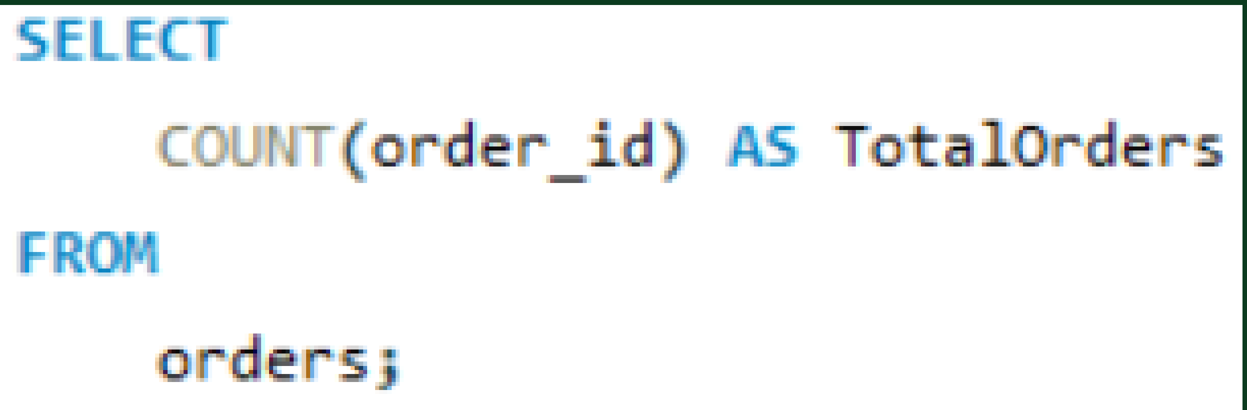
PIZZA SALE

SQL

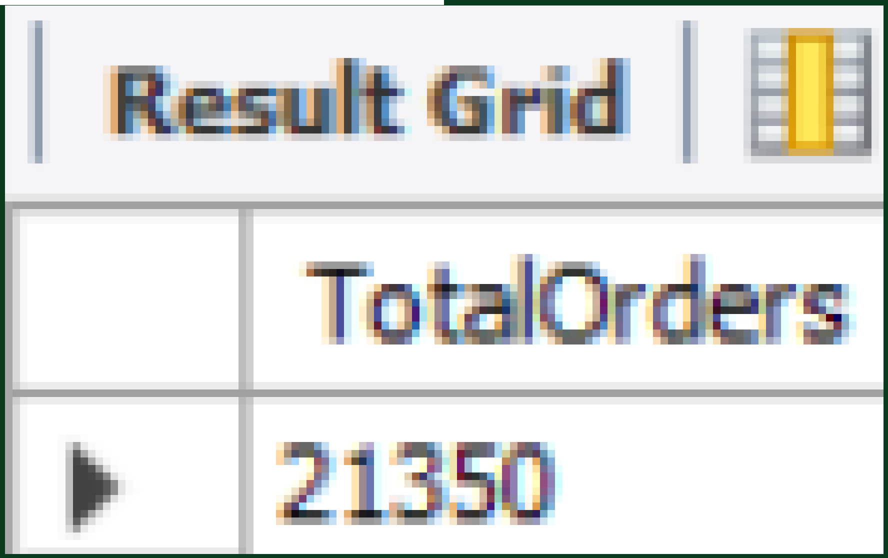
COMPREHENSIVE SQL ANALYSIS OF PIZZA SALES: FROM BASIC QUERIES TO ADVANCED TECHNIQUES

I HAVE CREATED A PROJECT ON SQL, WRITING VARIOUS QUERIES TO ANALYZE PIZZA SALES. THE QUERIES RANGE FROM BASIC, SUCH AS SELECT STATEMENTS FOR RETRIEVING DATA, TO INTERMEDIATE LEVEL WITH AGGREGATE FUNCTIONS, GROUP BY, AND ORDER BY CLAUSES, AND FINALLY TO ADVANCED QUERIES USING WINDOW FUNCTIONS AND SUBQUERIES.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED?

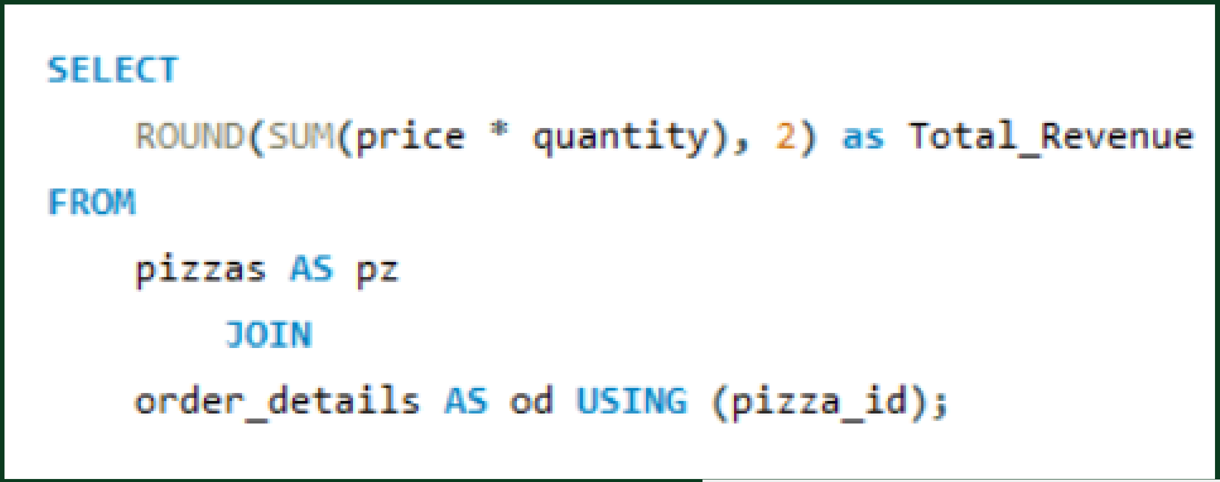


```sql
SELECT
    COUNT(order_id) AS TotalOrders
FROM
    orders;
```

