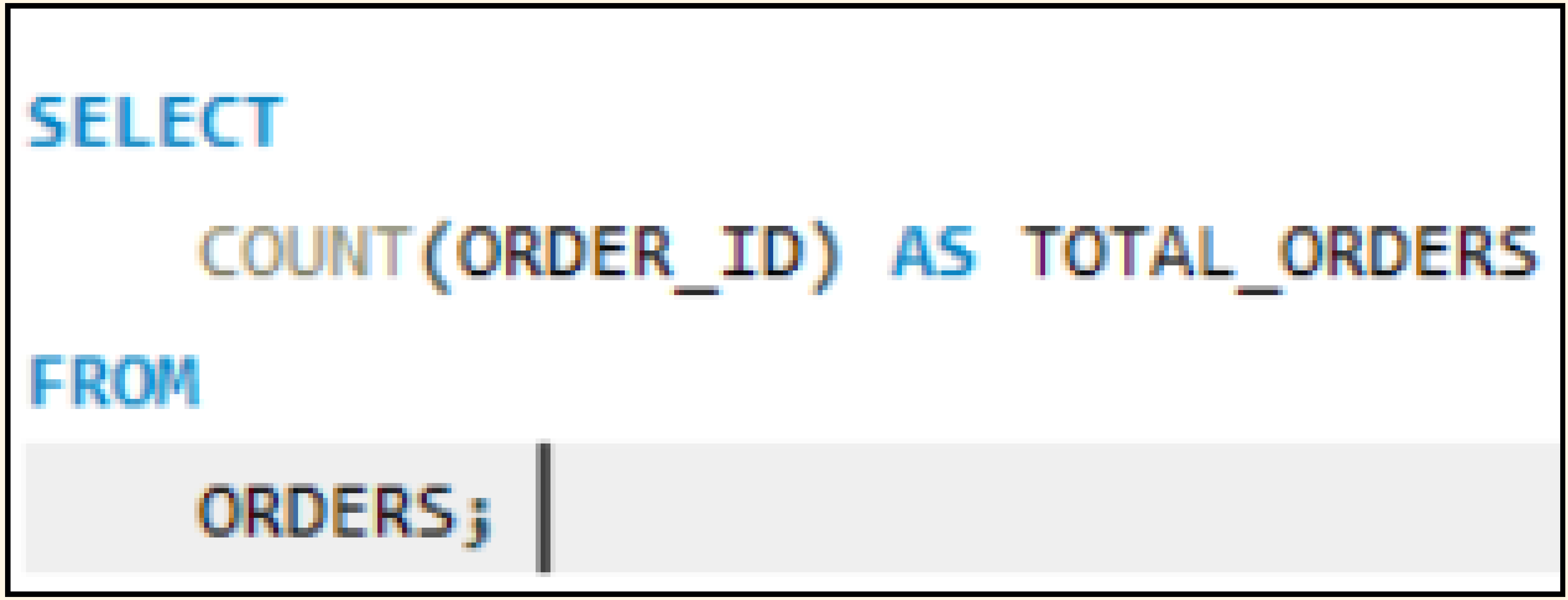
SQL PROJECT

ABOUT THE PROJECT & DATABASE

- This SQL project revolves around a comprehensive analysis of a pizza sales database.
- Initially, data is sourced from a CSV file and imported into MySQL server.
- The data includes pizza types, orders, prices, and quantities .
- It shows utilization of SQL joins and subqueries to extract meaningful insights from the database.

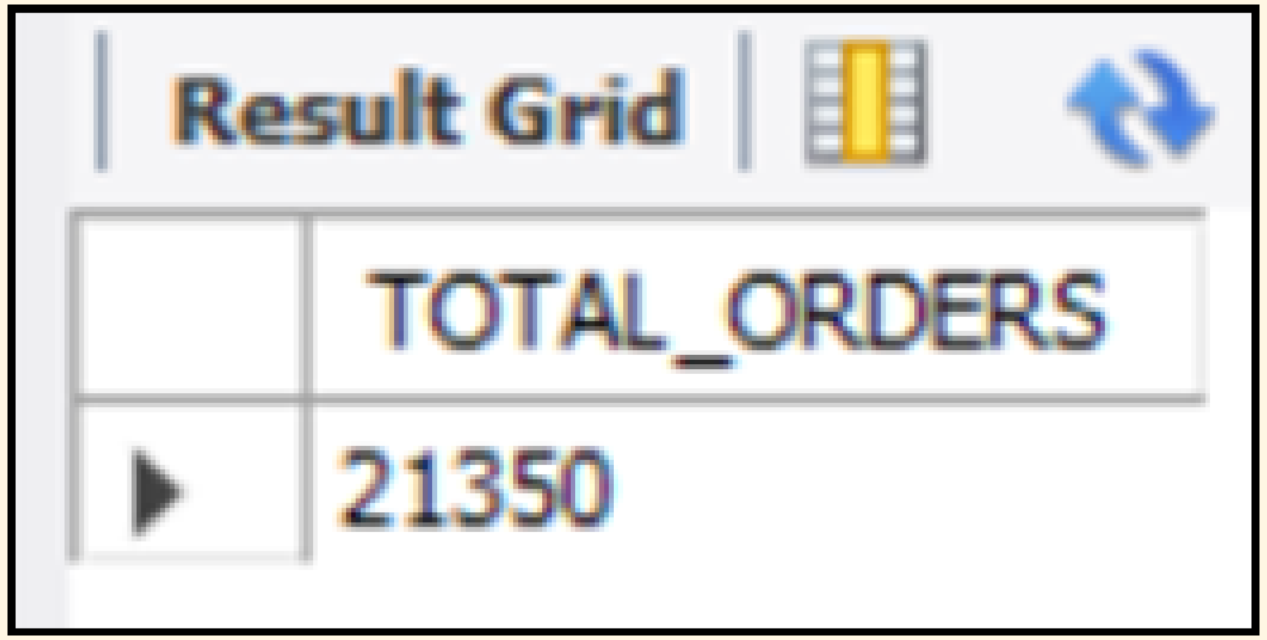
1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
SELECT
    COUNT(ORDER_ID) AS TOTAL_ORDERS
FROM
    ORDERS;
```

QUERY

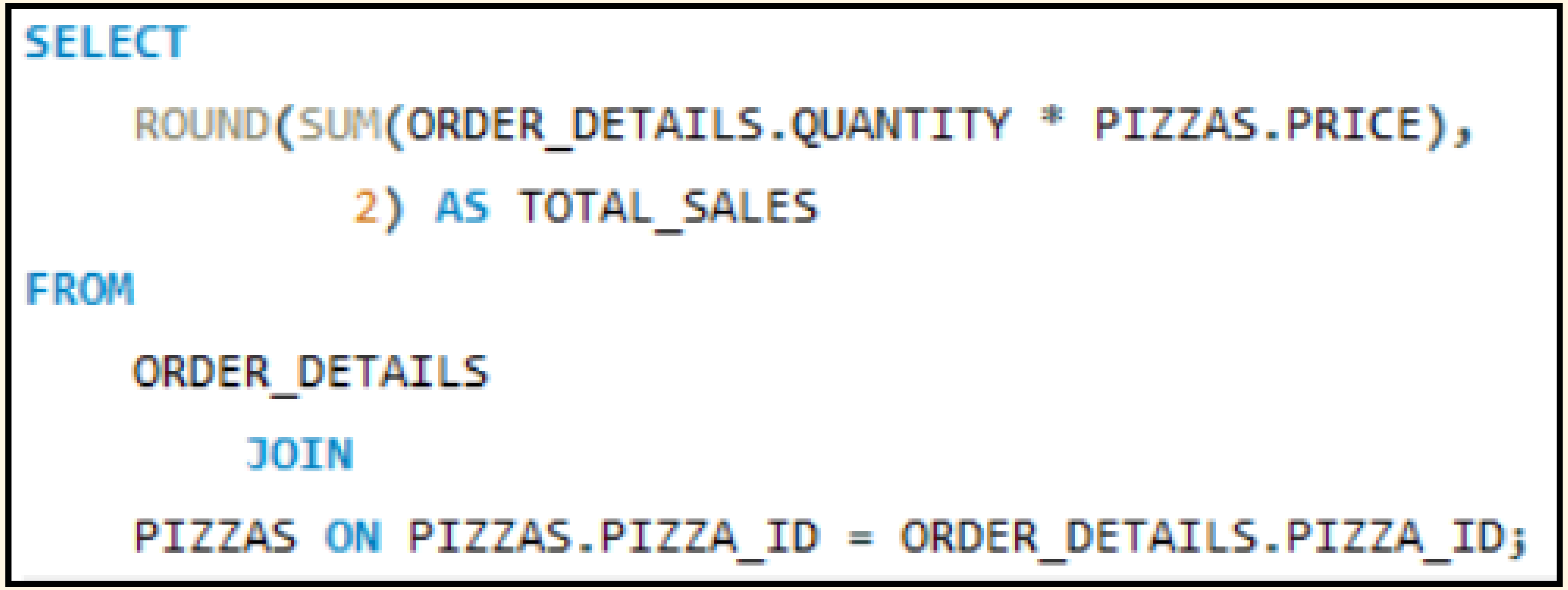
OUTPUT



	TOTAL_ORDERS
▶	21350

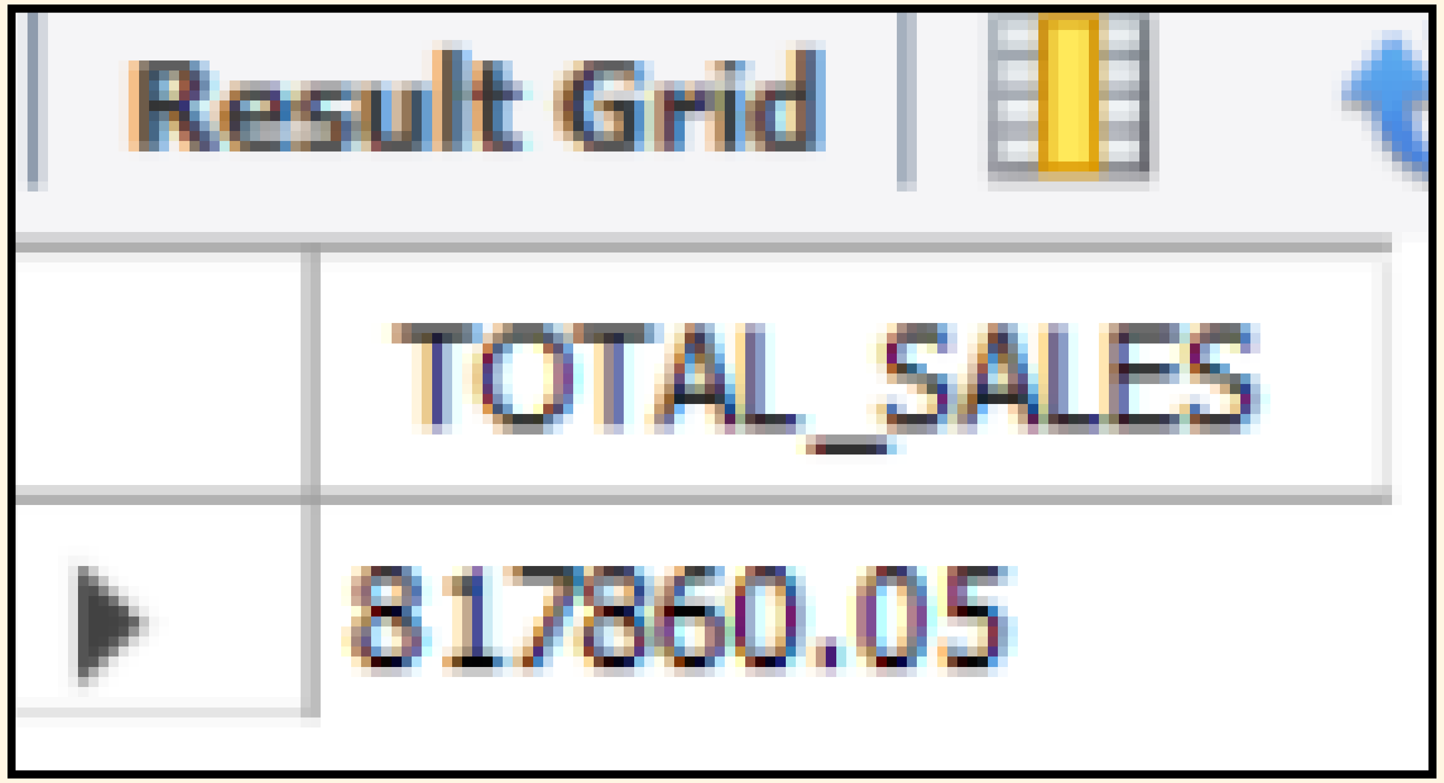
2 . CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES

QUERY



