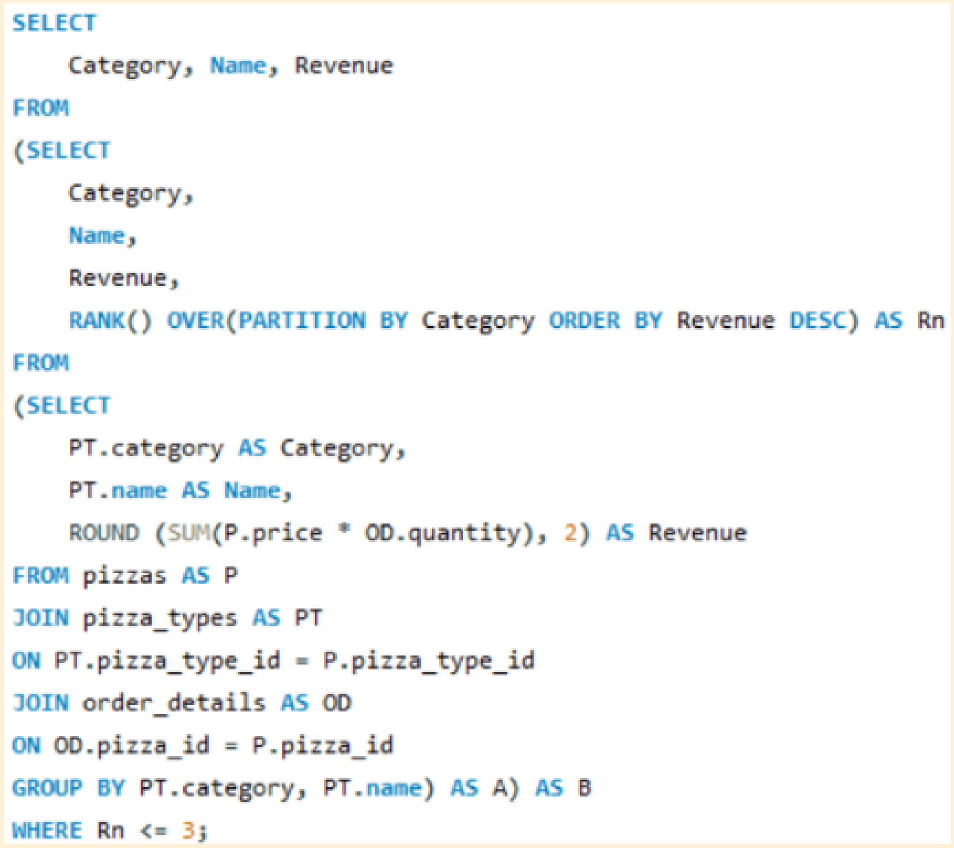
PIZZA SALES

SQL PROJECT

PRESENTED BY
ZIAUDDIN SHAH FAHAD

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

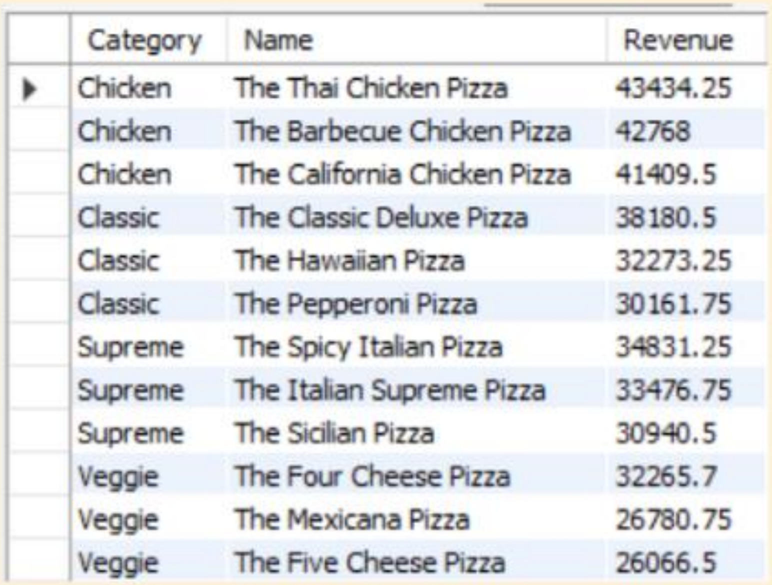
SQL Code



```
SELECT
    Category, Name, Revenue
FROM
(SELECT
    Category,
    Name,
    Revenue,
    RANK() OVER(PARTITION BY Category ORDER BY Revenue DESC) AS Rn
FROM
(SELECT
    PT.category AS Category,
    PT.name AS Name,
    ROUND (SUM(P.price * OD.quantity), 2) AS Revenue
FROM pizzas AS P
JOIN pizza_types AS PT
ON PT.pizza_type_id = P.pizza_type_id
JOIN order_details AS OD
ON OD.pizza_id = P.pizza_id
GROUP BY PT.category, PT.name) AS A) AS B
WHERE Rn <= 3;
```