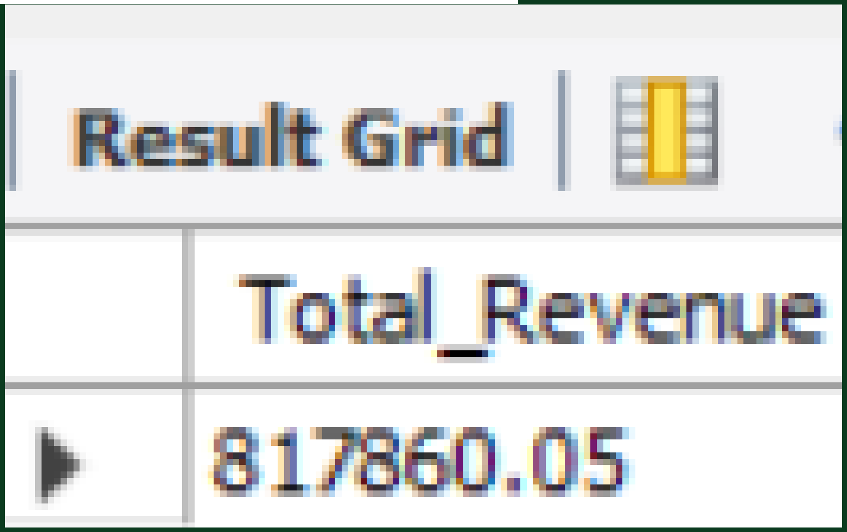


```
TotalOrders
21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES?

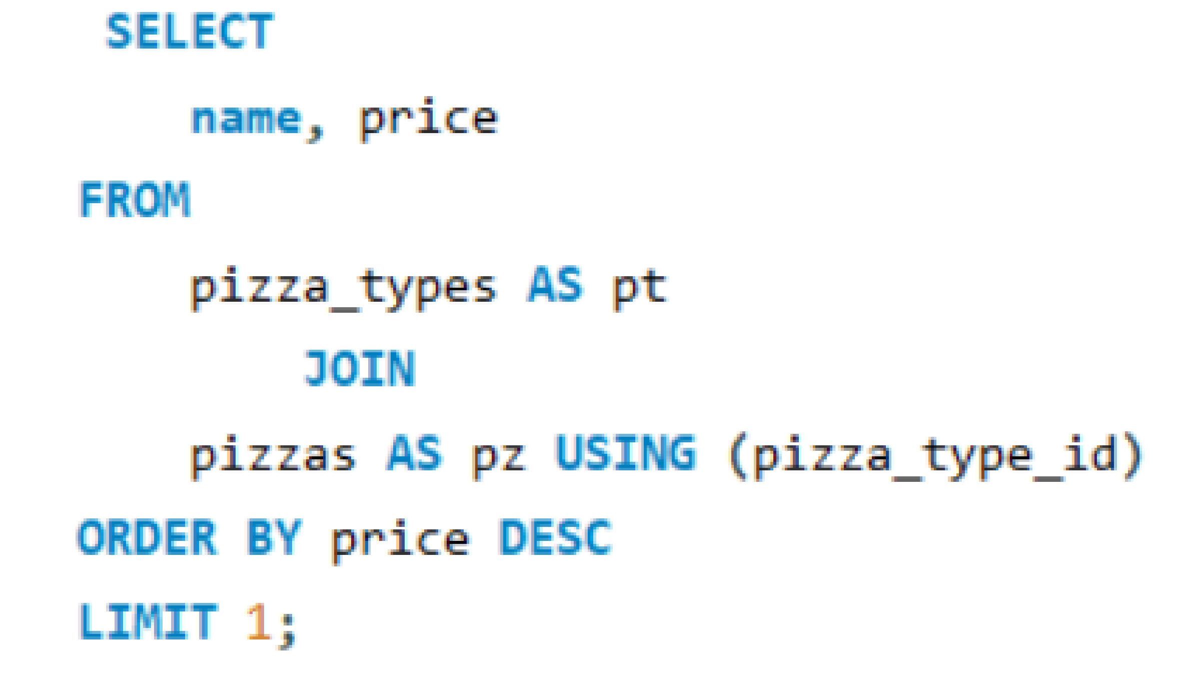


```
SELECT
    ROUND(SUM(price * quantity), 2) as Total_Revenue
FROM
    pizzas AS pz
        JOIN
    order_details AS od USING (pizza_id);
```

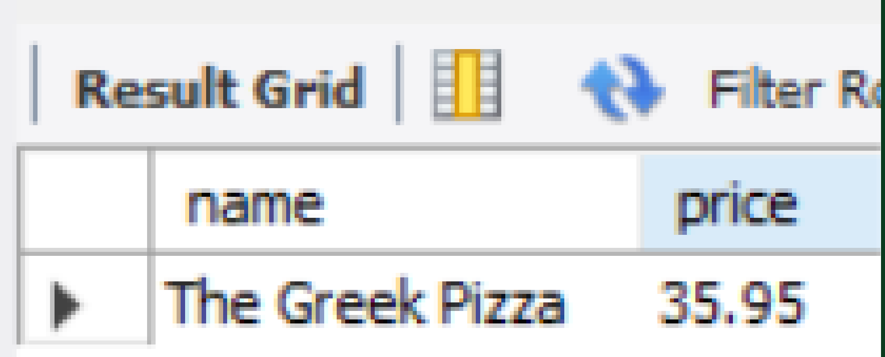


```
  Total_Revenue
▶ 817860.05
```

IDENTIFY THE HIGHEST-PRICED PIZZA?

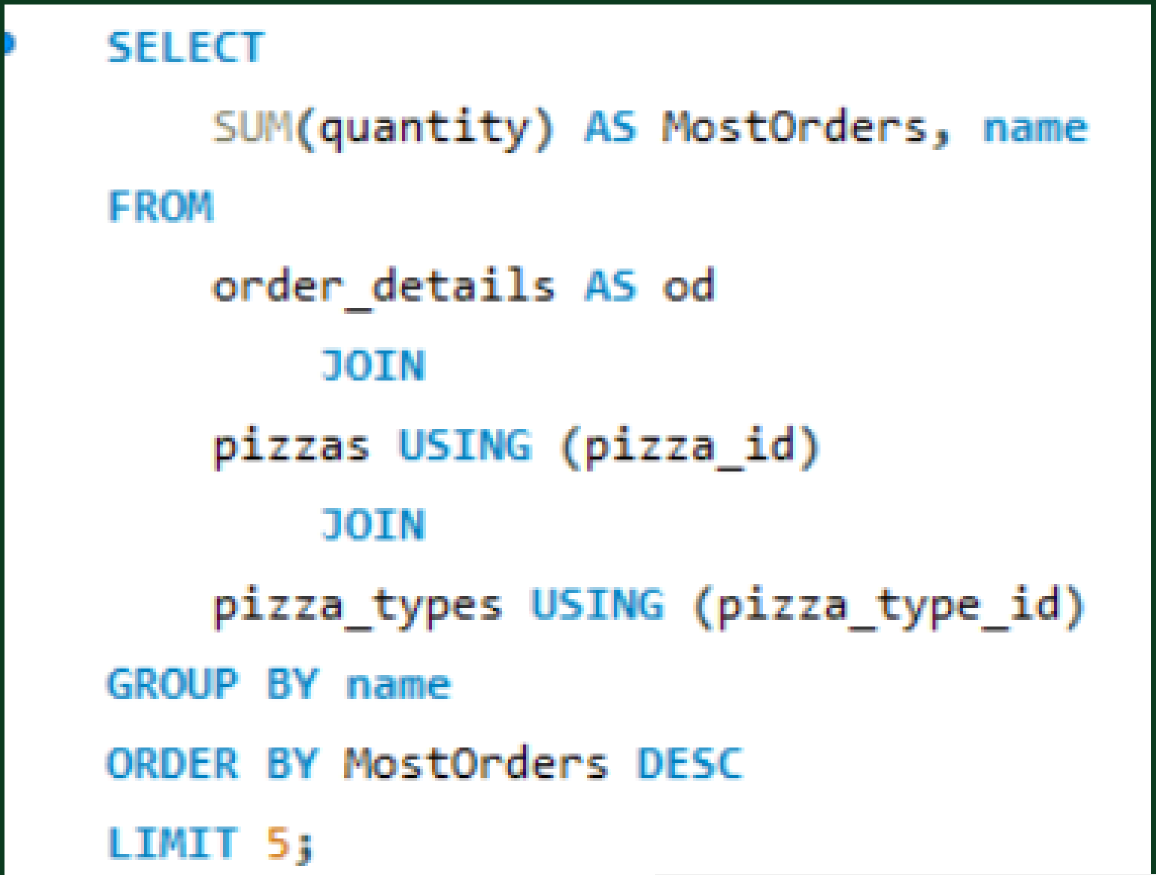