```
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;
```

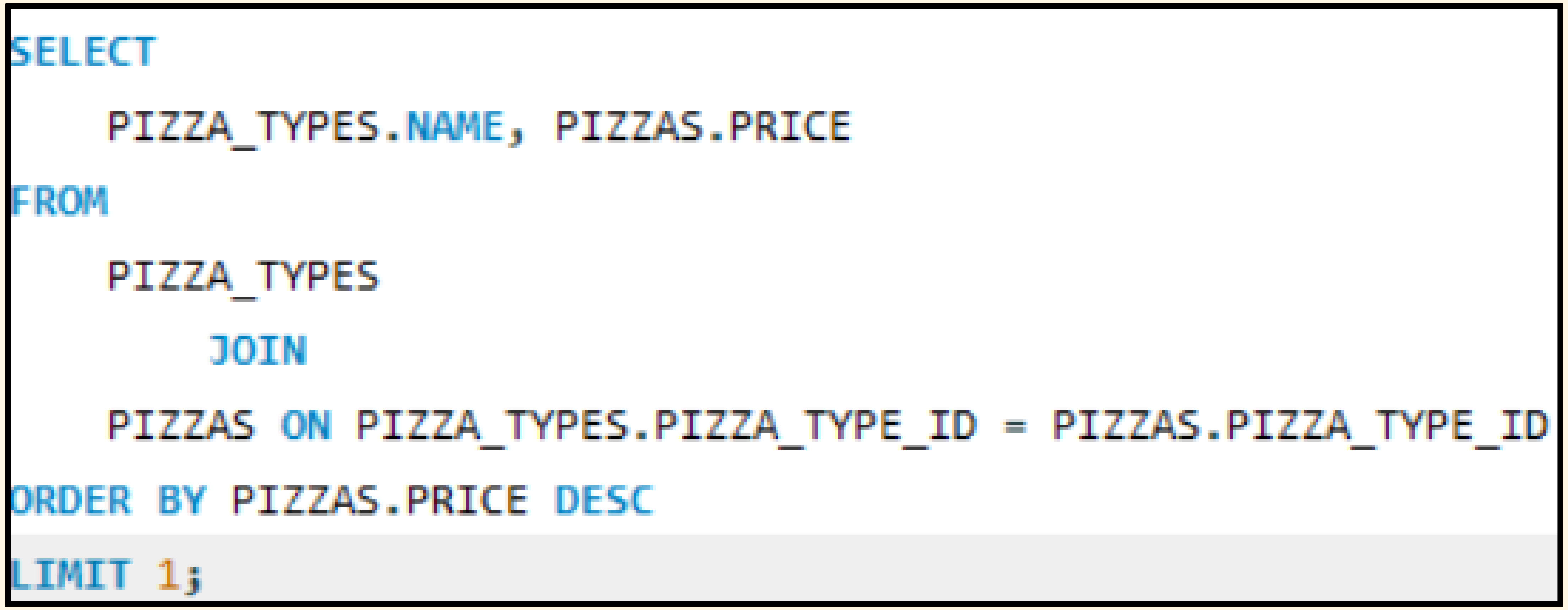
OUTPUT



```
TOTAL_SALES
817860.05
```

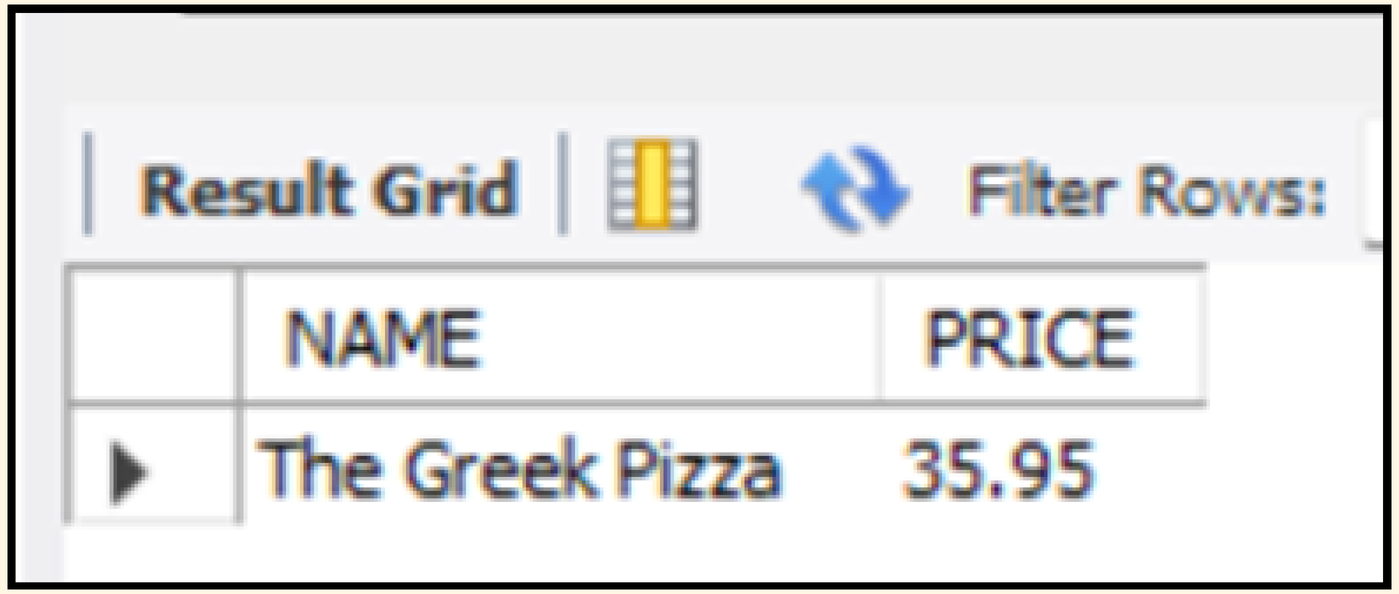
3 . IDENTIFY THE HIGHEST PRICED PIZZA

QUERY



```
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;
```

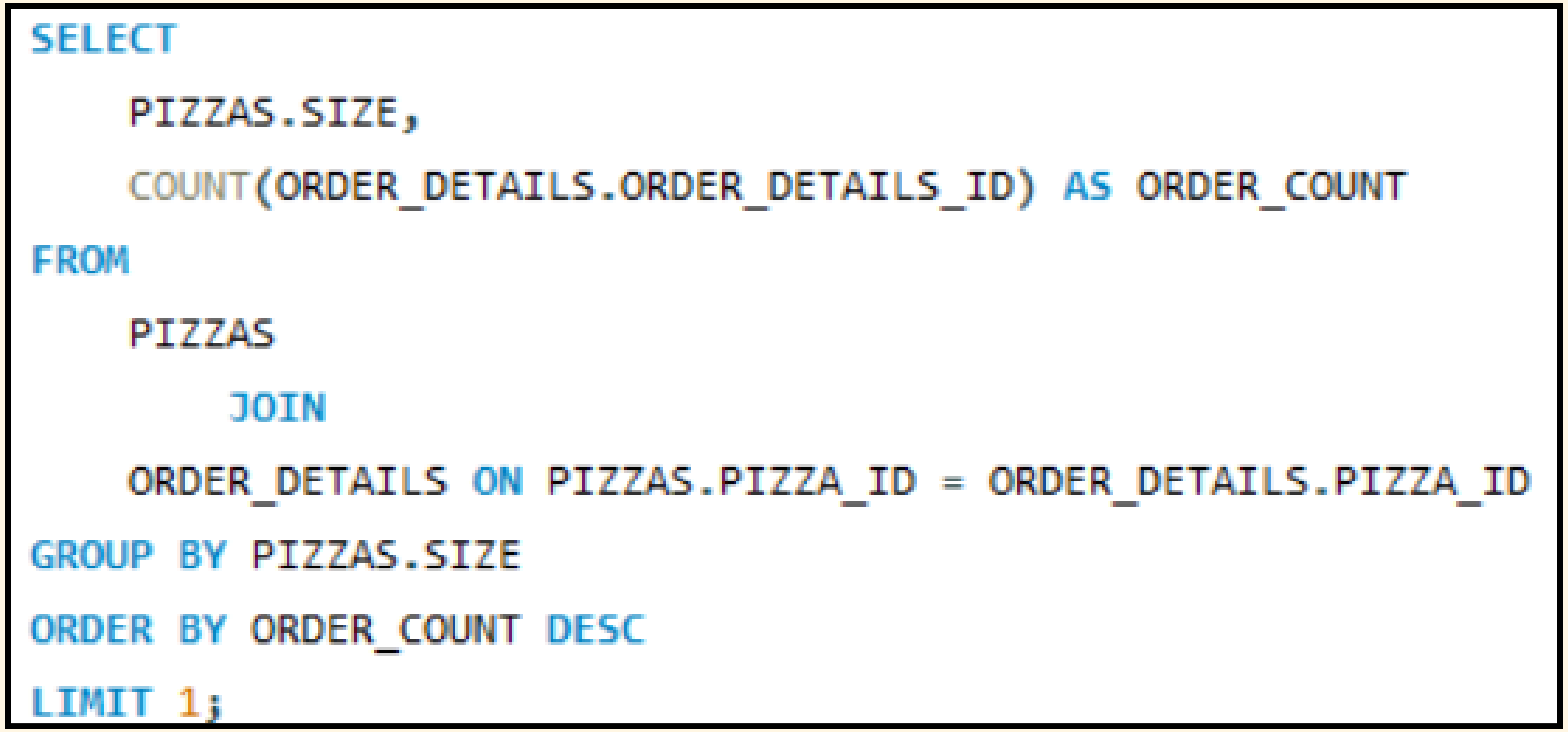
OUTPUT



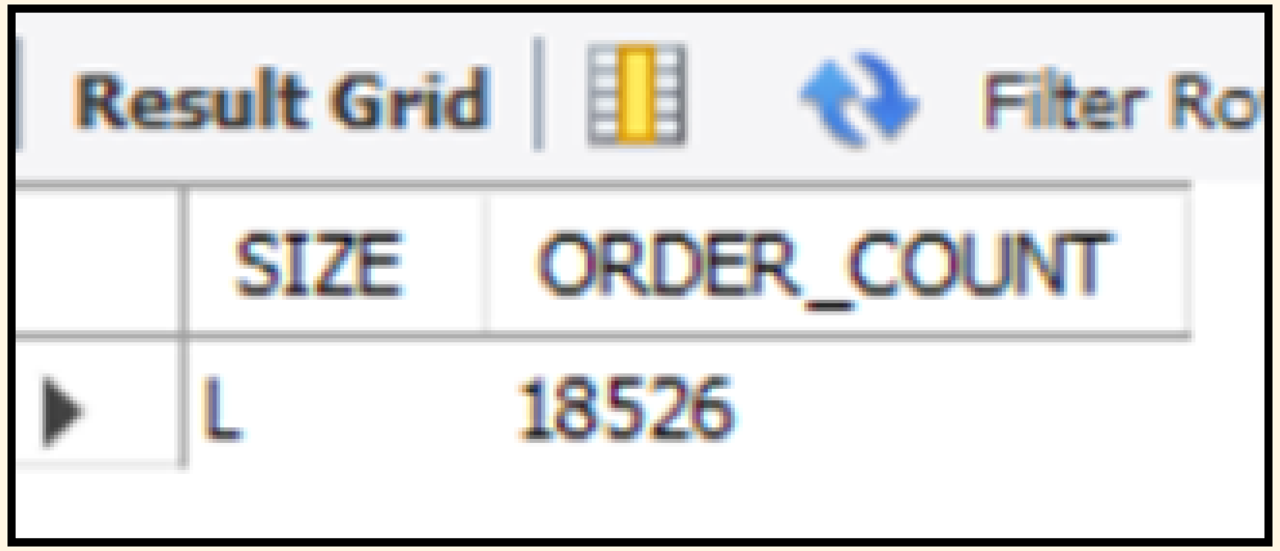
NAME	PRICE
The Greek Pizza	35.95

4 . IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

QUERY



