Delicious

PIZZA

SQL Project

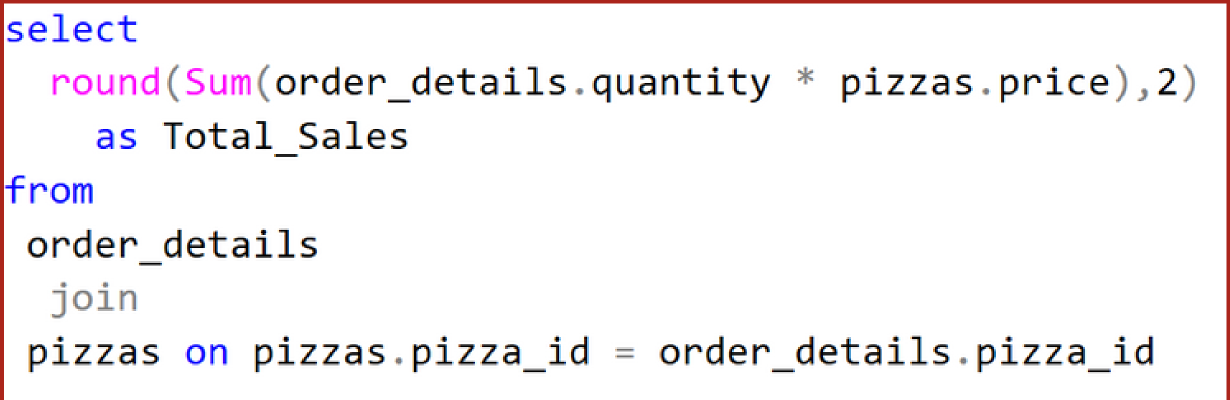
Pizza Sales

Hello !

My name is Arbaz Aslam and in this SQL project we analyze pizza sales data to uncover key insights on revenue and customer preferences. The goal is to identify the top-selling pizzas across different categories (e.g., Classic, Supreme) and determine the top 3 revenue-generating pizzas per category. Using SQL techniques such as joins, subqueries, and aggregate functions, we will extract actionable insights to help optimize business strategy and boost sales performance.

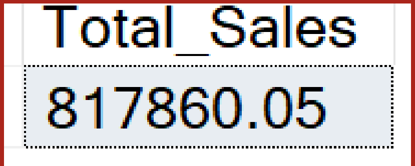
Calculate the total revenue generated from pizza sales.

Query:



```sql
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
```

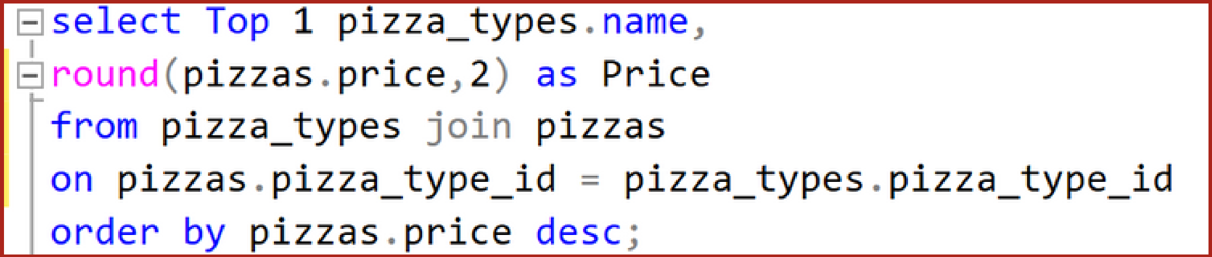
Output:



```
Total_Sales
817860.05
```

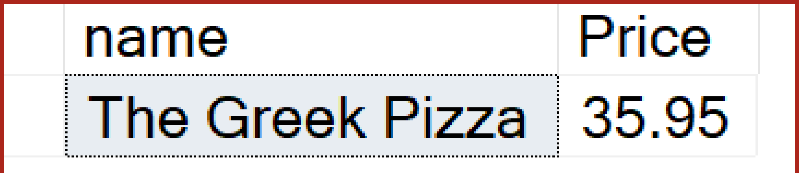
Identify the highest-priced pizza.

