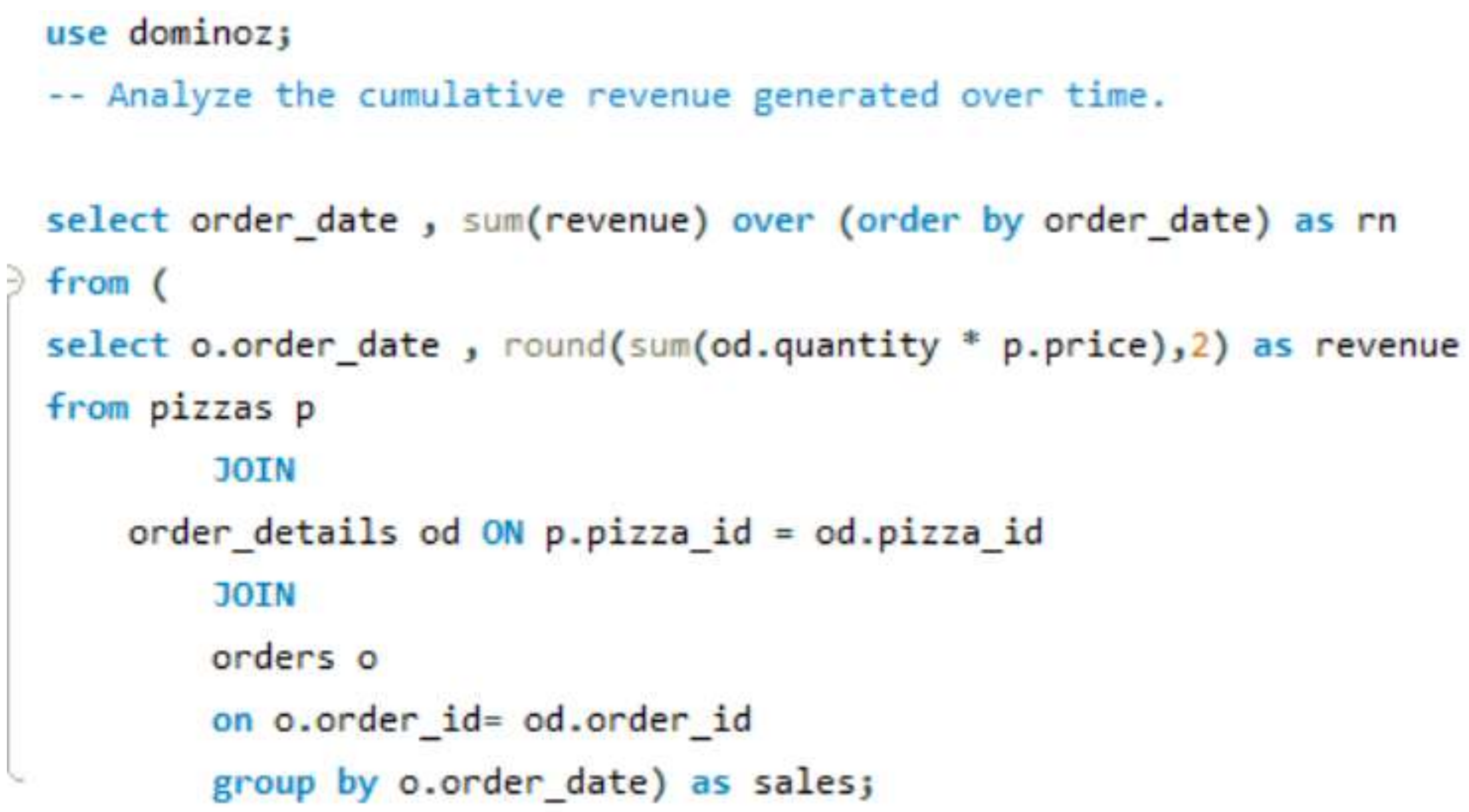


```
use dominoz;
-- Calculate the percentage contribution of each pizza type to total revenue.

SELECT
    pi.category,
    ROUND(SUM(o.quantity * p.price), 1) / (SELECT
            ROUND(SUM(p.price * o.quantity), 1)
        FROM
            order_details o
                JOIN
            pizzas p ON p.pizza_id = o.pizza_id)*100 as percentage
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pi ON pi.pizza_type_id = p.pizza_type_id
GROUP BY pi.category
ORDER BY percentage DESC;
```