```
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
LIMIT 1;
```

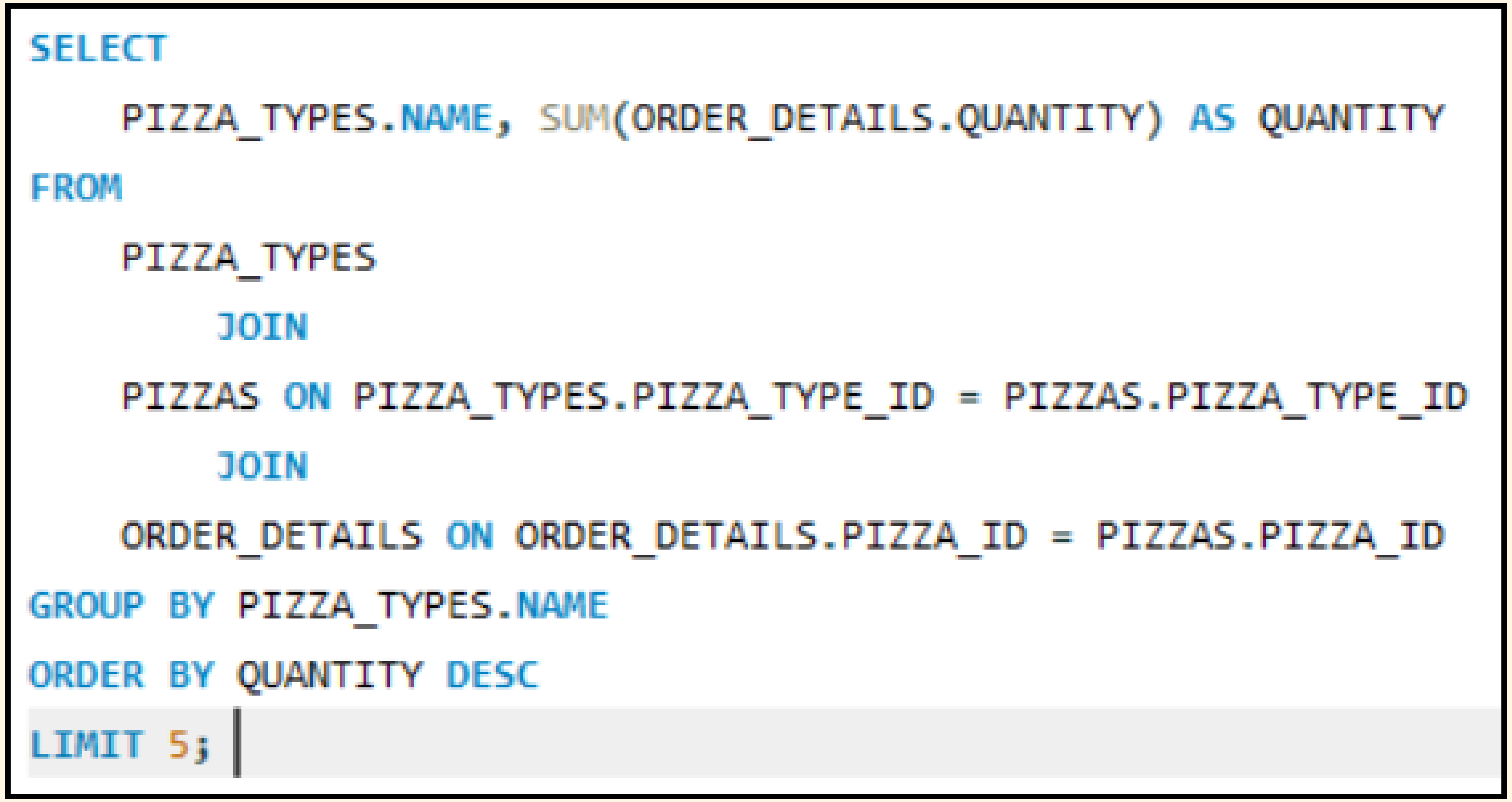


```
  SIZE   ORDER_COUNT
▶ L      18526
```

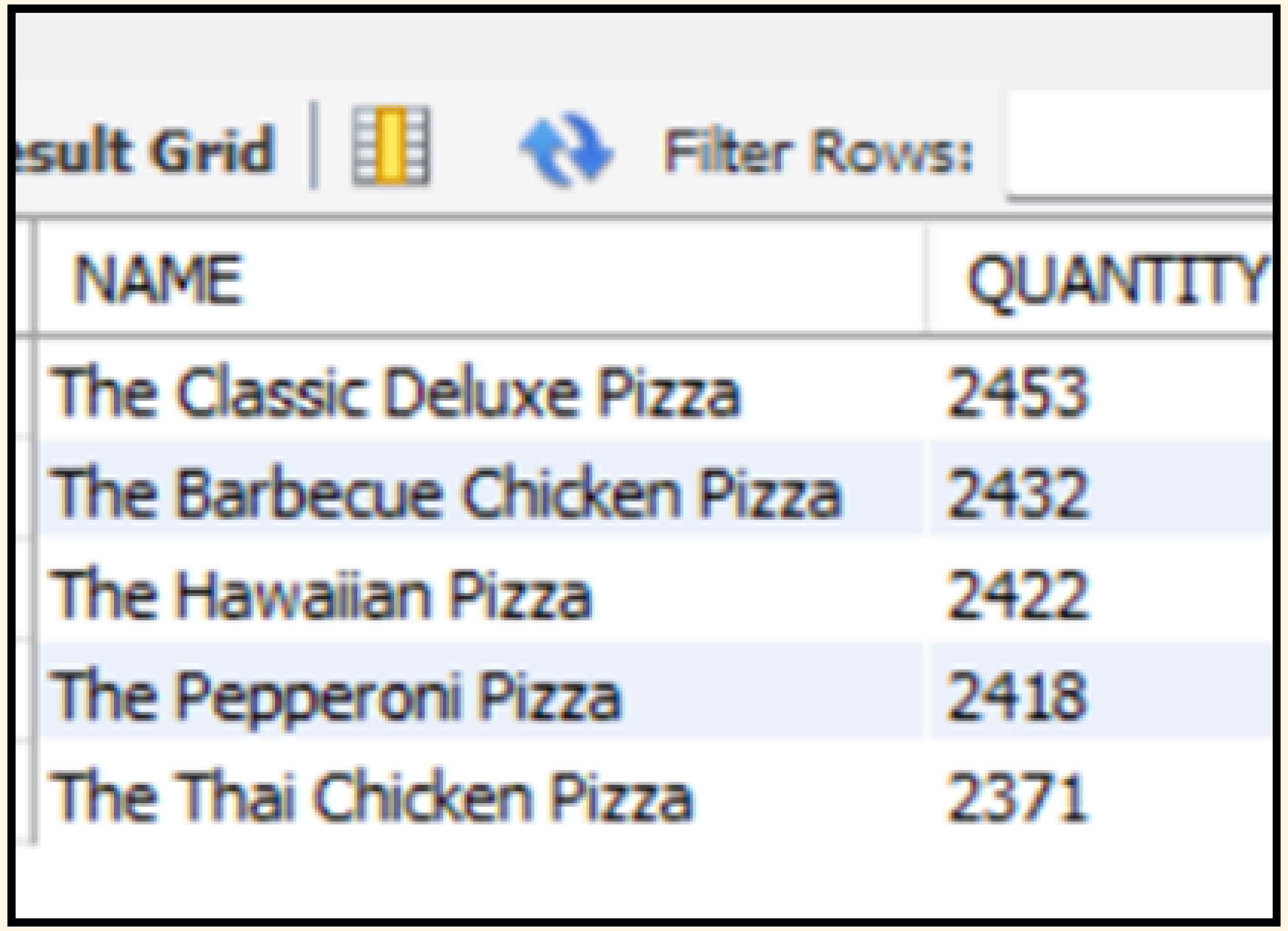
OUTPUT

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

QUERY