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

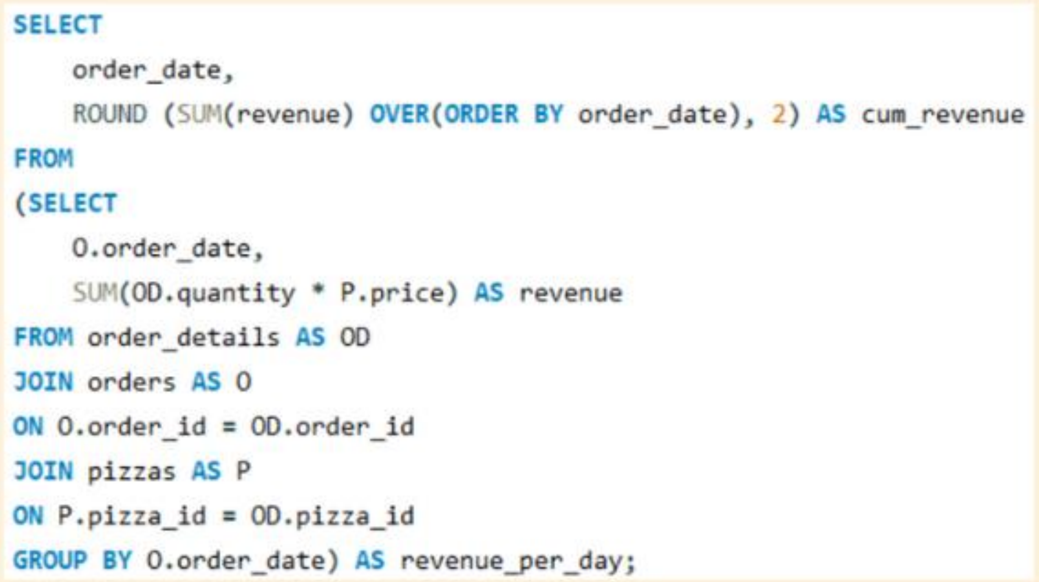
Output!



Category	Name	Revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

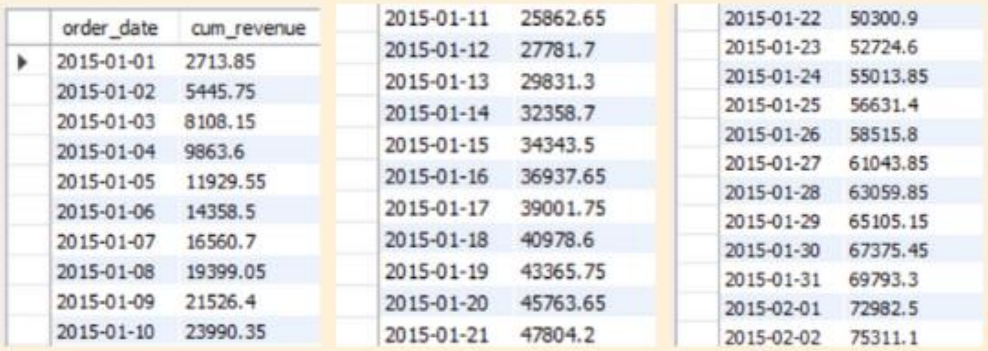
-- Analyze the cumulative revenue generated over time

SQL Code



```
SELECT
    order_date,
    ROUND (SUM(revenue) OVER(ORDER BY order_date), 2) AS cum_revenue
FROM
(SELECT
    O.order_date,
    SUM(OD.quantity * P.price) AS revenue
FROM order_details AS OD
JOIN orders AS O
ON O.order_id = OD.order_id
JOIN pizzas AS P
ON P.pizza_id = OD.pizza_id
GROUP BY O.order_date) AS revenue_per_day;
```

Output!

