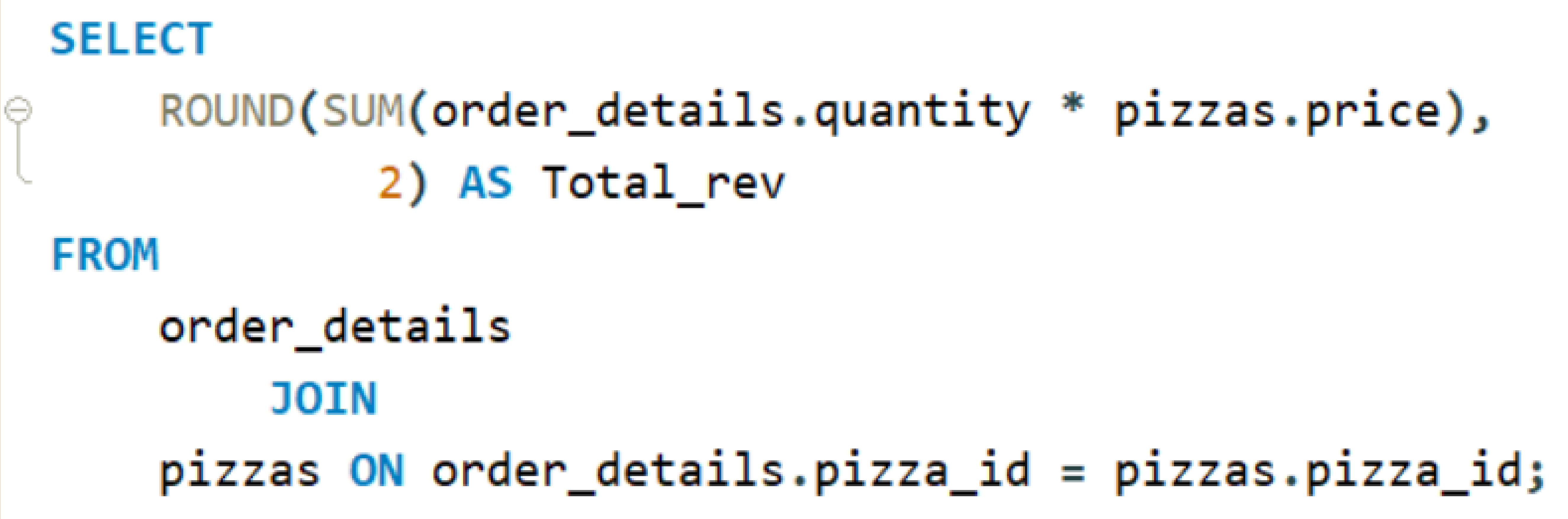
PIZZA SALES

INTRODUCTION

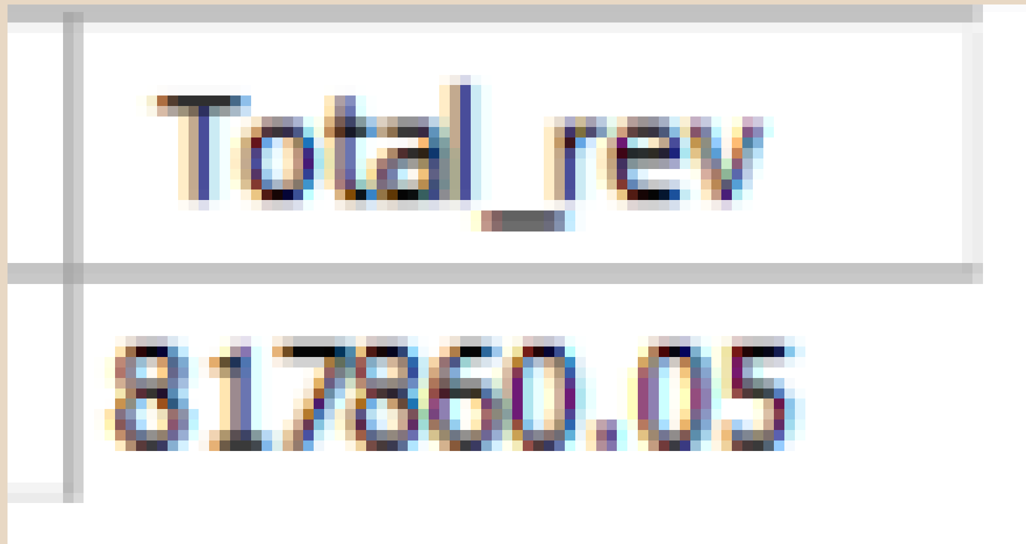
Hello, I am Tarushika Sharma, on this project i have utilized SQL queries to solve the questions that was related to pizza sales.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



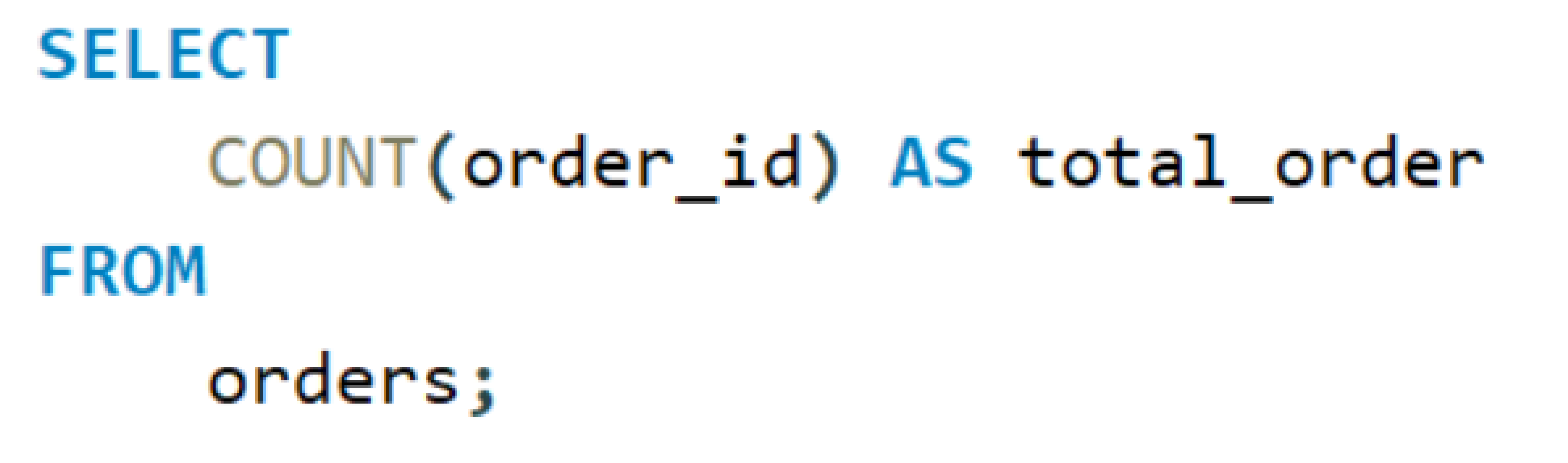
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_rev
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