Query:



```sql
select Top 1 pizza_types.name,
round(pizzas.price,2) as Price
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc;
```

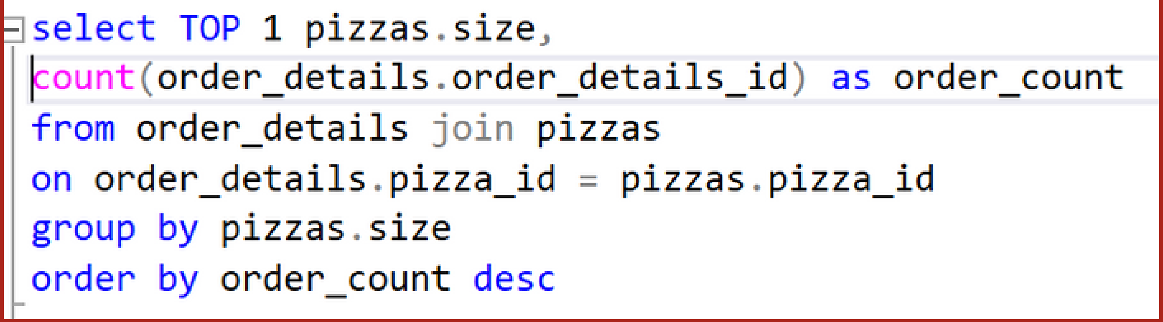
Output:



name	Price
The Greek Pizza	35.95

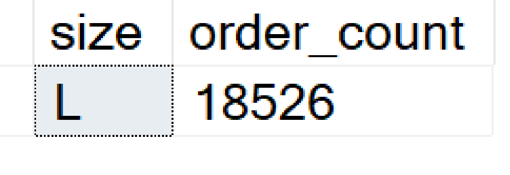
Identify the most common pizza size ordered.

Query:



```sql
select TOP 1 pizzas.size,
count(order_details.order_details_id) as order_count
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by order_count desc
```

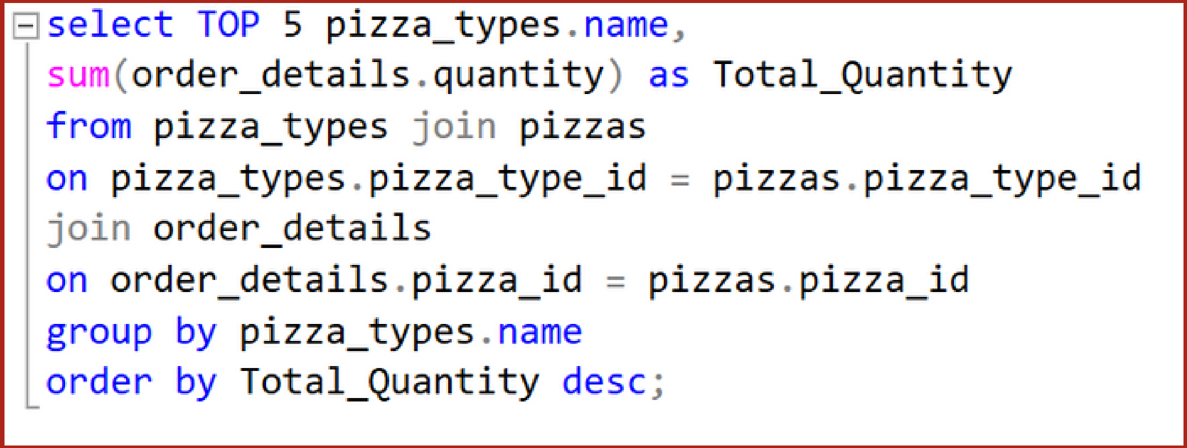
Output:



size	order_count
L	18526

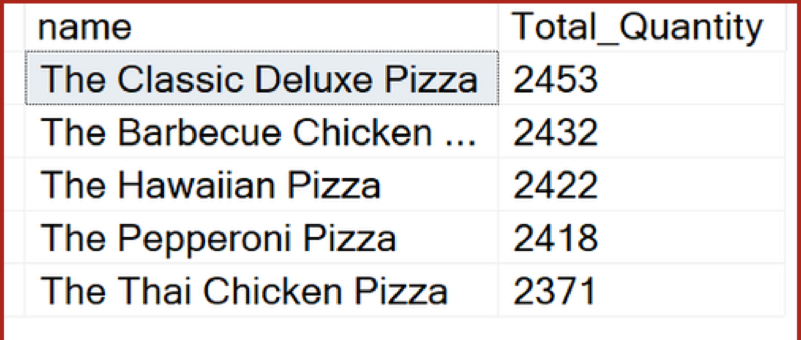
List the top 5 most ordered pizza types along with their quantities.

