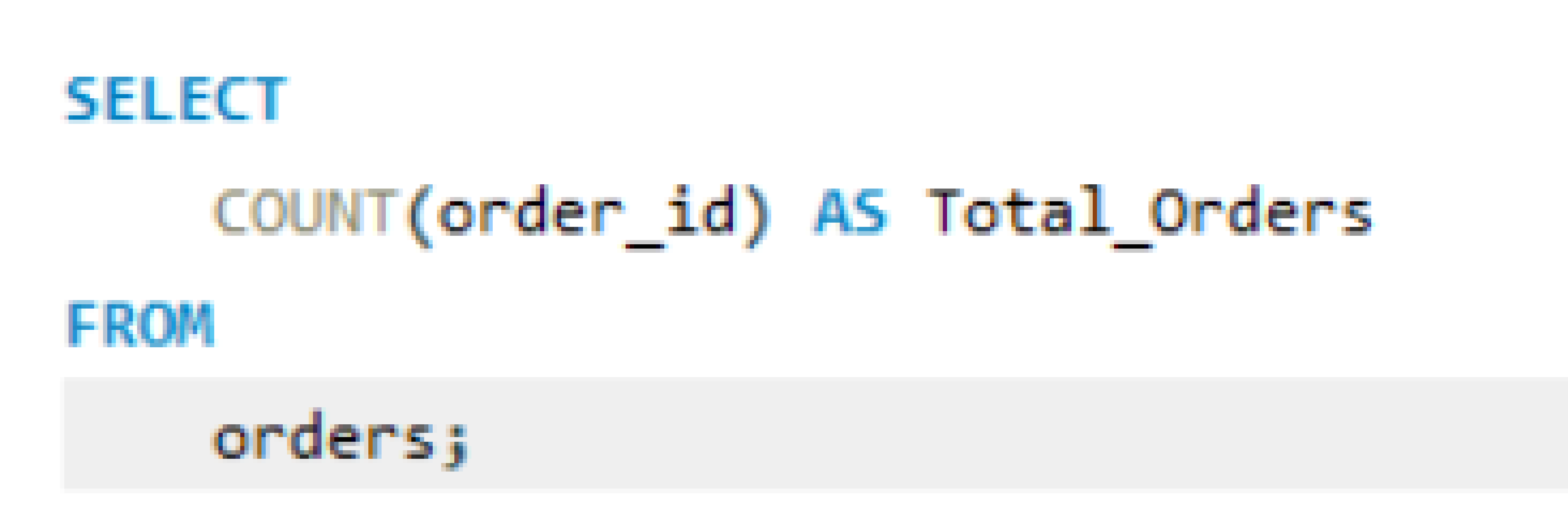
SQL PROJECT ON PIZZA SALES

HELLO!!!

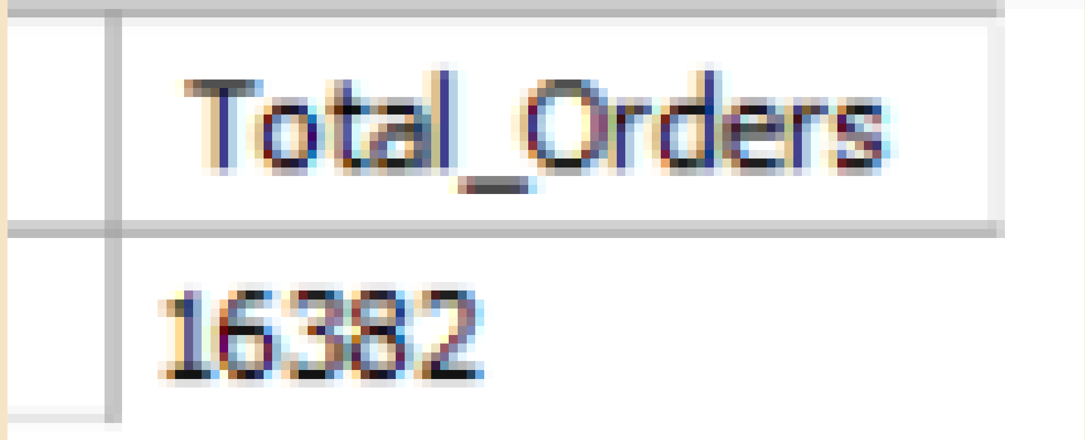
I'm Tusshar Singh, and this project uses SQL to study patterns in pizza sales.