OUTPUT-



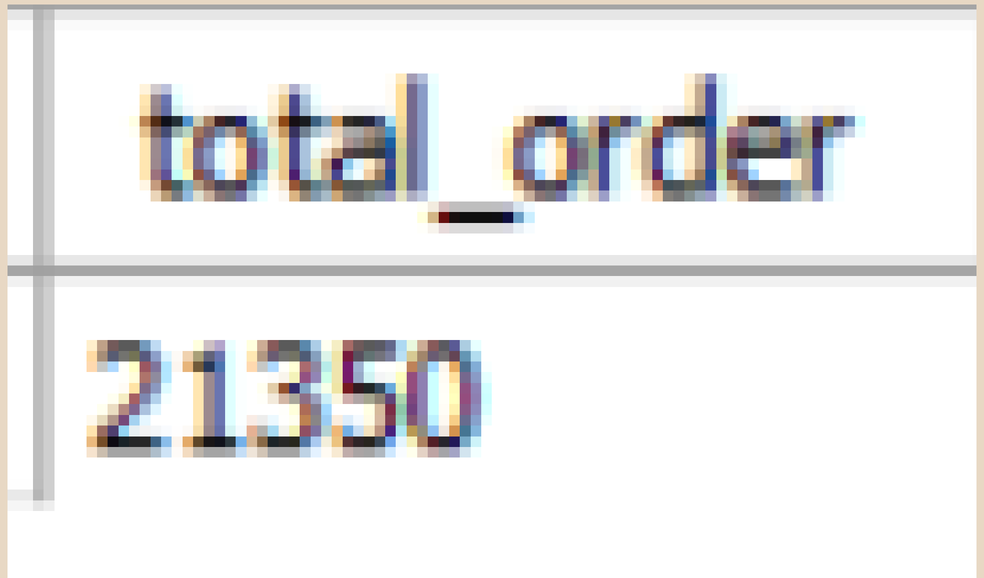
```
Total_rev
817860.05
```

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



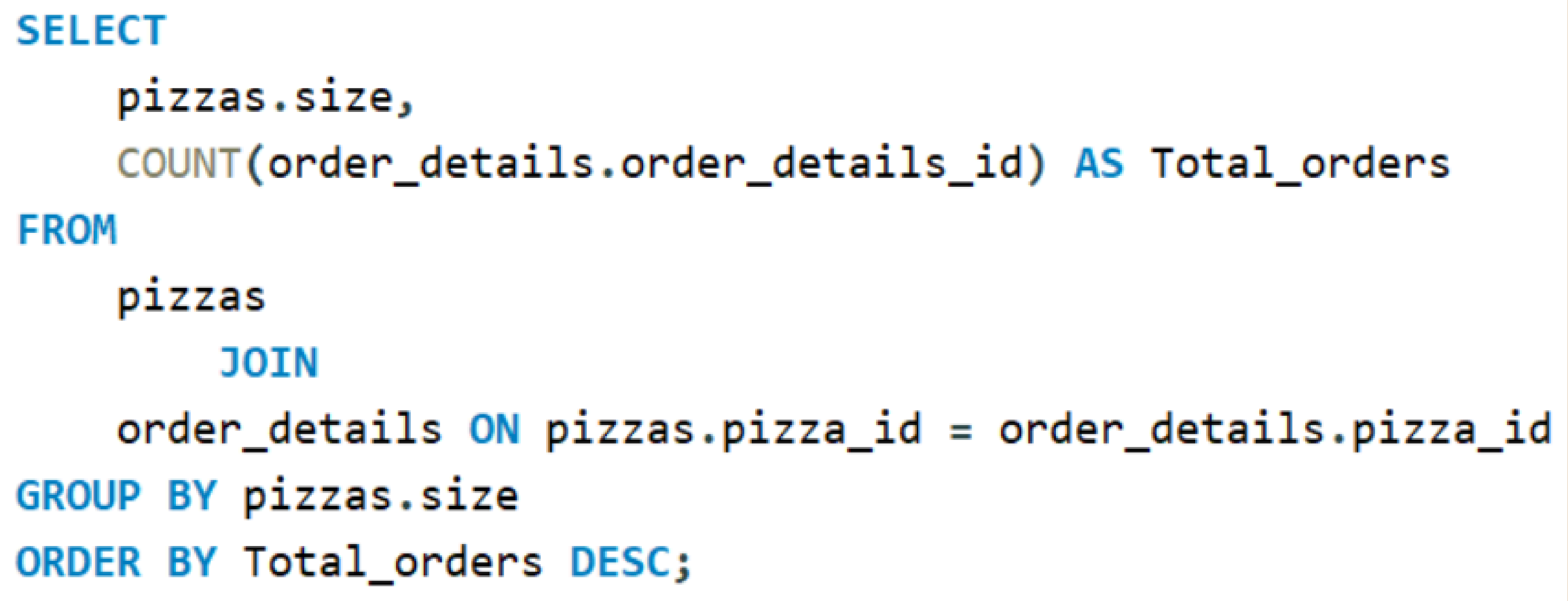
```sql
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```

OUTPUT-



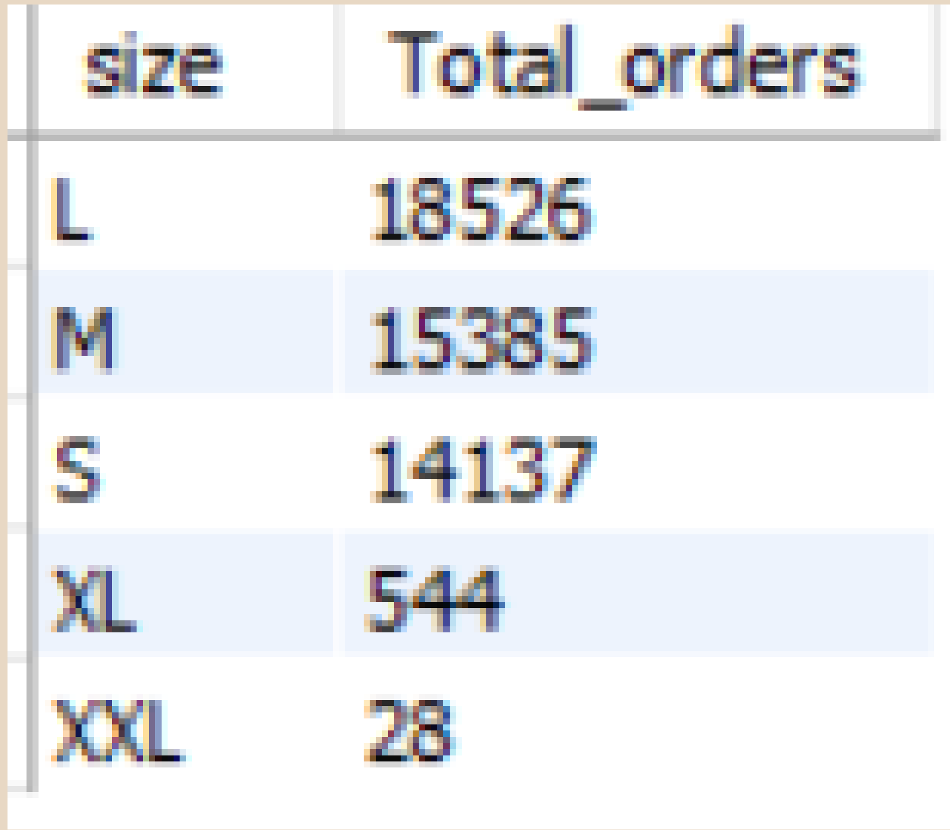
total_order
21350

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



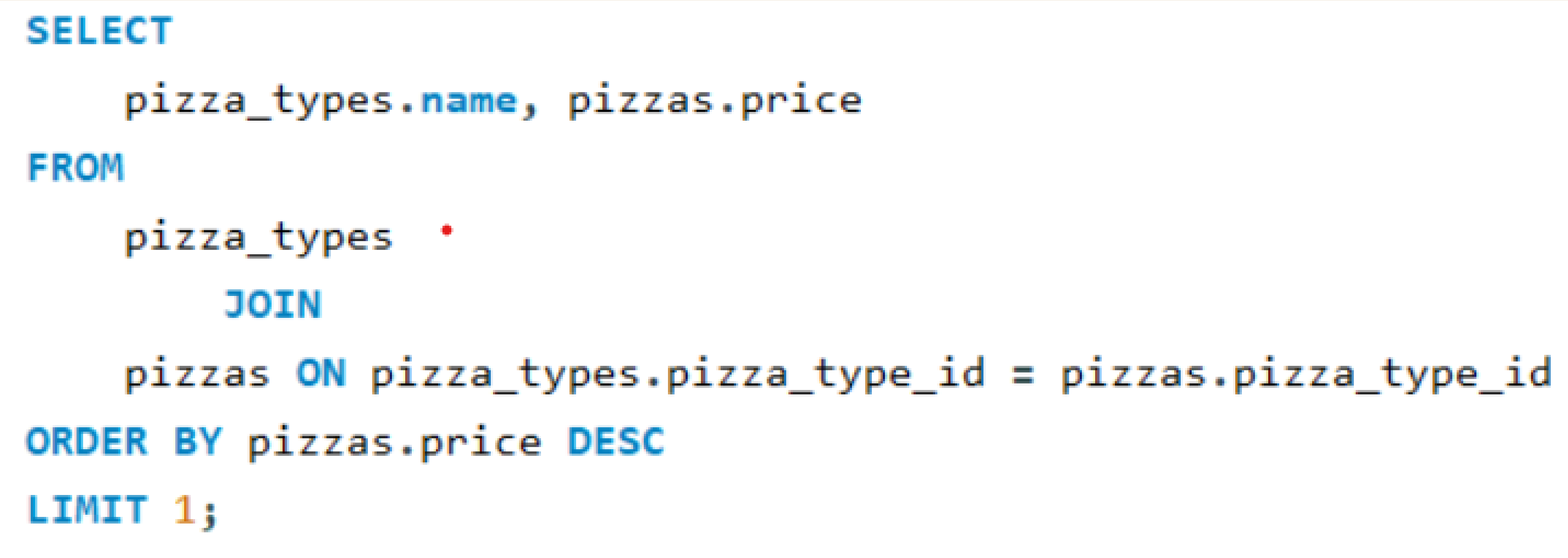
```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Total_orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Total_orders DESC;
