Comprehensive Analysis of Pizza Sales 🍕

Introduction:

In the highly competitive food industry, understanding sales patterns and customer preferences is essential for optimizing business strategies.

This **Pizza Sales Performance Analysis** aims to provide valuable insights into the sales performance of different pizza types, sizes, and categories. This project focuses on analyzing key metrics such as total orders, revenue generation, and the most popular pizza types. By leveraging data analytics, the goal is to help pizza businesses make informed decisions, improve customer satisfaction, and increase profitability.

Description:

This analysis uses sales data from a pizza company to uncover trends and performance indicators. By addressing the following questions, this project will highlight the most relevant aspects of the pizza sales process and help the business enhance its offerings and operational efficiency:

➞ Find out the total number of Pizza 🍕 orders placed:

Understanding how many orders have been placed is fundamental in measuring business activity and growth.

QUERY ⬇

```
select count(order_id) as Total_Orders
from orders
```

Results | Messages

	Total_Orders
1	21350

➞ Calculate the Total Revenue generated from 🍕 Pizza Sales:

Determining the total revenue will reveal the financial success of the pizza sales and help gauge the overall business performance.

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
```

	Total_Revenue
1	817860.05

➞ Calculate the highest-priced pizza 🍕 :

Knowing the highest-priced pizza helps in analyzing the pricing strategy and its impact on sales.

QUERY

```
select top 1 pizza_types.name, pizzas.price as highest_priced_pizza
from pizza_types
join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by highest_priced_pizza desc
```

	name	highest_priced_pizza
1	The Greek Pizza	35.9500007629395

➞ Identify the most common 🍕 pizza size ordered:

By identifying the most popular pizza size, the business can adjust inventory and production to meet customer demand.

QUERY

```
select top 1 pizzas.size, count(order_details.quantity)
as No_of_times_ordered
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by No_of_times_ordered desc
```

	size	No_of_times_ordered
1	L	18526

➞ List out the top 5 most ordered 🍕 pizza types along with their quantities:

Highlighting the most popular pizza types provides insight into customer preferences, allowing for better menu optimization.

QUERY

```
select top 5 pizza_types.name, sum(order_details.quantity)
as Top_5_most_ordered_pizza_types
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 Top_5_most_ordered_pizza_types desc;
```

	name	Top_5_most_ordered_pizza_types
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

➞ Find the total quantity of each pizza category ordered:

Combining different data tables enables a deeper understanding of how different pizza categories are performing in terms of orders.

QUERY ⬇