RETRIVE THE NO OF ORDERS PLACED



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

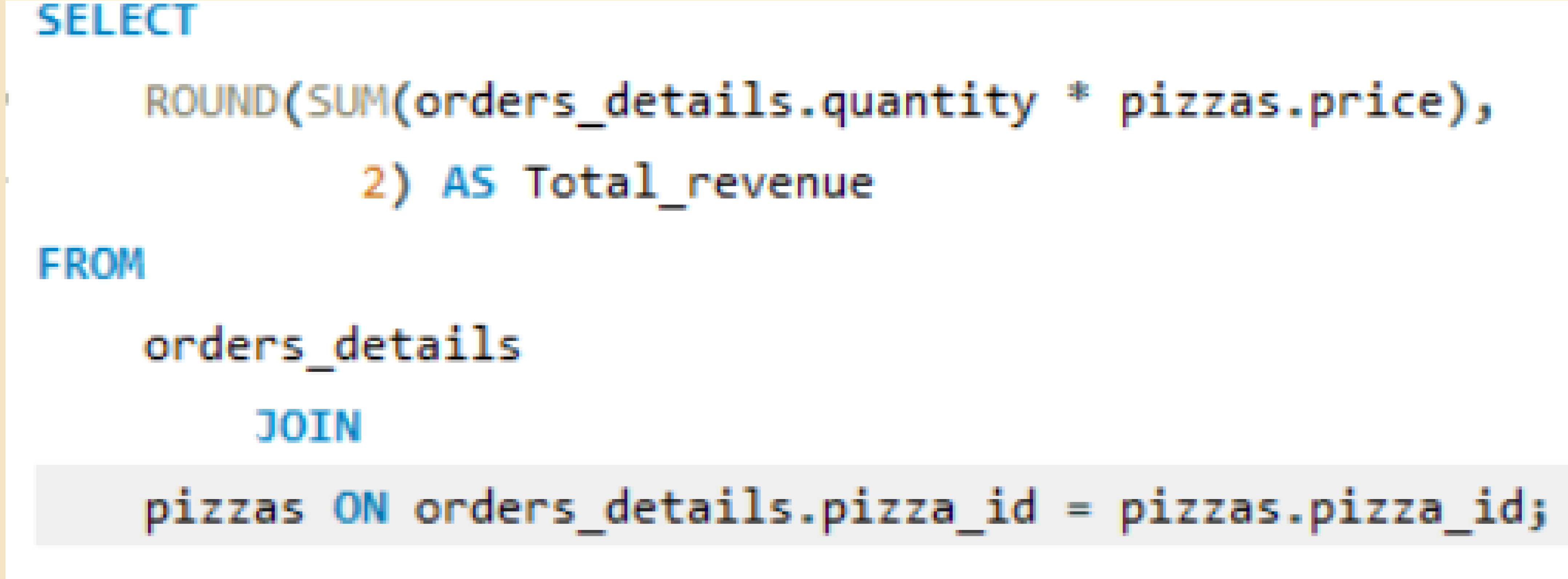
QUERY



Total_Orders
16382

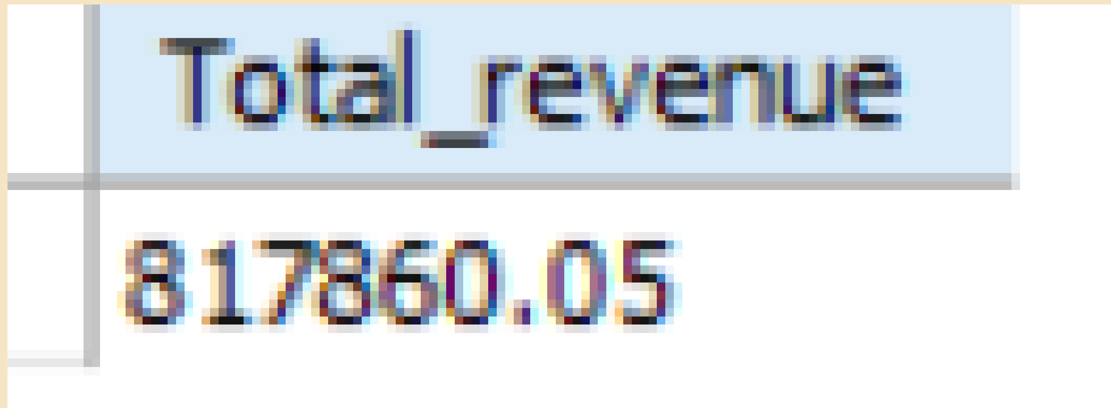
RESULT

CALCULATE THE TOTAL REVENUE GENERATED BY PIZZAS SALES



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

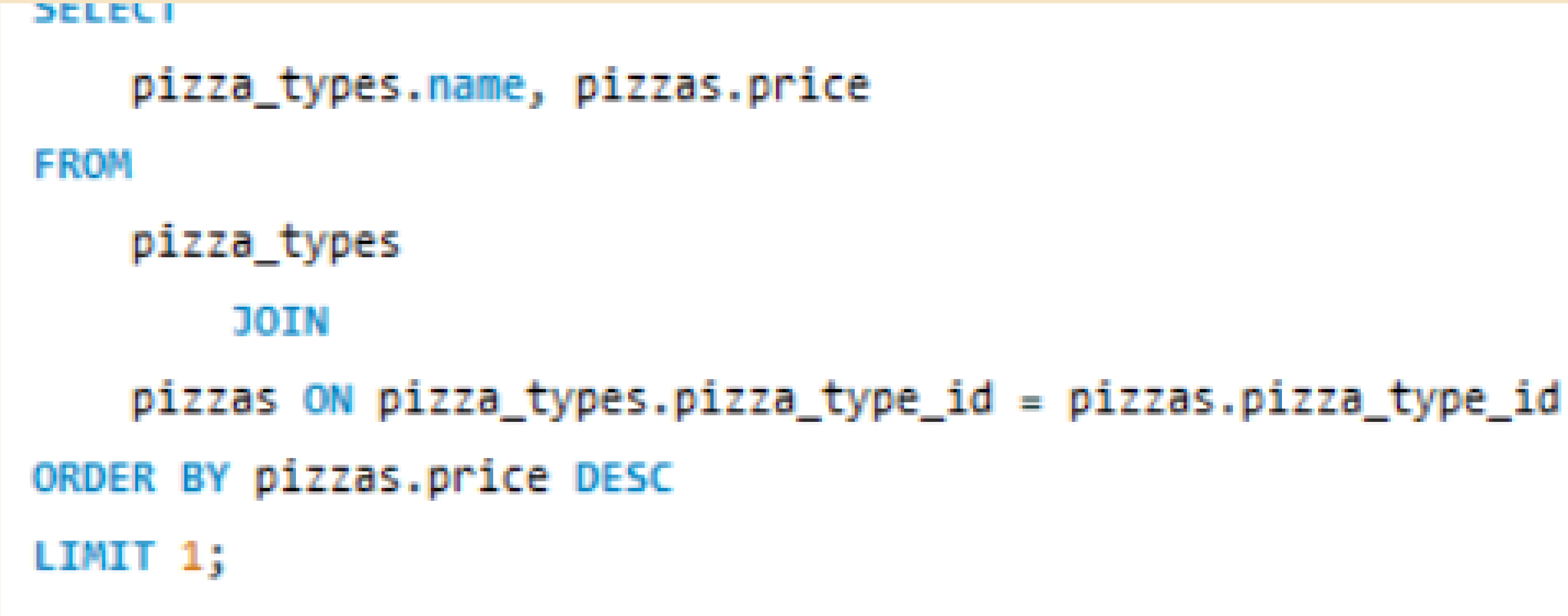
QUERY



Total_revenue
817860.05

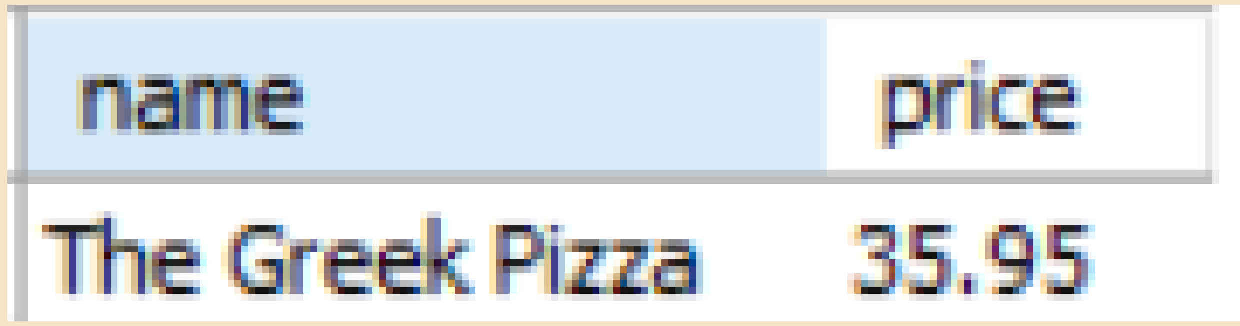
RESULT

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;
```

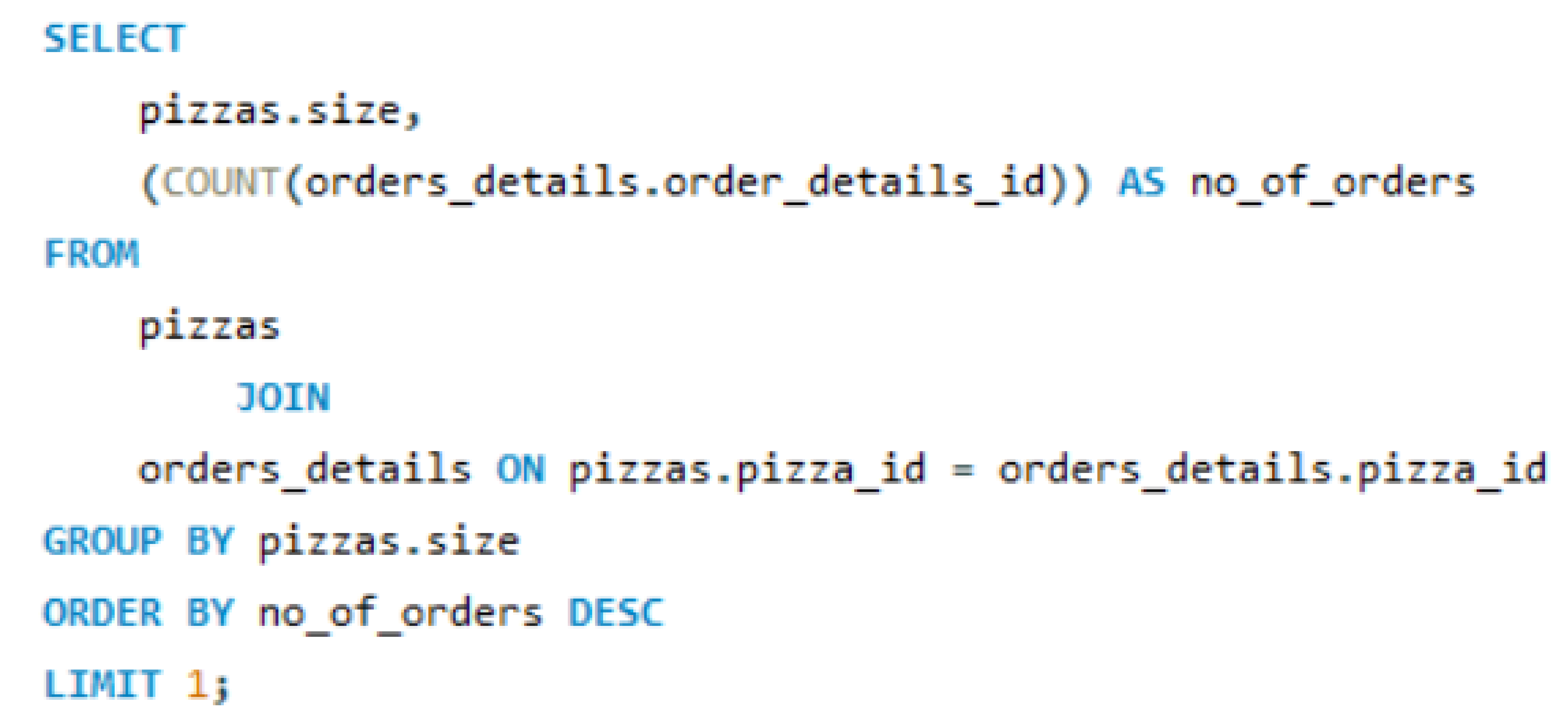