```sql
 SELECT
    name, price
FROM
    pizza_types AS pt
        JOIN
    pizzas AS pz USING (pizza_type_id)
ORDER BY price DESC
LIMIT 1;
```

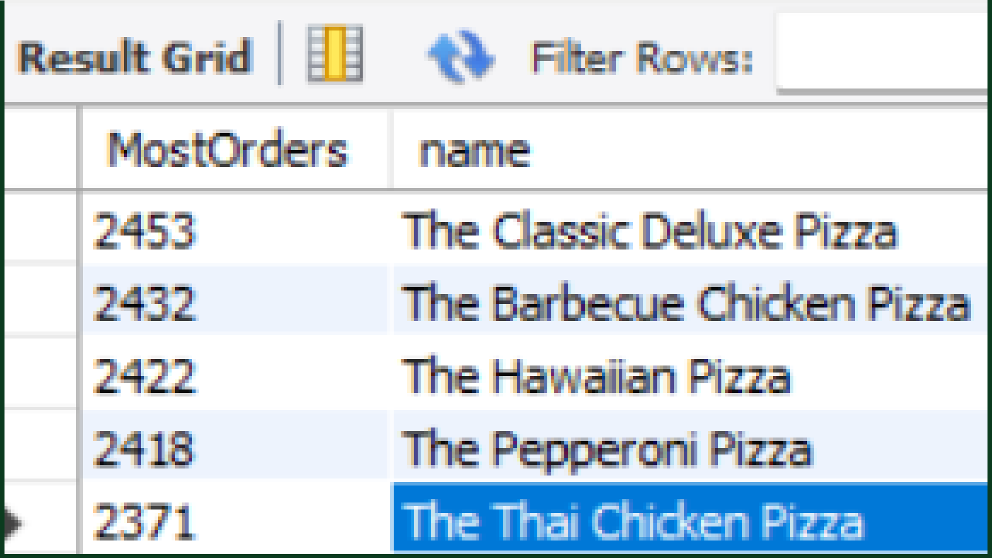


```
   name              price
▶  The Greek Pizza   35.95
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED?

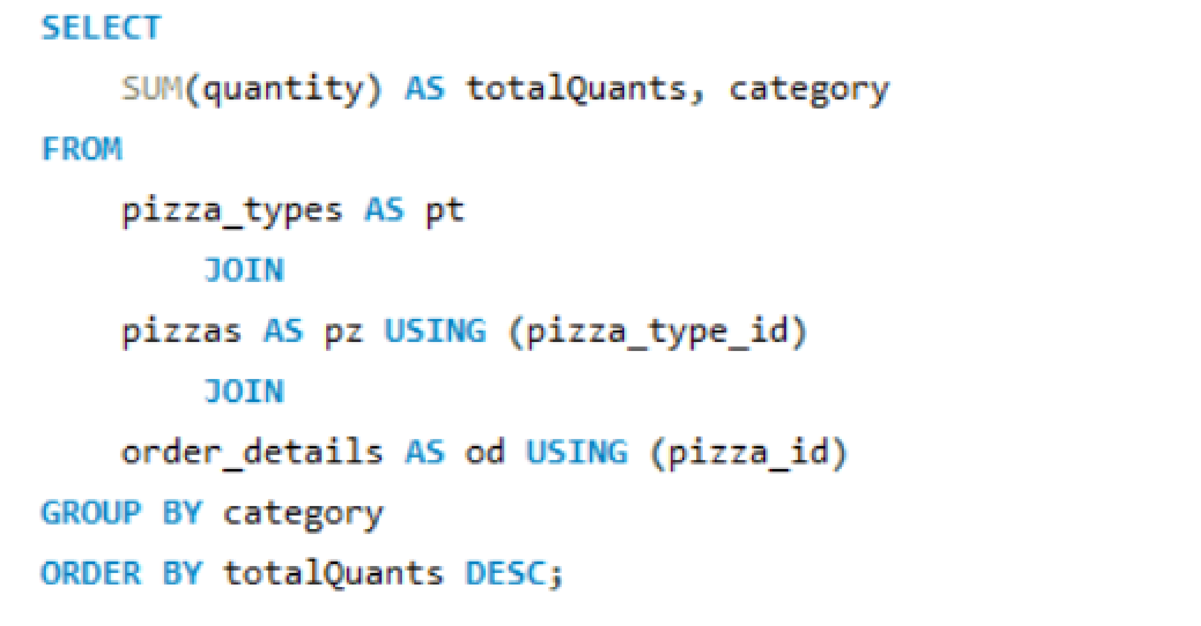


```
SELECT
    SUM(quantity) AS MostOrders, name
FROM
    order_details AS od
        JOIN
    pizzas USING (pizza_id)
        JOIN
    pizza_types USING (pizza_type_id)
GROUP BY name
ORDER BY MostOrders DESC
LIMIT 5;
```

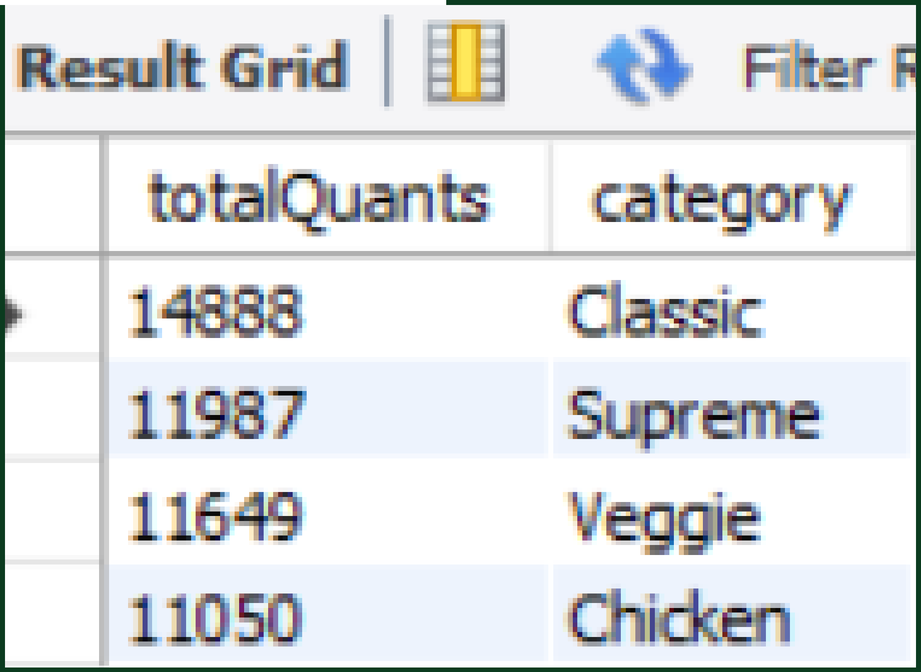


MostOrders	name
2453	The Classic Deluxe Pizza
2432	The Barbecue Chicken Pizza
2422	The Hawaiian Pizza
2418	The Pepperoni Pizza
2371	The Thai Chicken Pizza

FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED?