```
select pizza_types.category, sum(order_details.quantity)
as Total_Quantity_by_Pizzaa_Category 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 Total_Quantity_by_Pizzaa_Category desc;
```

	category	Total_Quantity_by_Pizzaa_Category
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

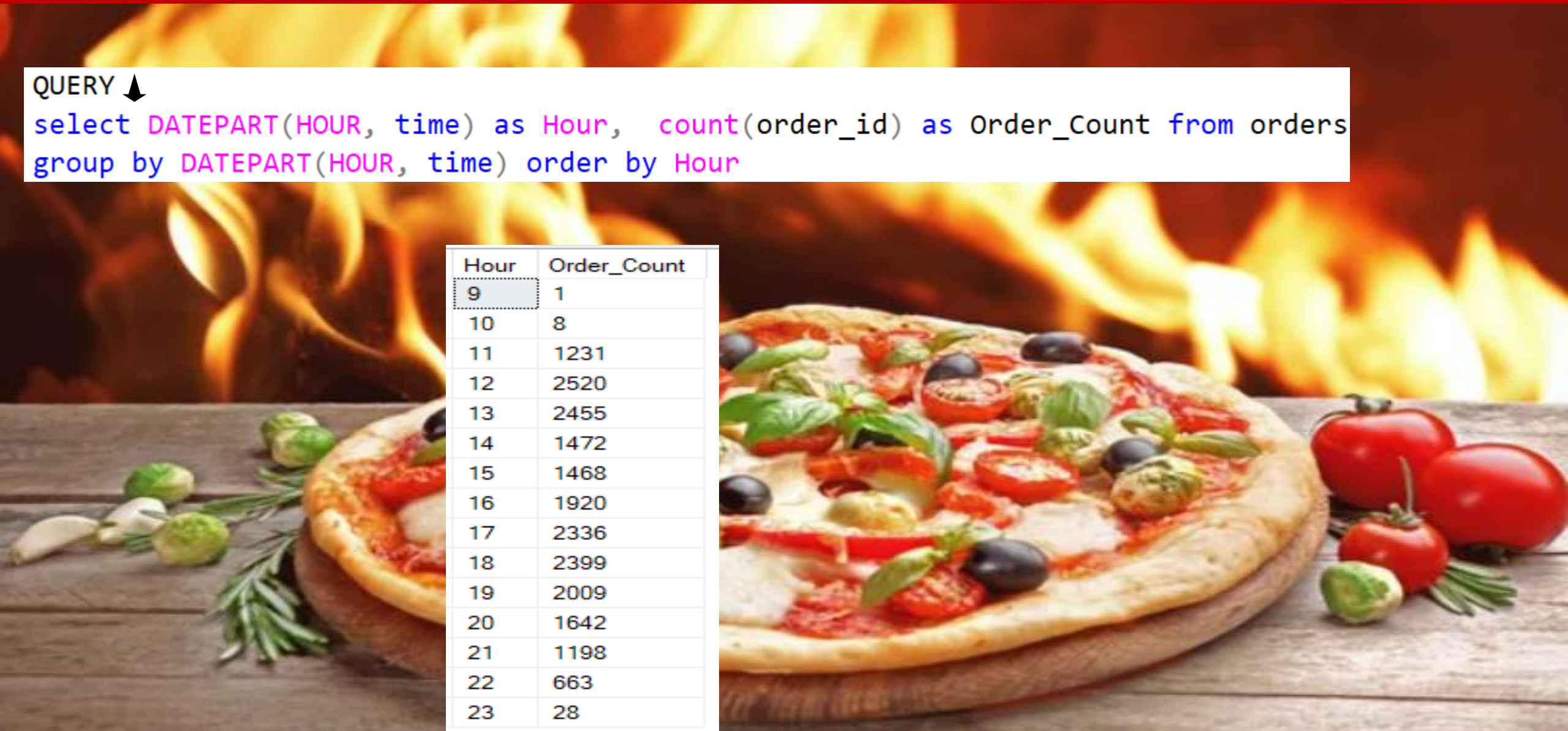
➞ Calculate the distribution of orders by hour of the day:

Analyzing order distribution by time helps in identifying peak business hours and optimizing staffing and supply chain logistics.

QUERY ⬇