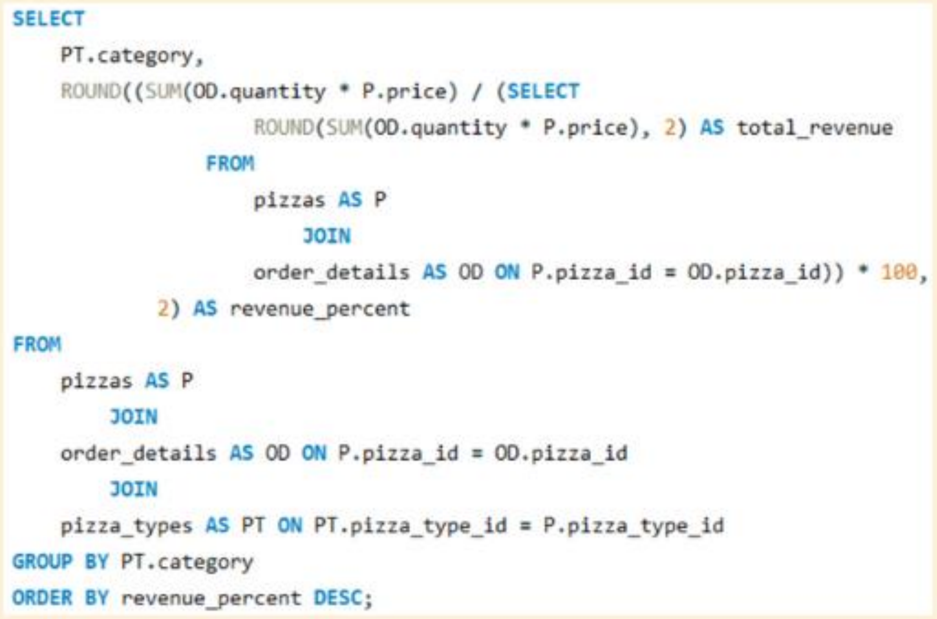


order_date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1

●●●

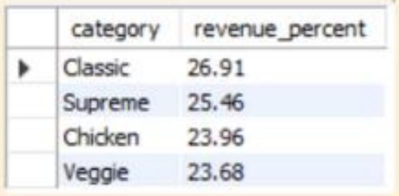
-- Calculate the percentage contribution of each pizza type to total revenue.

SQL Code



```
SELECT
    PT.category,
    ROUND((SUM(OD.quantity * P.price) / (SELECT
                ROUND(SUM(OD.quantity * P.price), 2) AS total_revenue
            FROM
                pizzas AS P
                    JOIN
                order_details AS OD ON P.pizza_id = OD.pizza_id)) * 100,
          2) AS revenue_percent
FROM
    pizzas AS P
        JOIN
    order_details AS OD ON P.pizza_id = OD.pizza_id
        JOIN
    pizza_types AS PT ON PT.pizza_type_id = P.pizza_type_id
GROUP BY PT.category
ORDER BY revenue_percent DESC;
```

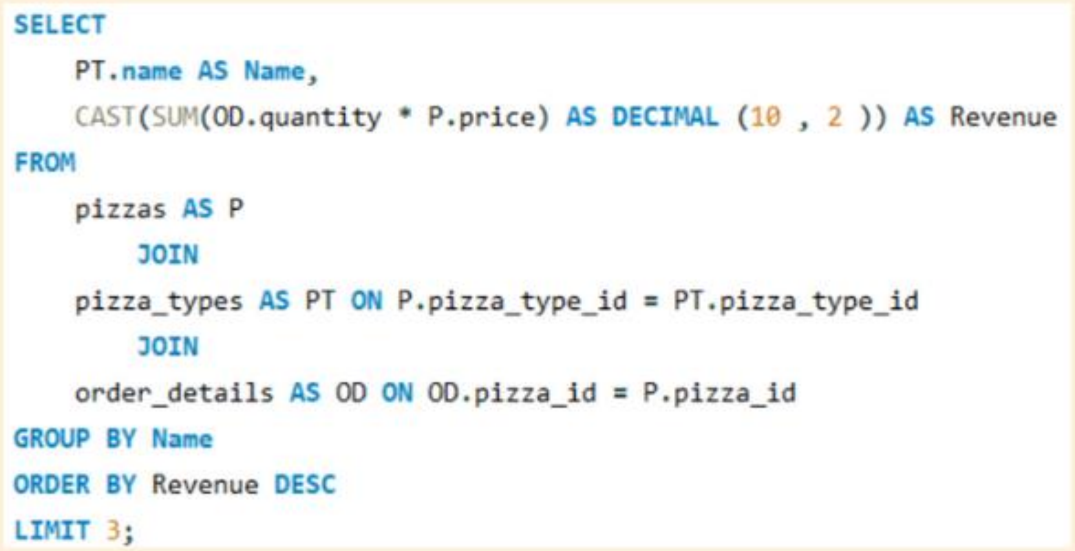
Output!