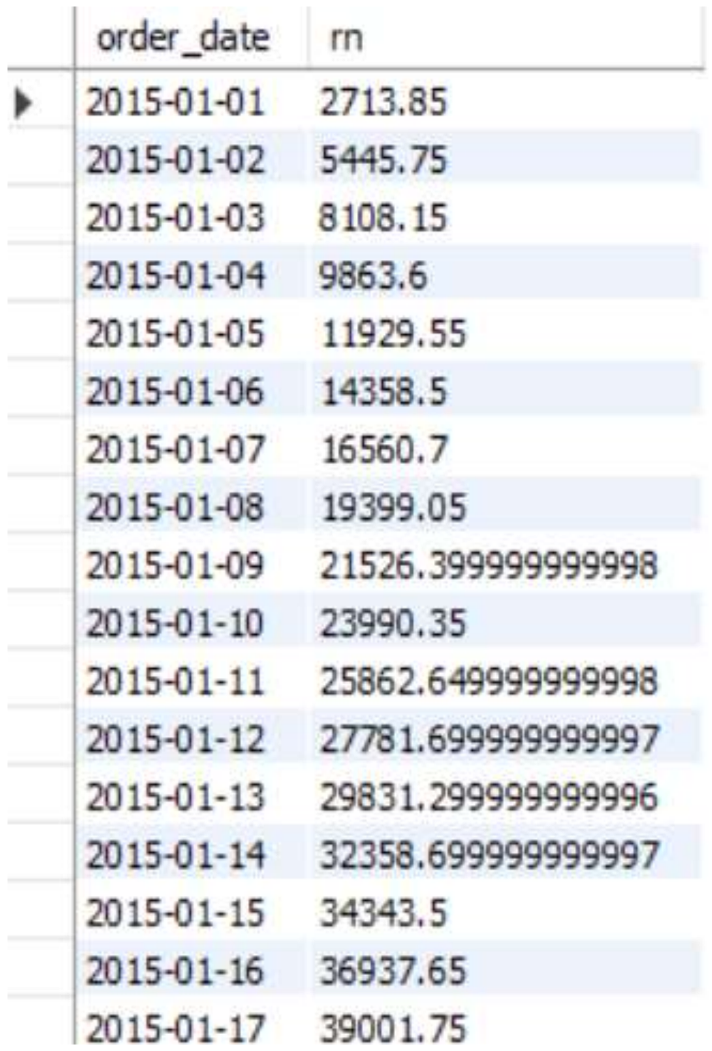


category	percentage
Classic	26.905961900569782
Supreme	25.45631281637444
Chicken	23.955139021348398
Veggie	23.682598488738904

Analyze the cumulative revenue generated over time.