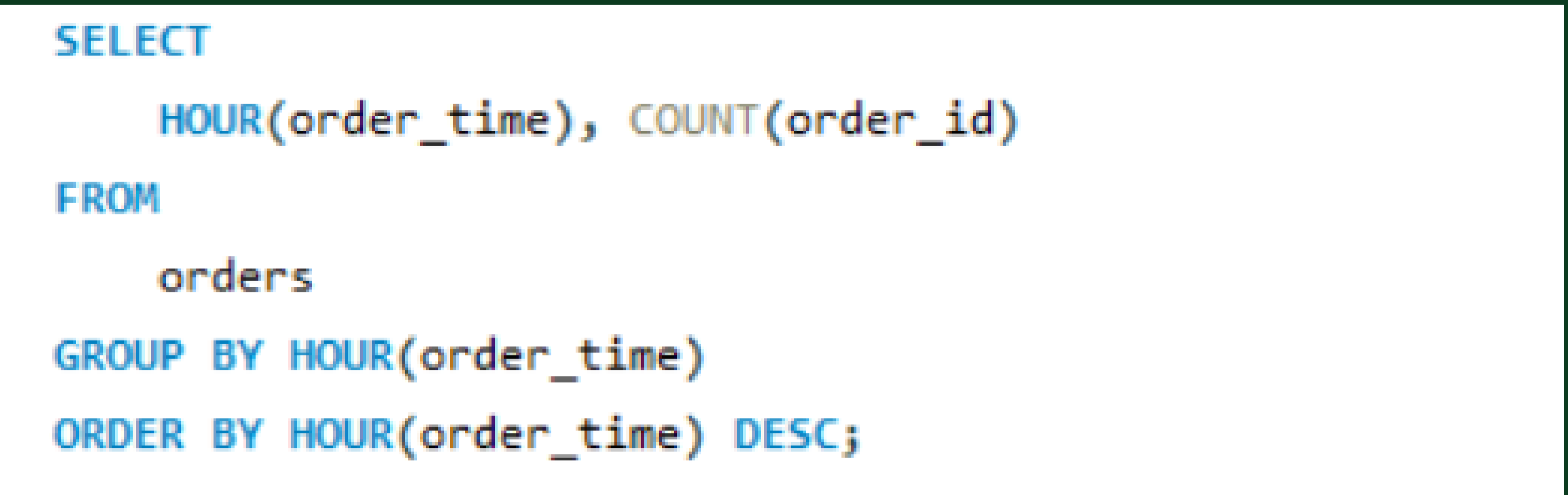


```
SELECT
    SUM(quantity) AS totalQuants, category
FROM
    pizza_types AS pt
        JOIN
    pizzas AS pz USING (pizza_type_id)
        JOIN
    order_details AS od USING (pizza_id)
GROUP BY category
ORDER BY totalQuants DESC;
```

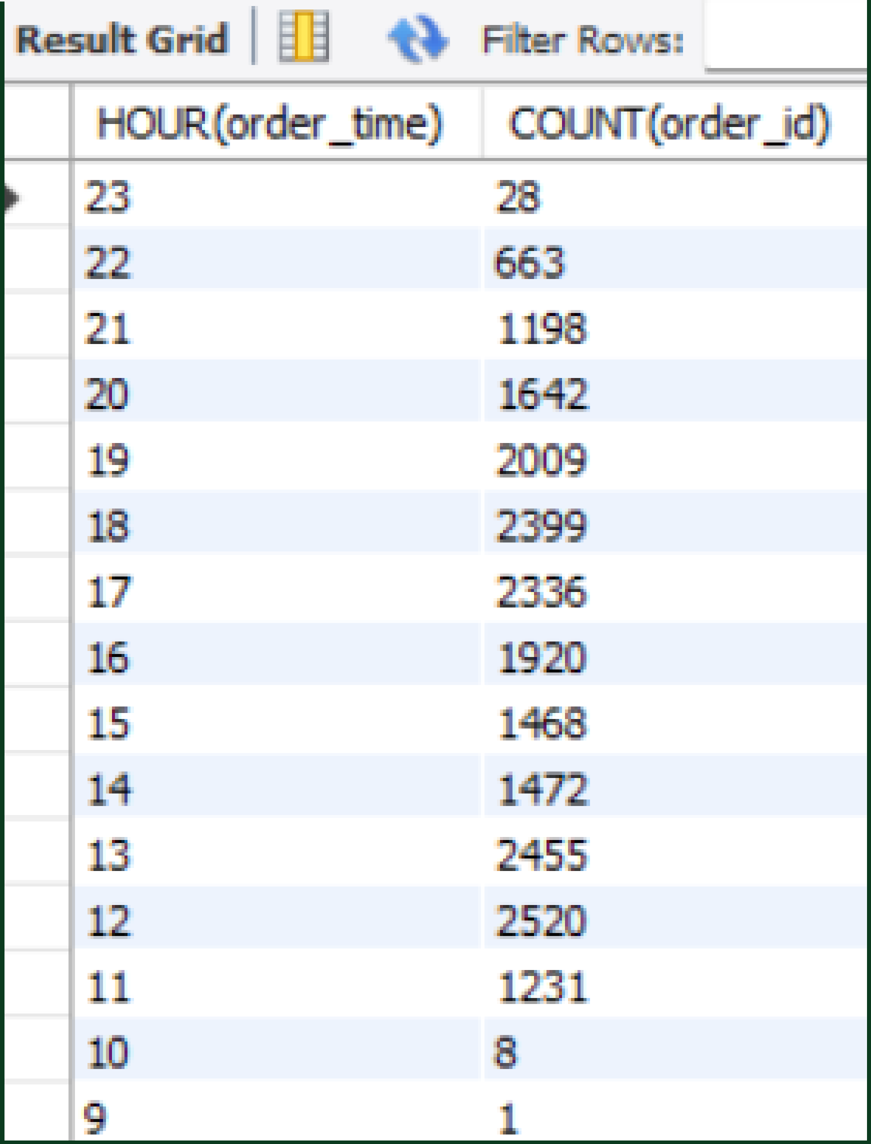


totalQuants	category
14888	Classic
11987	Supreme
11649	Veggie
11050	Chicken

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY?

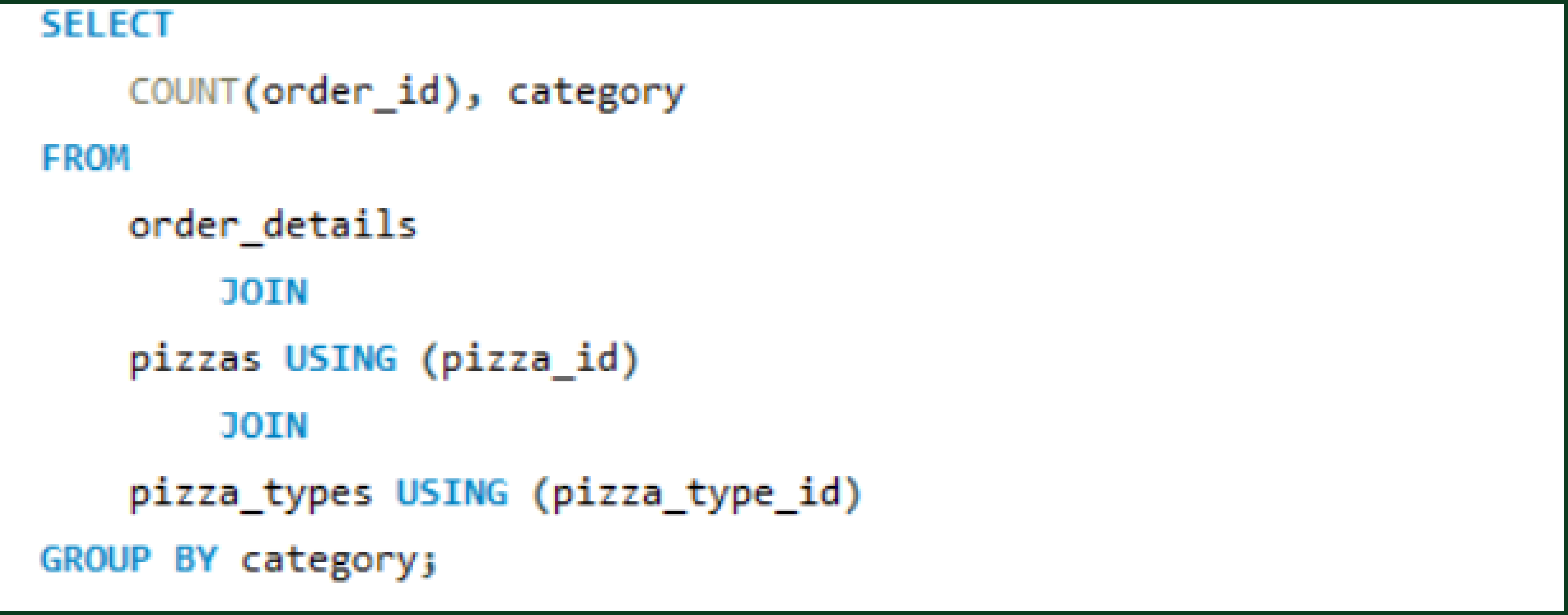


```sql
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY HOUR(order_time) DESC;
```