	category	revenue_percent
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

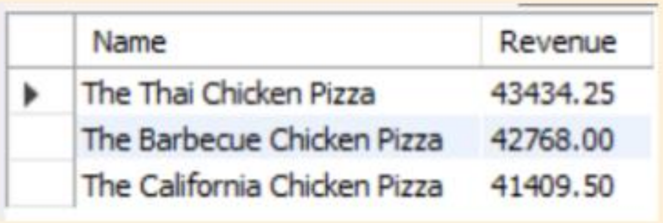
-- Determine the top 3 most ordered pizza types based on revenue.

SQL Code



```
SELECT
    PT.name AS Name,
    CAST(SUM(OD.quantity * P.price) AS DECIMAL (10 , 2 )) AS Revenue
FROM
    pizzas AS P
        JOIN
    pizza_types AS PT ON P.pizza_type_id = PT.pizza_type_id
        JOIN
    order_details AS OD ON OD.pizza_id = P.pizza_id
GROUP BY Name
ORDER BY Revenue DESC
LIMIT 3;
```

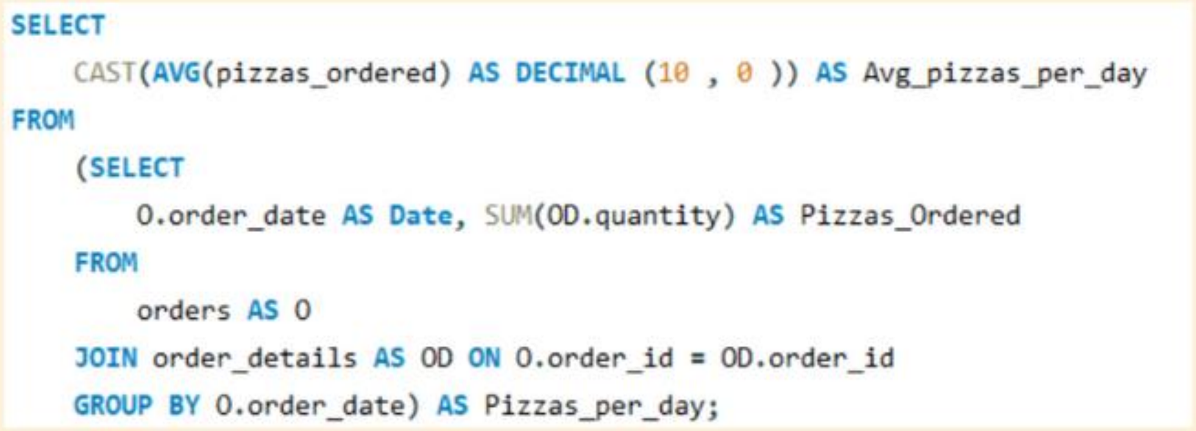
Output!



	Name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768.00
	The California Chicken Pizza	41409.50

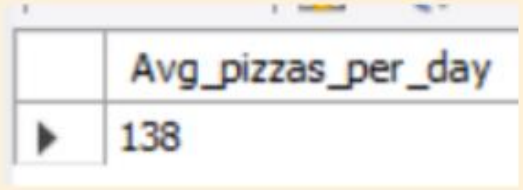
-- Group the orders by date and calculate the average number of pizzas ordered per day.