```

OUTPUT-



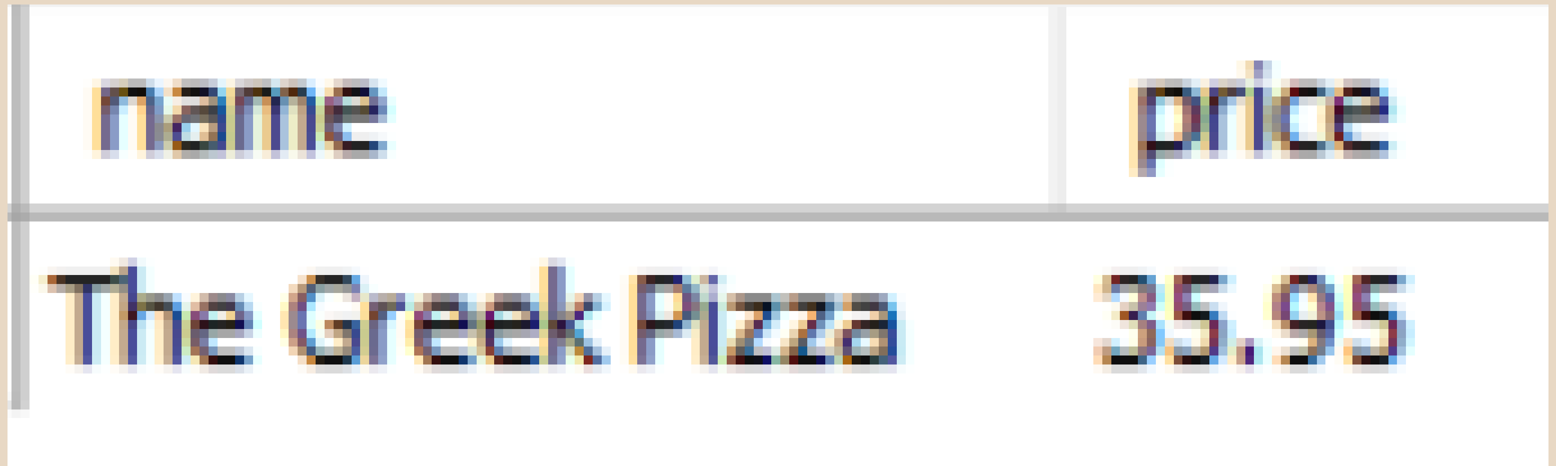
size	Total_orders
L	18526
M	15385
S	14137
XL	544
XXL	28

IDENTIFY THE HIGHEST-PRICED PIZZA.



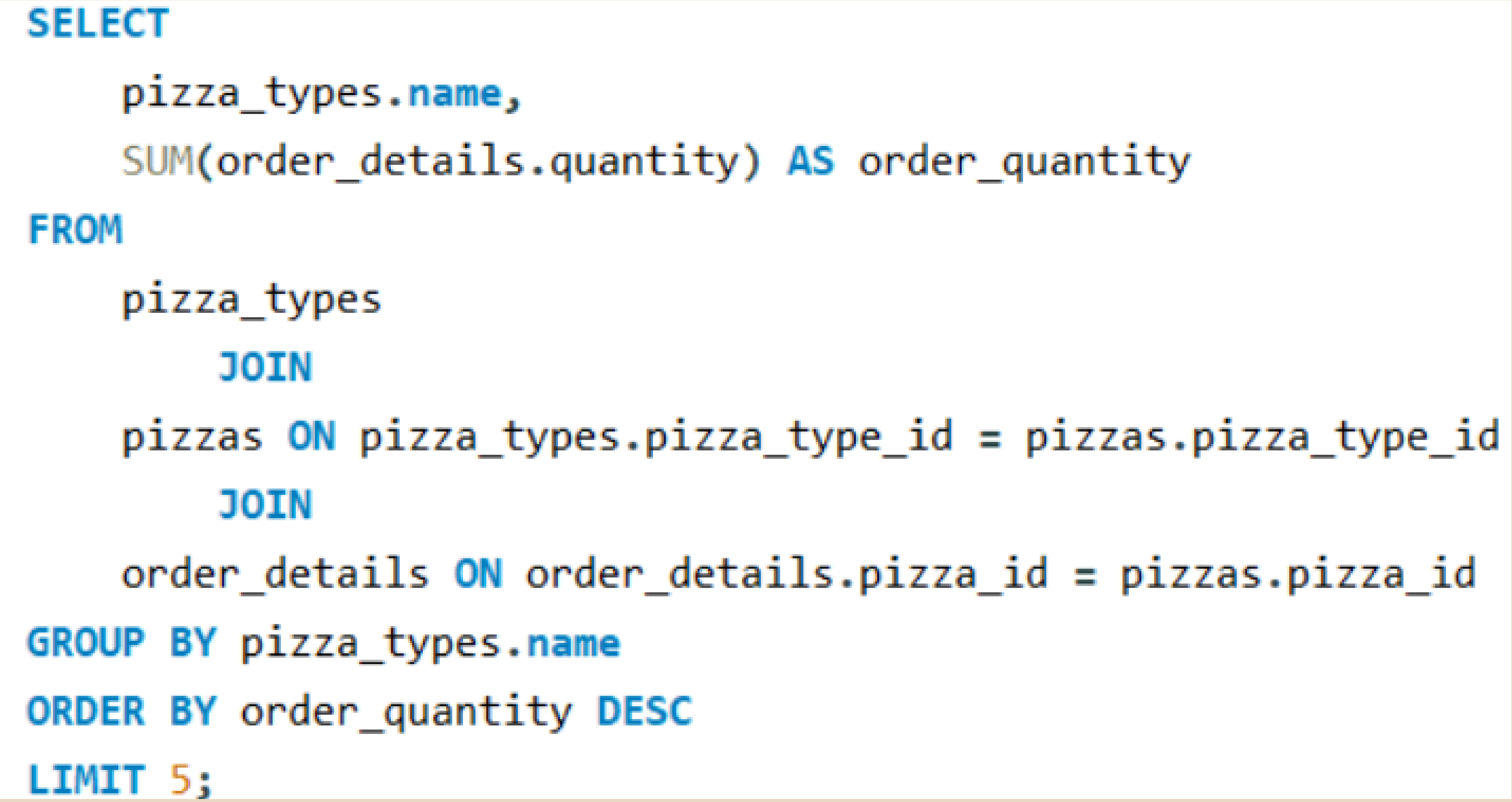
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

OUTPUT-



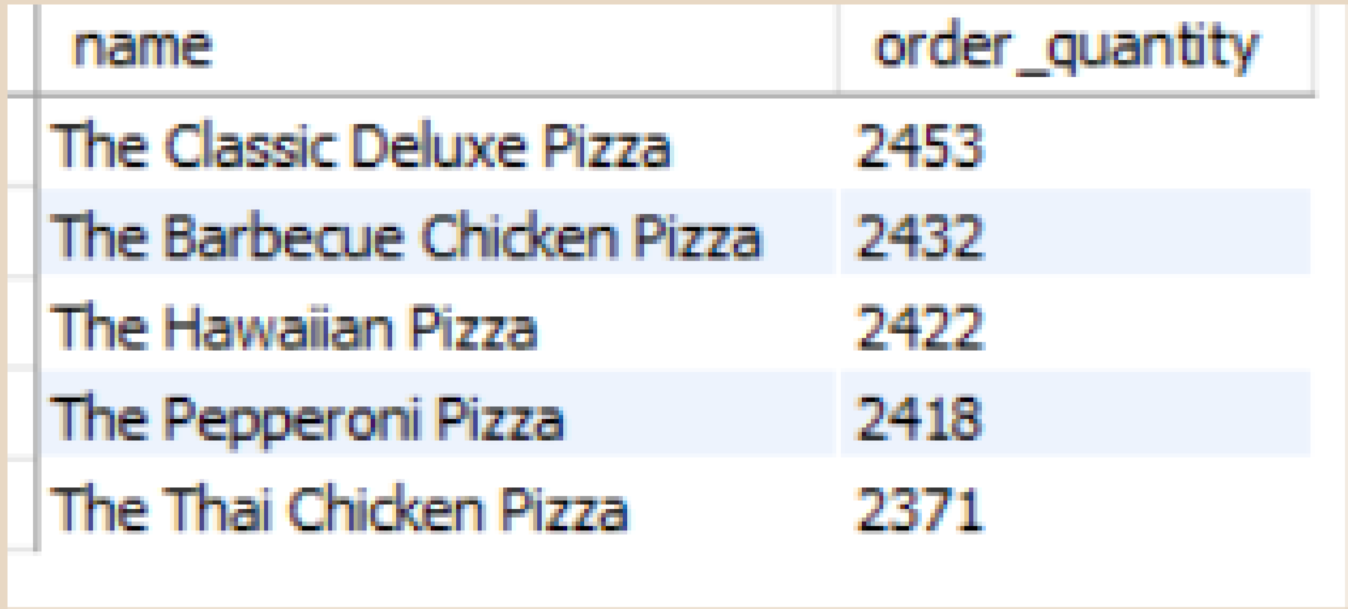
name	price
The Greek Pizza	35.95

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



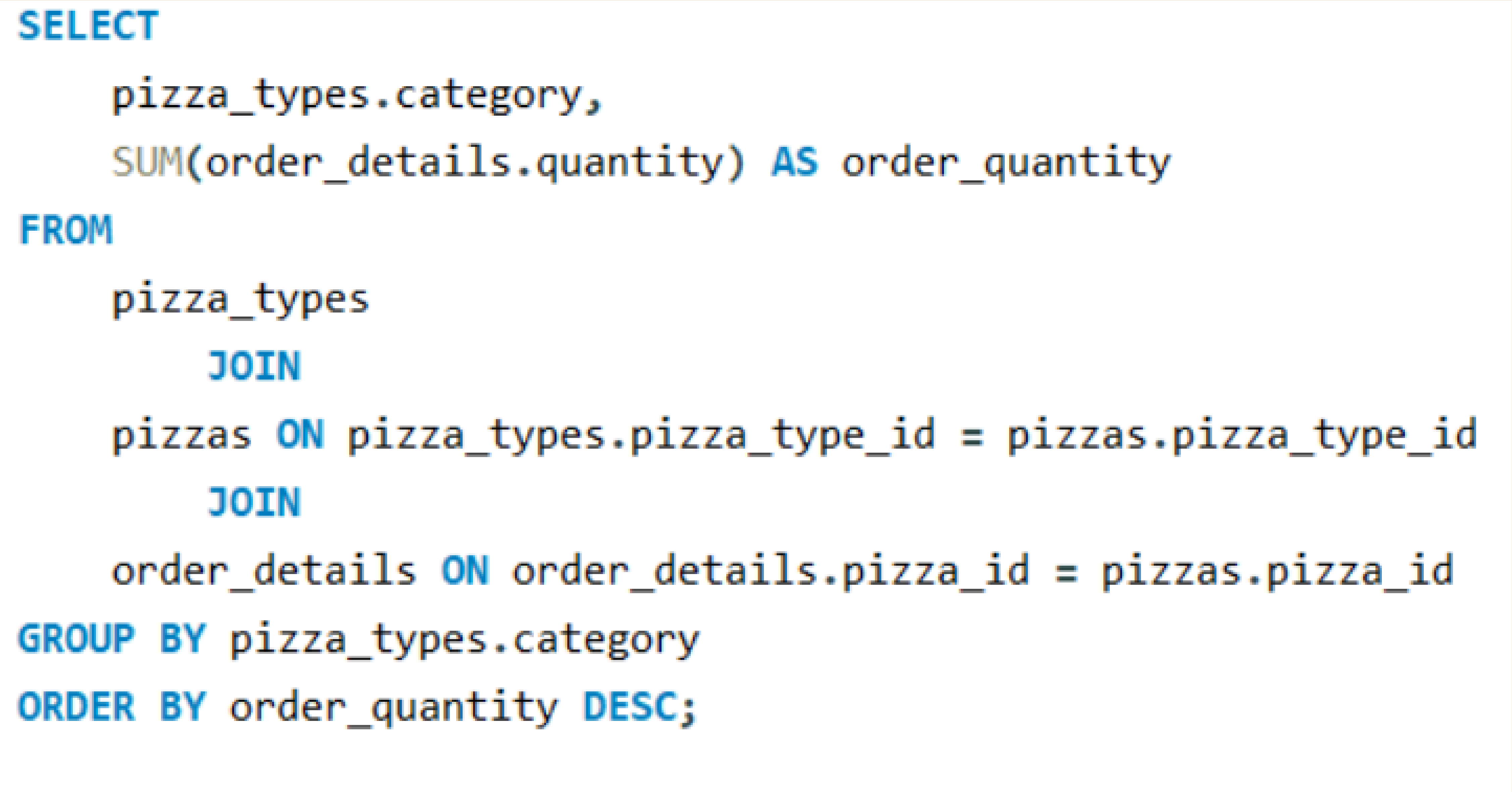