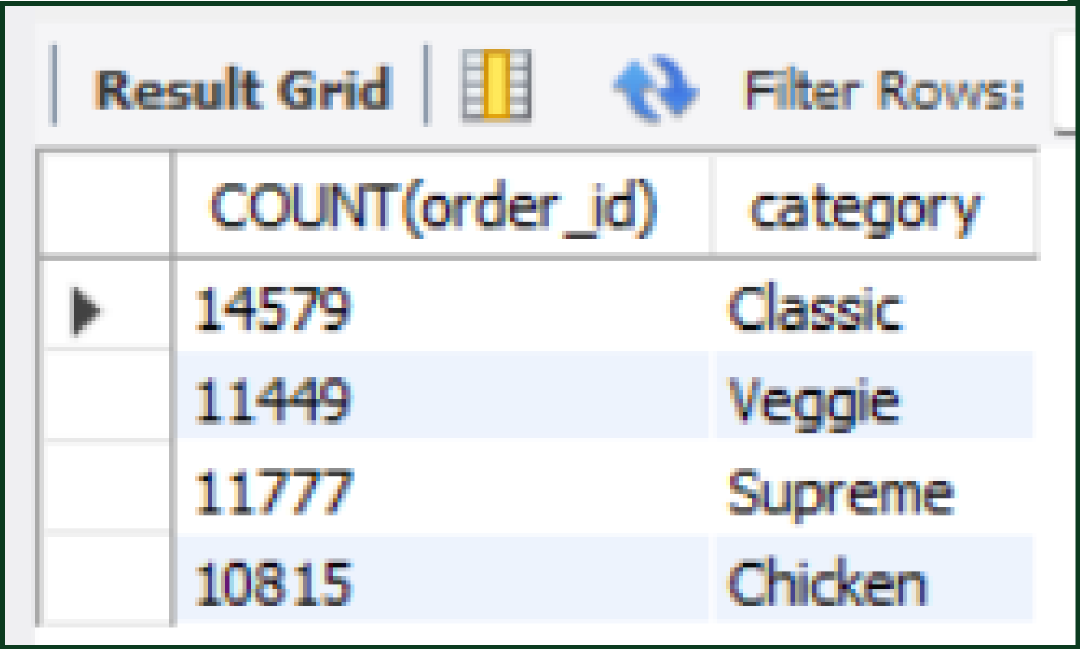


HOUR(order_time)	COUNT(order_id)
23	28
22	663
21	1198
20	1642
19	2009
18	2399
17	2336
16	1920
15	1468
14	1472
13	2455
12	2520
11	1231
10	8
9	1

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

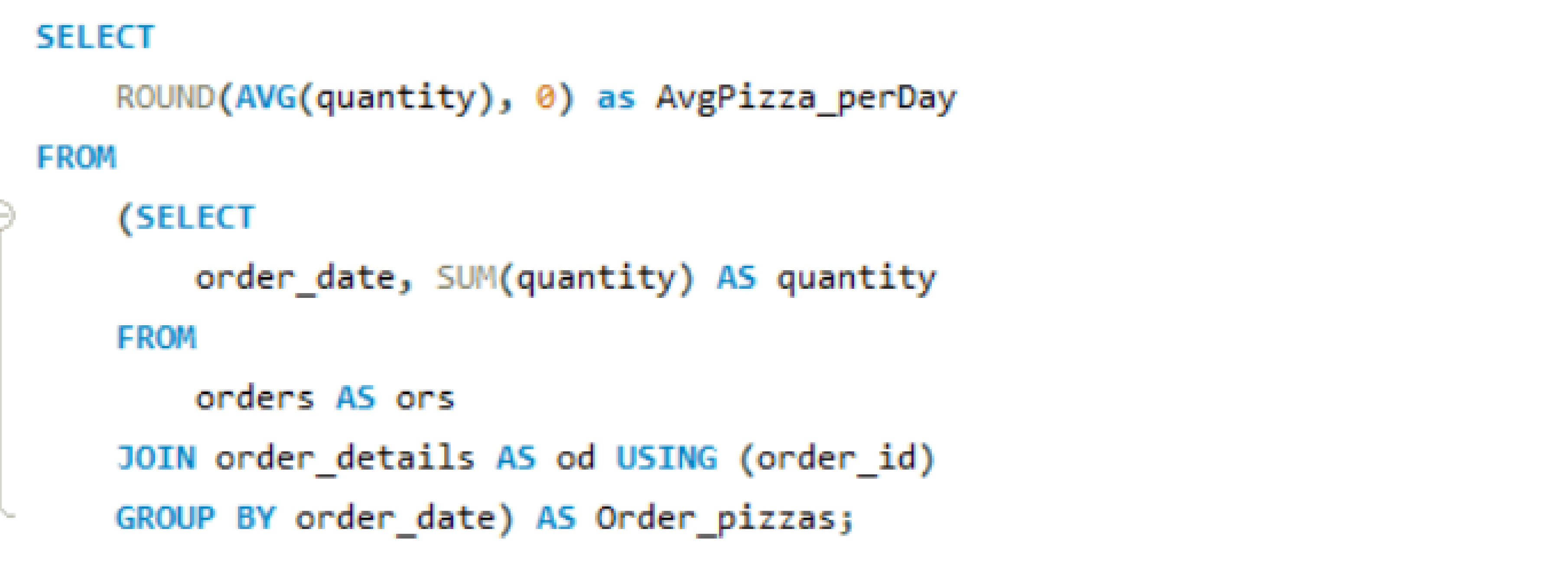


```sql
SELECT
    COUNT(order_id), category
FROM
    order_details
        JOIN
    pizzas USING (pizza_id)
        JOIN
    pizza_types USING (pizza_type_id)
GROUP BY category;
```