```
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;
```

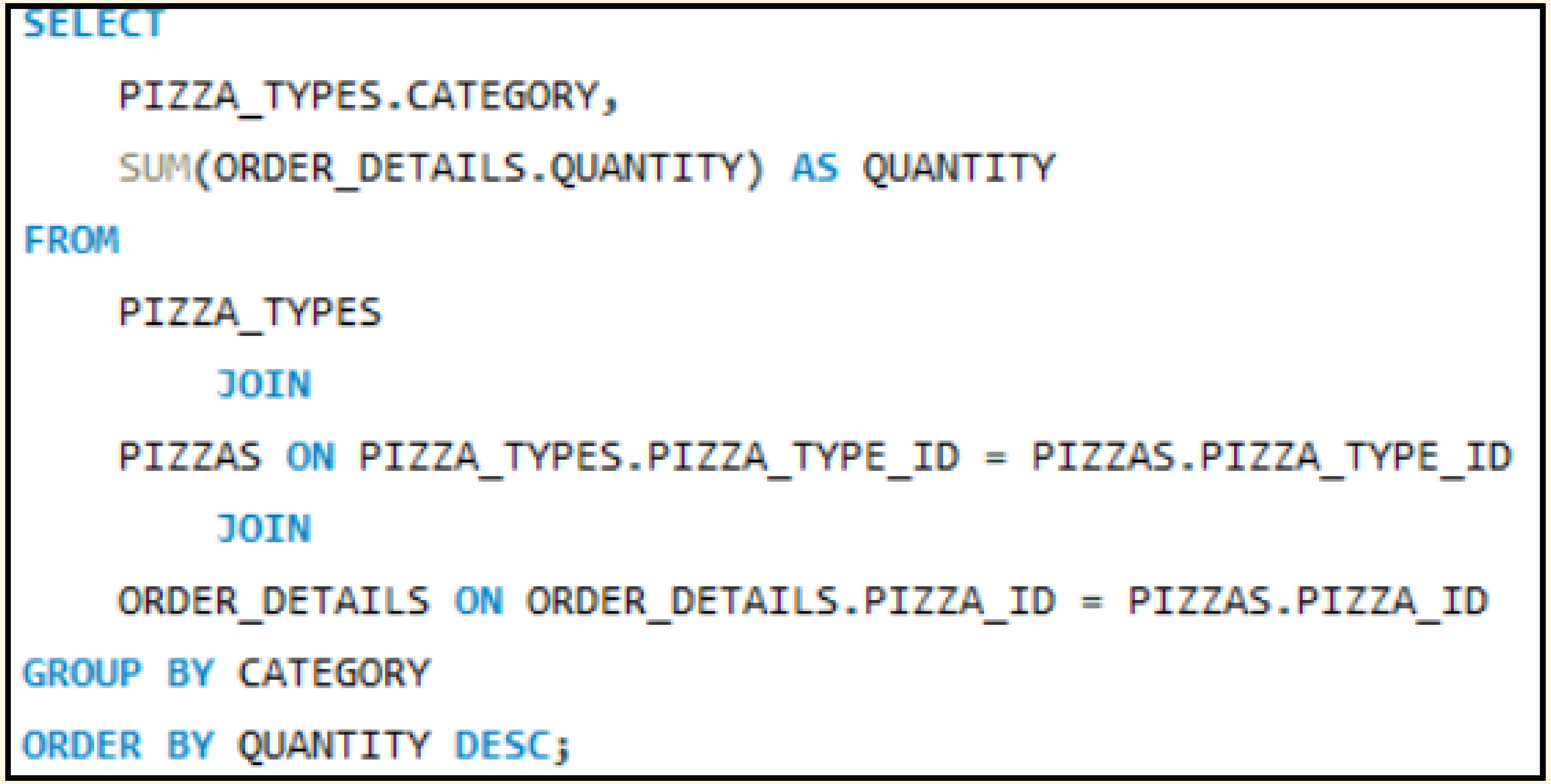


NAME	QUANTITY
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

OUTPUT

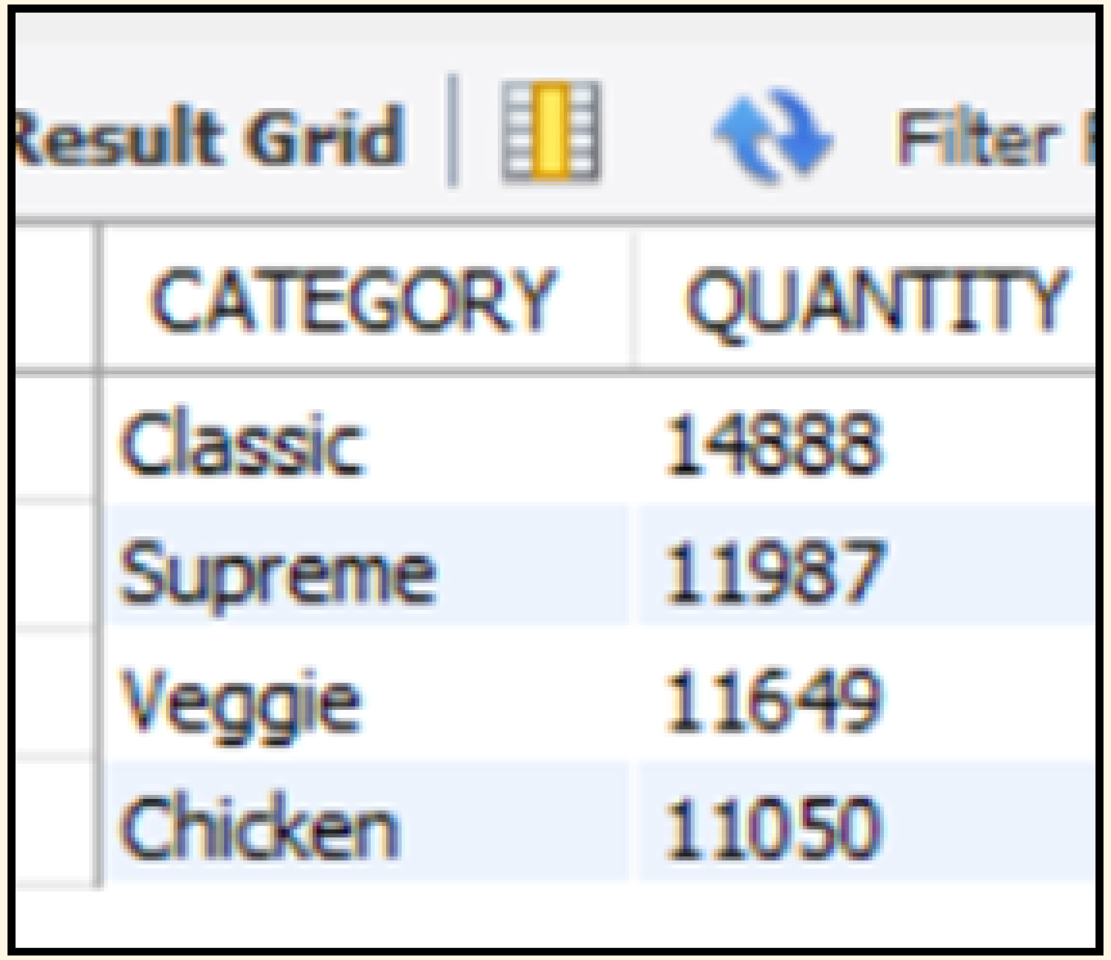
6 . JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

QUERY