```
select DATEPART(HOUR, time) as Hour,  count(order_id) as Order_Count from orders
group by DATEPART(HOUR, time) order by Hour
```

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



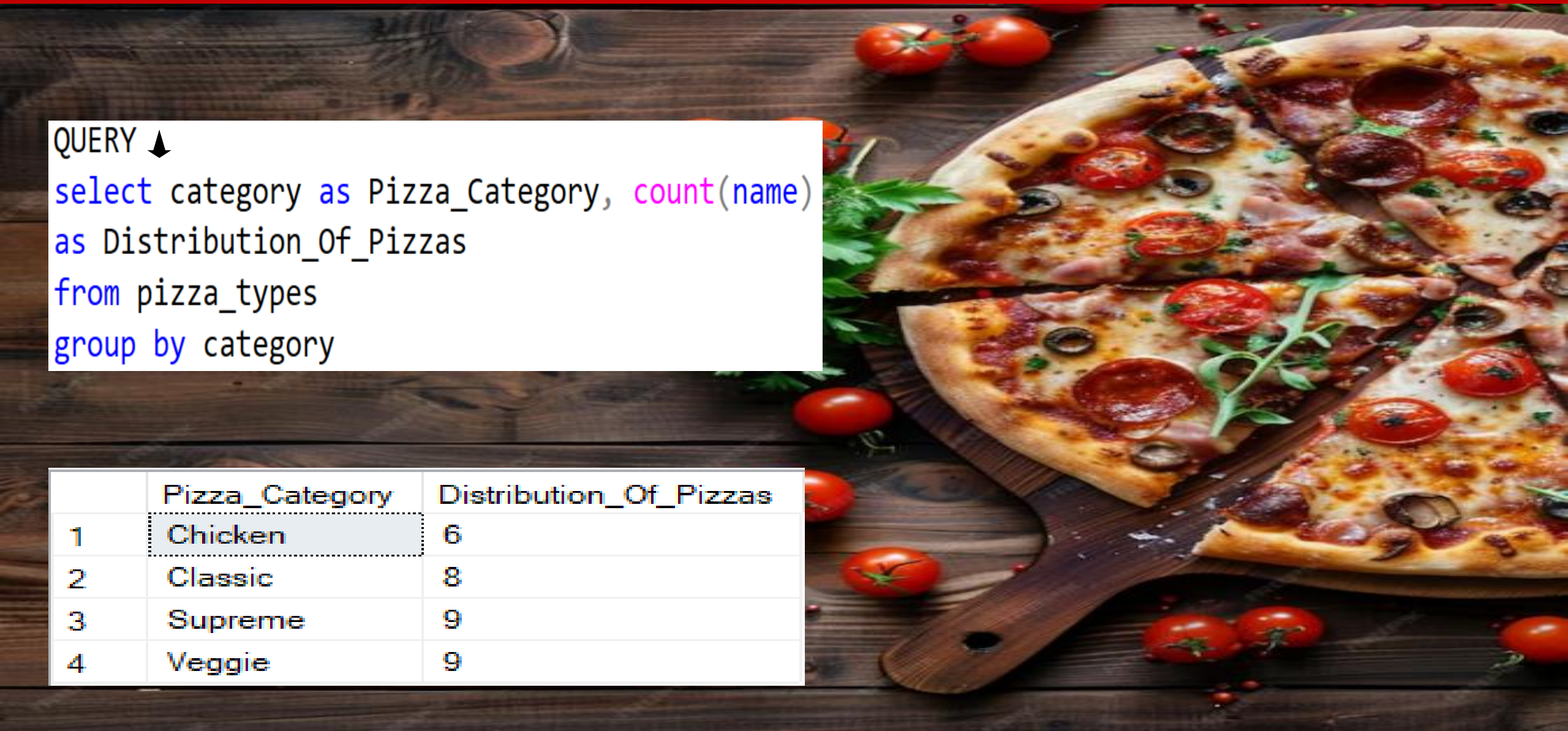
➞ Find the category-wise distribution of 🍕pizzas:

This step will further analyze the sales performance by category, offering insights into which categories are more popular.

QUERY ⬇

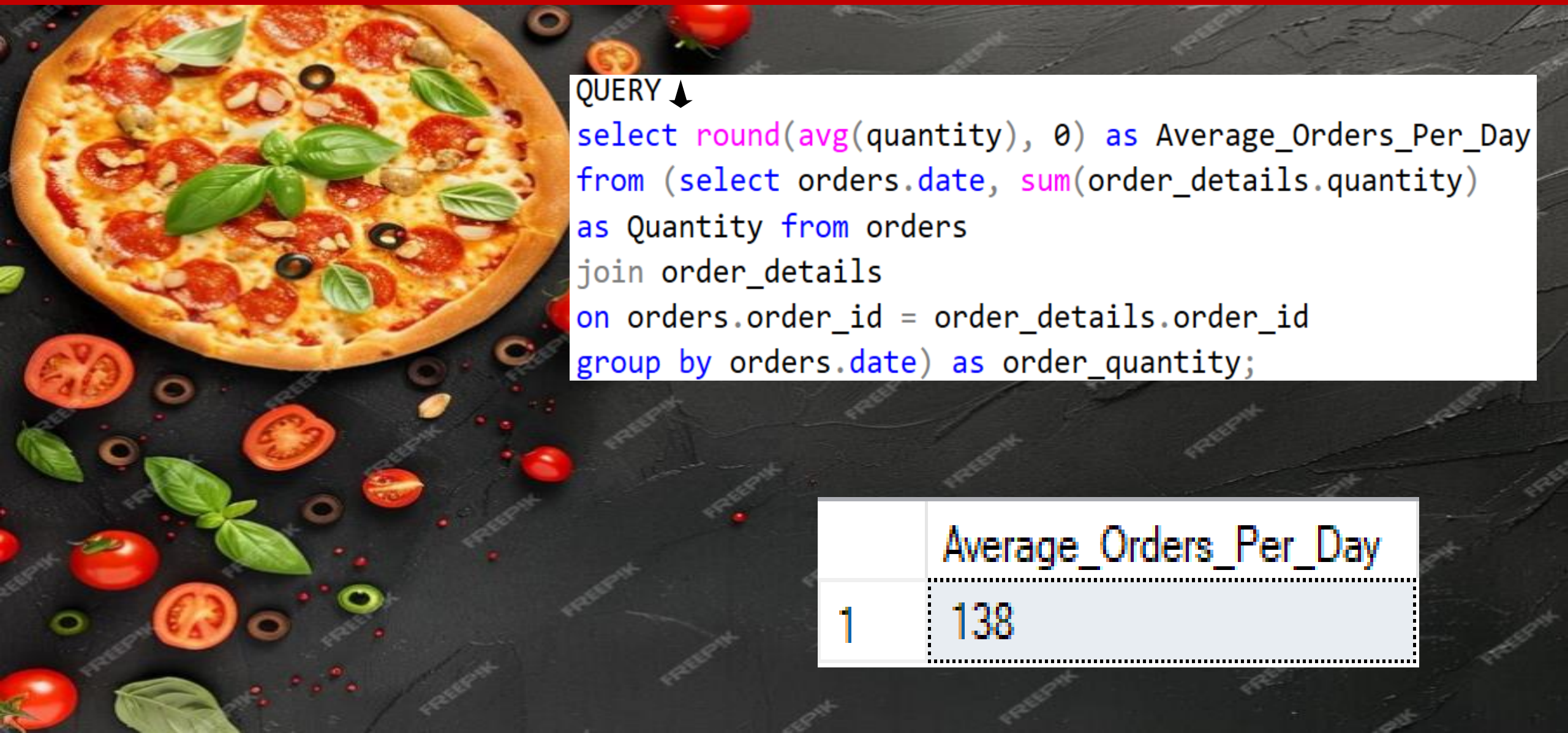
```
select category as Pizza_Category, count(name)
as Distribution_Of_Pizzas
from pizza_types
group by category
```

	Pizza_Category	Distribution_Of_Pizzas
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9



➞ Find the orders by date and calculate average number of Pizzas ordered per day:

Grouping orders by date and calculating averages allows for trend analysis and can help predict future demand.



QUERY