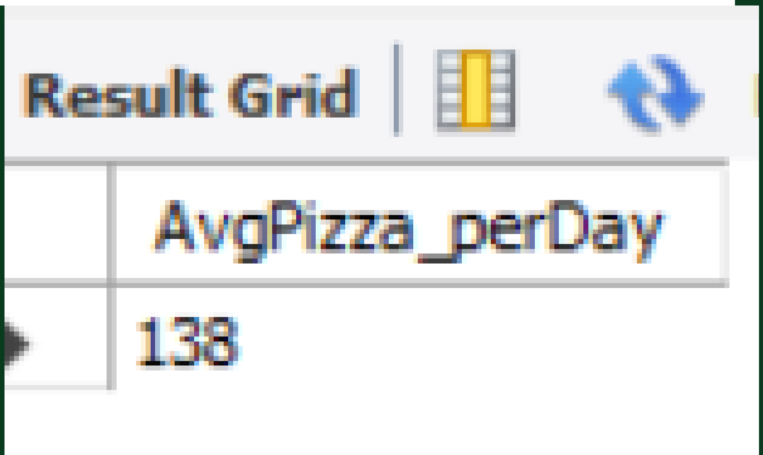


COUNT(order_id)	category
14579	Classic
11449	Veggie
11777	Supreme
10815	Chicken

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

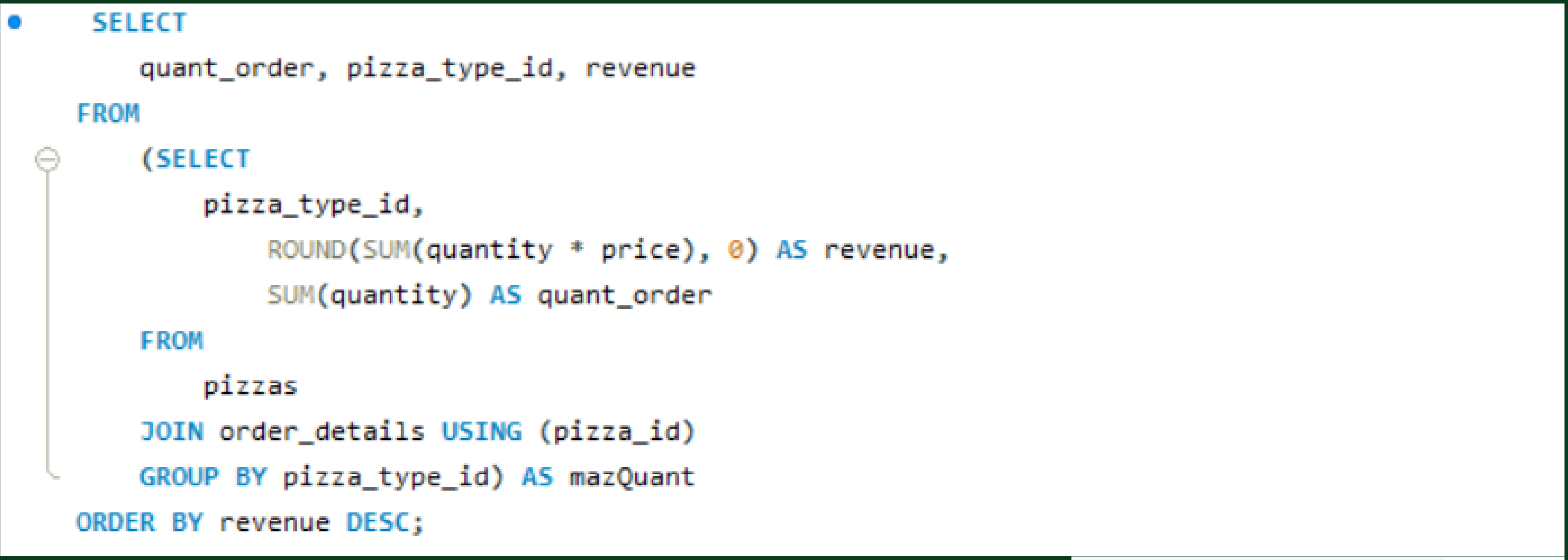


```sql
SELECT
    ROUND(AVG(quantity), 0) as AvgPizza_perDay
FROM
    (SELECT
        order_date, SUM(quantity) AS quantity
    FROM
        orders AS ors
    JOIN order_details AS od USING (order_id)
    GROUP BY order_date) AS Order_pizzas;
```

