



# ***Pizza Sales Analysis Using SQL***

BY  
MALIHA SHEHERIN



# Project Objective



- Analyze pizza sales data to extract actionable business insights.
- Use SQL to answer key questions related to revenue, popular items, and customer behavior.
- Identify patterns and trends to help improve business decisions.



# Dataset Overview

- **Source:** GitHub dataset (pizza\_sales.zip)
- **Tables:**
  - orders: order ID, date,time
  - order\_details: order\_details\_id, order\_id,pizza\_id, quantity
  - pizzas: Pizza\_id, pizza\_type\_id,size, price
  - pizza\_types: pizza\_type\_id,name, category, ingredients





# Retrieve the total number of orders placed.

Query:

```
SELECT
```

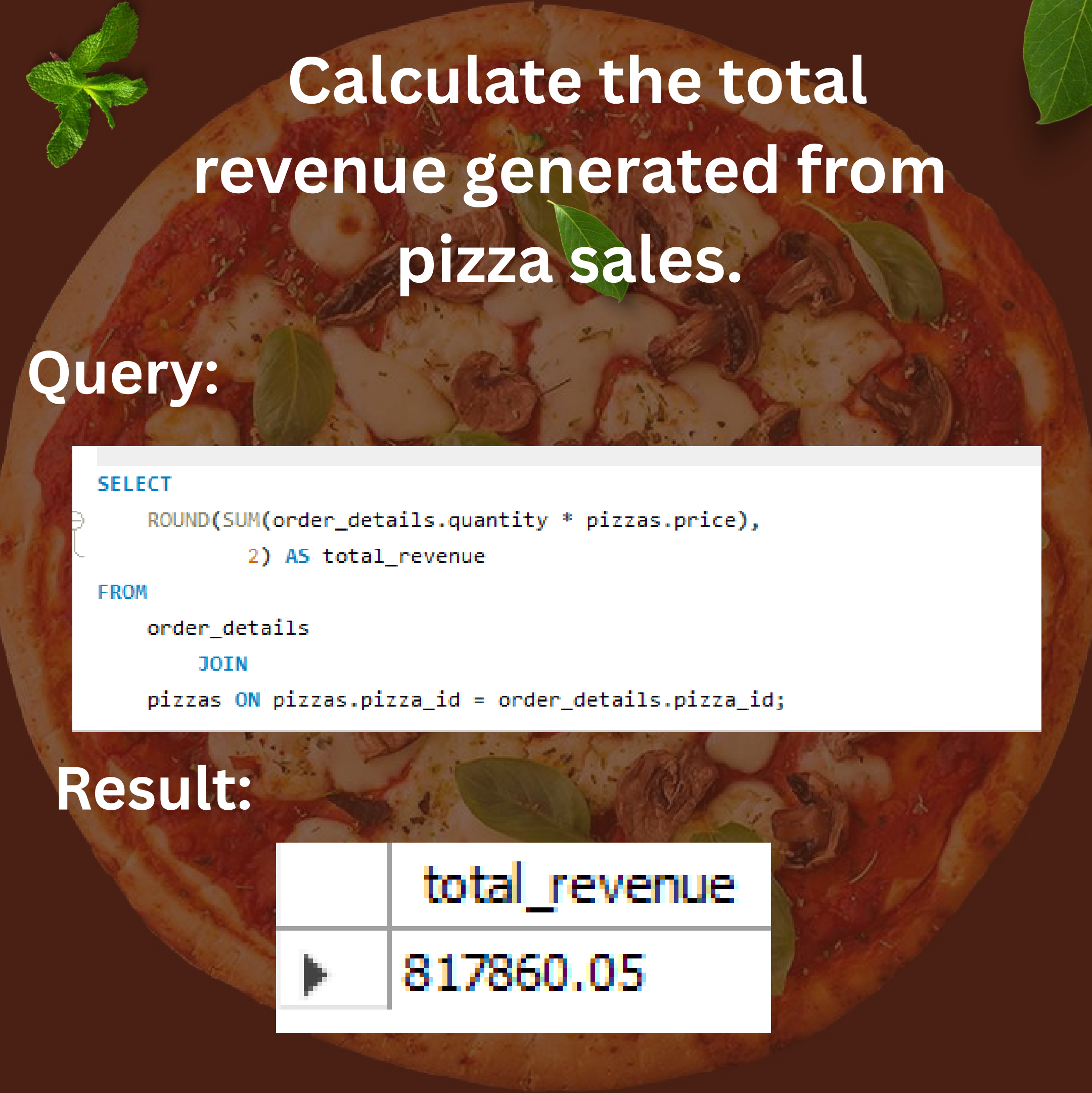
```
    COUNT(order_id) AS total_orders
```

```
FROM
```

```
orders;
```

Result:

	total_orders
▶	21350



# Calculate the total revenue generated from pizza sales.

Query:

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

Result:

	total_revenue
▶	817860.05



# 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;
```

Result:

	name	price
►	The Greek Pizza	35.95



# Identify the most frequently ordered pizza size.

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;
```