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS order_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY order_quantity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

OUTPUT-



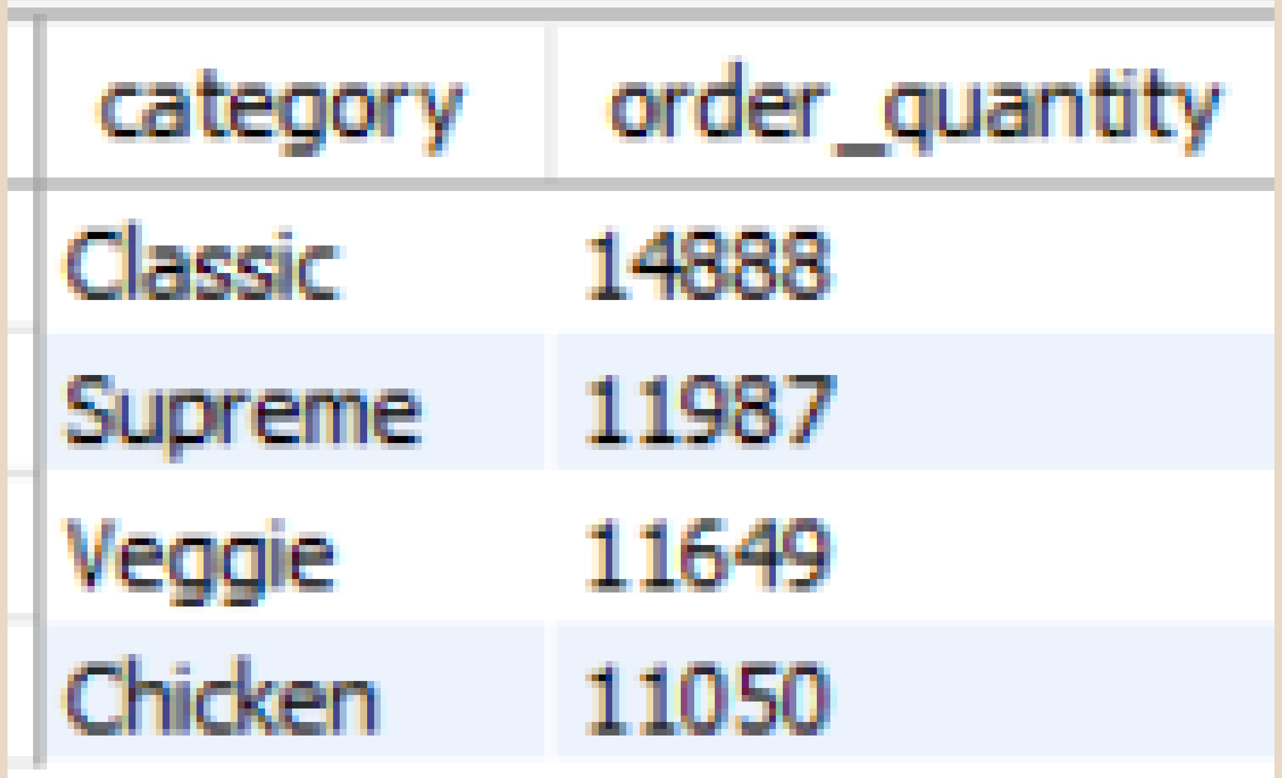
name	order_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



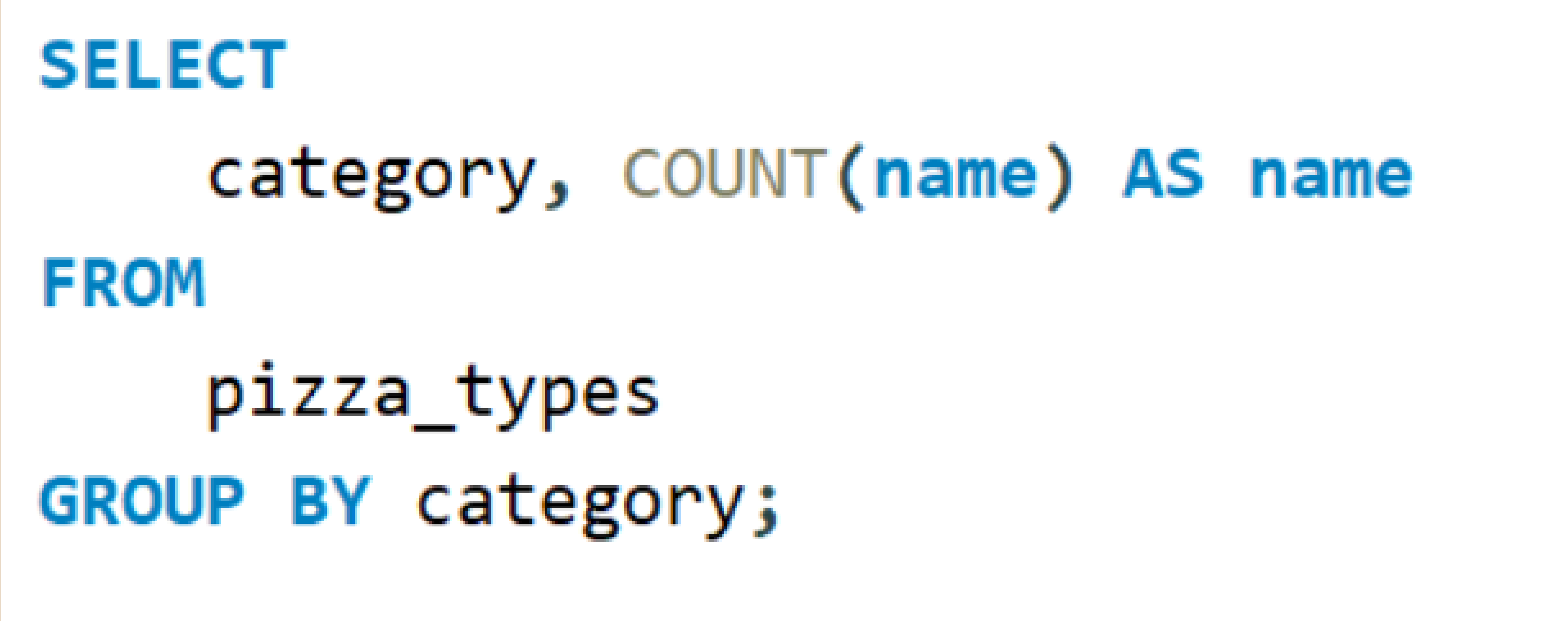
```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS order_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY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