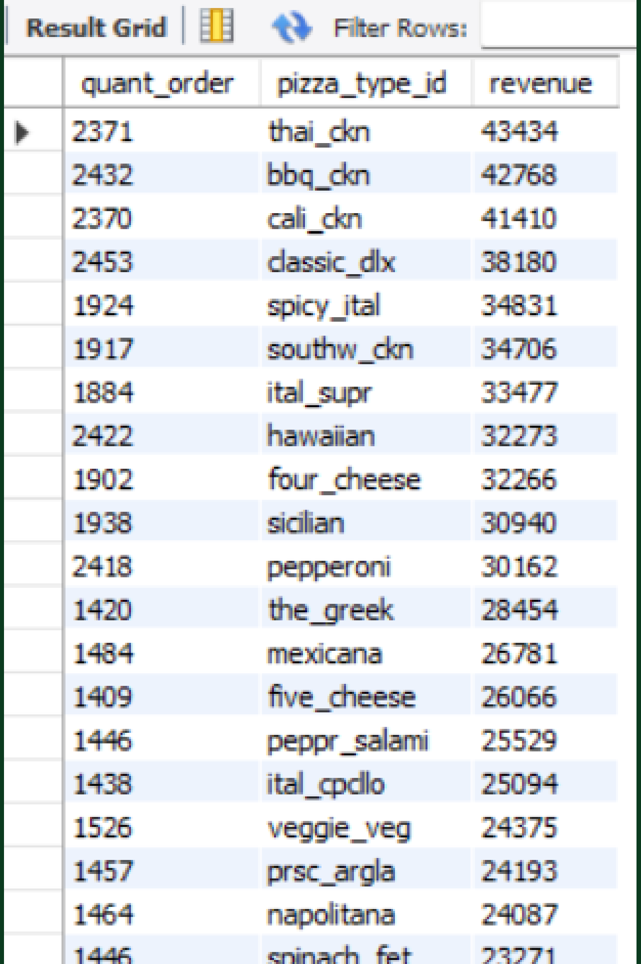


AvgPizza_perDay
138

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

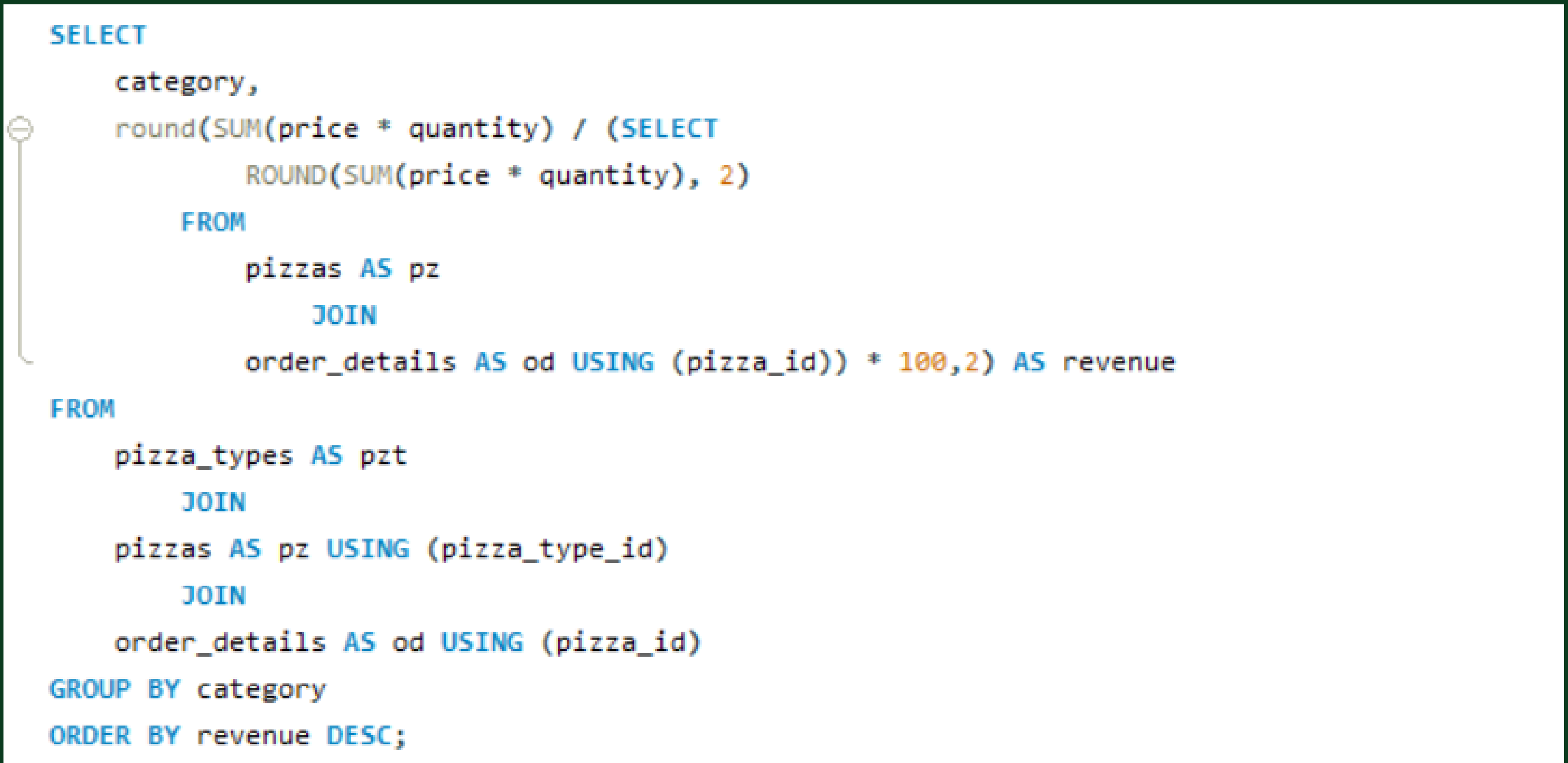


```
SELECT
    quant_order, pizza_type_id, revenue
FROM
    (SELECT
        pizza_type_id,
            ROUND(SUM(quantity * price), 0) AS revenue,
            SUM(quantity) AS quant_order
    FROM
        pizzas
    JOIN order_details USING (pizza_id)
    GROUP BY pizza_type_id) AS mazQuant
ORDER BY revenue DESC;
```

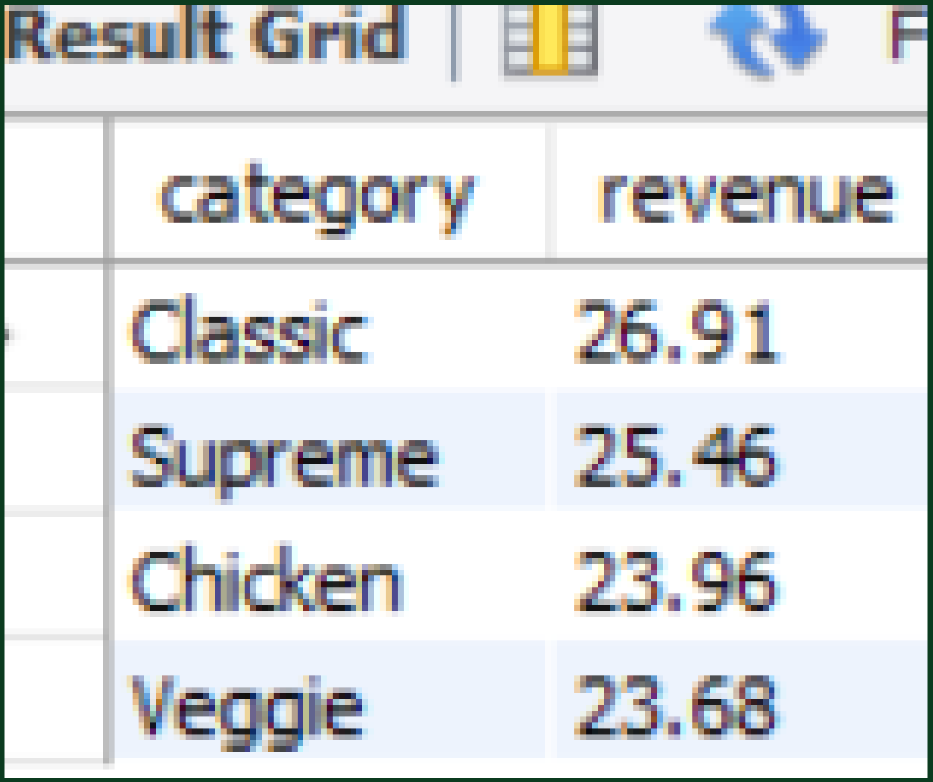


quant_order	pizza_type_id	revenue
2371	thai_ckn	43434
2432	bbq_ckn	42768
2370	cali_ckn	41410
2453	classic_dlx	38180
1924	spicy_ital	34831
1917	southw_ckn	34706
1884	ital_supr	33477
2422	hawaiian	32273
1902	four_cheese	32266
1938	sicilian	30940
2418	pepperoni	30162
1420	the_greek	28454
1484	mexicana	26781
1409	five_cheese	26066
1446	peppr_salami	25529
1438	ital_cpdlo	25094
1526	veggie_veg	24375
1457	prsc_argla	24193
1464	napolitana	24087
1446	spinach_fet	23271

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

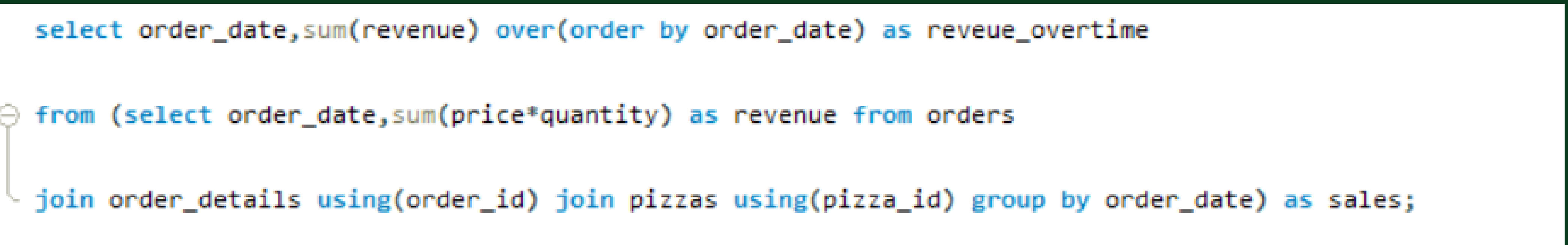


```
SELECT
    category,
    round(SUM(price * quantity) / (SELECT
            ROUND(SUM(price * quantity), 2)
        FROM
            pizzas AS pz
                JOIN
            order_details AS od USING (pizza_id)) * 100,2) AS revenue
FROM
    pizza_types AS pzt
        JOIN
    pizzas AS pz USING (pizza_type_id)
        JOIN
    order_details AS od USING (pizza_id)
GROUP BY category
ORDER BY revenue DESC;
```