QUERY



name	price
The Greek Pizza	35.95

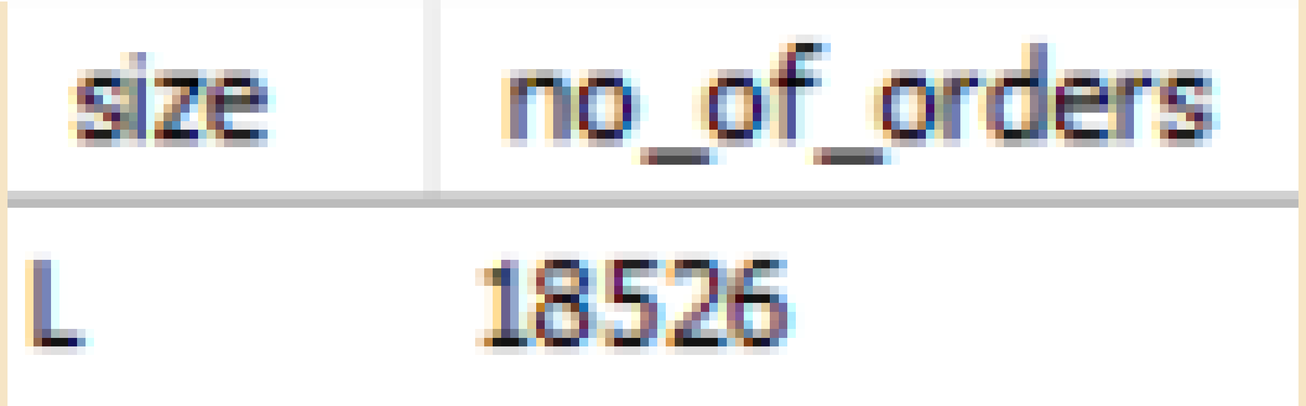
RESULT

IDENTIFY THE MOST COMMONLY OREDERED PIZZA SIZE



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

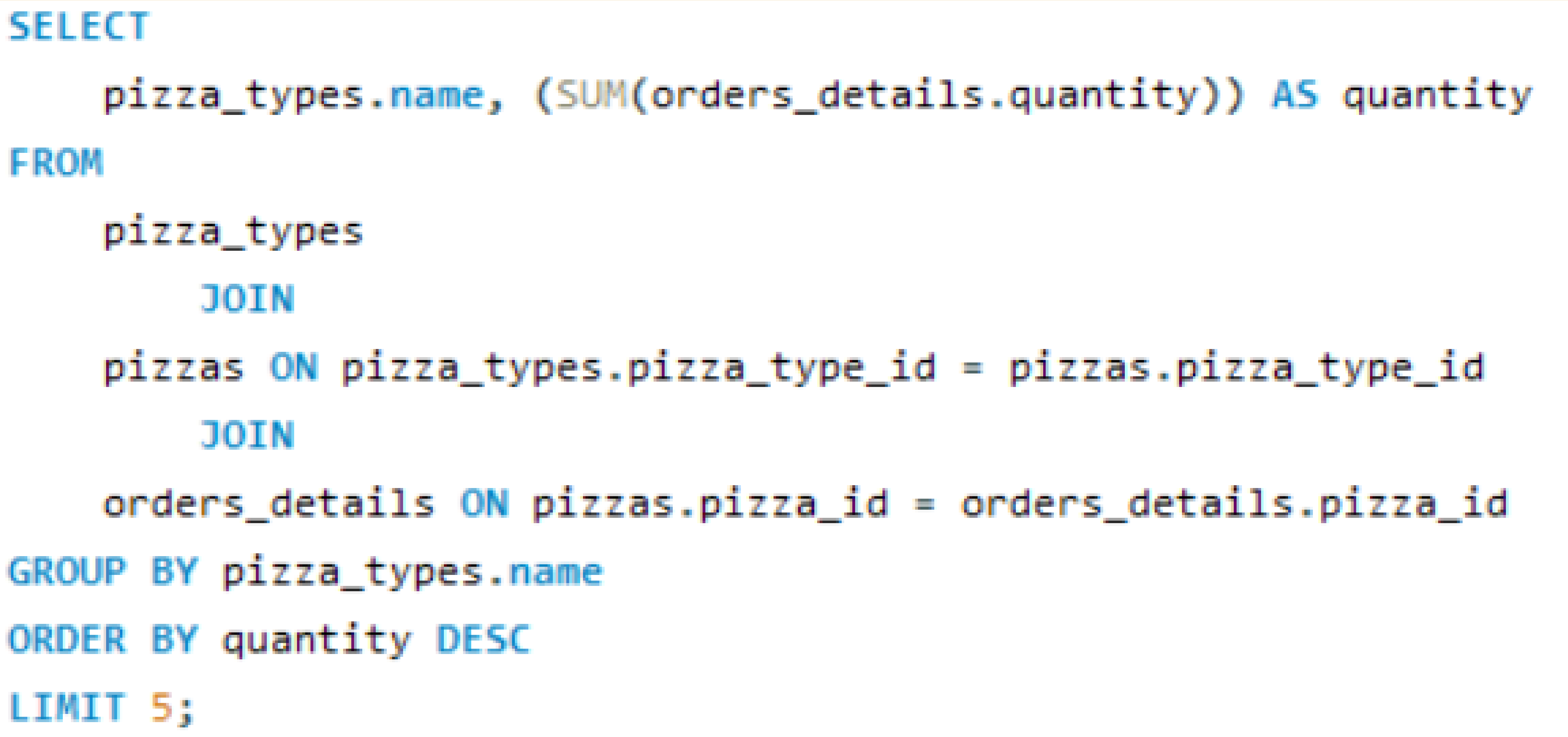
QUERY



size	no_of_orders
L	18526

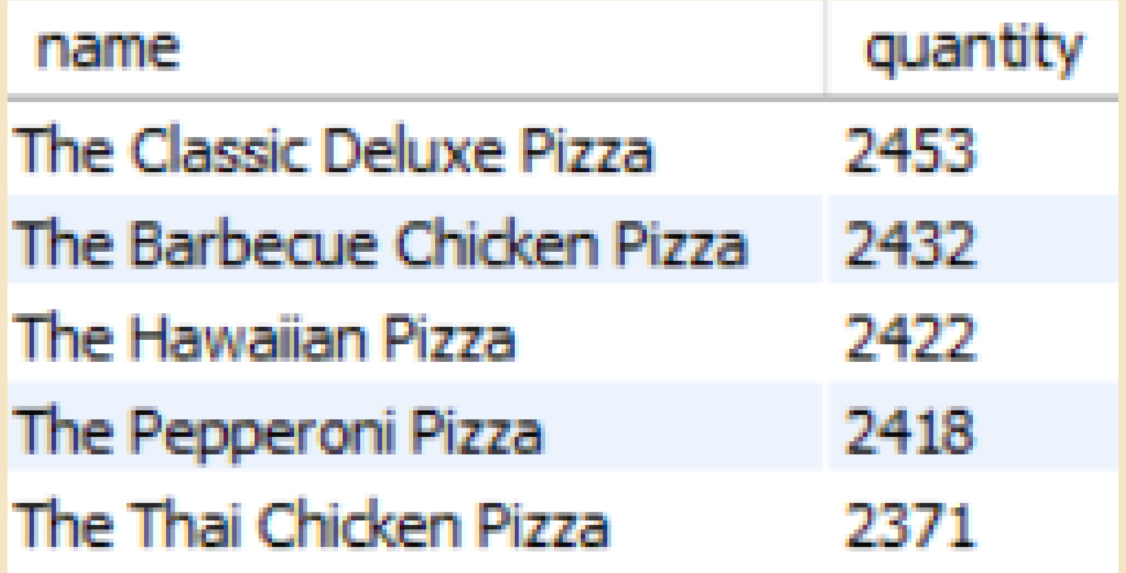
RESULT

LIST THE TOP 5 MOST ORDERED PIZZAS TYPES ALONG WITH THEIR QUANTITIES



```
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 pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