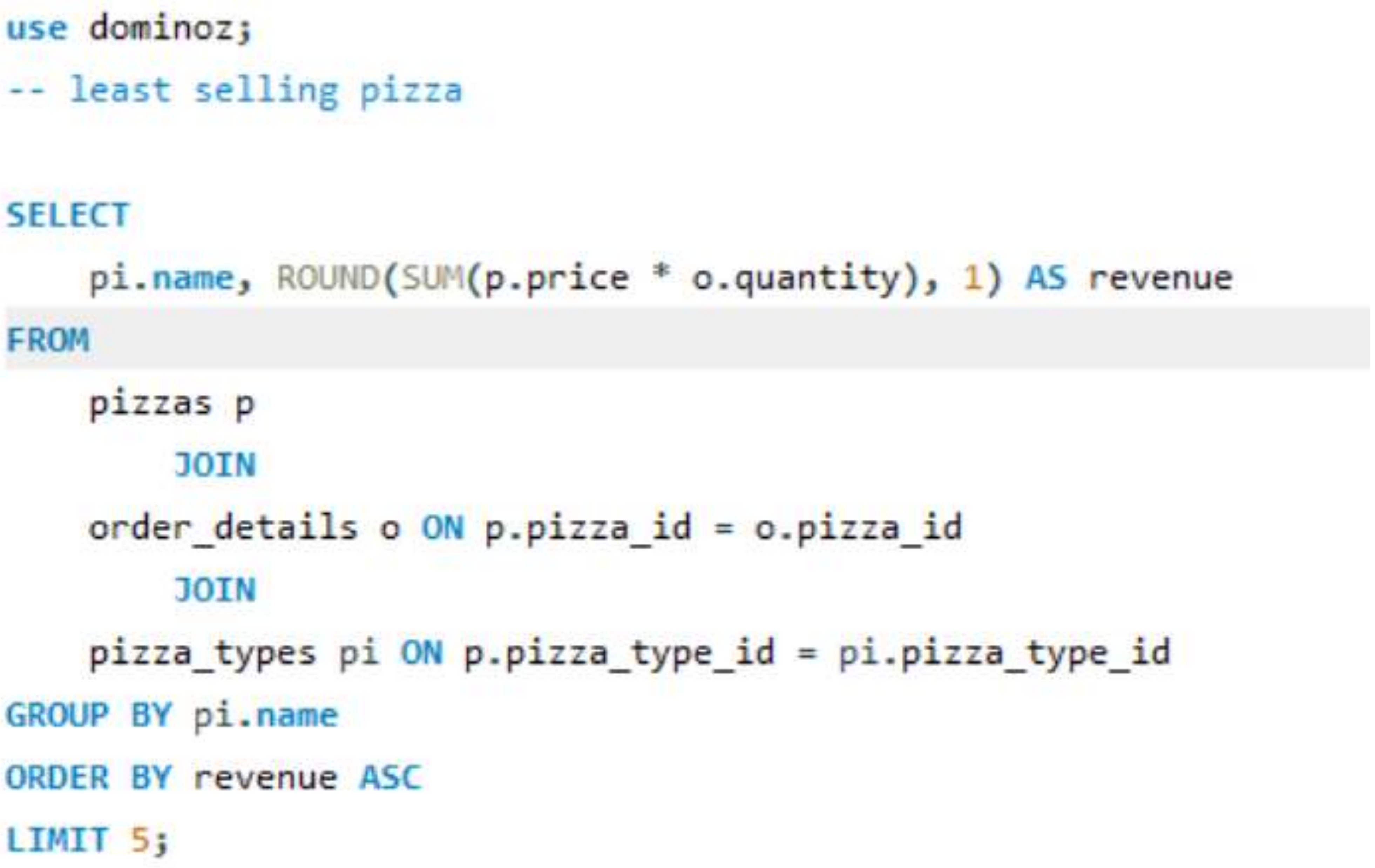


```
use dominoz;
-- Analyze the cumulative revenue generated over time.

select order_date , sum(revenue) over (order by order_date) as rn
from (
select o.order_date , round(sum(od.quantity * p.price),2) as revenue
from pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
        JOIN
        orders o
        on o.order_id= od.order_id
        group by o.order_date) as sales;
```