SQL Code



```
SELECT
    CAST(AVG(pizzas_ordered) AS DECIMAL (10 , 0 )) AS Avg_pizzas_per_day
FROM
    (SELECT
        O.order_date AS Date, SUM(OD.quantity) AS Pizzas_Ordered
    FROM
        orders AS O
    JOIN order_details AS OD ON O.order_id = OD.order_id
    GROUP BY O.order_date) AS Pizzas_per_day;
```

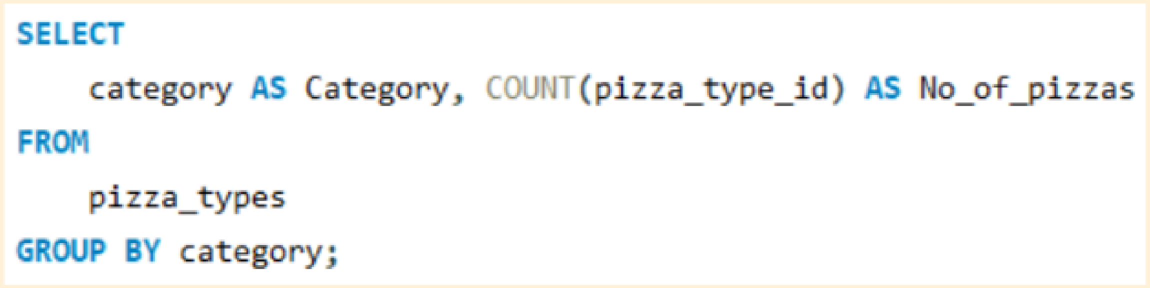
Output!



```
  Avg_pizzas_per_day
▶ 138
```

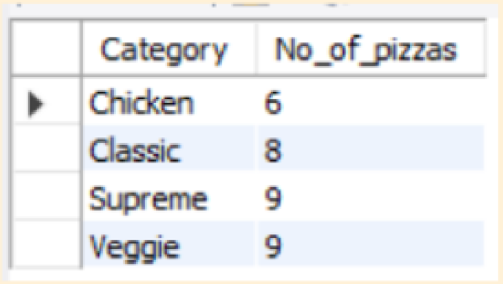
-- Join relevant tables to find the category-wise distribution of pizzas.

SQL Code



```
SELECT
    category AS Category, COUNT(pizza_type_id) AS No_of_pizzas
FROM
    pizza_types
GROUP BY category;
```

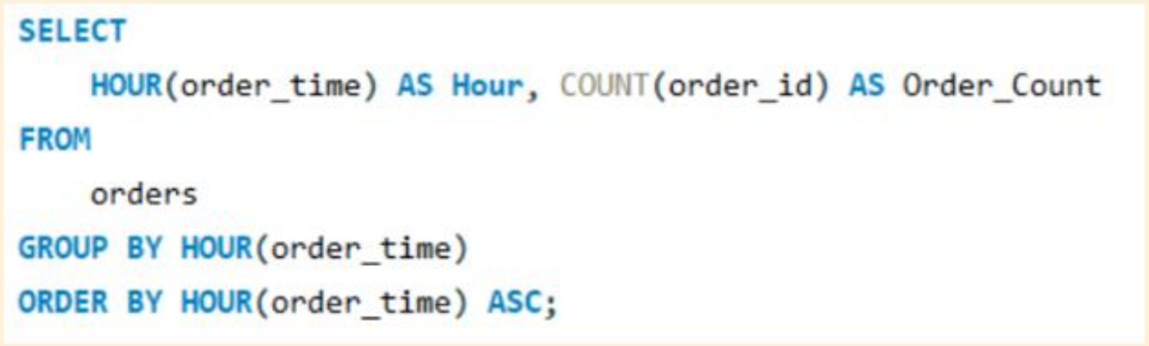
Output!



Category	No_of_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

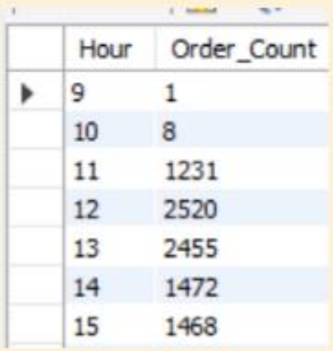
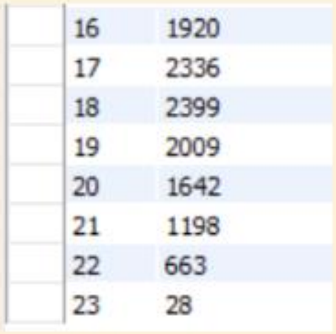
-- Determine the distribution of orders by hour of the day.