OUTPUT-



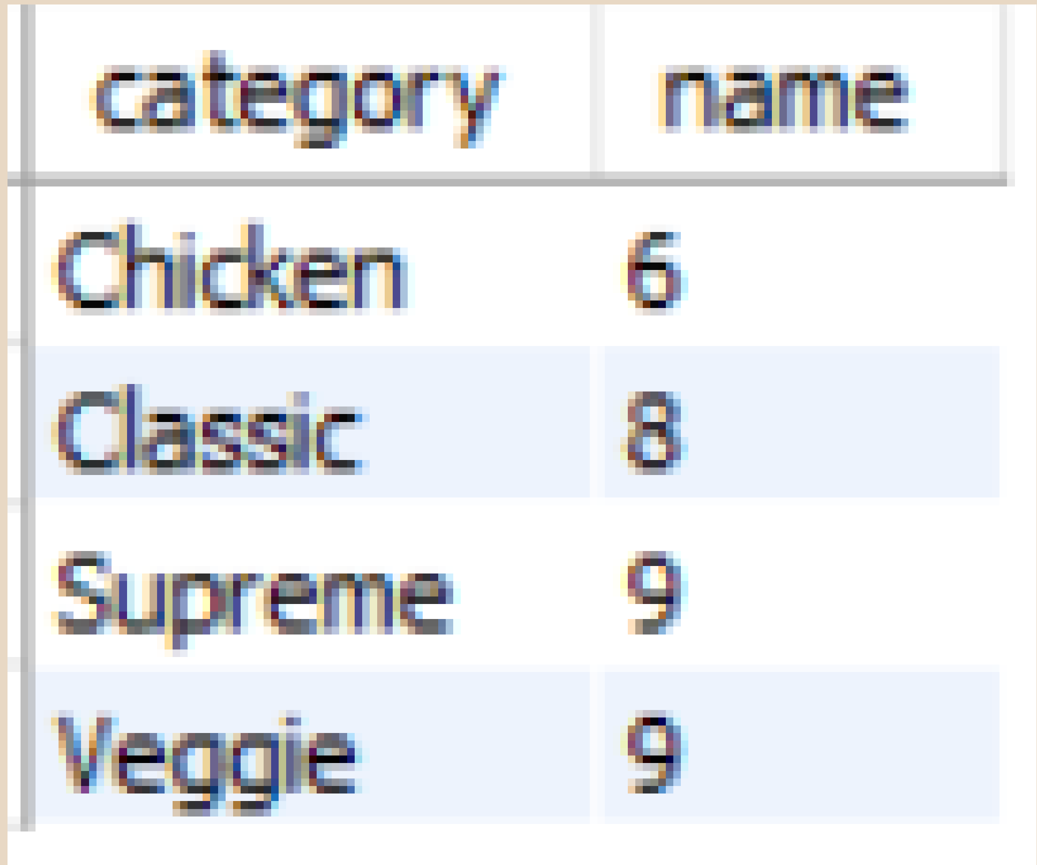
category	order_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



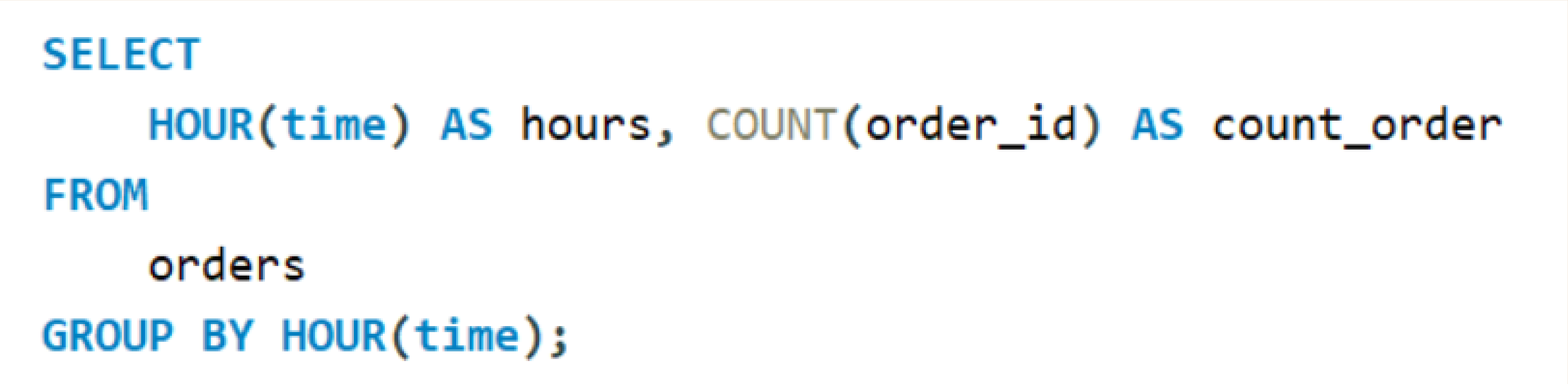
```
SELECT
    category, COUNT(name) AS name
FROM
    pizza_types
GROUP BY category;
```

OUTPUT-



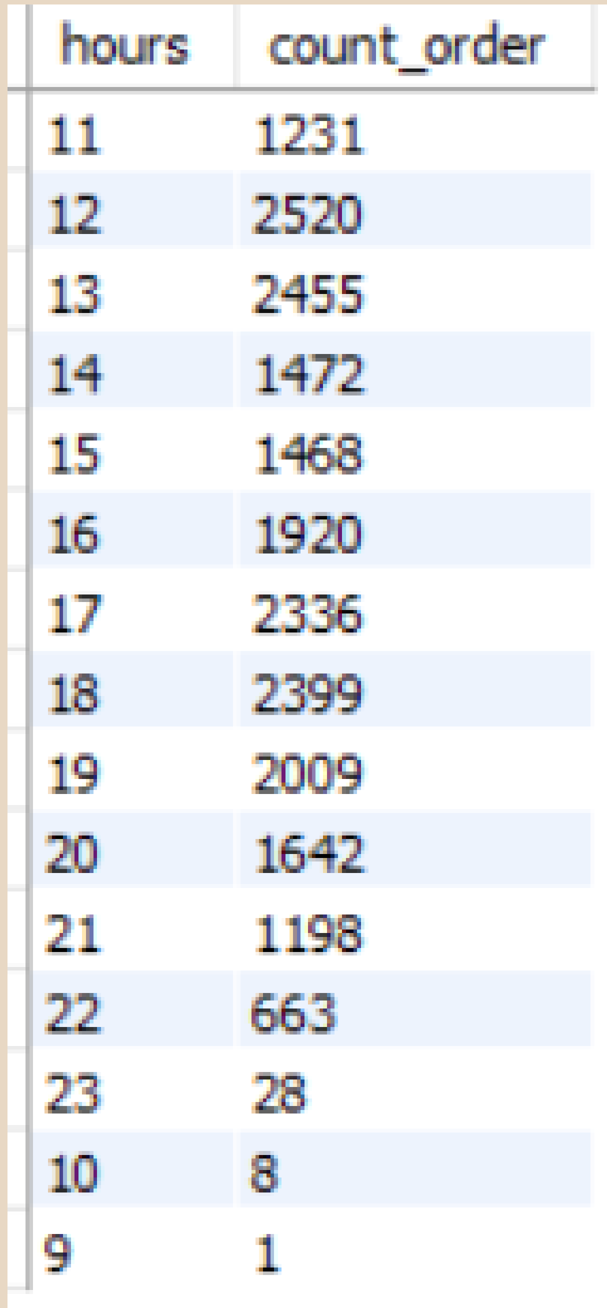
category	name
Chicken	6
Classic	8
Supreme	9
Veggie	9