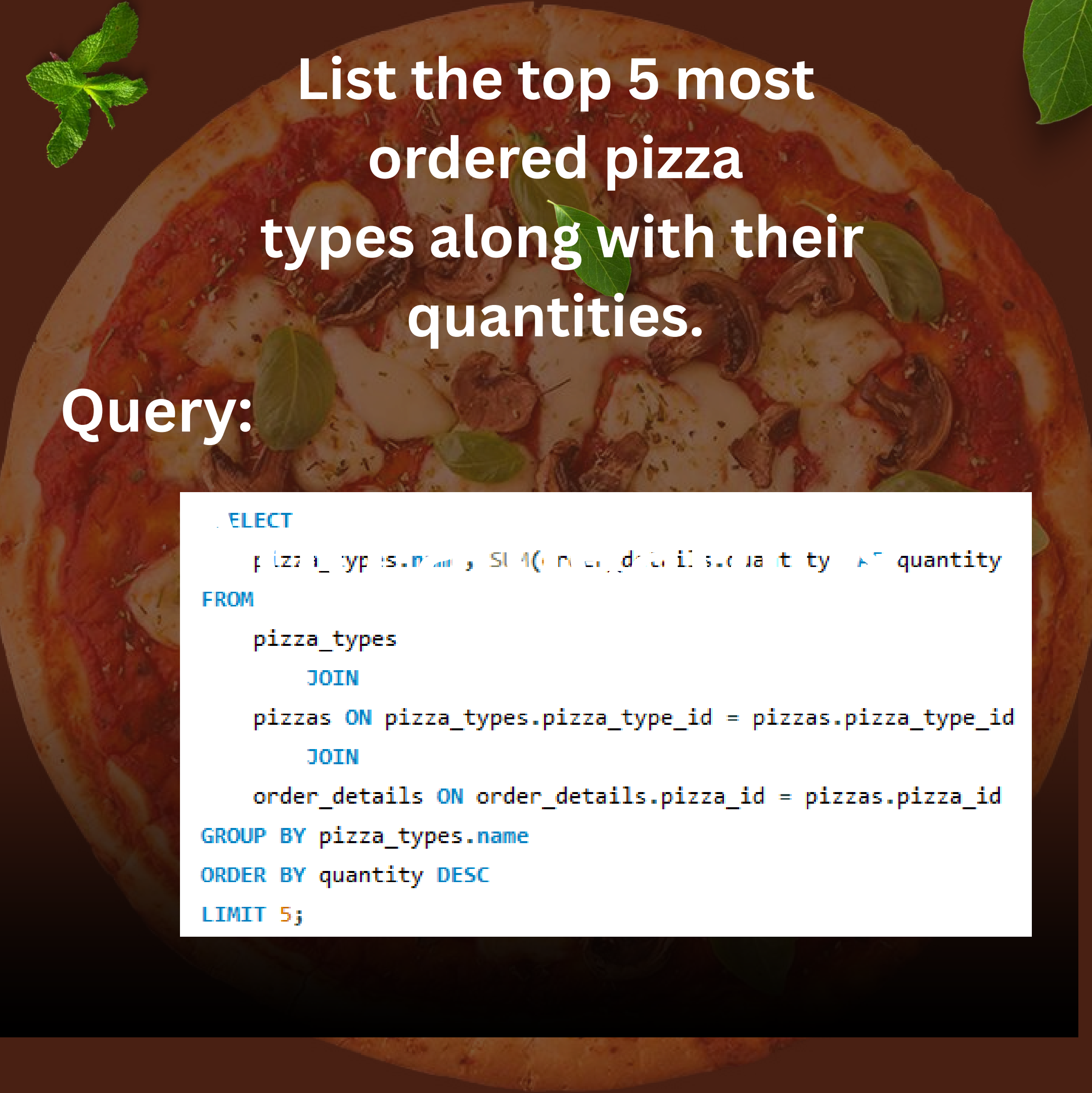




Result:

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28






# 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;
```

Result:

	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



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

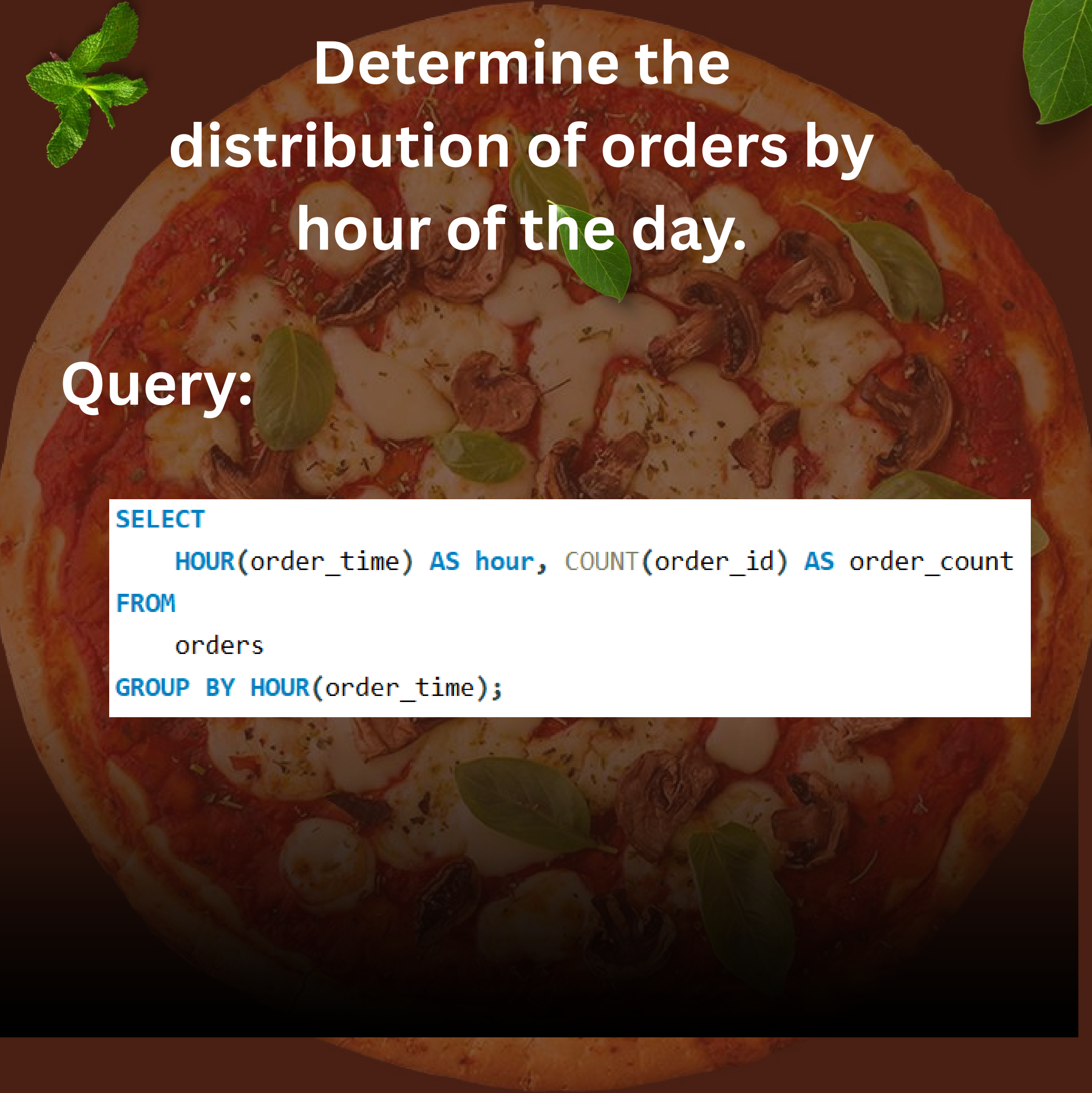
Query:

```
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 pizza_types.category
ORDER BY quantity DESC;
```



Result:

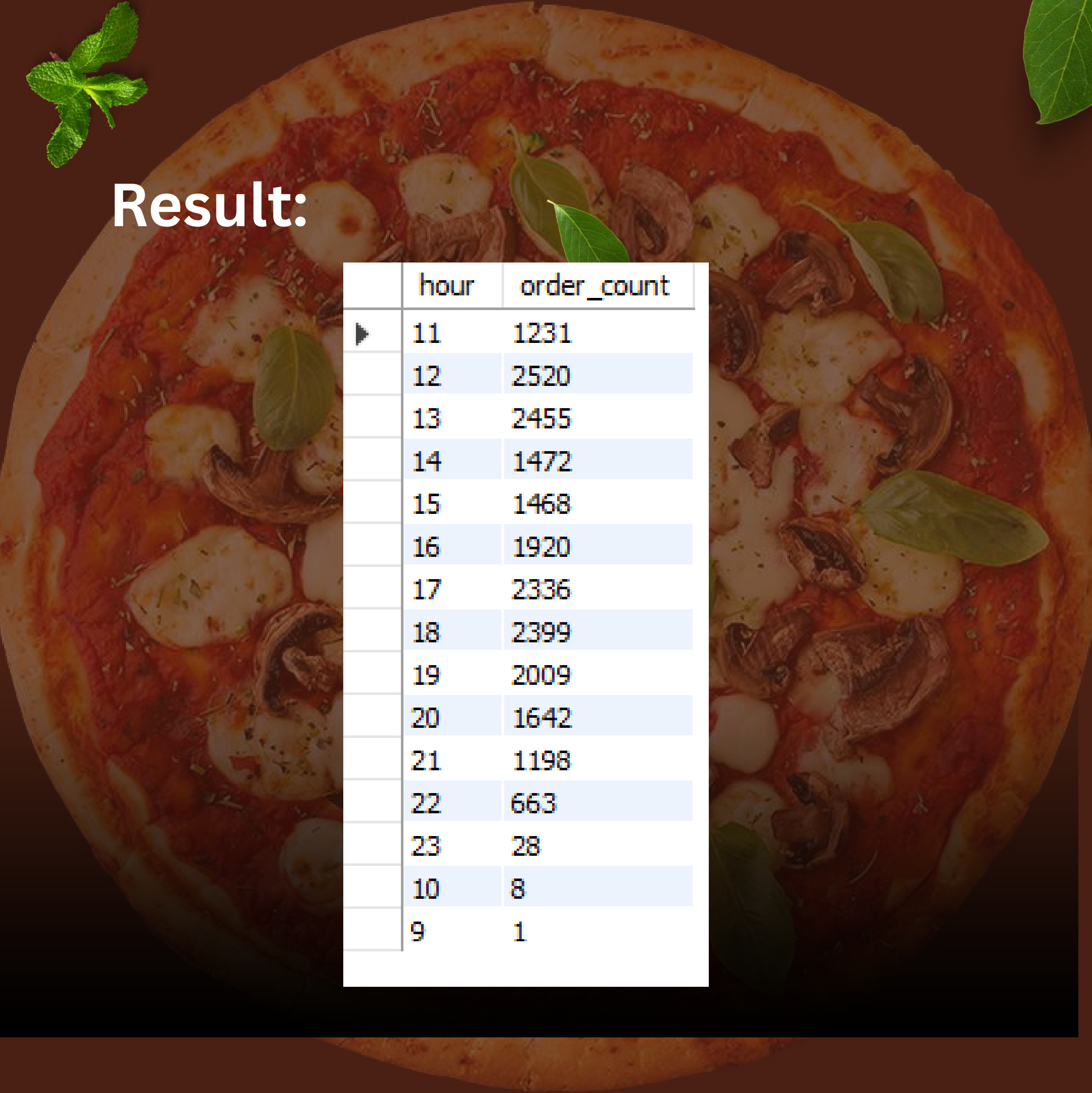
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the  
distribution of orders by  
hour of the day.

Query:

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```



# Result:

	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
	10	8
	9	1





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

Query:

```
SELECT
```

```
    category, COUNT(name)
```

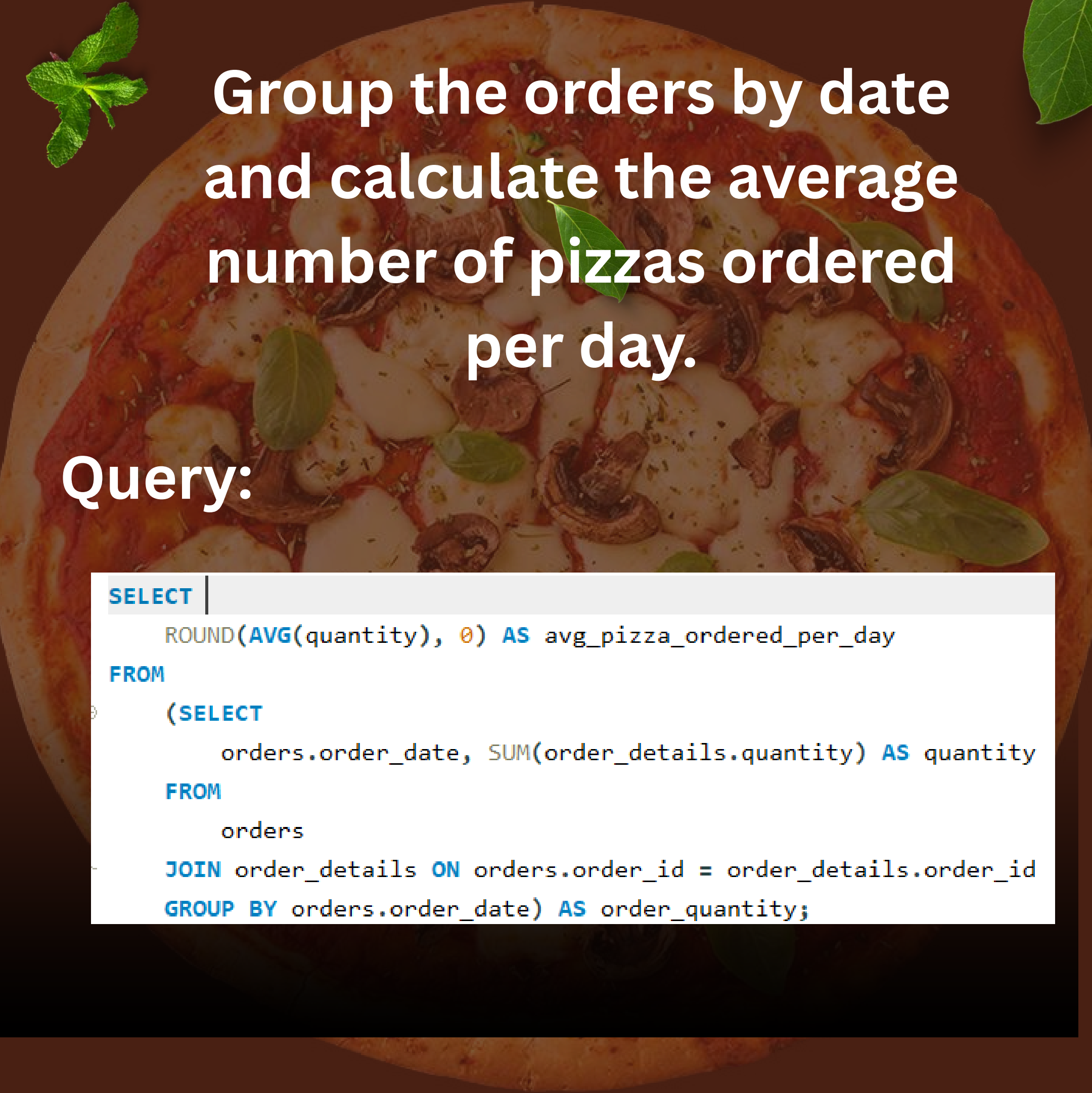
```
FROM
```