Query:



```
select TOP 5 pizza_types.name,
sum(order_details.quantity) as Total_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 Total_Quantity desc;
```

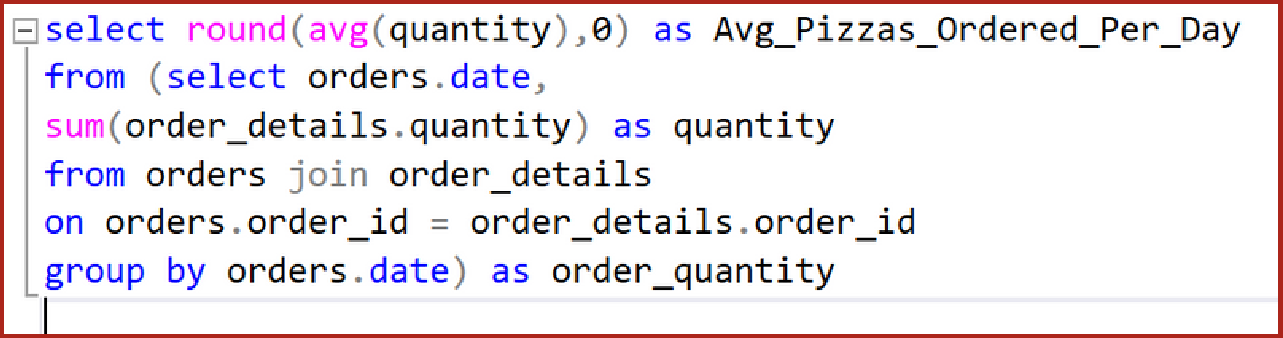
Output:



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

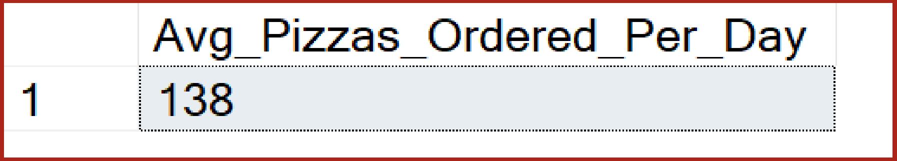
Group the orders by date and calculate the average number of pizzas ordered per day.

Query



```sql
select round(avg(quantity),0) as Avg_Pizzas_Ordered_Per_Day
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 order_quantity
```

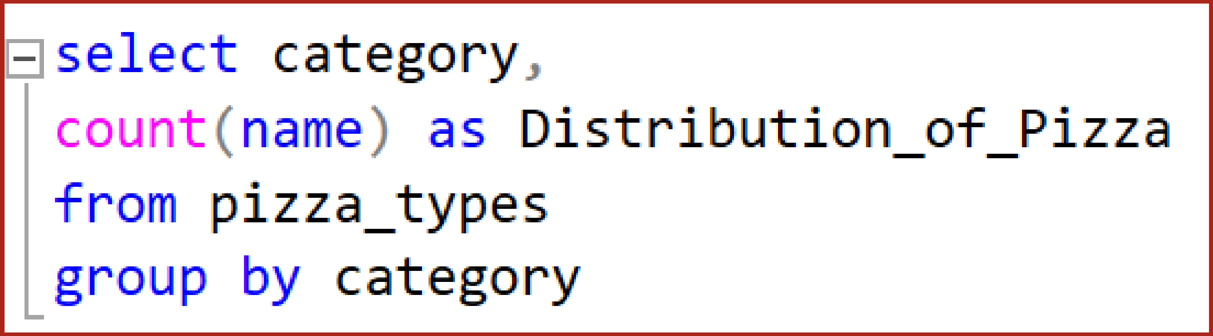
Output:



	Avg_Pizzas_Ordered_Per_Day
1	138

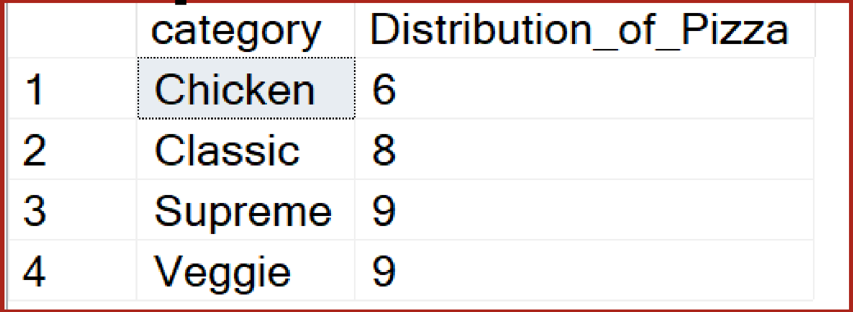
Find the category-wise distribution of pizzas.