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



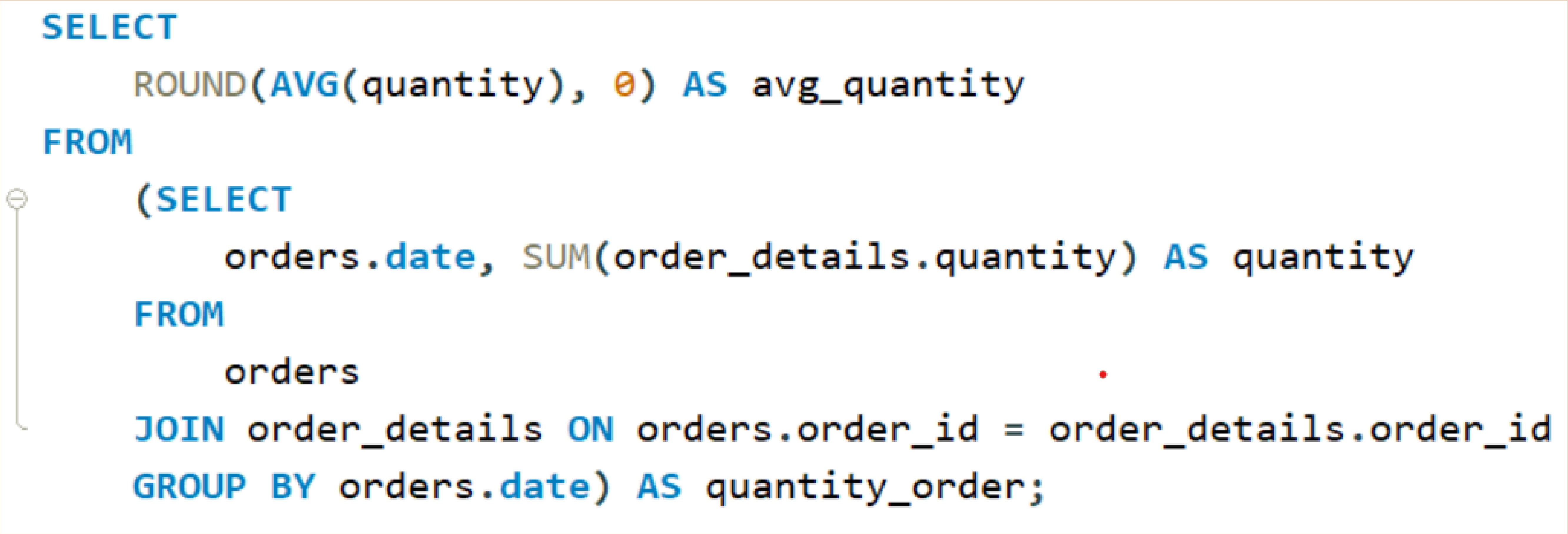
```
SELECT
    HOUR(time) AS hours, COUNT(order_id) AS count_order
FROM
    orders
GROUP BY HOUR(time);
```

OUTPUT-



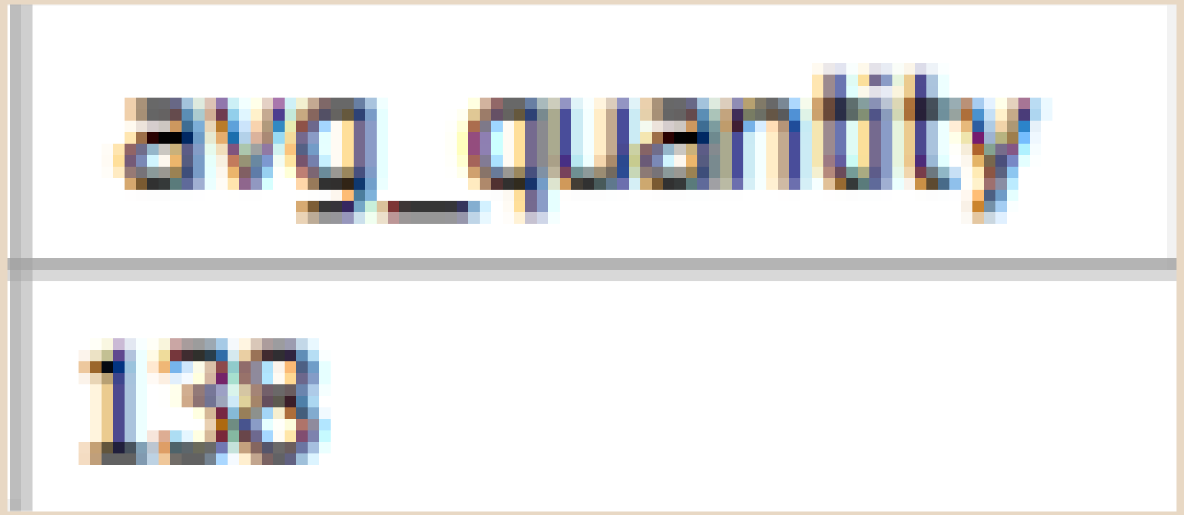
hours	count_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



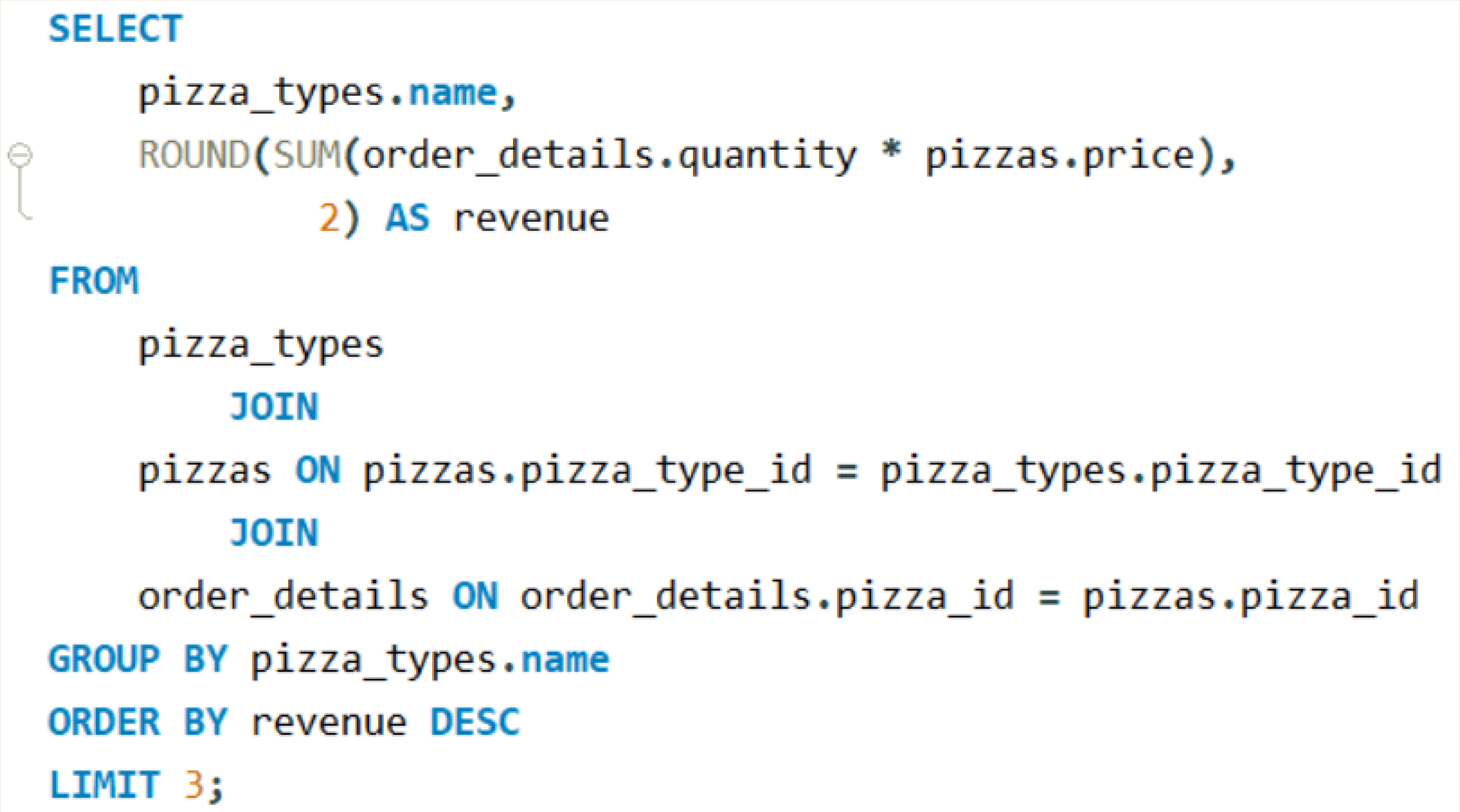
```
SELECT
    ROUND(AVG(quantity), 0) AS avg_quantity
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS quantity_order;
```