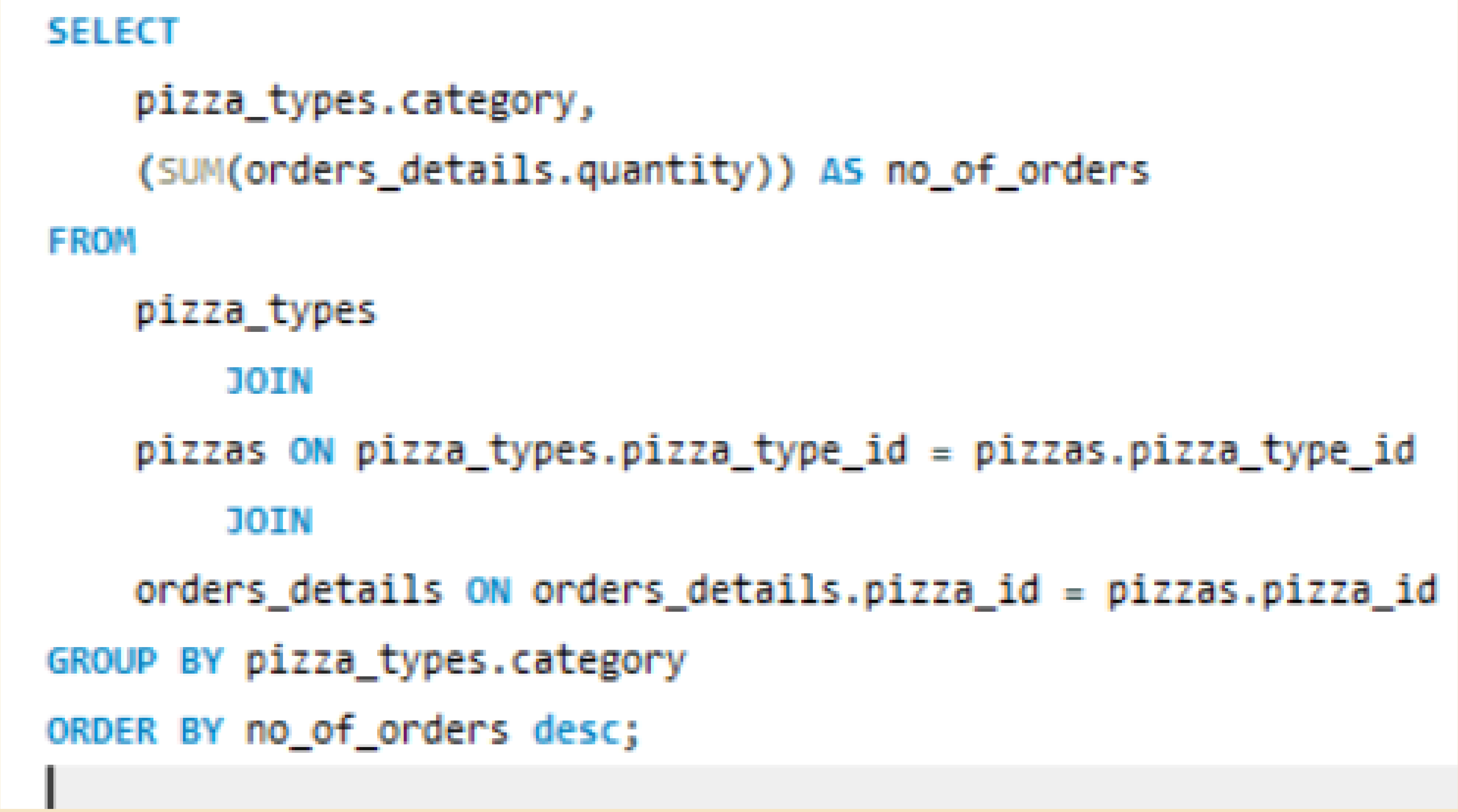
QUERY



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

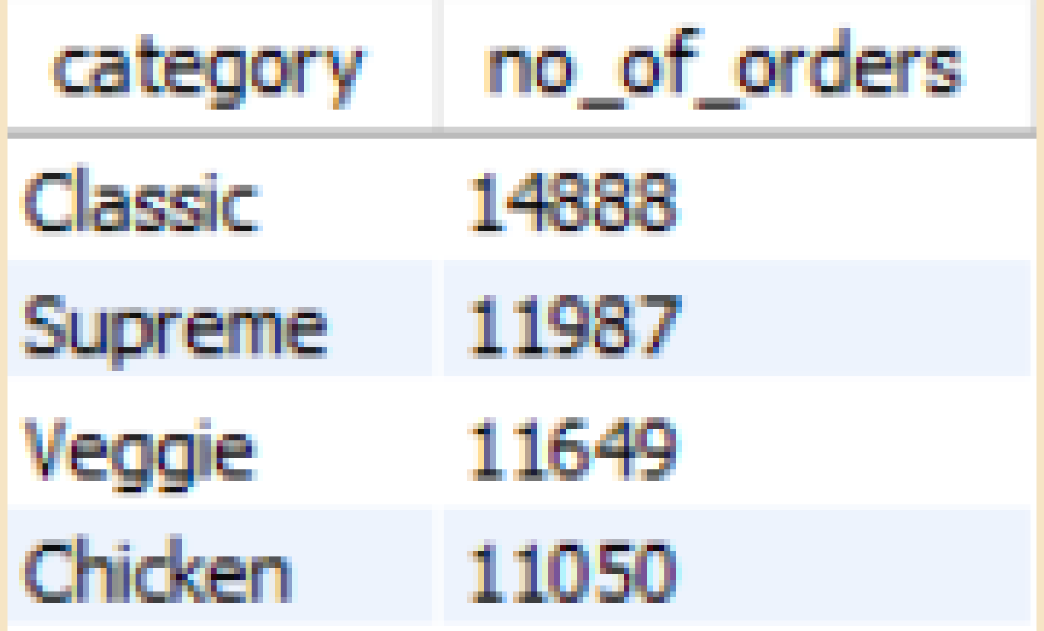
RESULT

JOIN THE NECCESSARY TABLES TO FIND THE TOATAL QUANTITY OF EACH PIZZA CATAGORY



```
SELECT
    pizza_types.category,
    (SUM(orders_details.quantity)) AS no_of_orders
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 no_of_orders desc;
```