Query:



```
select category,
count(name) as Distribution_of_Pizza
from pizza_types
group by category
```

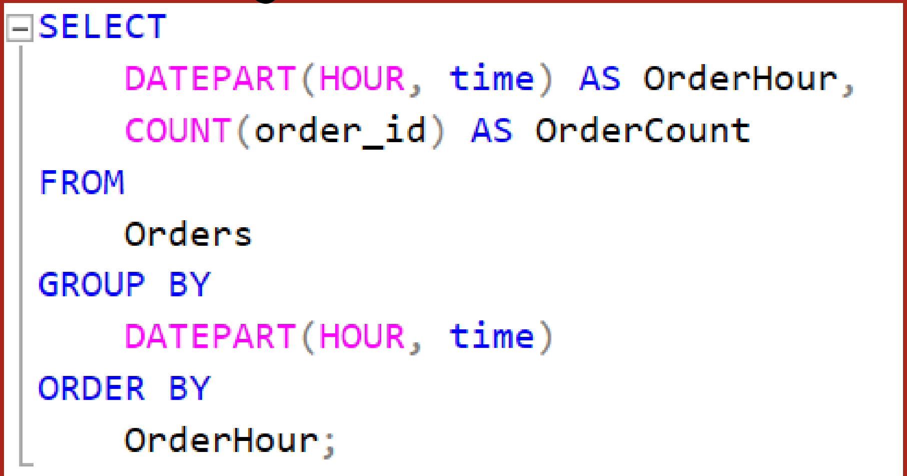
Output:



	category	Distribution_of_Pizza
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

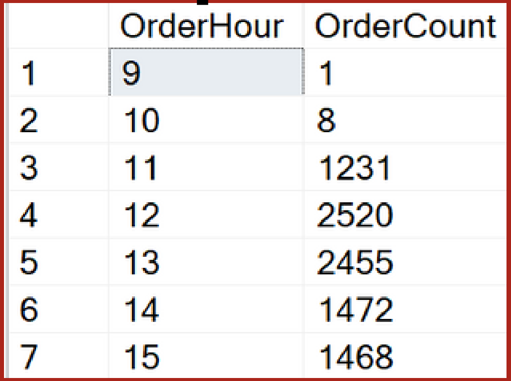
Determine the distribution of orders by hour of the day.

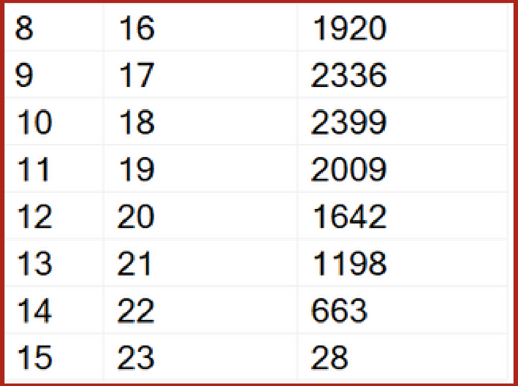
Query:



```
SELECT
    DATEPART(HOUR, time) AS OrderHour,
    COUNT(order_id) AS OrderCount
FROM
    Orders
GROUP BY
    DATEPART(HOUR, time)
ORDER BY
    OrderHour;
```

Output:

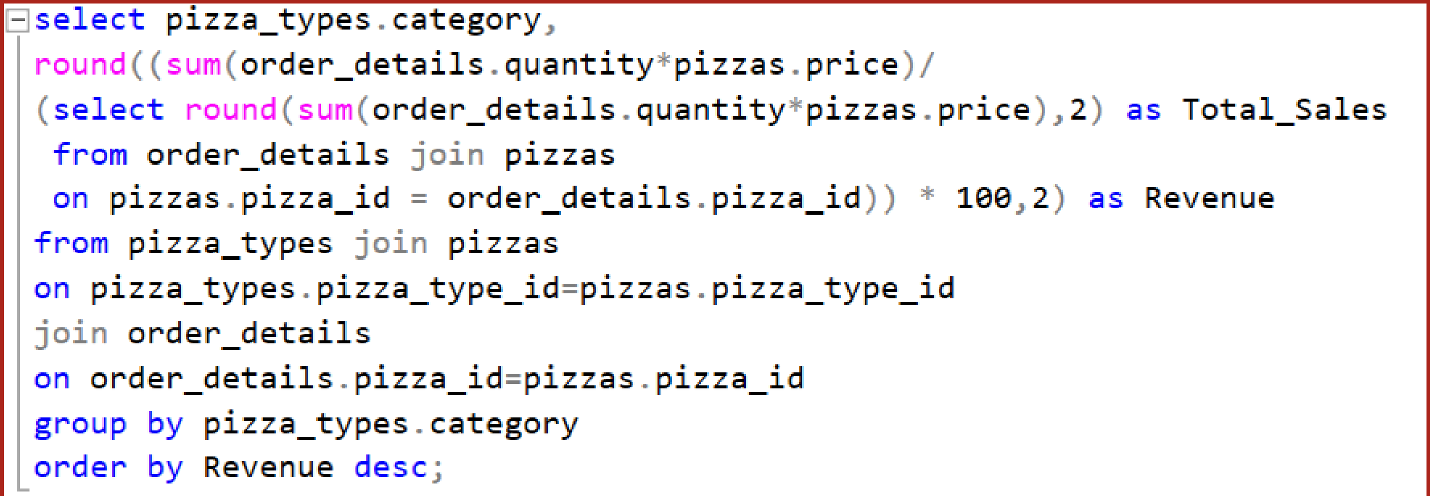




	OrderHour	OrderCount
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

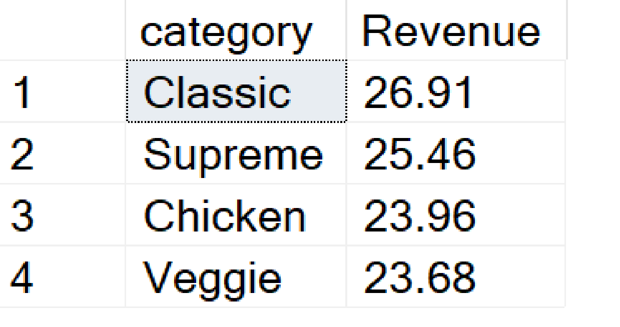
Calculate the percentage contribution of each pizza type to total revenue.

Query:



```
select pizza_types.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 pizza_types.category
order by Revenue desc;
```

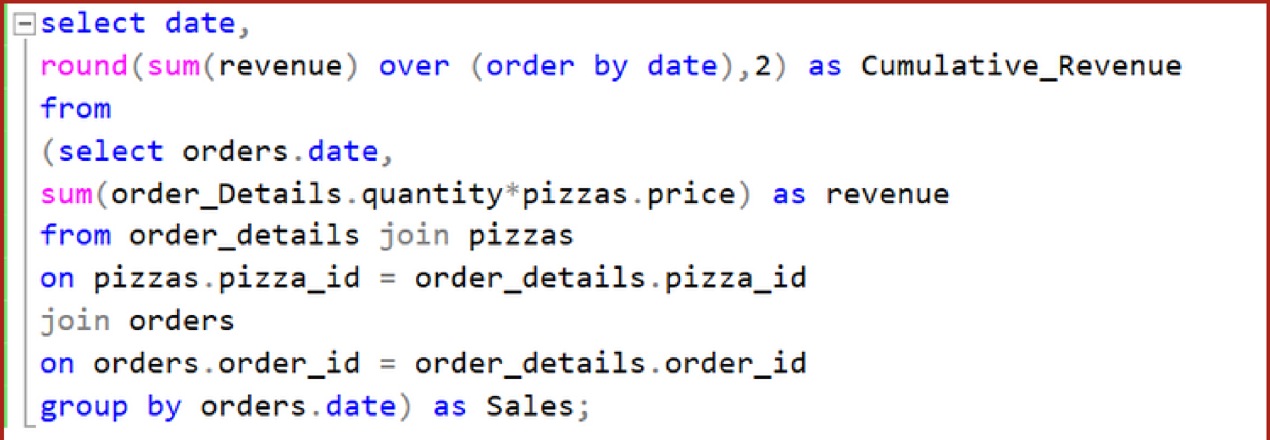
Output:



	category	Revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

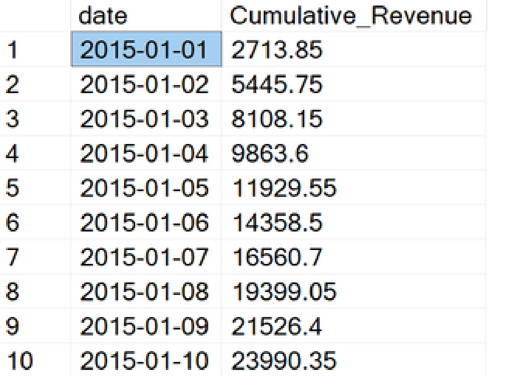
Analyze the cumulative revenue generated over time.

Query:



```sql
select date,
round(sum(revenue) over (order by date),2) as Cumulative_Revenue
from
(select orders.date,
sum(order_Details.quantity*pizzas.price) as revenue
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as Sales;
```

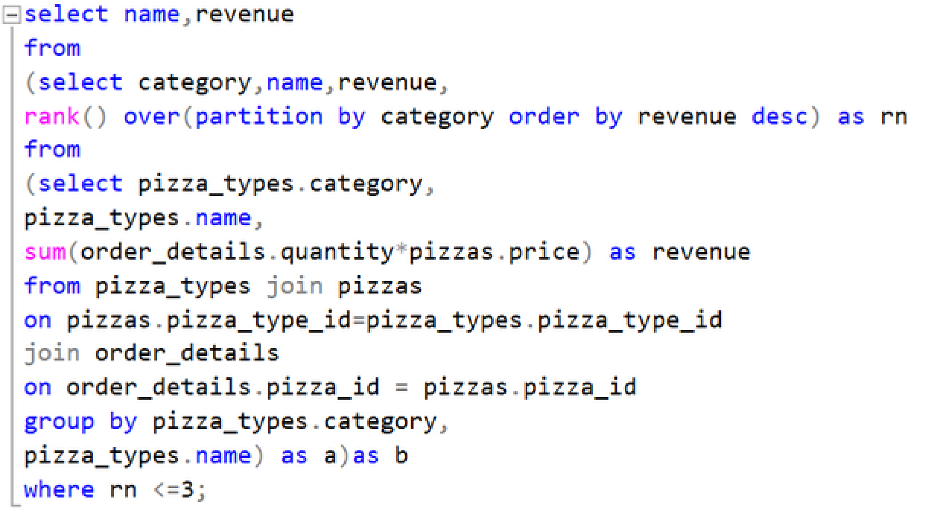
Output:



	date	Cumulative_Revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.35

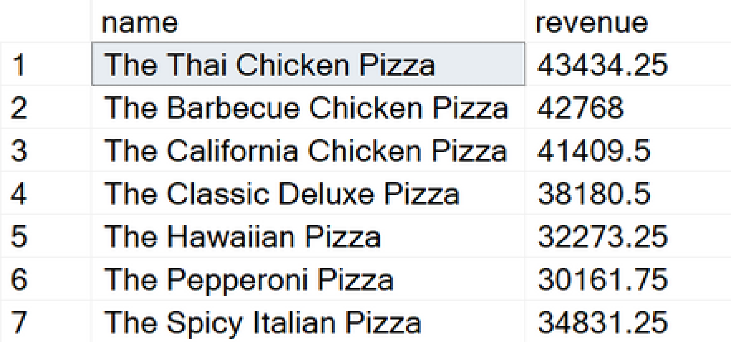
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

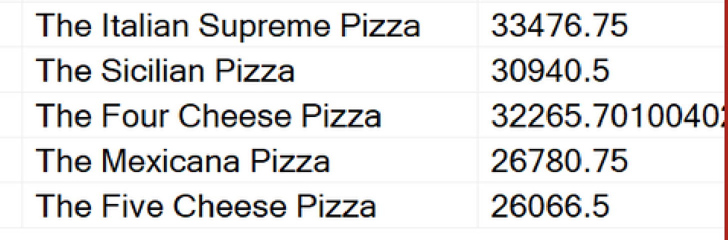
Query:



```
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 pizzas.pizza_type_id=pizza_types.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;
```

Output:





	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.7010040
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5