OUTPUT-



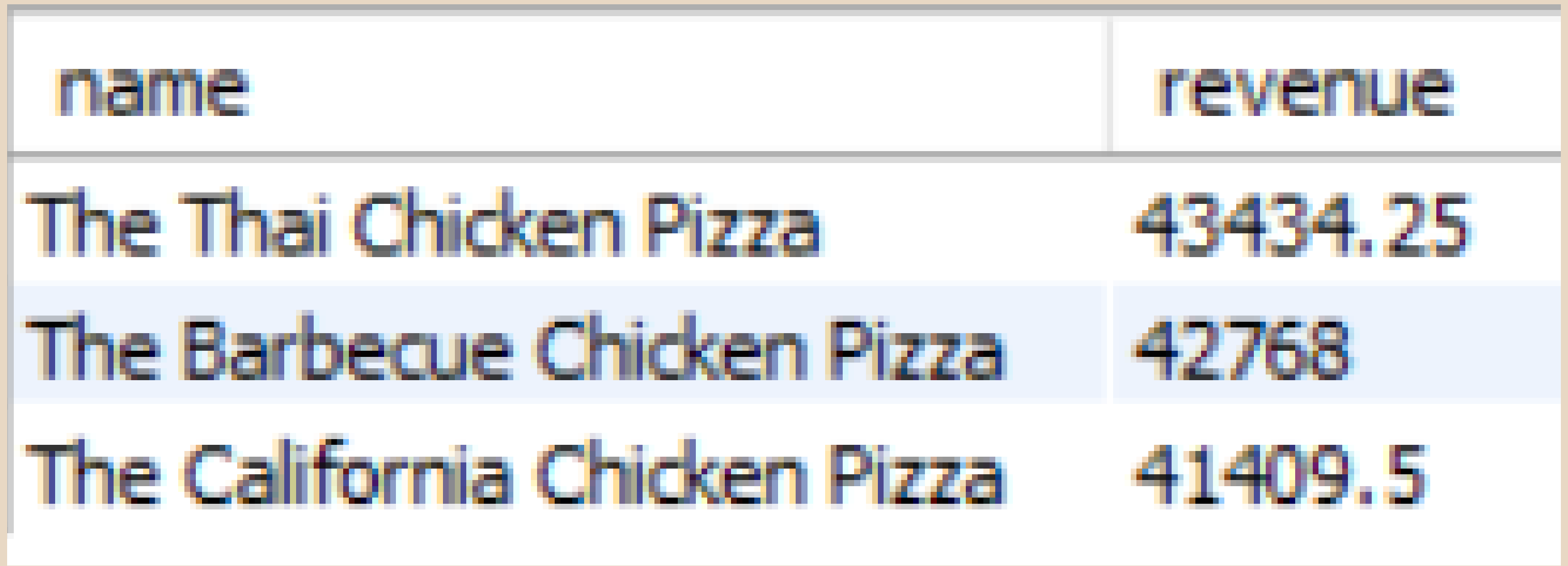
avg_quantity
138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