```
    pizza_types
```

```
GROUP BY category;
```

Result:

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

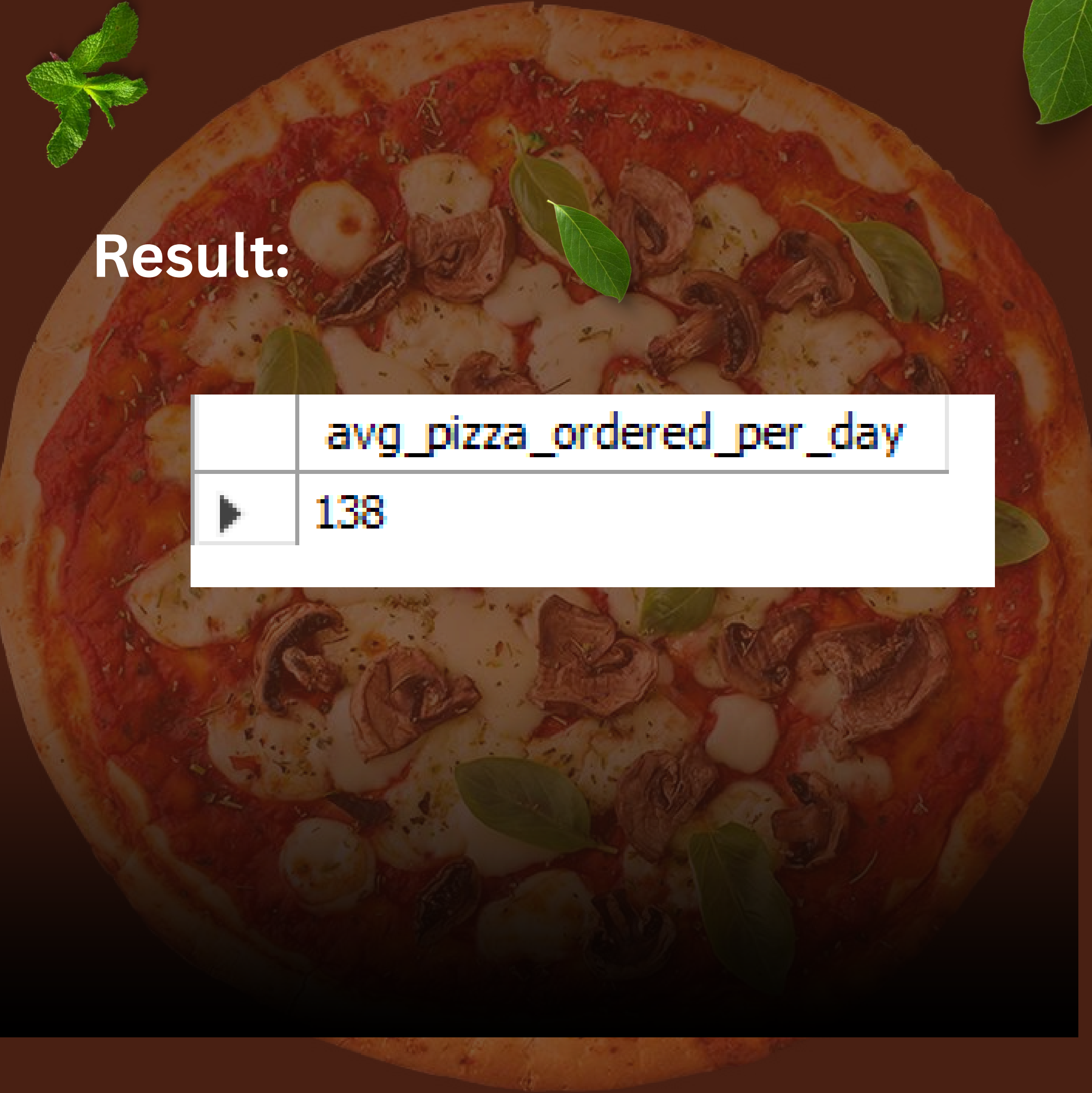


Group the orders by date  
and calculate the average  
number of pizzas ordered  
per day.

Query:

```
SELECT  
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```





Result:

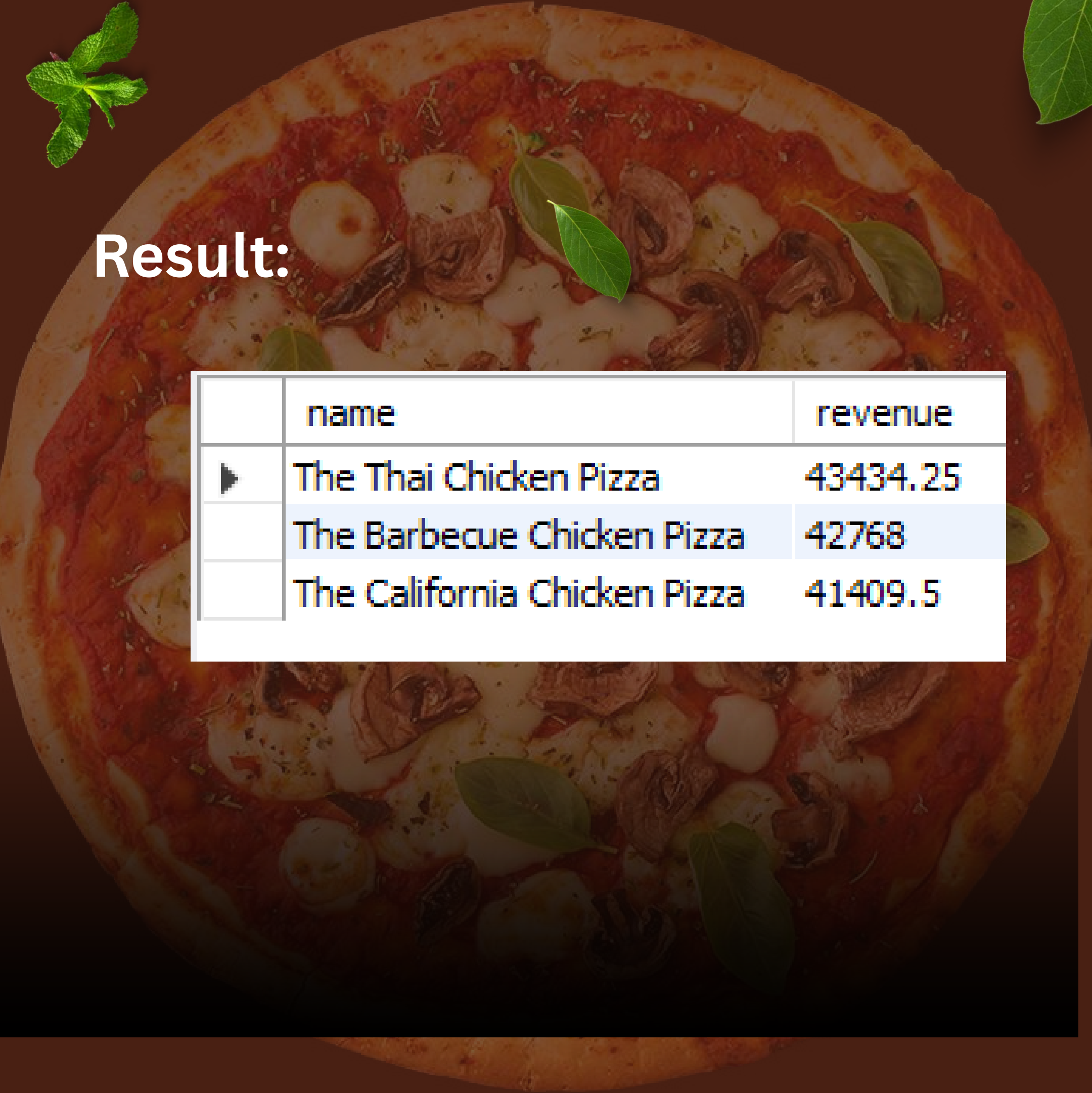
	avg_pizza_ordered_per_day
▶	138



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

Query:

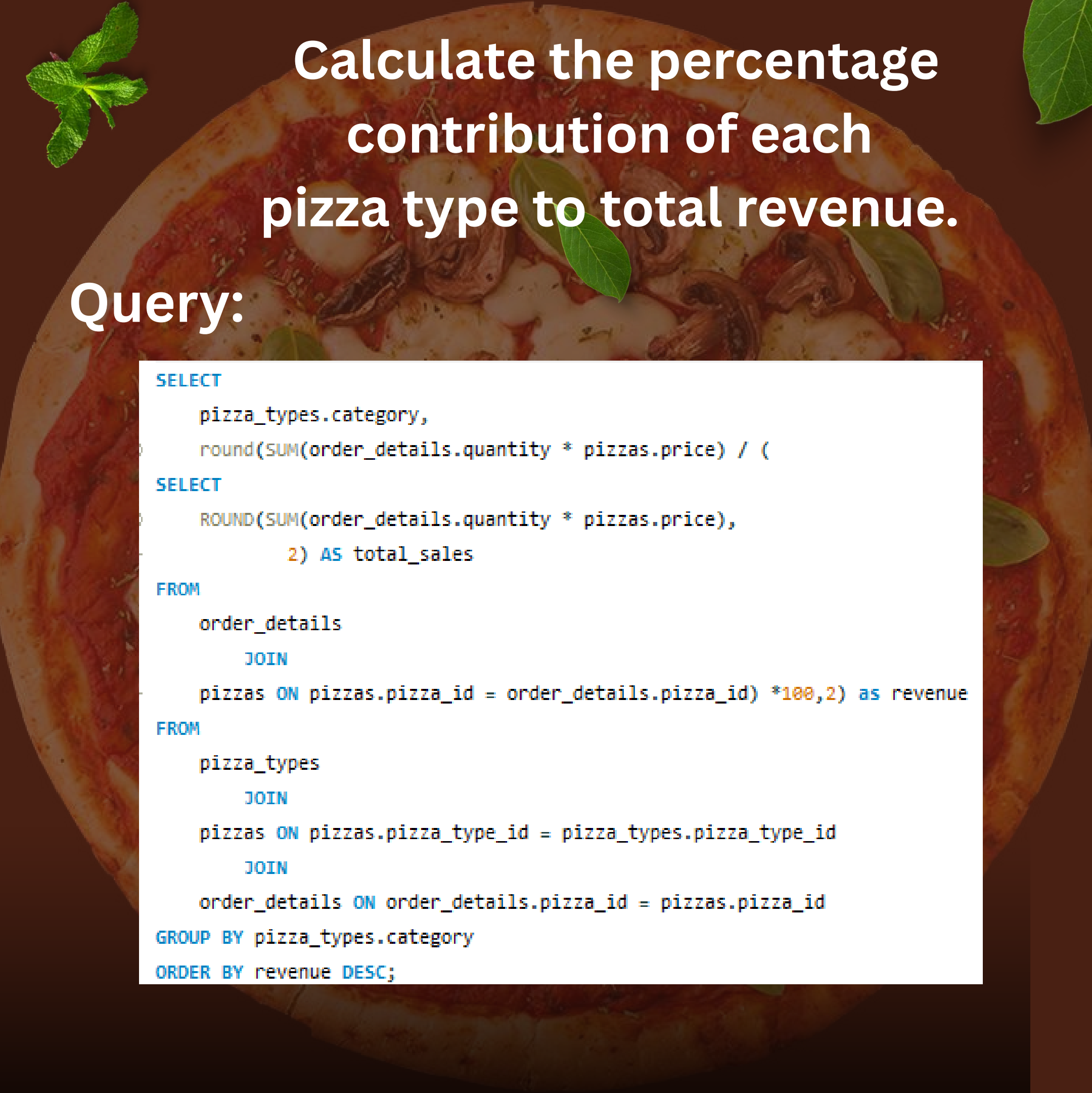
```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```



Result:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5






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

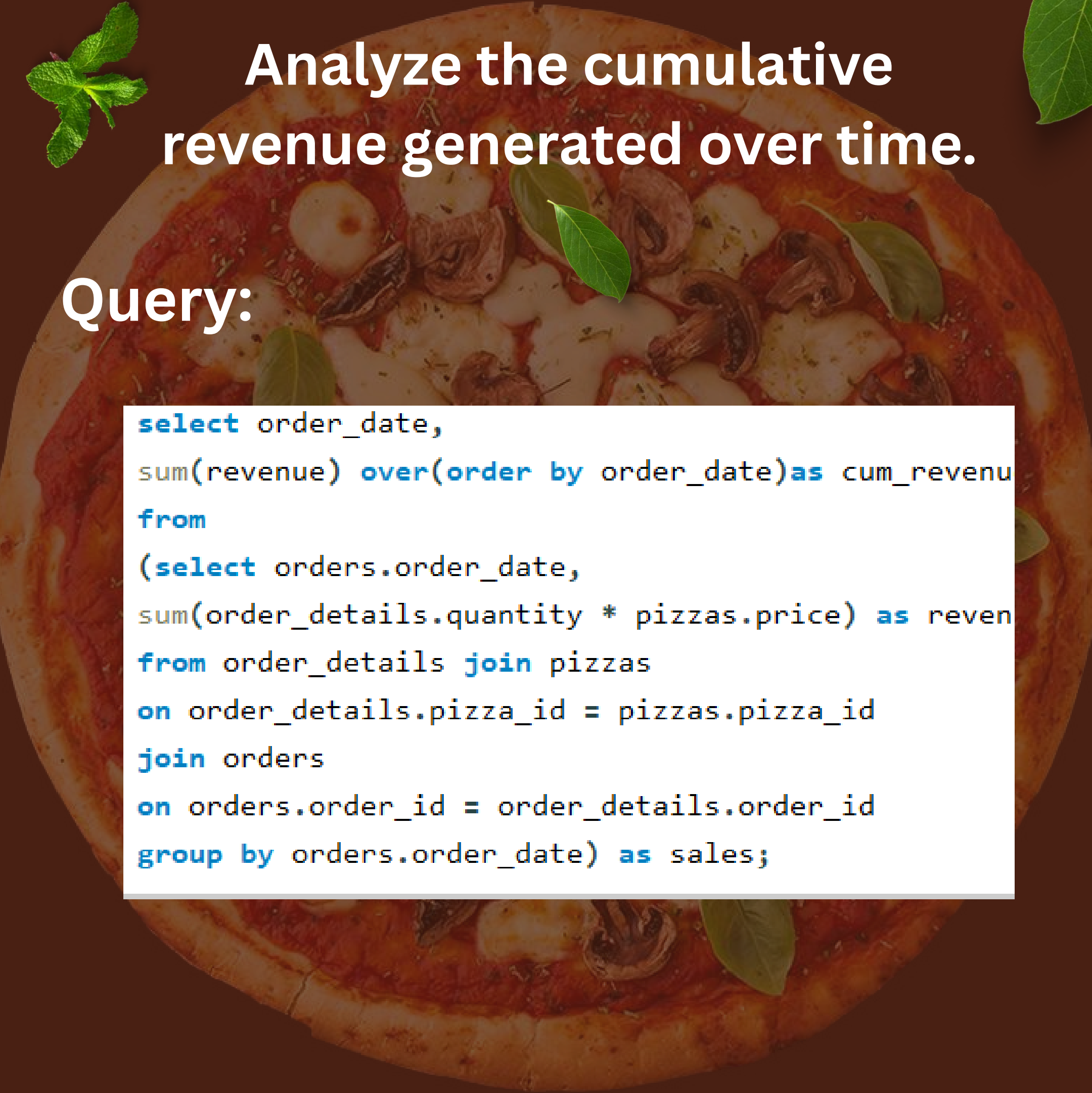
## Query:

```
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 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
ORDER BY revenue DESC;
```



Result:

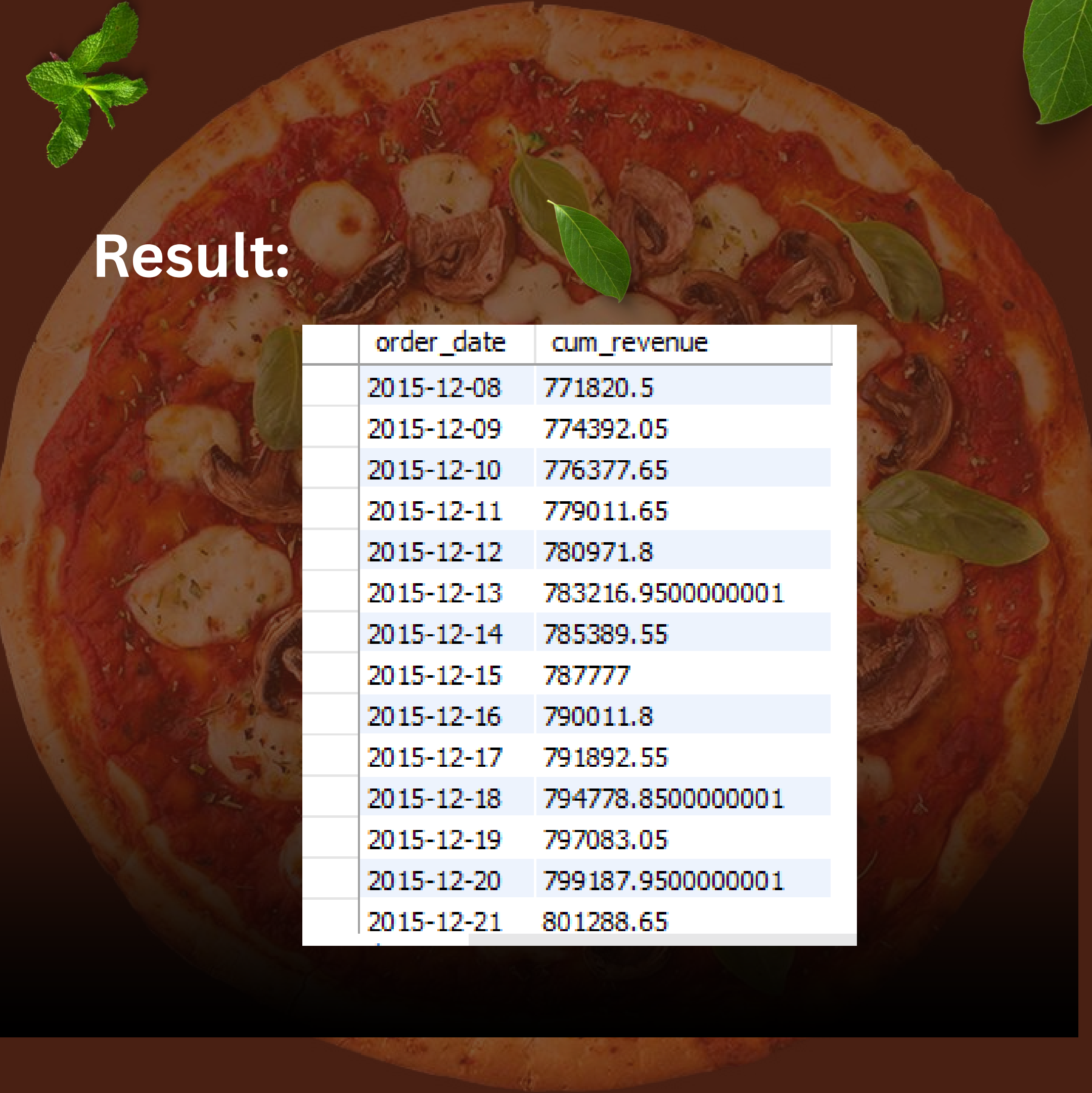
	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



# Analyze the cumulative revenue generated over time.

Query:

```
select order_date,  
sum(revenue) over(order by order_date)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;
```



# Result:

	order_date	cum_revenue
	2015-12-08	771820.5
	2015-12-09	774392.05
	2015-12-10	776377.65
	2015-12-11	779011.65
	2015-12-12	780971.8
	2015-12-13	783216.9500000001
	2015-12-14	785389.55
	2015-12-15	787777
	2015-12-16	790011.8
	2015-12-17	791892.55
	2015-12-18	794778.8500000001
	2015-12-19	797083.05
	2015-12-20	799187.9500000001
	2015-12-21	801288.65





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;
```



# Result:

	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 SupThe Sicilian Pizza	76.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



# Key Insights Summary



- Total revenue: \$817,860.05
- Best-selling pizza: Classic Deluxe Medium
- Peak order time: 12 PM – 1 PM
- Friday has highest number of orders
- Most revenue generated from Classic and Supreme categories





# Recommendations

- Focus marketing campaigns during peak hours and weekends.
- Offer combo deals for top-selling pizzas.
- Stock up inventory in July and December for high demand.
- Introduce loyalty programs to boost weekday sales.





THANK YOU