SQL Code



```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Order_Count
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY HOUR(order_time) ASC;
```

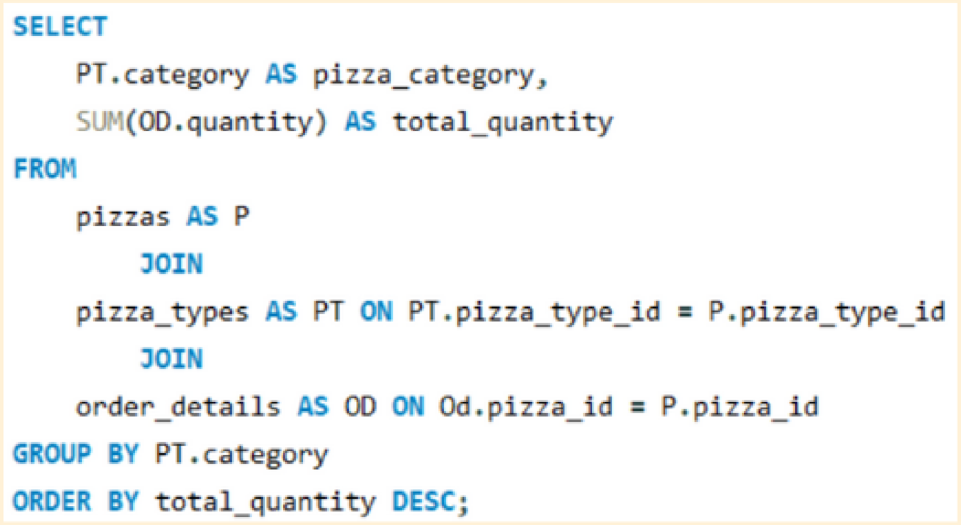
Output!

Hour	Order_Count
9	1
10	8
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

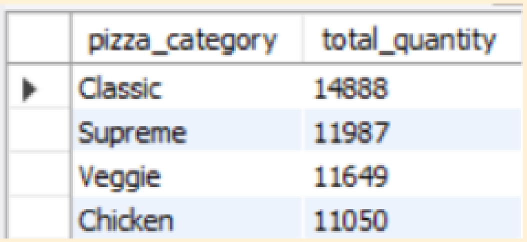
-- Join the necessary tables to find the total quantity of each pizza category ordered.

SQL Code



```
SELECT
    PT.category AS pizza_category,
    SUM(OD.quantity) AS total_quantity
FROM
    pizzas AS P
        JOIN
    pizza_types AS PT ON PT.pizza_type_id = P.pizza_type_id
        JOIN
    order_details AS OD ON Od.pizza_id = P.pizza_id
GROUP BY PT.category
ORDER BY total_quantity DESC;
```

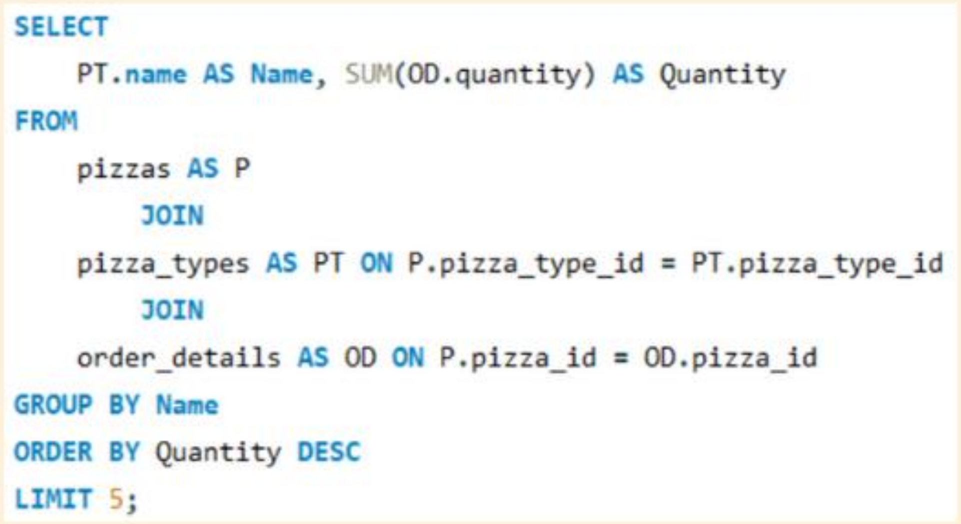
Output!



pizza_category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

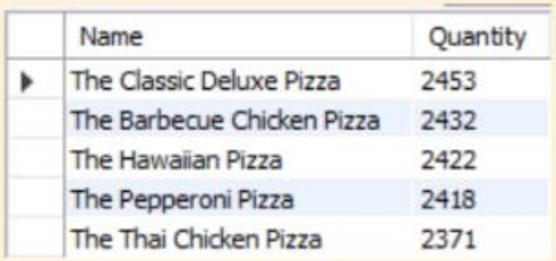
-- List the top 5 most ordered pizza types along with their quantities.