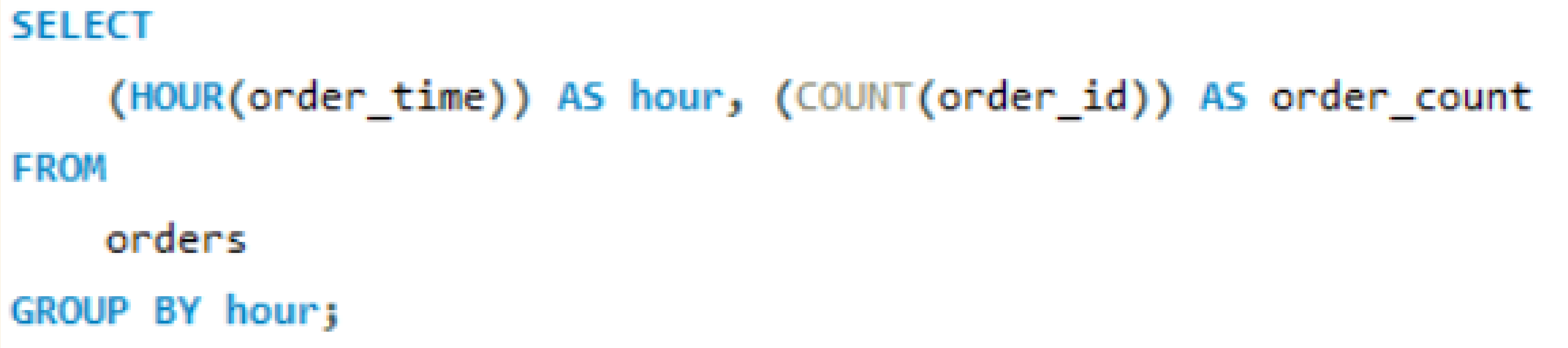
QUERY



category	no_of_orders
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

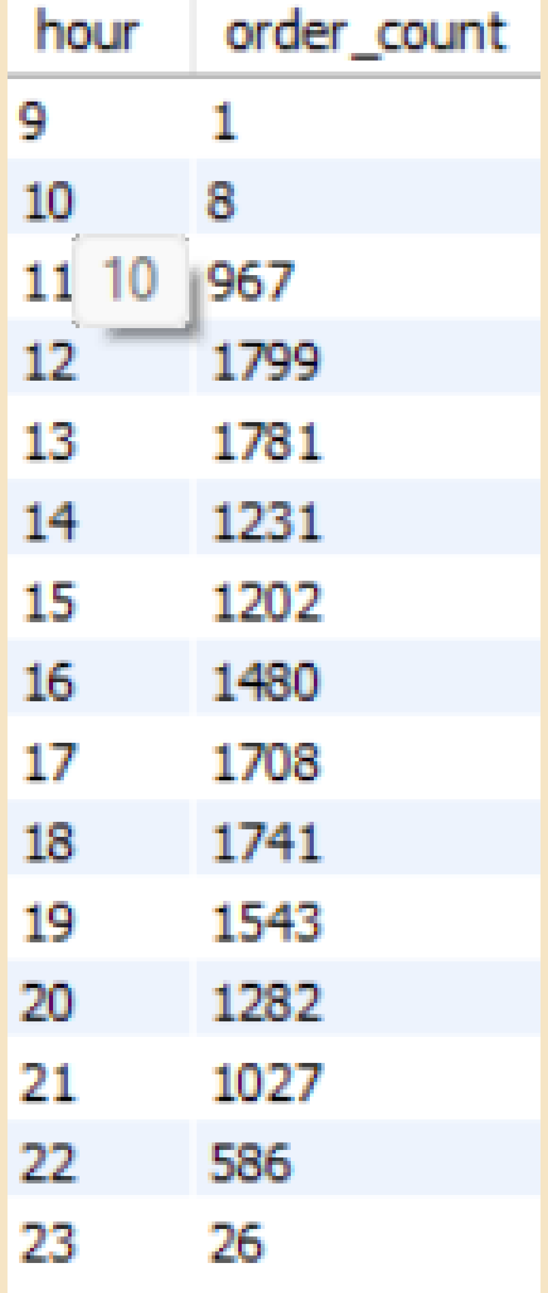
RESULT

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;
```

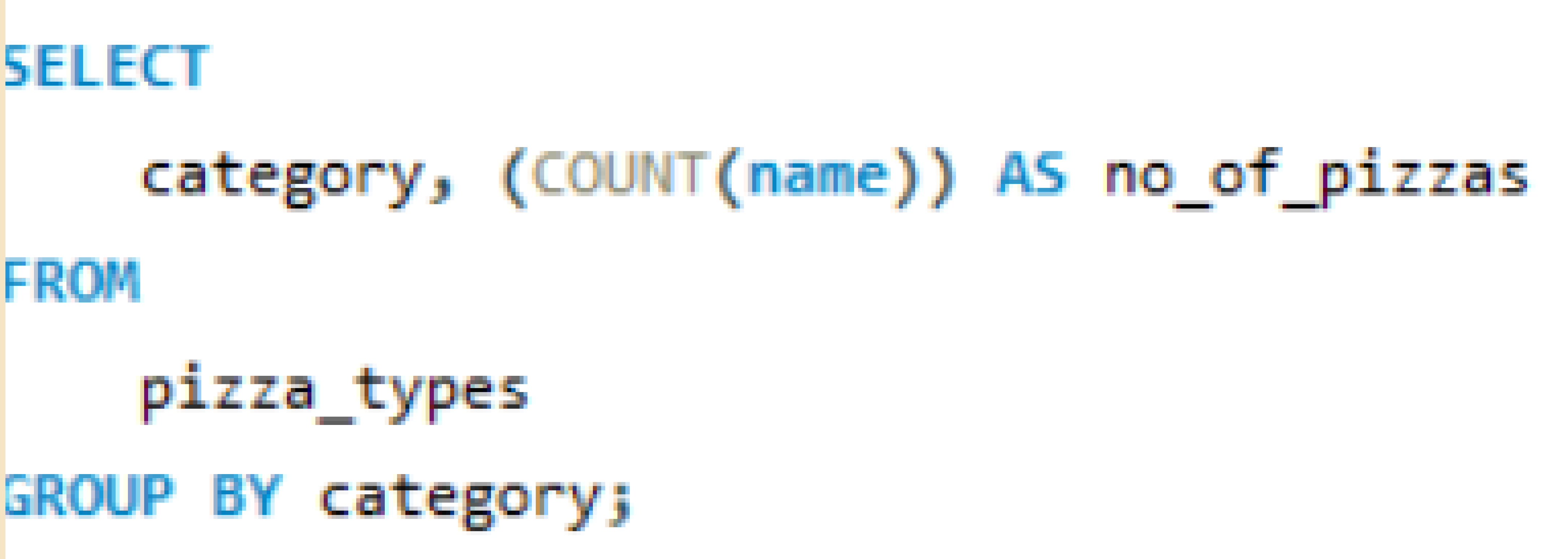
QUERY



hour	order_count
9	1
10	8
11	967
12	1799
13	1781
14	1231
15	1202
16	1480
17	1708
18	1741
19	1543
20	1282
21	1027
22	586
23	26

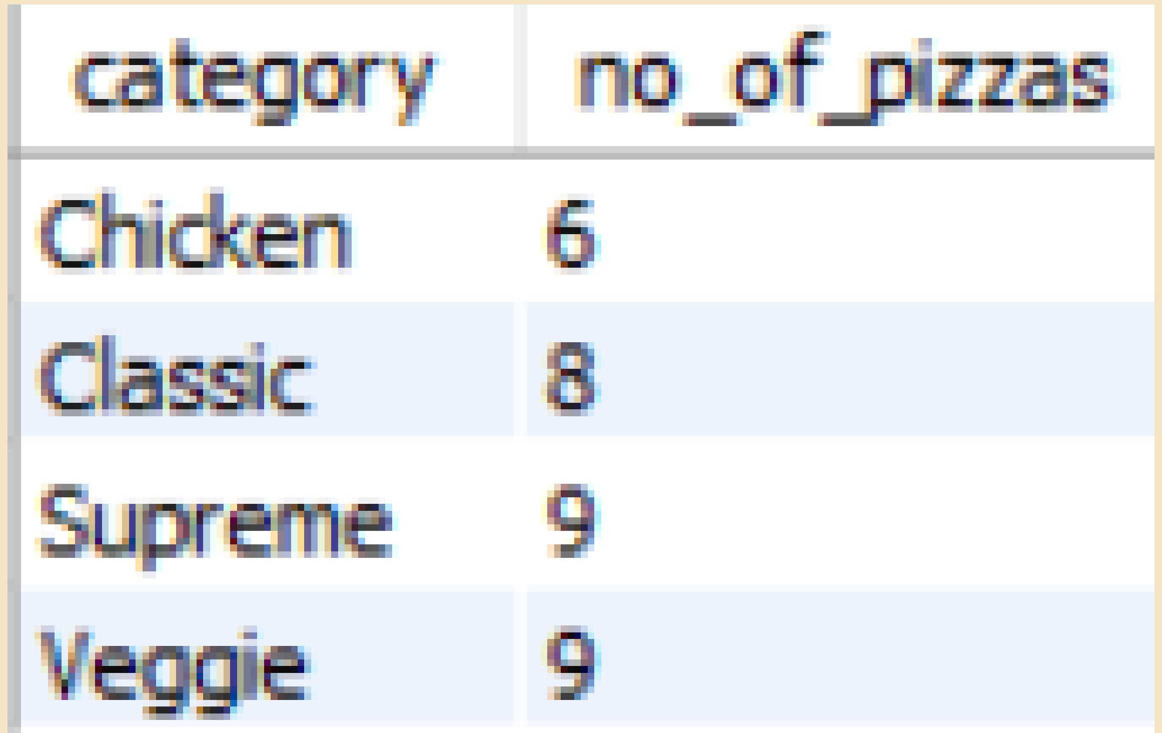
RESULT

JOIN THE RELEVANT TABLES TO FIND THE CATAGORY WISE DISTRIBUTION OF PIZZAS



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

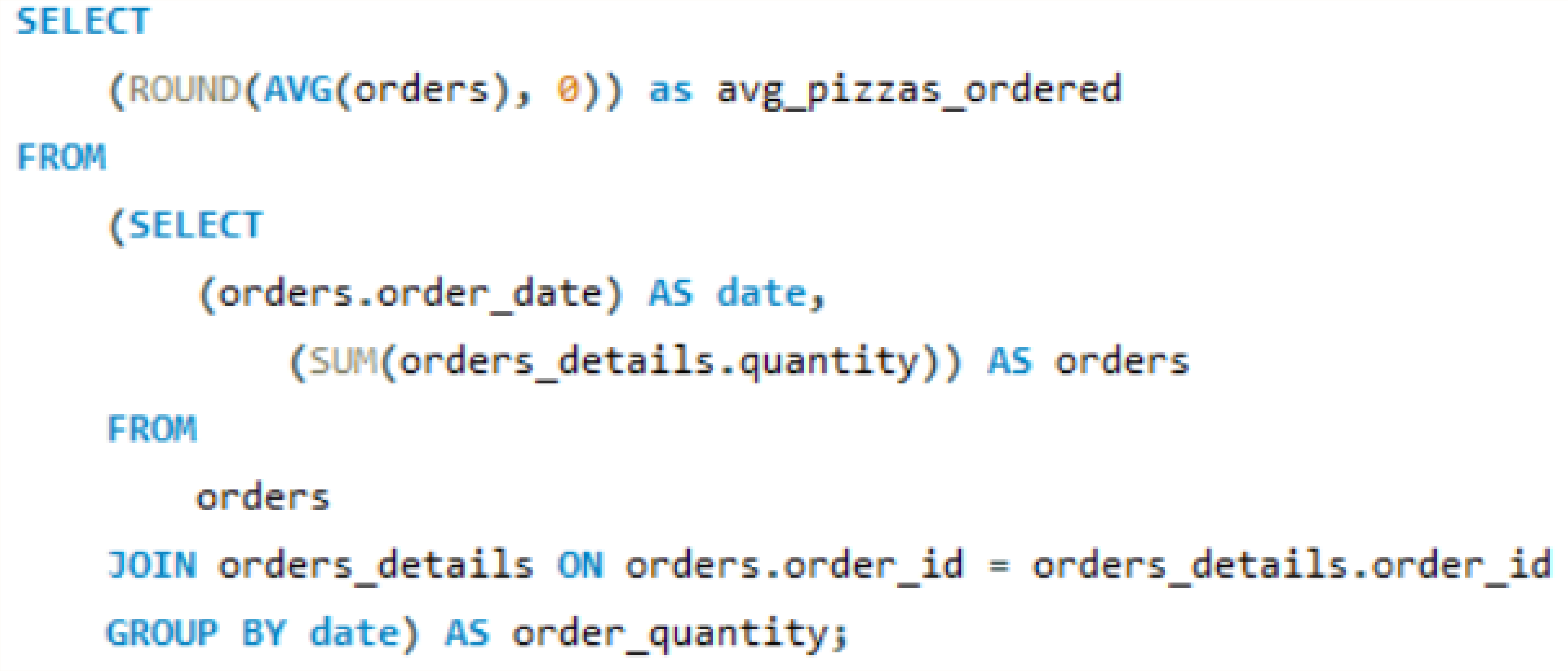
QUERY



category	no_of_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

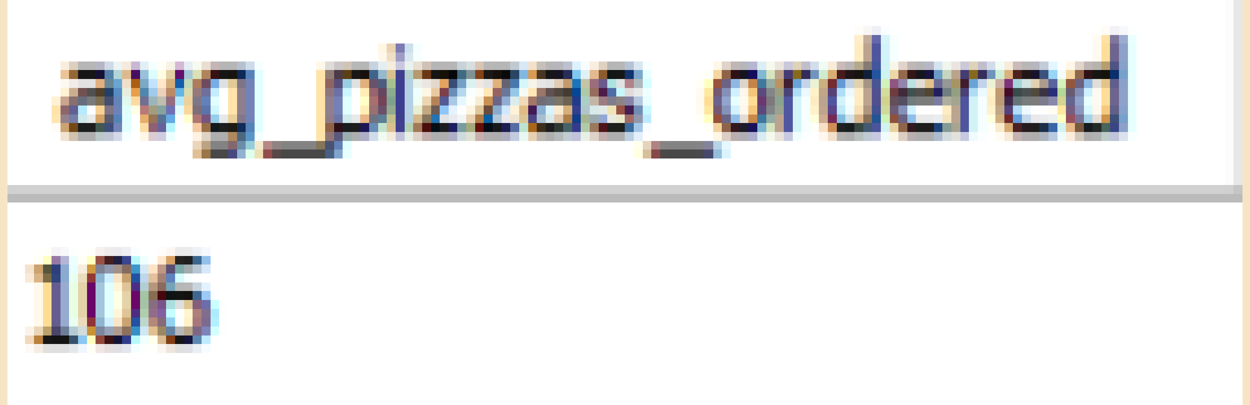
RESULT

GROUP THE ORDERS BY DATE AND CAL THE AVG NO. OF PIZZAS ORDERED PER DAY



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

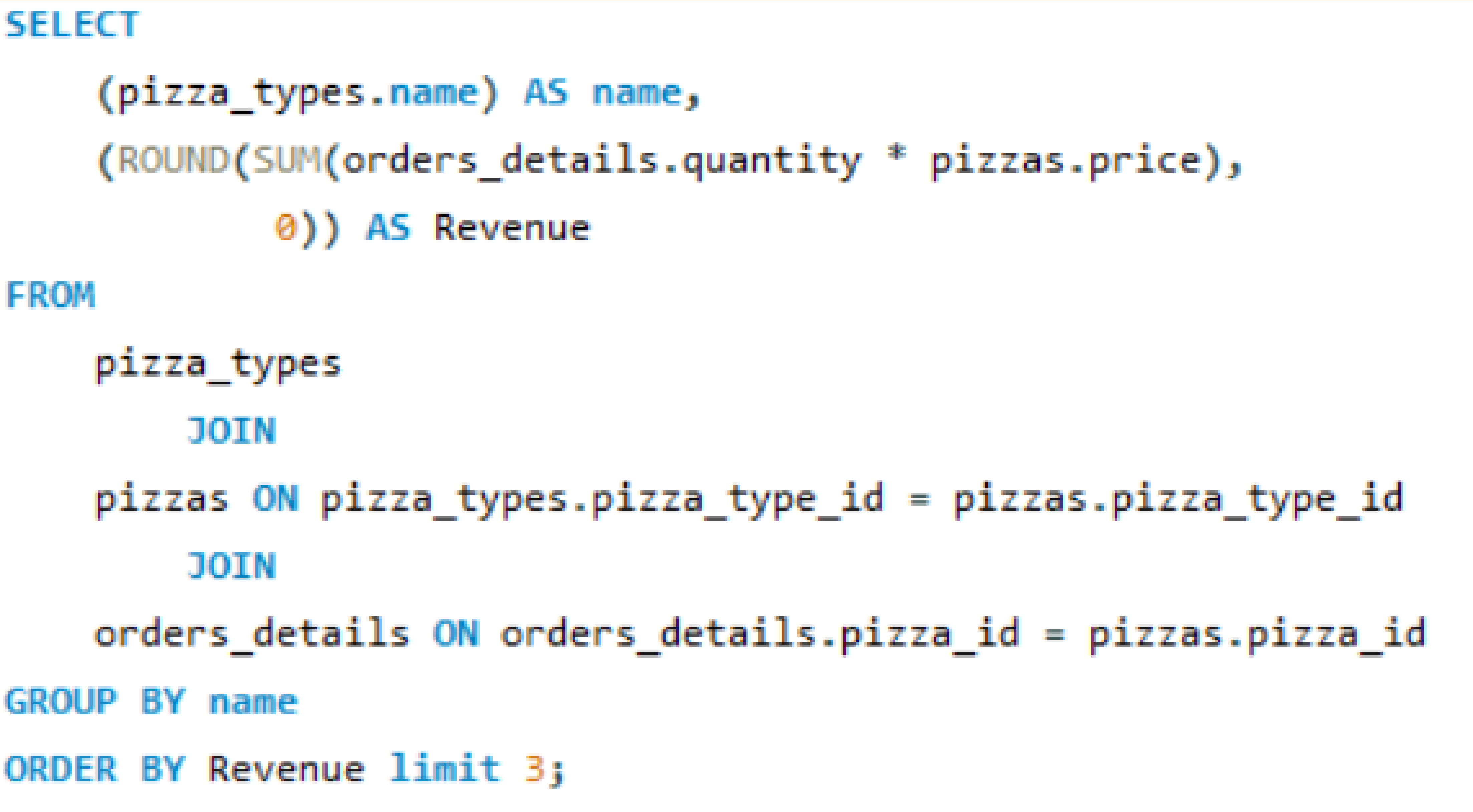
QUERY