```
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 CATEGORY
ORDER BY QUANTITY DESC;
```

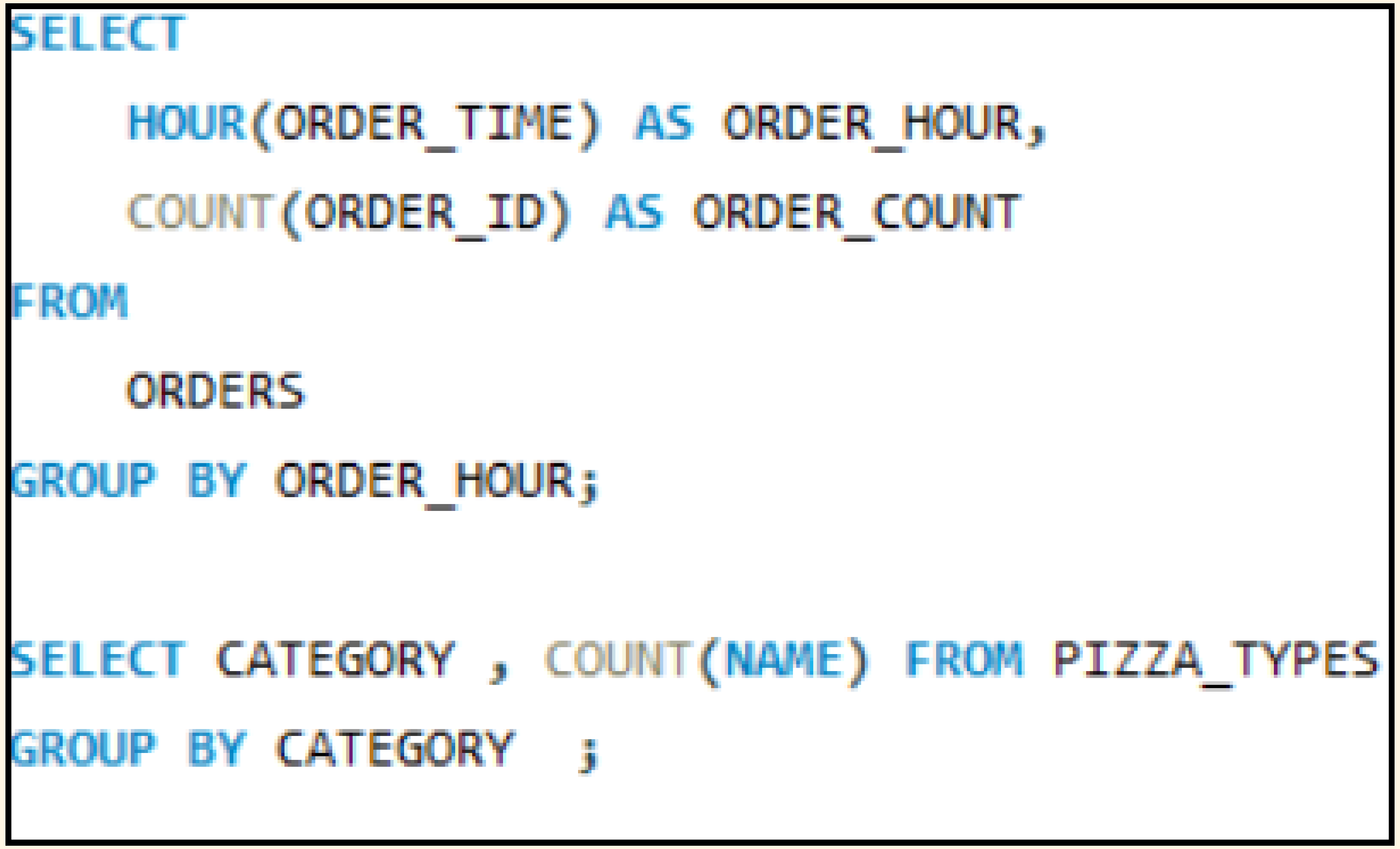
OUTPUT



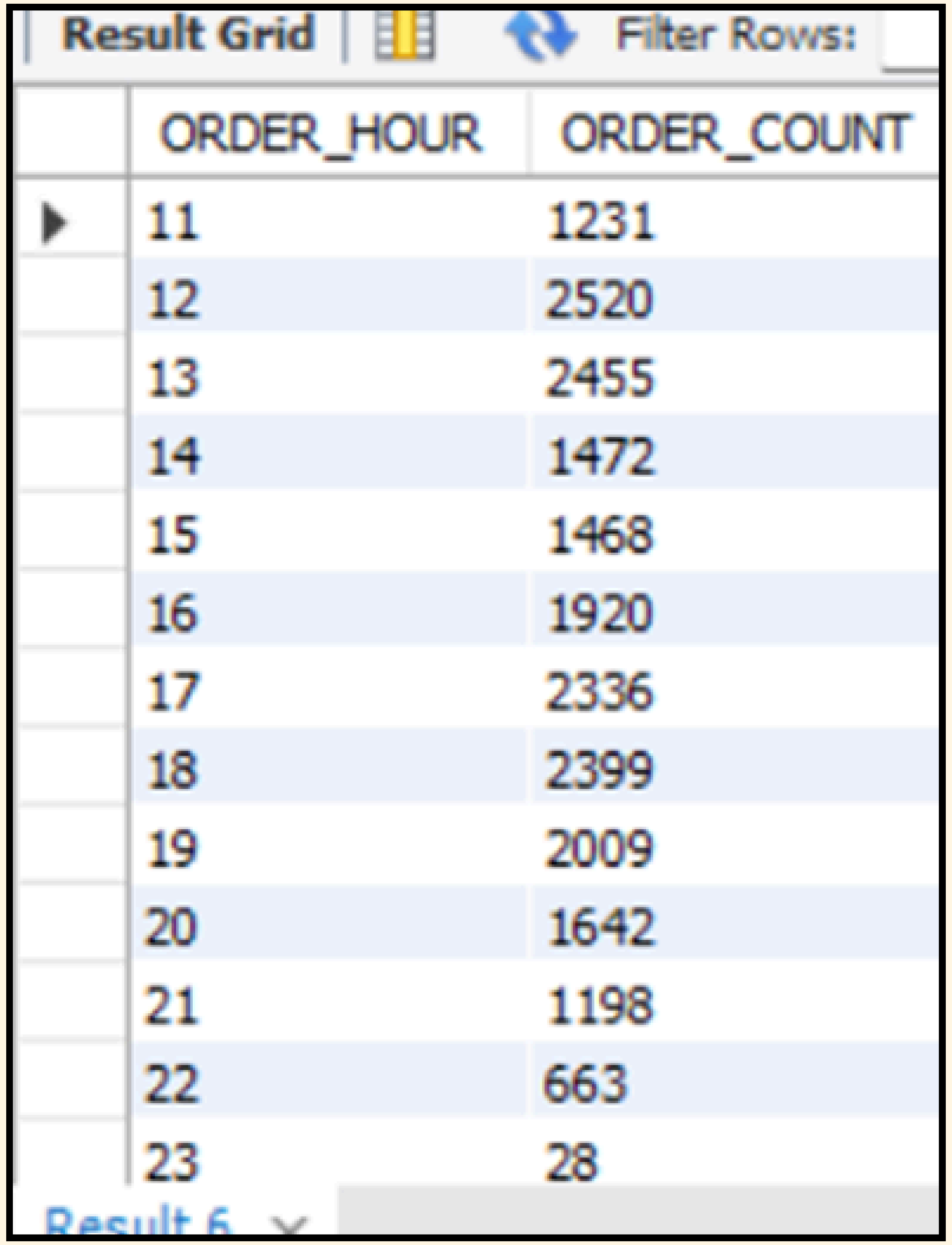
CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7 . DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

QUERY



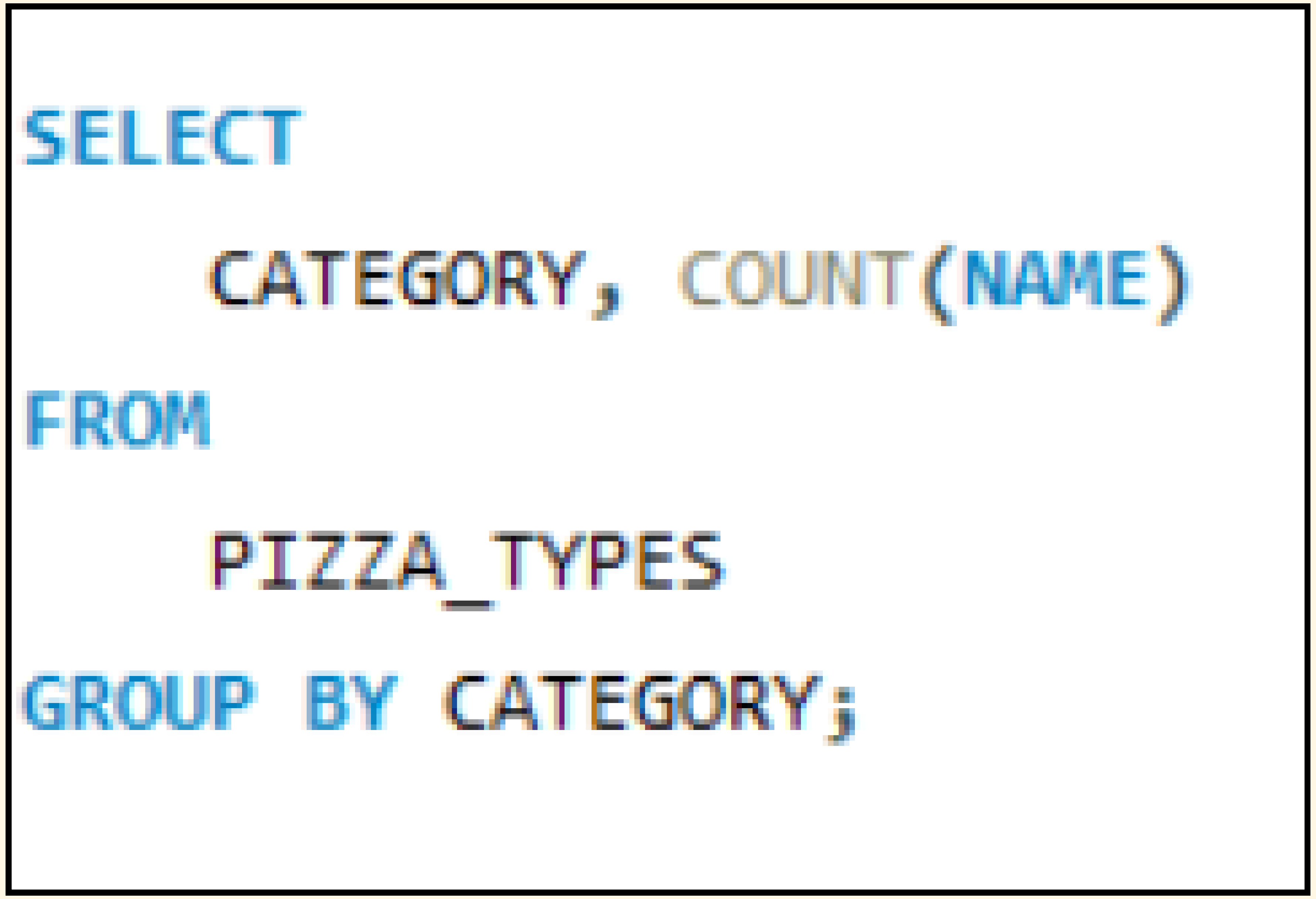
```
SELECT
    HOUR(ORDER_TIME) AS ORDER_HOUR,
    COUNT(ORDER_ID) AS ORDER_COUNT
FROM
    ORDERS
GROUP BY ORDER_HOUR;


SELECT CATEGORY , COUNT(NAME) FROM PIZZA_TYPES
GROUP BY CATEGORY  ;
```



ORDER_HOUR	ORDER_COUNT
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

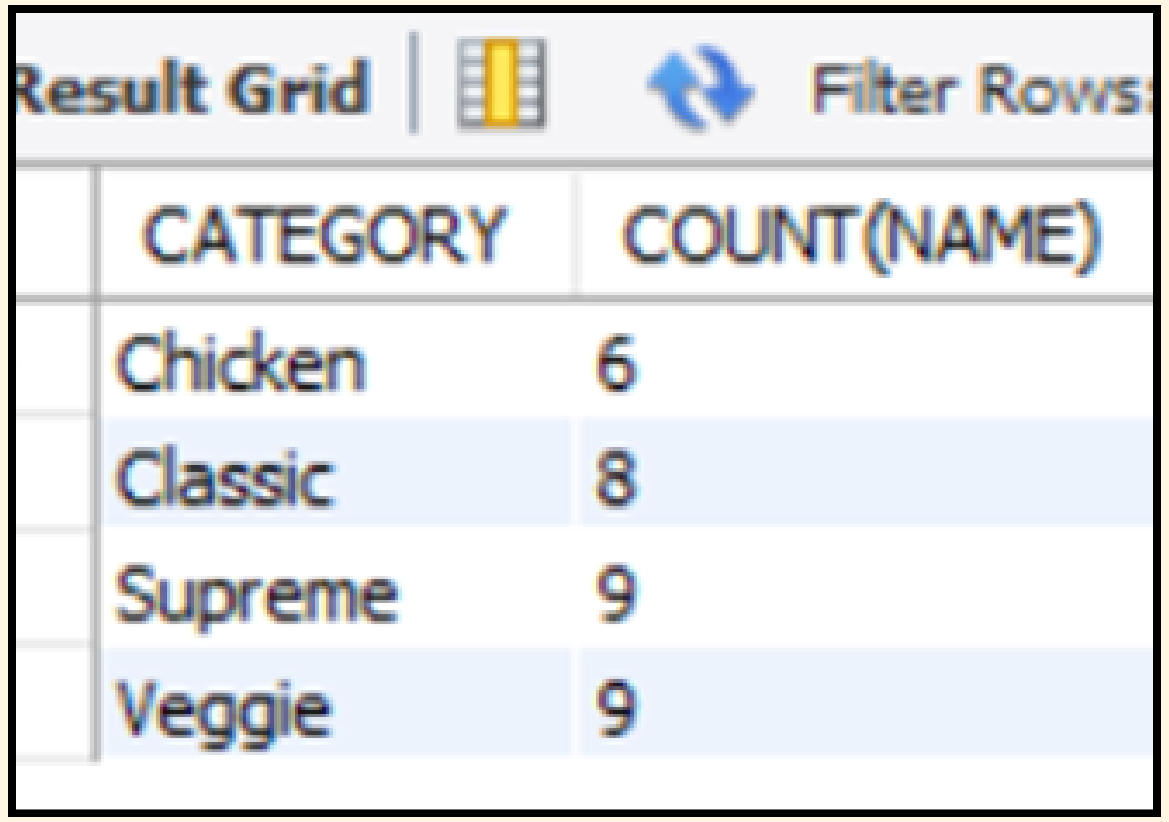
OUTPUT

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