SQL Code



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

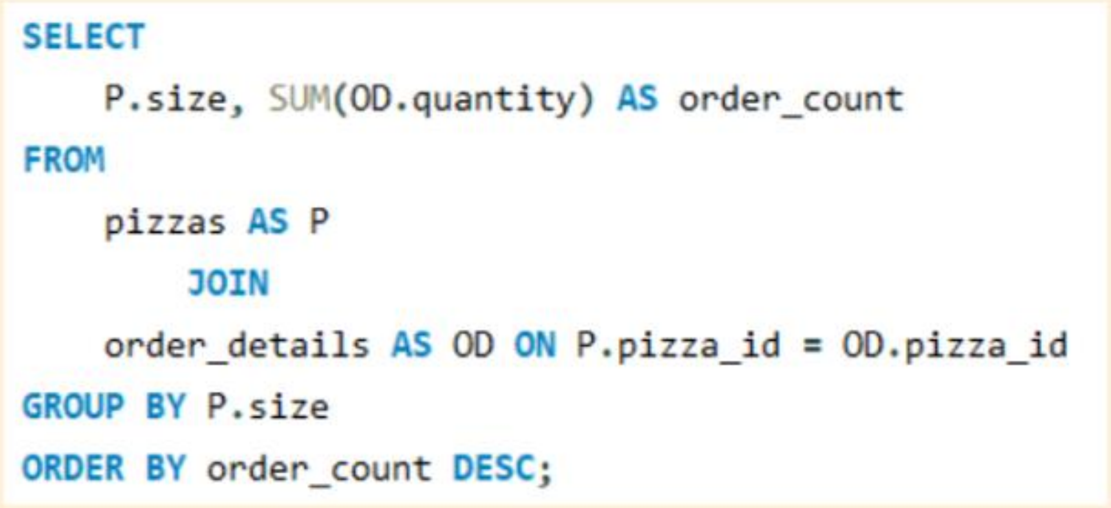
Output!



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

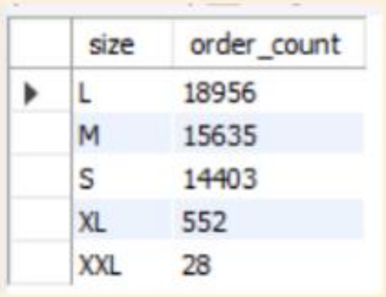
-- Identify the most common pizza size ordered.

SQL Code



```sql
SELECT
    P.size, SUM(OD.quantity) AS order_count
FROM
    pizzas AS P
        JOIN
    order_details AS OD ON P.pizza_id = OD.pizza_id
GROUP BY P.size
ORDER BY order_count DESC;
```

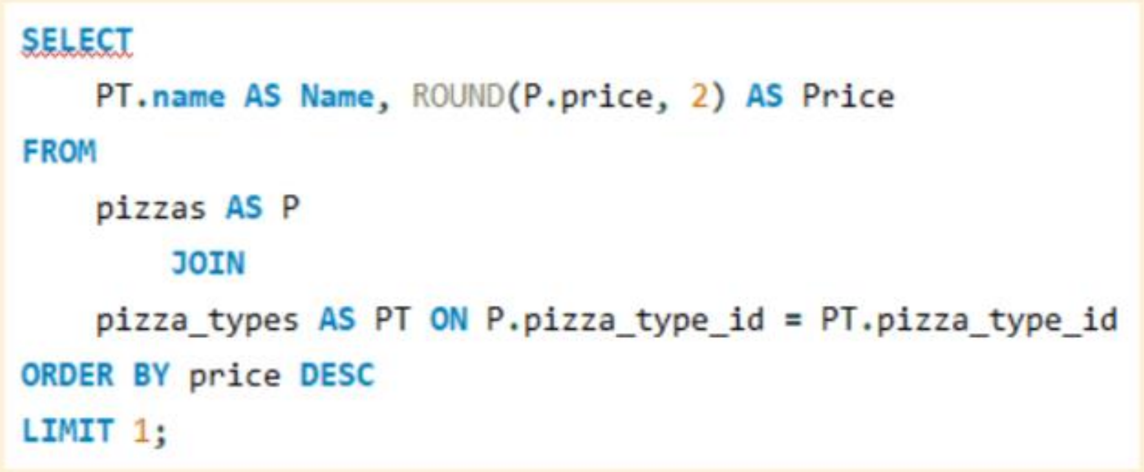
Output!