```
avg_pizzas_ordered
106
```

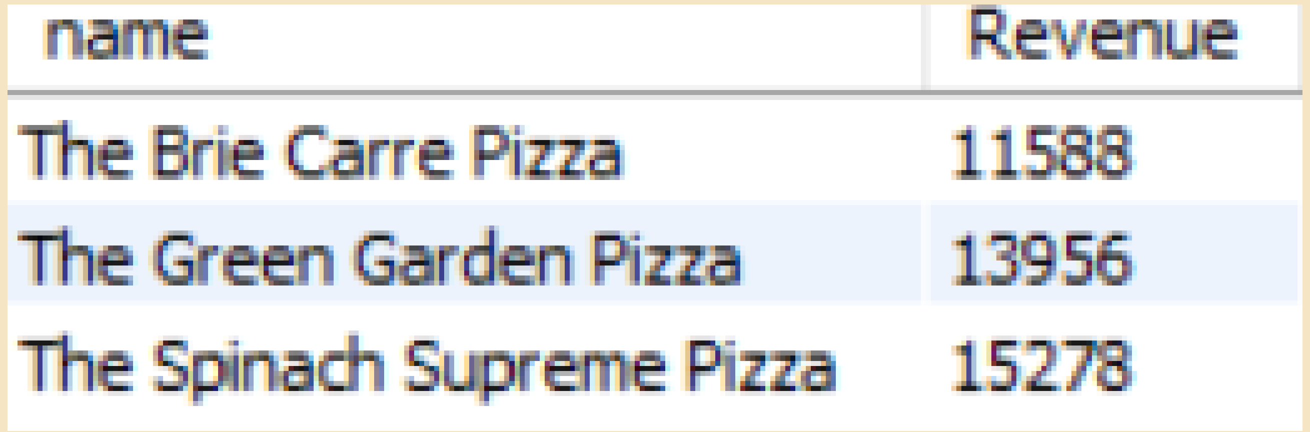
RESULT

DETERMINE THE TOP 3 PIZZAS BASED ON THE REVENUE