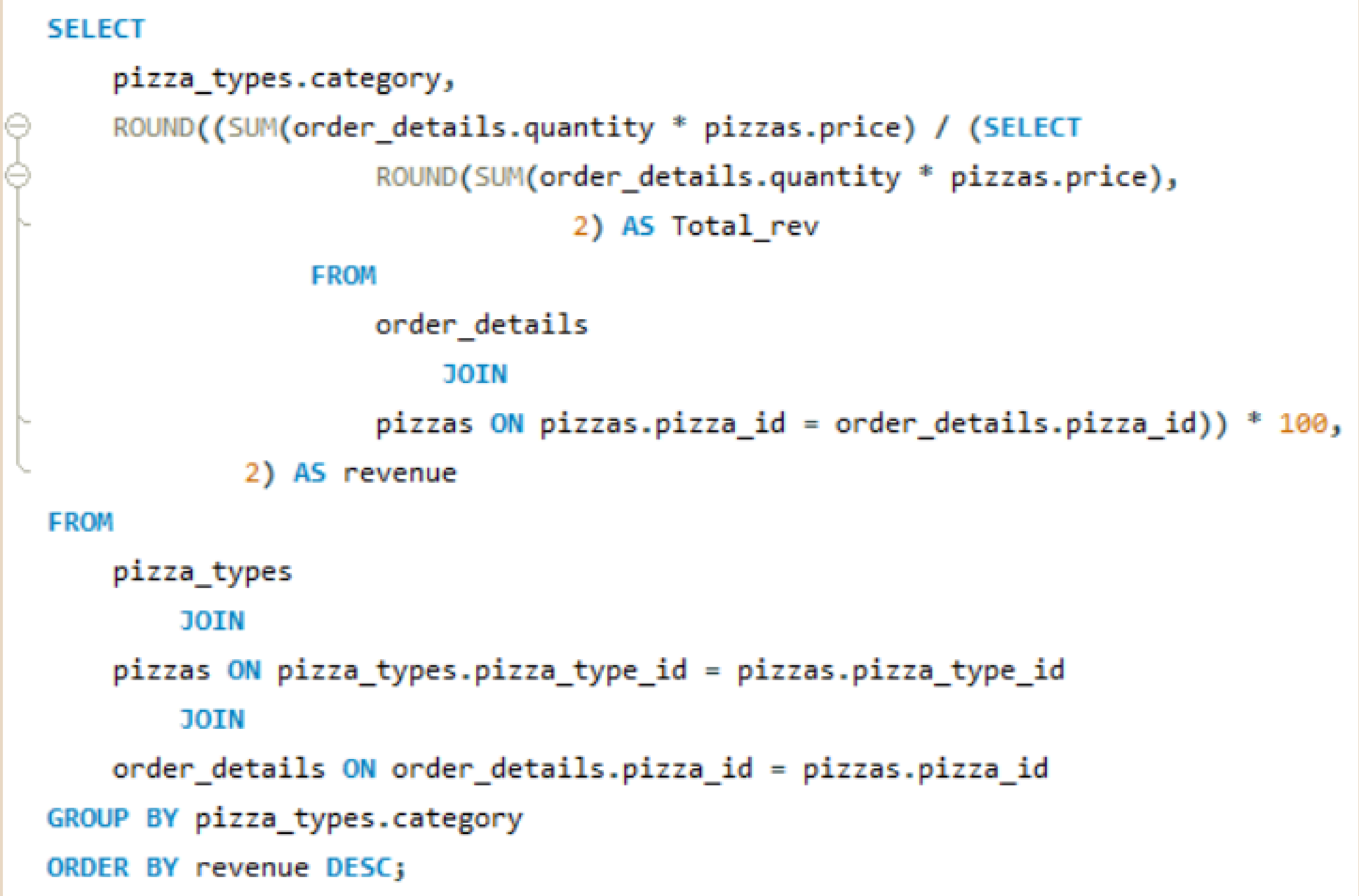
```
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name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

OUTPUT-



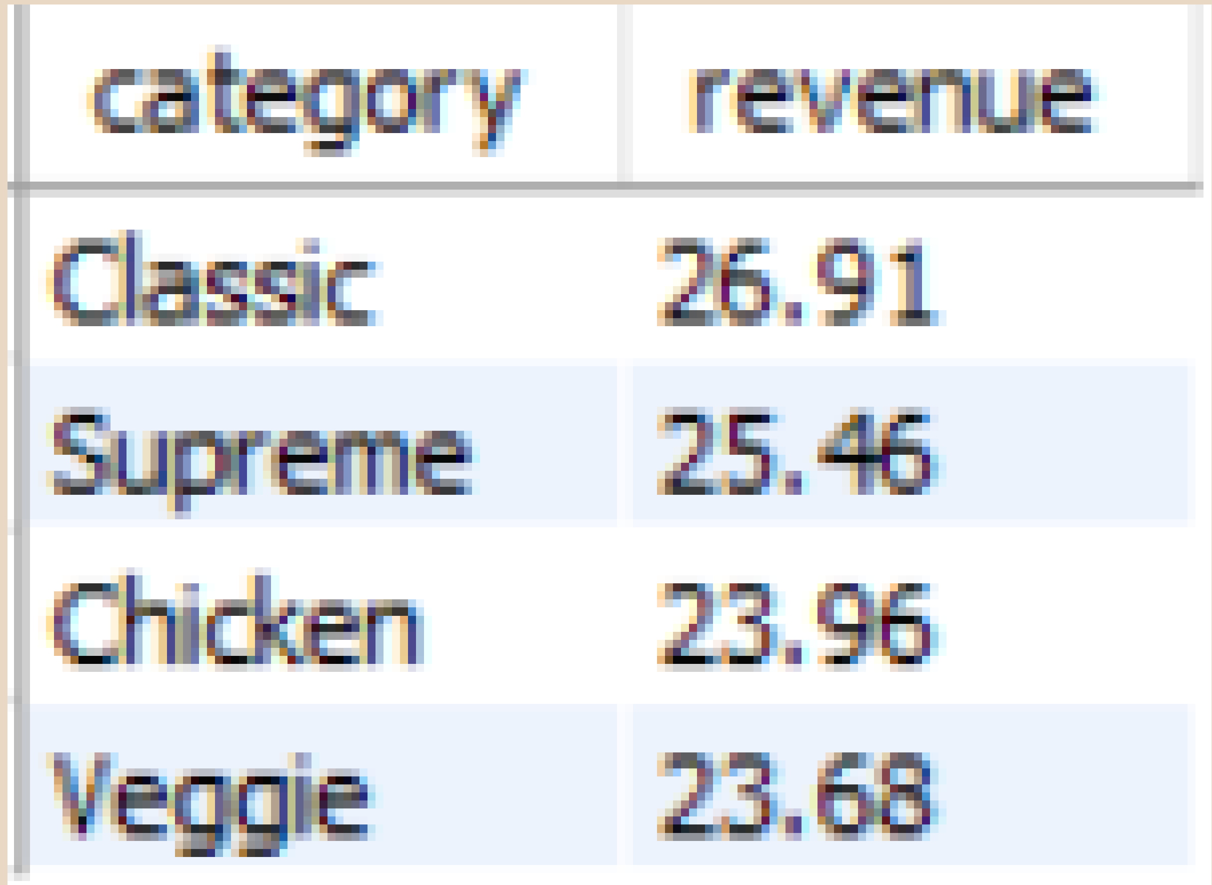
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



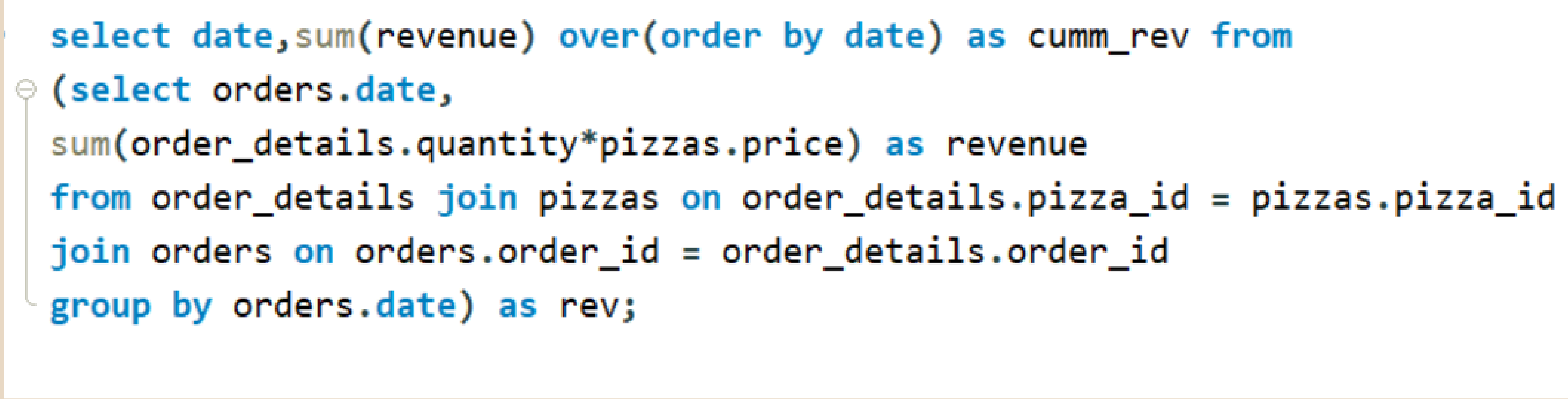
```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS Total_rev
                FROM
                    order_details
                        JOIN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

OUTPUT-



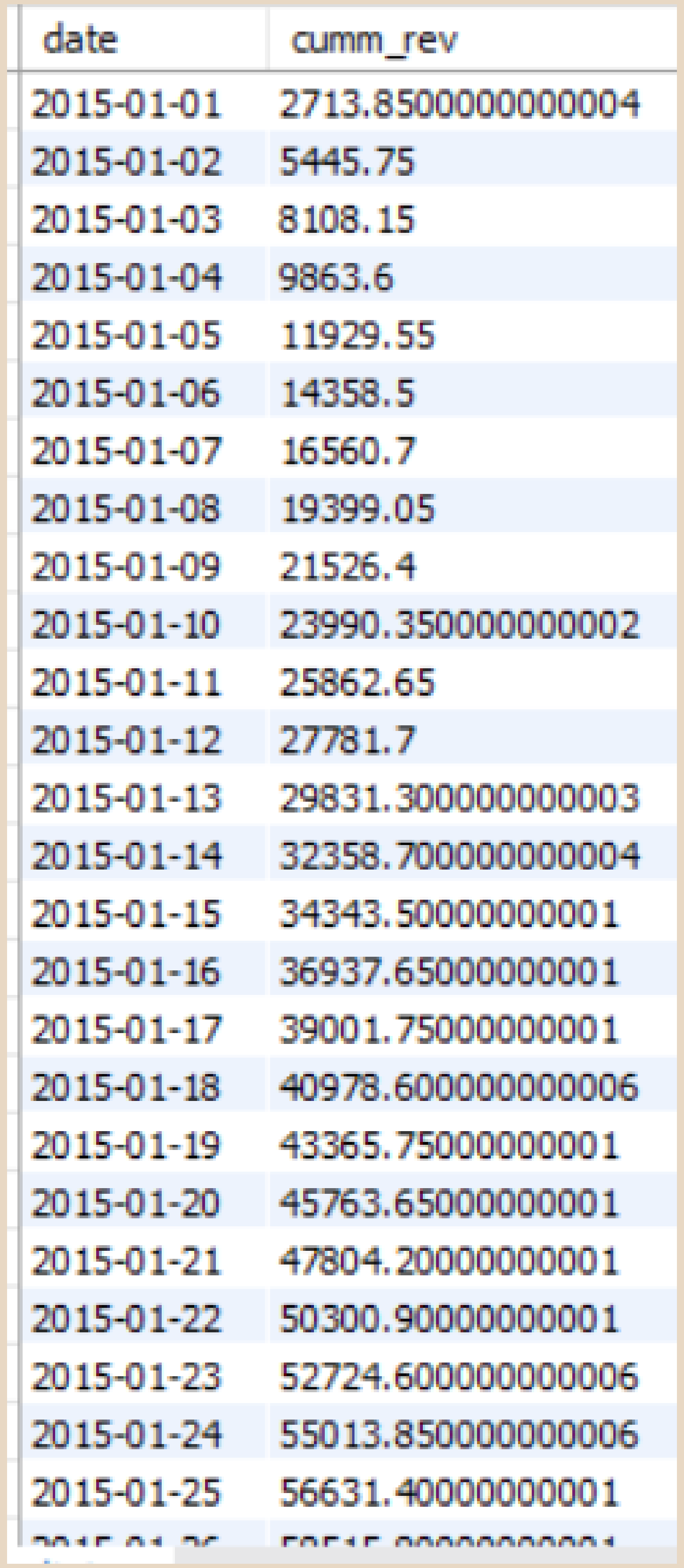
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
select date,sum(revenue) over(order by date) as cumm_rev from
(select orders.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.date) as rev;
```

OUTPUT-



date	cumm_rev
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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001

THANK YOU

BY-TARUSHIKA
SHARMA