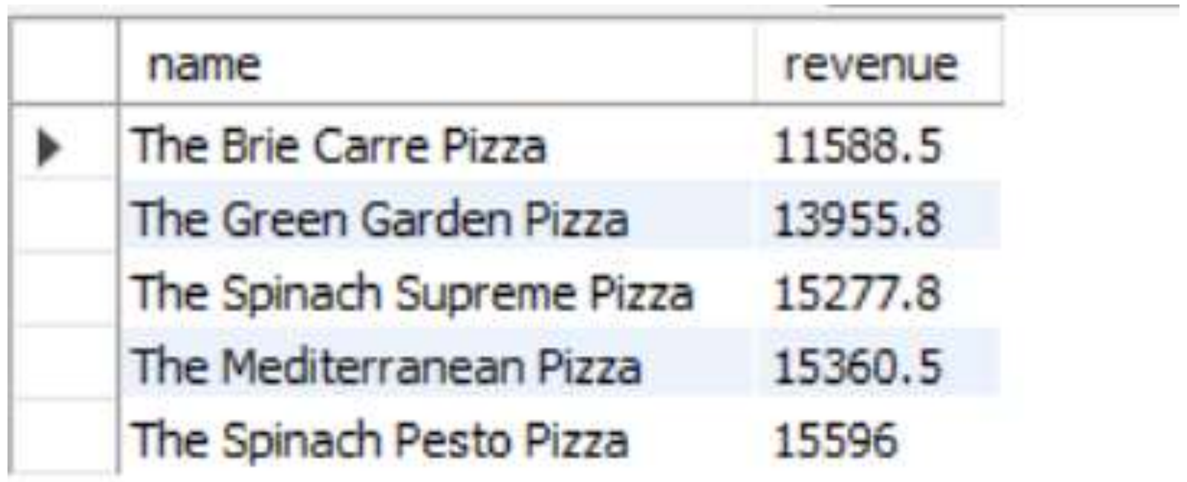


order_date	rn
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75

least selling pizza

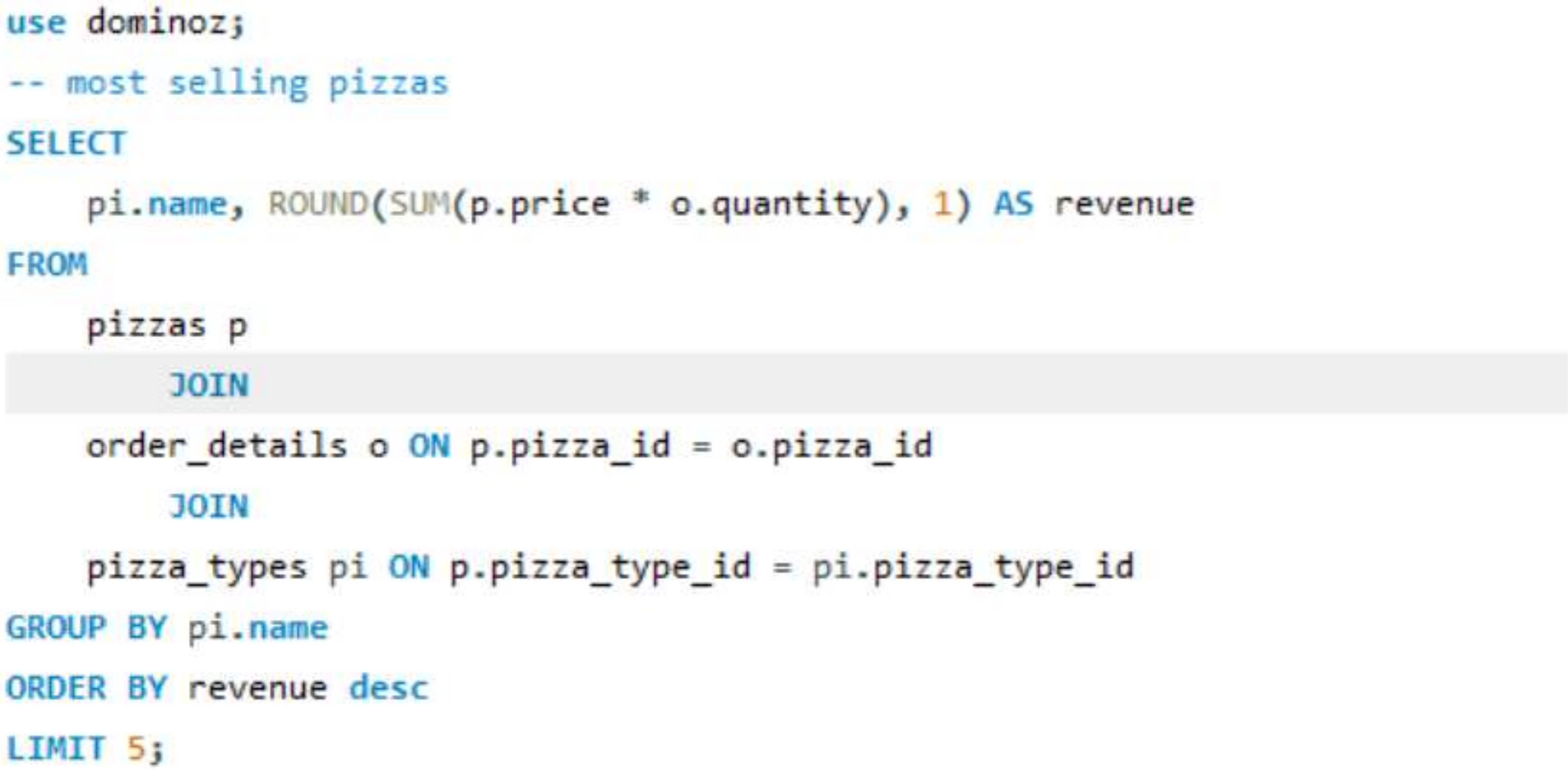


```
use dominoz;
-- least selling pizza

SELECT
    pi.name, ROUND(SUM(p.price * o.quantity), 1) AS revenue
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pi ON p.pizza_type_id = pi.pizza_type_id
GROUP BY pi.name
ORDER BY revenue ASC
LIMIT 5;
```