```
SELECT
    (pizza_types.name) AS name,
    (ROUND(SUM(orders_details.quantity * pizzas.price),
            0)) 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 name
ORDER BY Revenue limit 3;
```

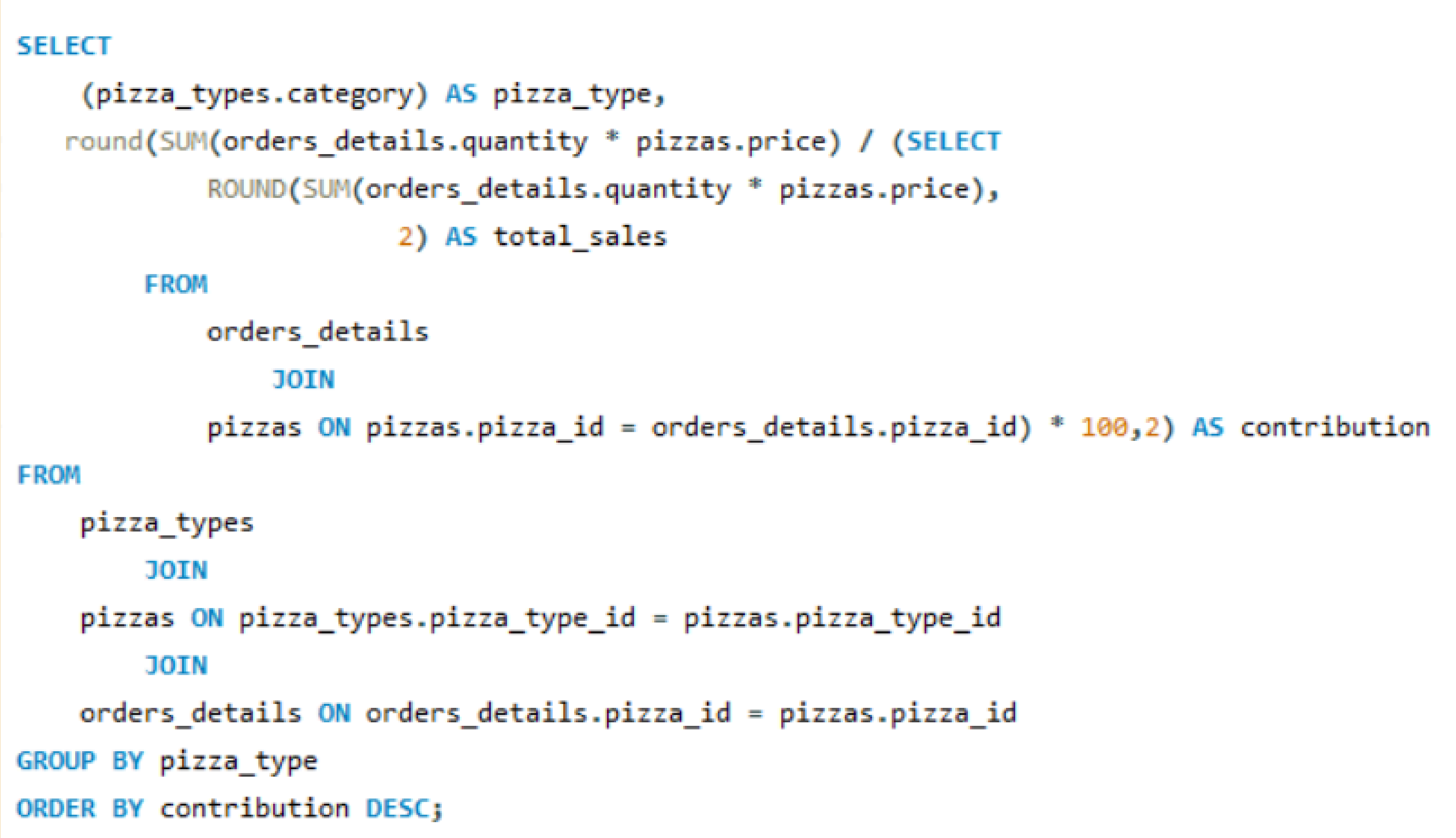
QUERY



name	Revenue
The Brie Carre Pizza	11588
The Green Garden Pizza	13956
The Spinach Supreme Pizza	15278

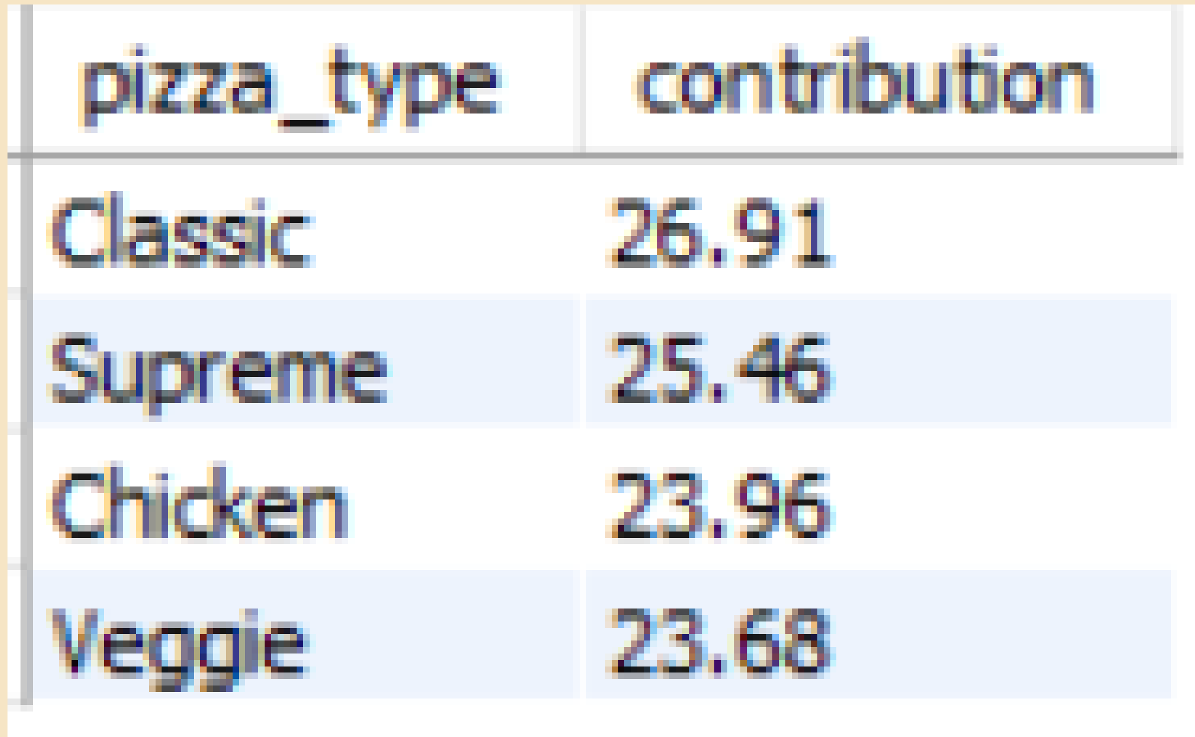
RESULT

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE