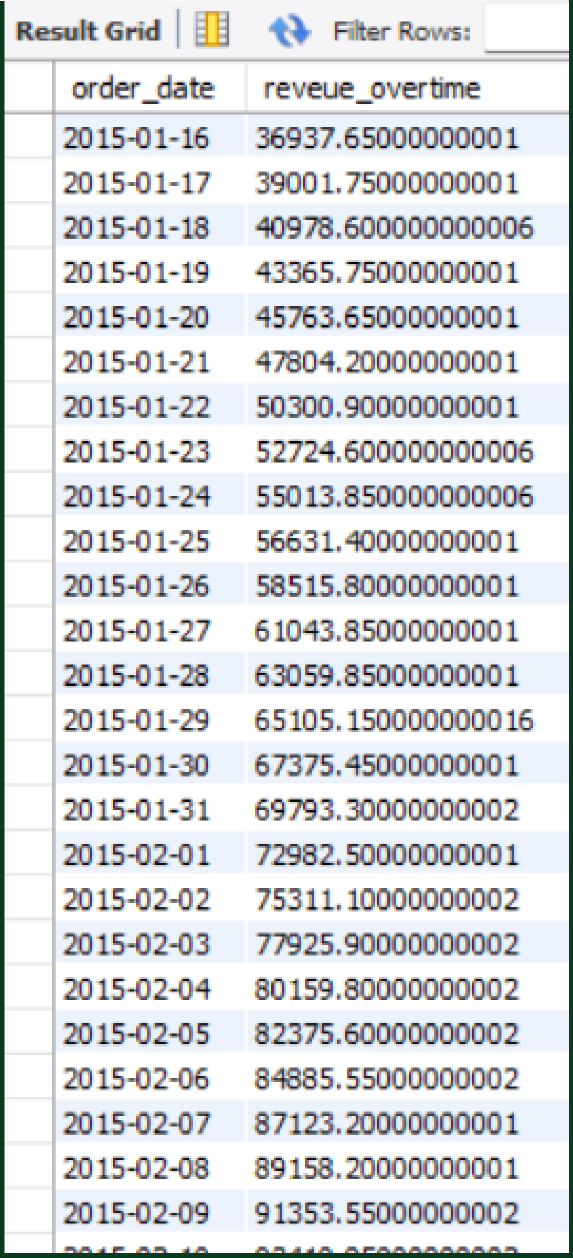


category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME?

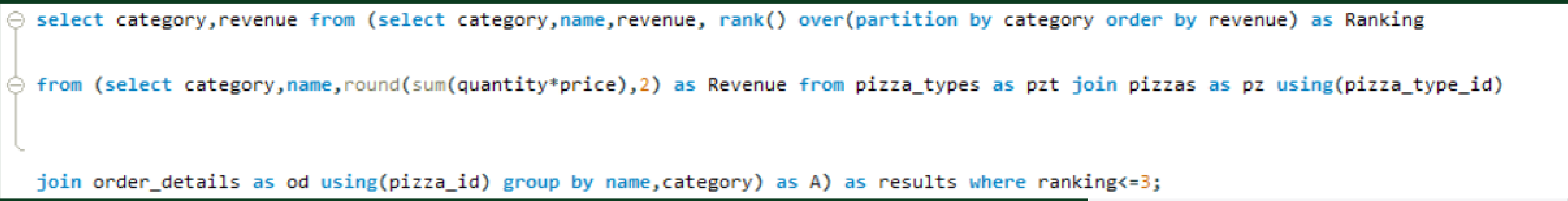


```
select order_date,sum(revenue) over(order by order_date) as reveue_overtime

from (select order_date,sum(price*quantity) as revenue from orders

join order_details using(order_id) join pizzas using(pizza_id) group by order_date) as sales;
```

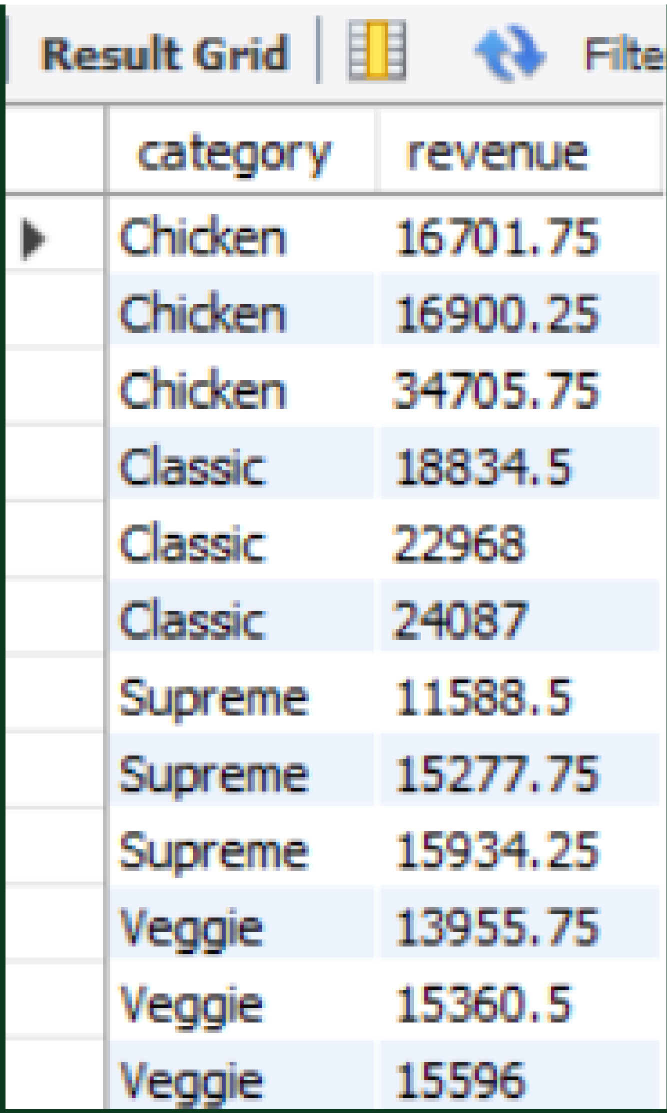


order_date	reveue_overtime
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001
2015-01-27	61043.85000000001
2015-01-28	63059.85000000001
2015-01-29	65105.150000000016
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002
2015-02-01	72982.50000000001
2015-02-02	75311.10000000002
2015-02-03	77925.90000000002
2015-02-04	80159.80000000002
2015-02-05	82375.60000000002
2015-02-06	84885.55000000002
2015-02-07	87123.20000000001
2015-02-08	89158.20000000001
2015-02-09	91353.55000000002

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



```
select category,revenue from (select category,name,revenue, rank() over(partition by category order by revenue) as Ranking

from (select category,name,round(sum(quantity*price),2) as Revenue from pizza_types as pzt join pizzas as pz using(pizza_type_id)


join order_details as od using(pizza_id) group by name,category) as A) as results where ranking<=3;
```



category	revenue
Chicken	16701.75
Chicken	16900.25
Chicken	34705.75
Classic	18834.5
Classic	22968
Classic	24087
Supreme	11588.5
Supreme	15277.75
Supreme	15934.25
Veggie	13955.75
Veggie	15360.5
Veggie	15596

MORE TO COME.

THANK YOU