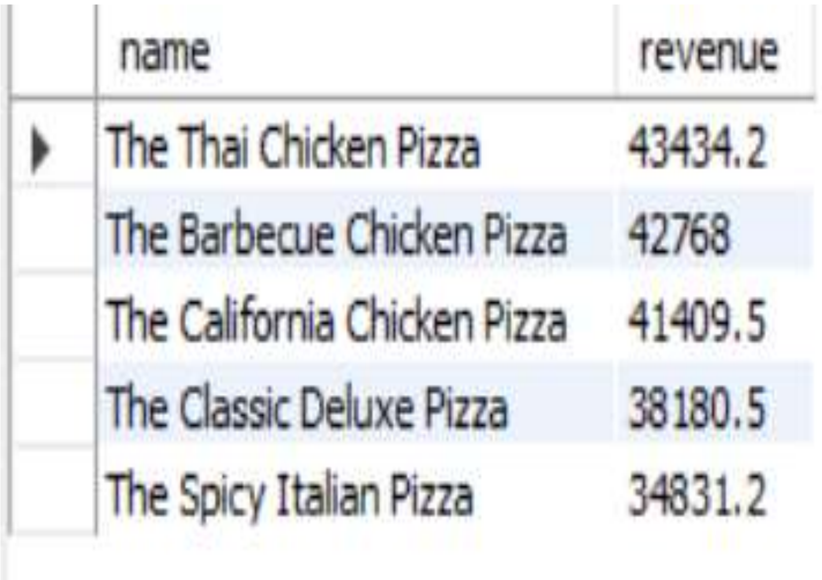


	name	revenue
▶	The Brie Carre Pizza	11588.5
	The Green Garden Pizza	13955.8
	The Spinach Supreme Pizza	15277.8
	The Mediterranean Pizza	15360.5
	The Spinach Pesto Pizza	15596

most selling pizzas



```
use dominoz;
-- most selling pizzas
SELECT
    pi.name, ROUND(SUM(p.price * o.quantity), 1) AS revenue
FROM
    pizzas p
        JOIN
    order_details o ON p.pizza_id = o.pizza_id
        JOIN
    pizza_types pi ON p.pizza_type_id = pi.pizza_type_id
GROUP BY pi.name
ORDER BY revenue desc
LIMIT 5;
```



	name	revenue
▶	The Thai Chicken Pizza	43434.2
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Spicy Italian Pizza	34831.2