```
SELECT
    CATEGORY, COUNT(NAME)
FROM
    PIZZA_TYPES
GROUP BY CATEGORY;
```

QUERY

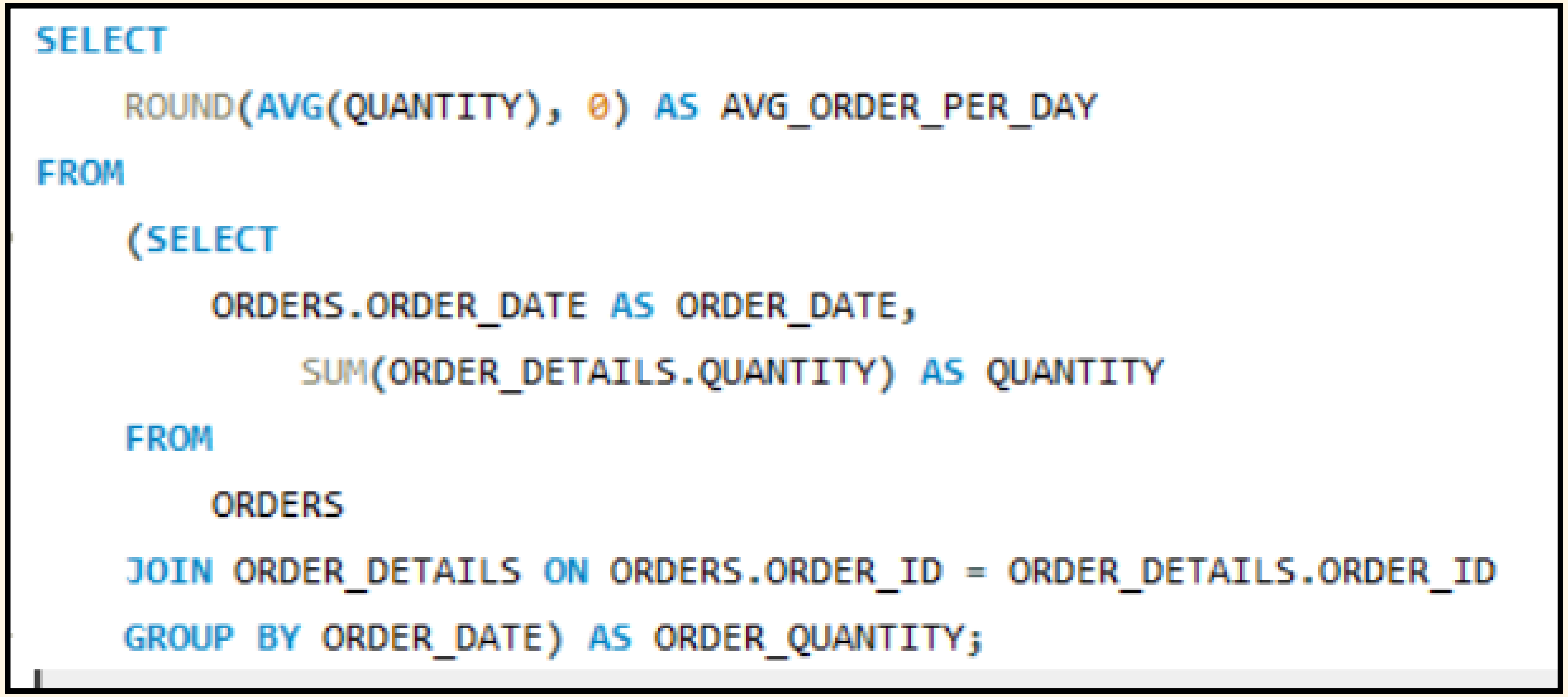


```
CATEGORY   COUNT(NAME)
Chicken    6
Classic    8
Supreme    9
Veggie     9
```

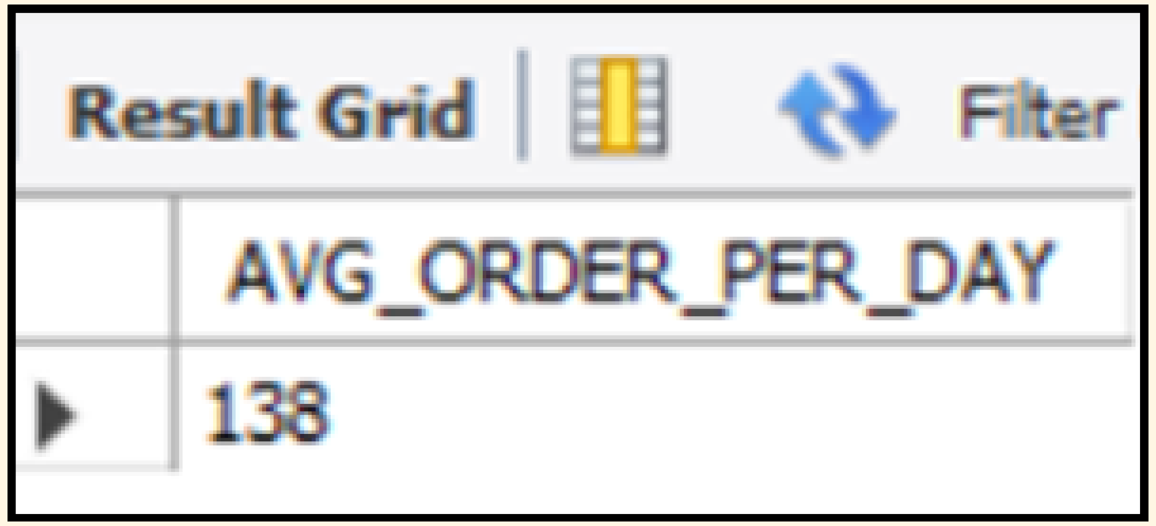
OUTPUT

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

QUERY



```
SELECT
    ROUND(AVG(QUANTITY), 0) AS AVG_ORDER_PER_DAY
FROM
    (SELECT
        ORDERS.ORDER_DATE AS ORDER_DATE,
            SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
    FROM
        ORDERS
    JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
    GROUP BY ORDER_DATE) AS ORDER_QUANTITY;
```

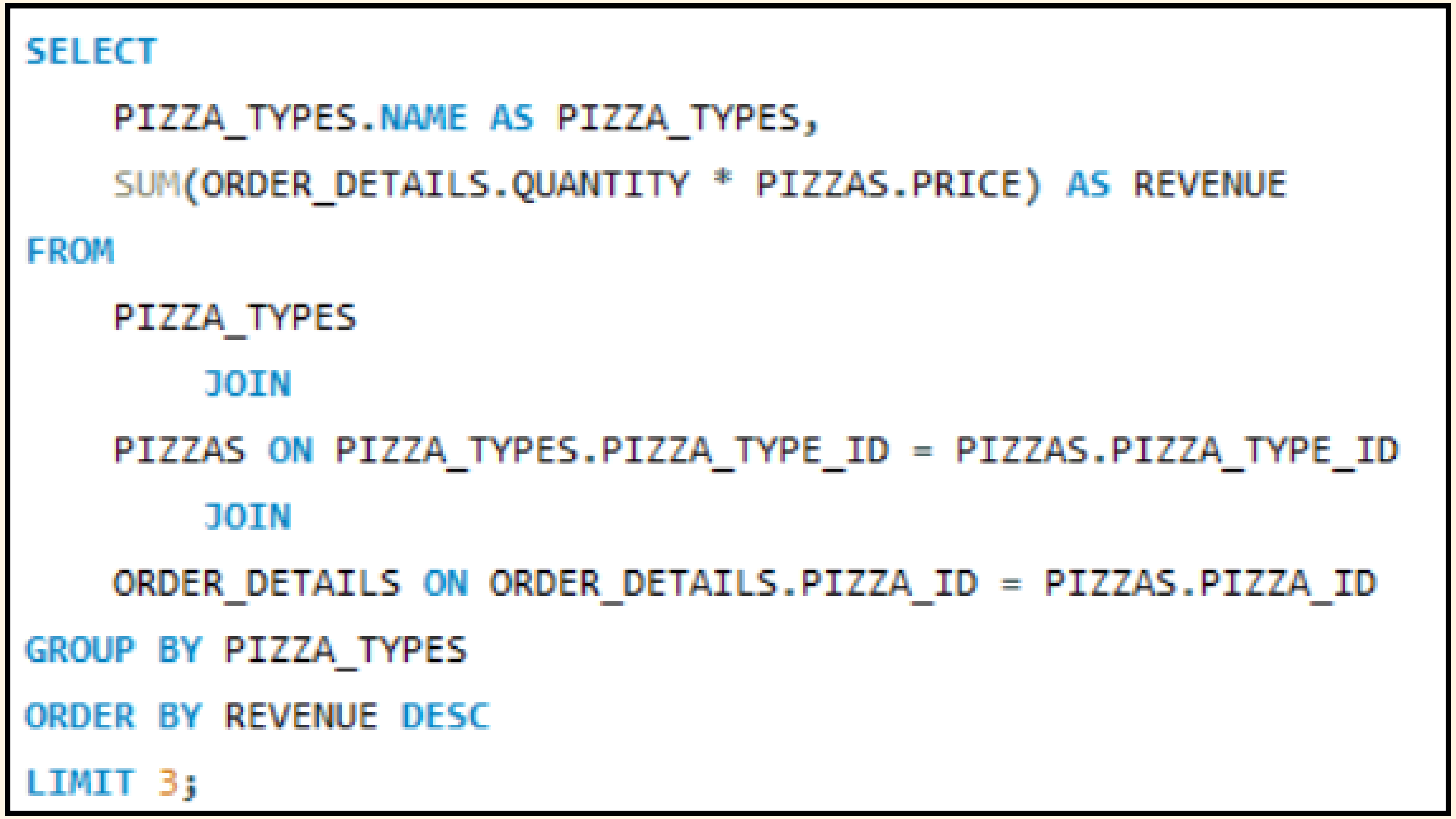


AVG_ORDER_PER_DAY
138

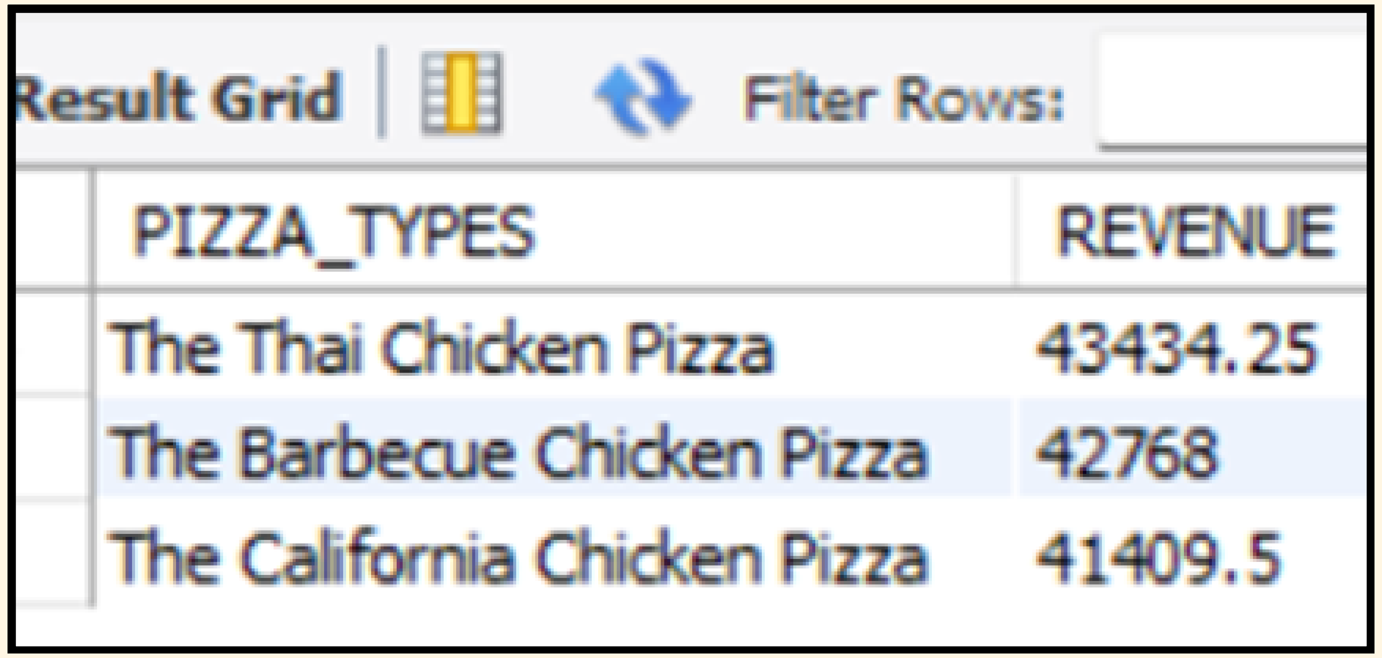
OUTPUT

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

QUERY