```sql
SELECT
    (pizza_types.category) AS pizza_type,
   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 contribution
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_type
ORDER BY contribution DESC;
```

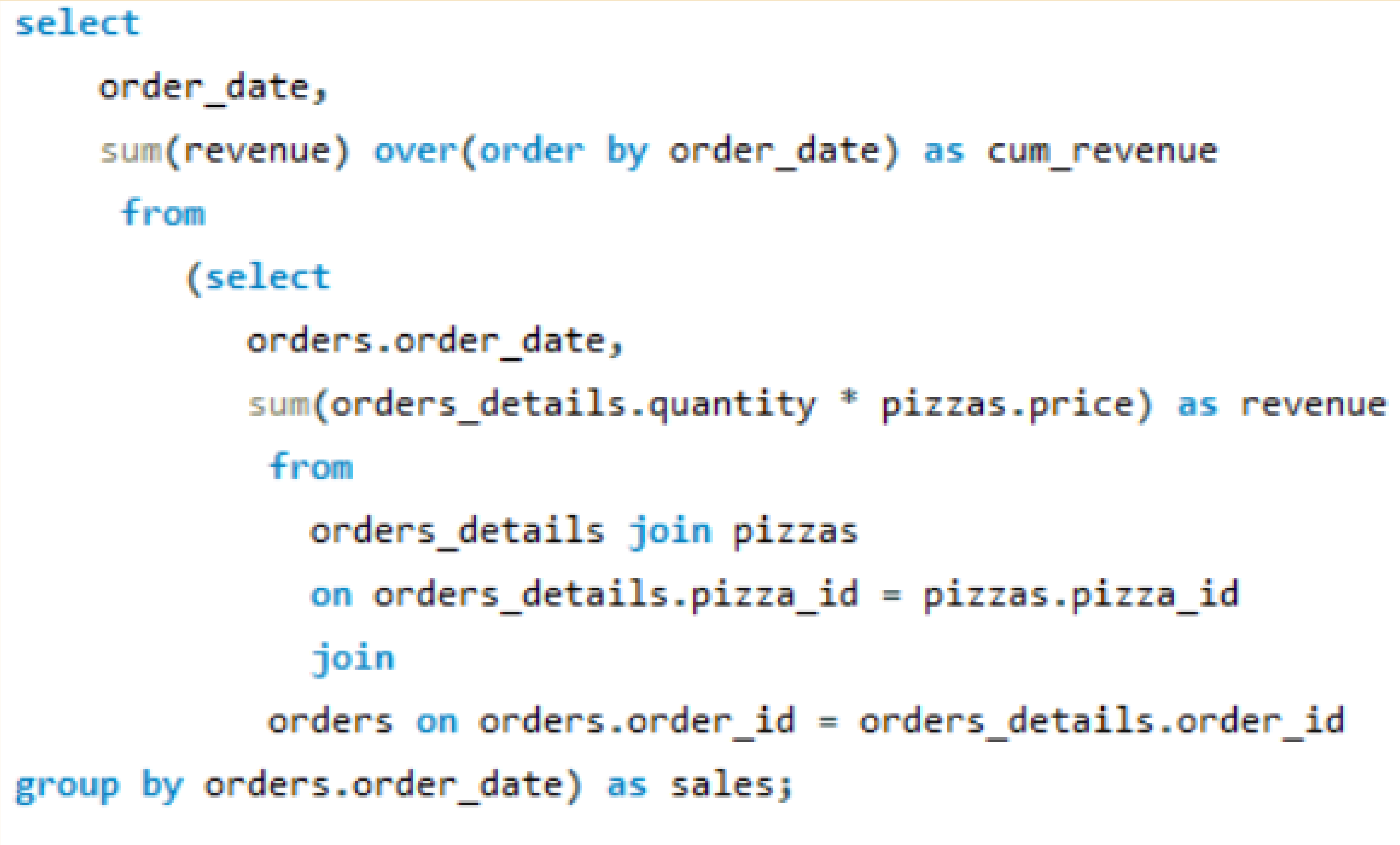
QUERY



pizza_type	contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

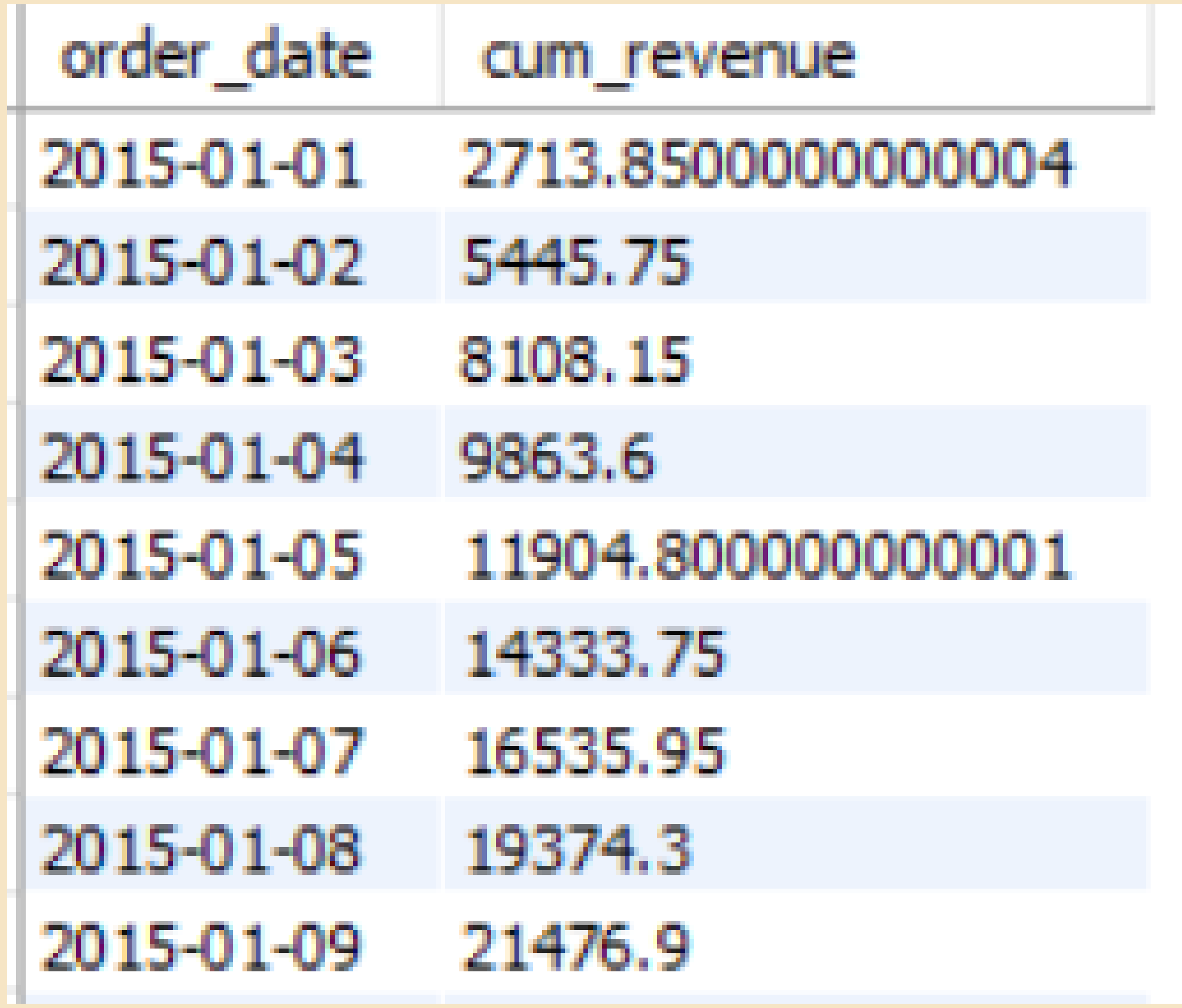
RESULT

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(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;
```

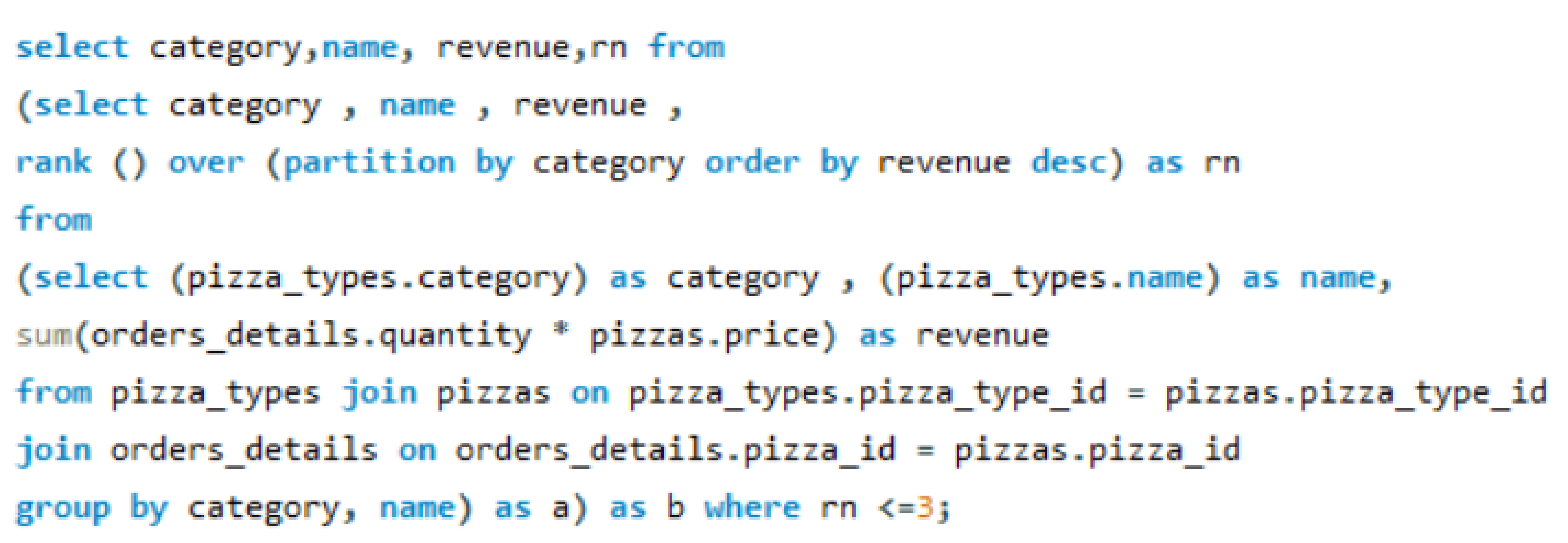
QUERY



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	11904.80000000001
2015-01-06	14333.75
2015-01-07	16535.95
2015-01-08	19374.3
2015-01-09	21476.9

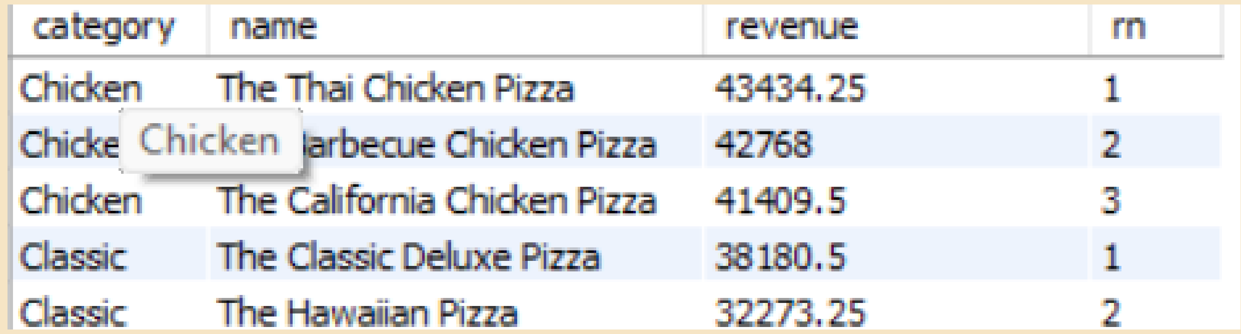
RESULT

DETERMINE THE TOP 3 MOST ORDERED PIZZAS BASED ON REVENUE FOR EACH PIZZA CATEGORY



```
select category,name, revenue,rn from
(select category , name , revenue ,
rank () over (partition by category order by revenue desc) as rn
from
(select (pizza_types.category) as category , (pizza_types.name) as 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 category, name) as a) as b where rn <=3;
```

QUERY



category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicke Chicken	arbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2

RESULT

THANK YOU :3

Instagram - https://www.instagram.com/shim_kwai?igsh=NjFqdGQzbWlpZTgx

GitHub - https://github.com/Tusshar-web

linkedin - https://www.linkedin.com/in/tusshar-singh-a4561b29a?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app