size	order_count
L	18956
M	15635
S	14403
XL	552
XXL	28

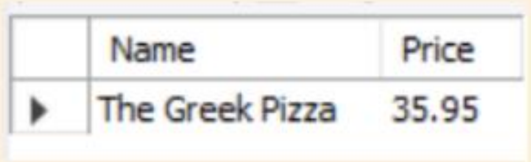
-- Identify the highest-priced pizza.

SQL Code



```
SELECT
    PT.name AS Name, ROUND(P.price, 2) AS Price
FROM
    pizzas AS P
        JOIN
    pizza_types AS PT ON P.pizza_type_id = PT.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

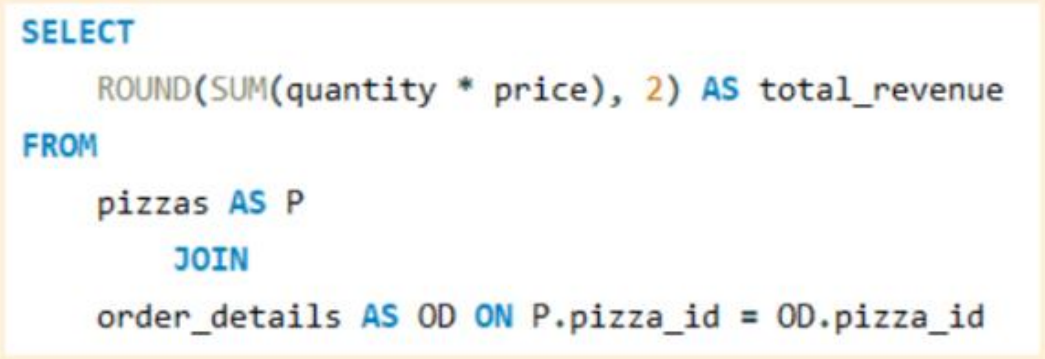
Output!



	Name	Price
▶	The Greek Pizza	35.95

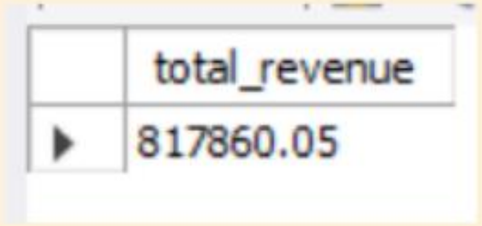
-- Calculate the total revenue generated from pizza sales.

SQL Code



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

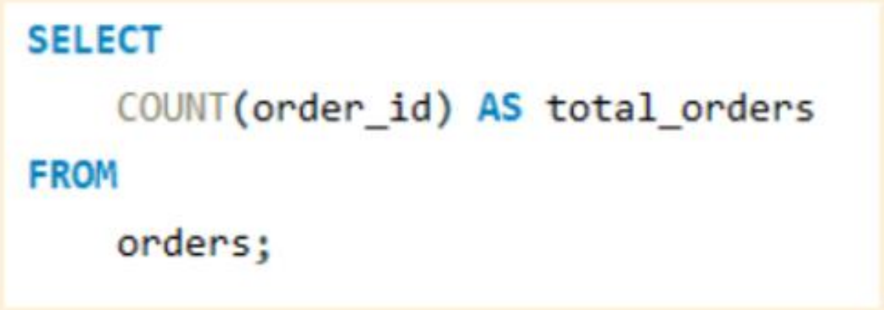
Output!



```
  | total_revenue
▶ | 817860.05
```

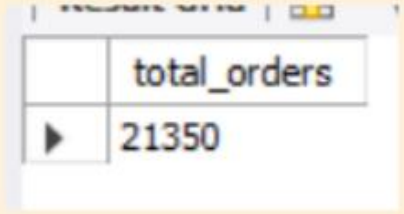
-- Retrieve the total number of orders placed.

SQL Code



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

Output!



```
   total_orders
▶  21350
```