```
SELECT
    PIZZA_TYPES.NAME AS PIZZA_TYPES,
    SUM(ORDER_DETAILS.QUANTITY * PIZZAS.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
ORDER BY REVENUE DESC
LIMIT 3;
```

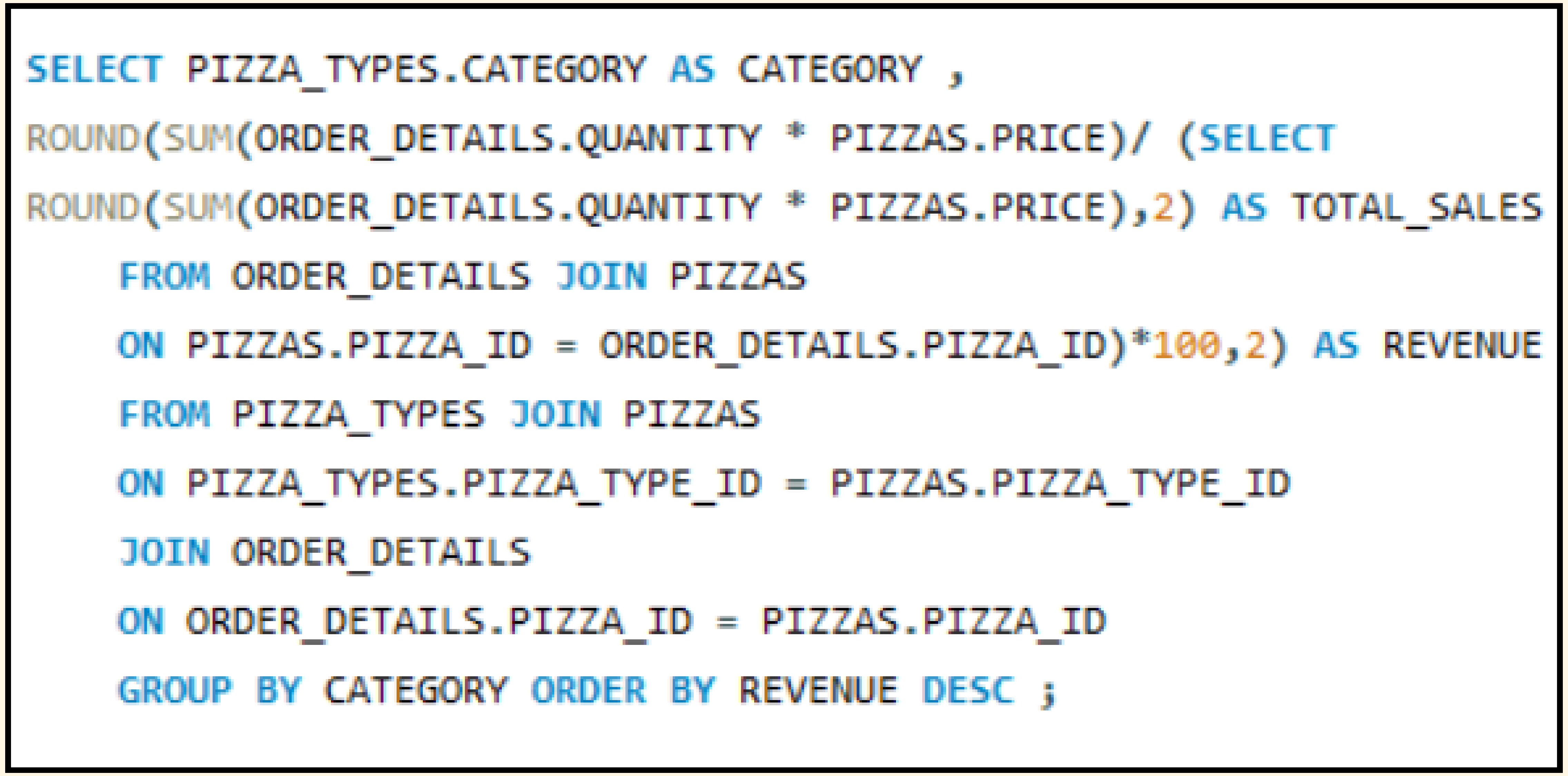


PIZZA_TYPES	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

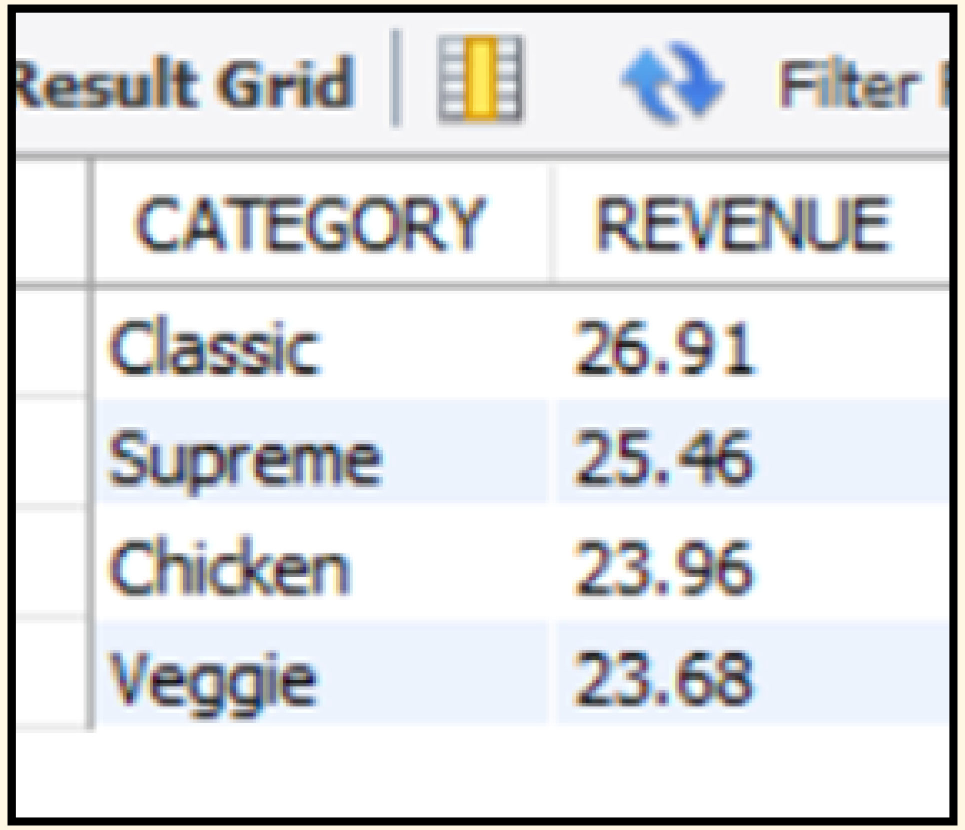
OUTPUT

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

QUERY



```
SELECT PIZZA_TYPES.CATEGORY AS 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_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 CATEGORY ORDER BY REVENUE DESC ;
```

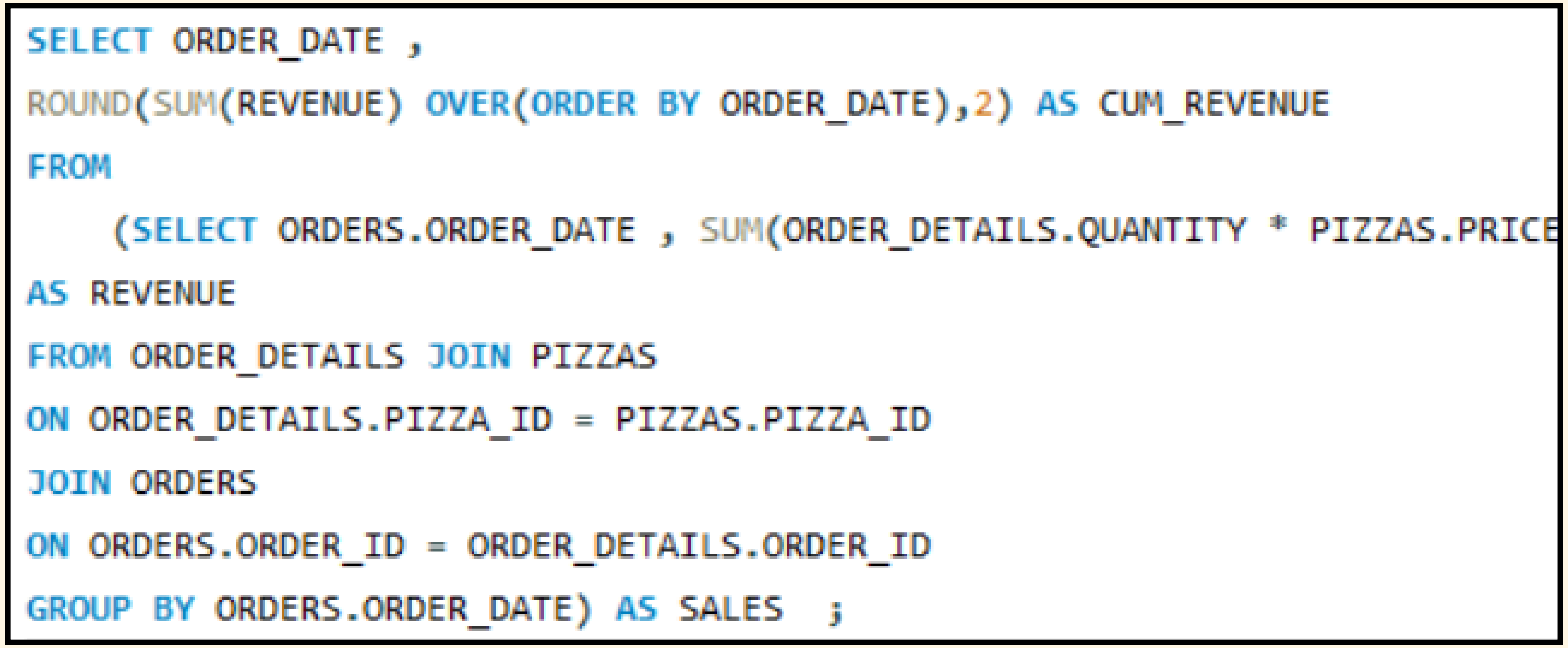


CATEGORY	REVENUE
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

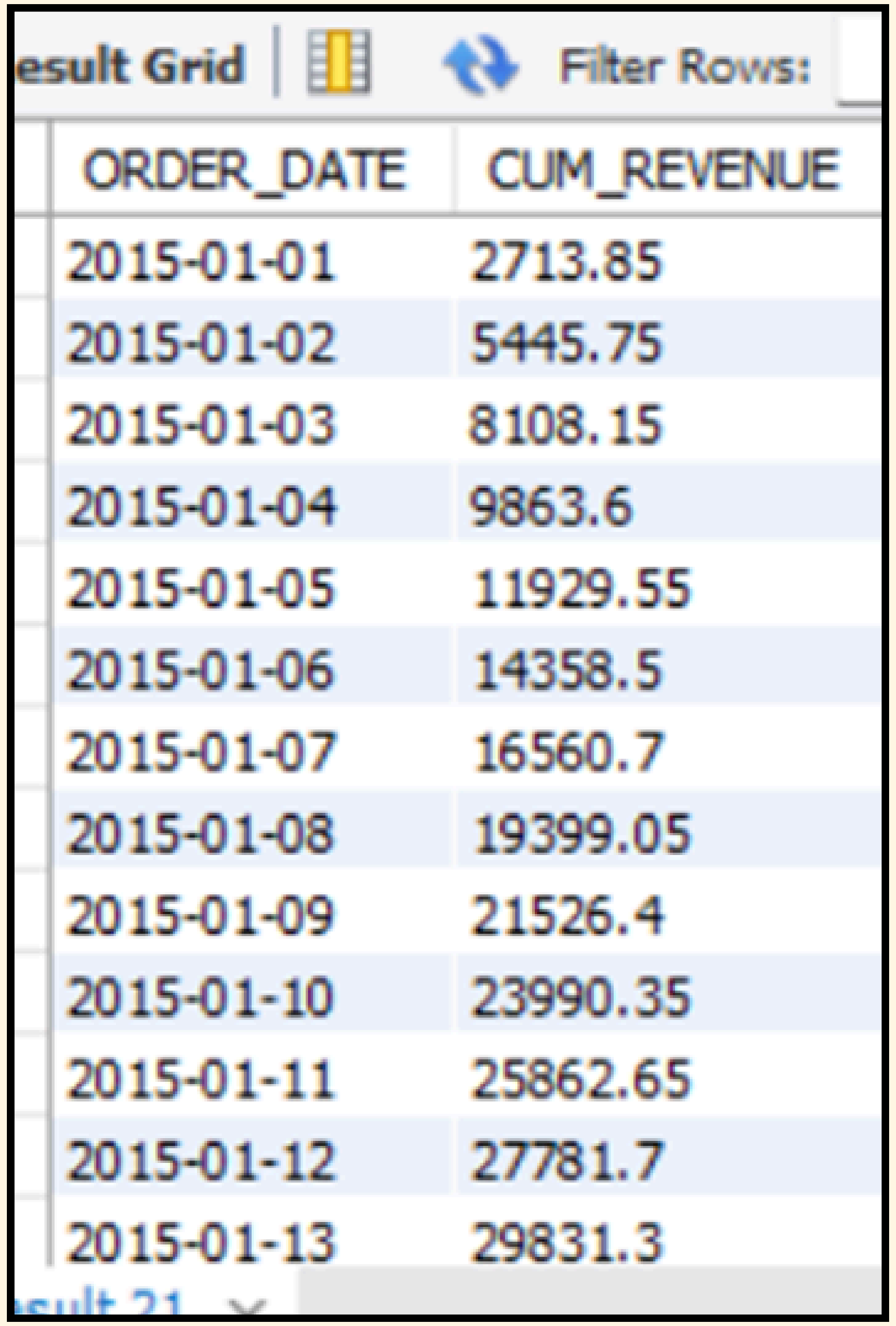
OUTPUT

12 . ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY



```
SELECT ORDER_DATE ,
ROUND(SUM(REVENUE) OVER(ORDER BY ORDER_DATE),2) 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.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS SALES  ;
```

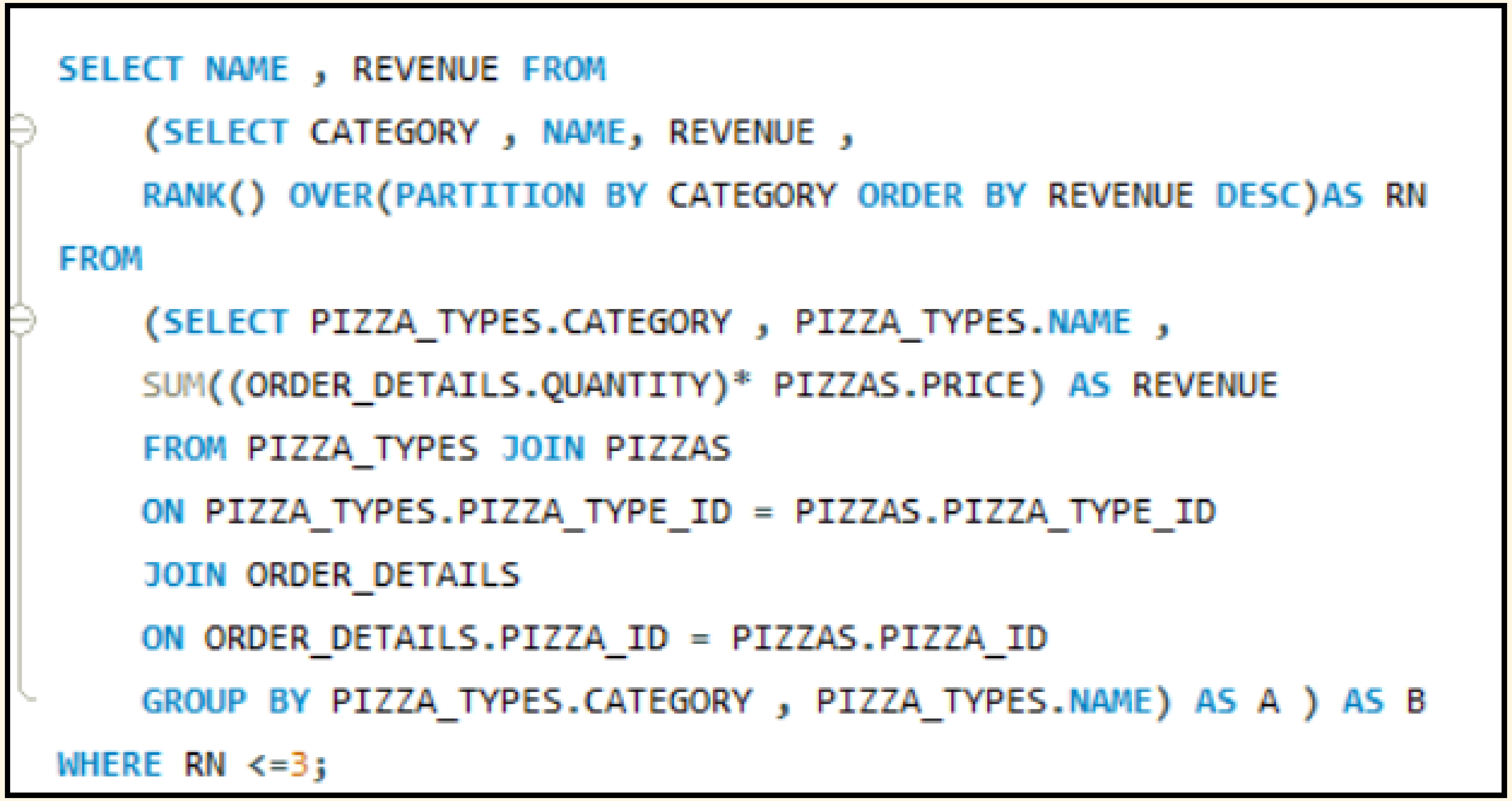


ORDER_DATE	CUM_REVENUE
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3

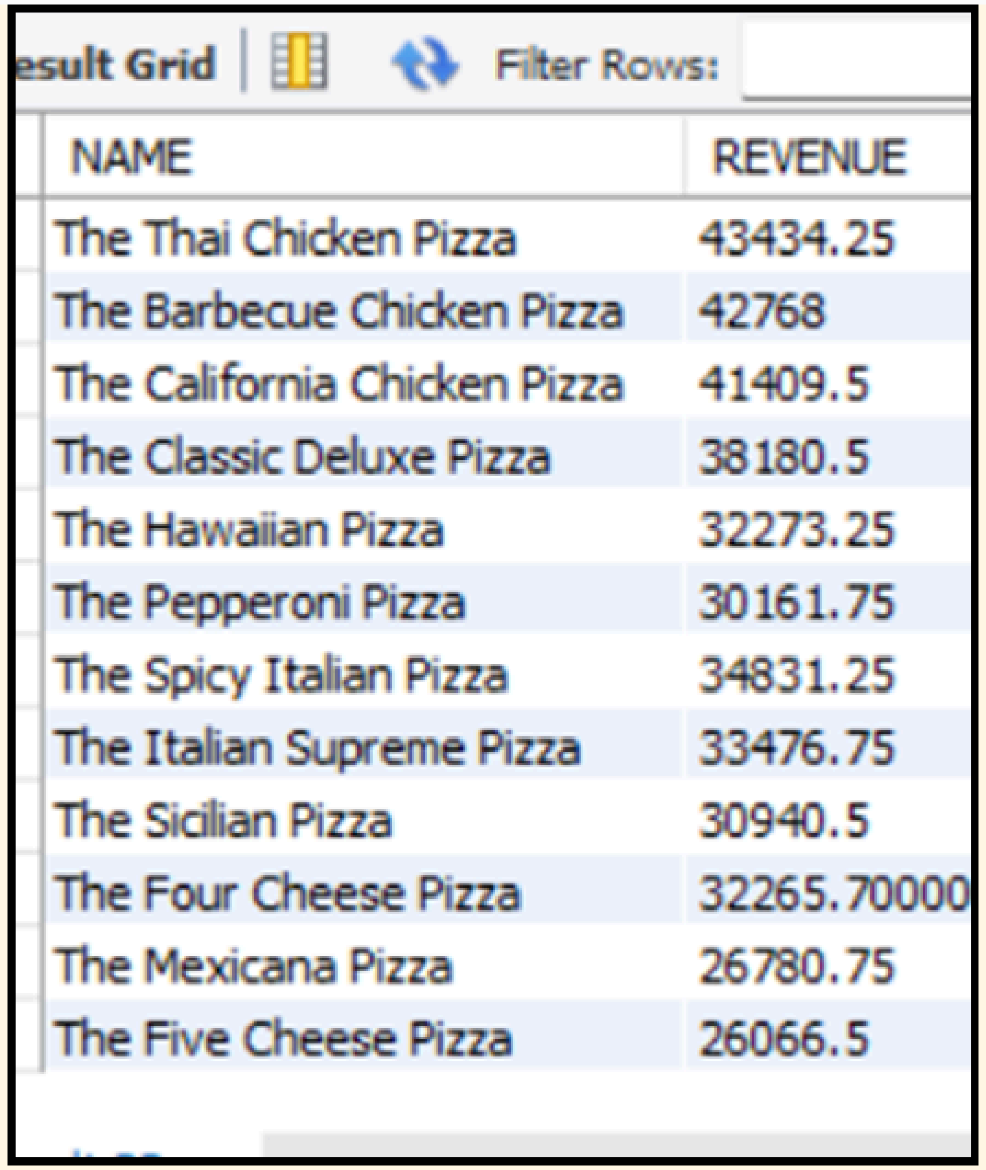
OUTPUT

13 . DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

QUERY



```
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)* PIZZAS.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;
```



NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

OUTPUT