```
select round(avg(quantity), 0) as Average_Orders_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;
```

	Average_Orders_Per_Day
1	138

➞ Determine the top 3 most ordered 🍕 Pizza types based on revenue:

Identifying the top revenue-generating pizzas helps the business focus on promoting its best-performing products.

QUERY ⬇

```
select top 3 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
```

	name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

➞ Evaluate the % contribution of each pizza type to total revenue:

This analysis will show the significance of each pizza type in the overall revenue and help make decisions on promotions or menu changes.

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;
```

	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:

Cumulative revenue analysis provides a timeline view of financial performance, identifying trends or periods of growth and decline.

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 order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as Sales;
```

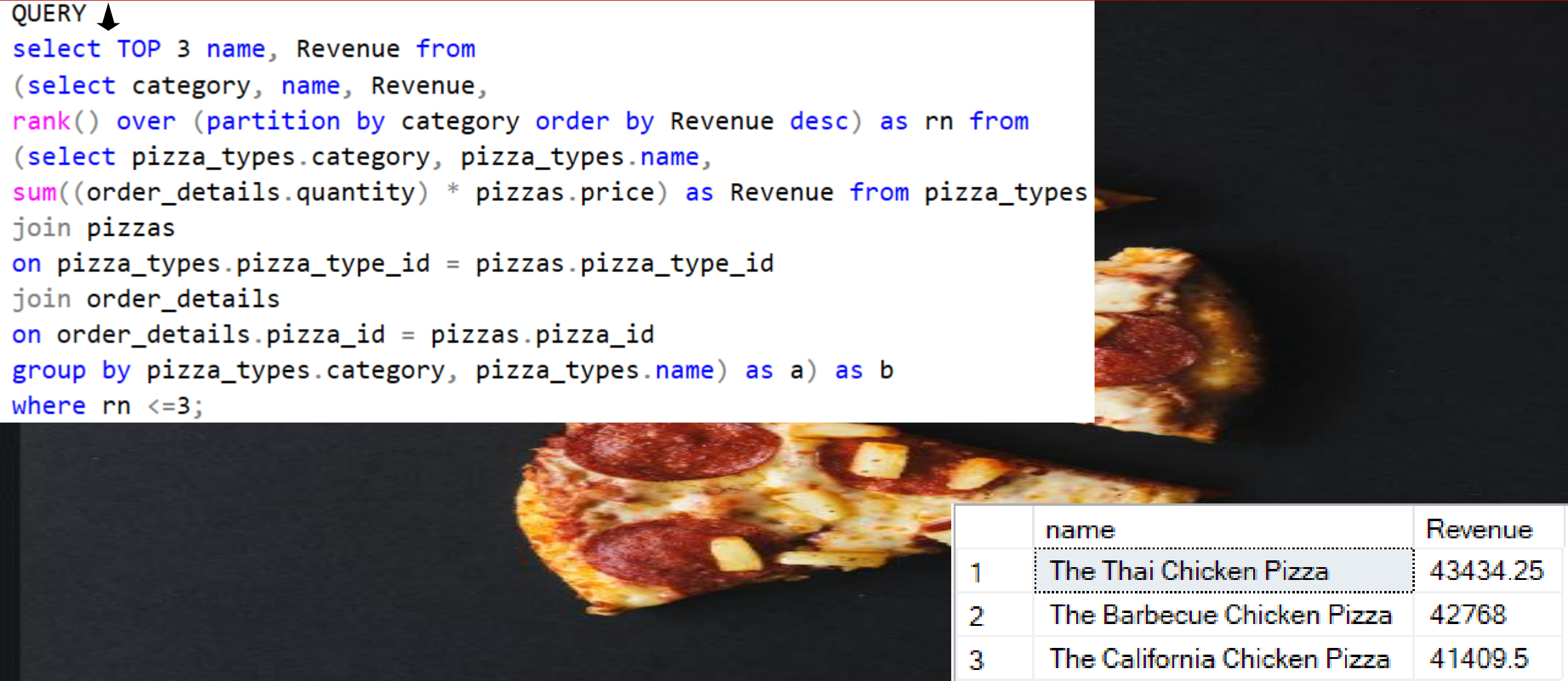
	date	Cumulative_Revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.3
14	2015-01-14	32358.7
15	2015-01-15	34343.5
16	2015-01-16	36937.65
17	2015-01-17	39001.75
18	2015-01-18	40978.6
19	2015-01-19	43365.75
20	2015-01-20	45763.65
21	2015-01-21	47804.2
22	2015-01-22	50300.9
23	2015-01-23	52724.6
24	2015-01-24	55013.85
25	2015-01-25	56631.4
26	2015-01-26	58515.8
27	2015-01-27	61043.85
28	2015-01-28	63059.85
29	2015-01-29	65105.15
30	2015-01-30	67375.45
31	2015-01-31	69793.3
32	2015-02-01	72982.5
33	2015-02-02	75311.1
34	2015-02-03	77925.9
35	2015-02-04	80159.8
36	2015-02-05	82375.6

➞ **Determine the top 3 most ordered 🍕 Pizza types based on revenue for each pizza category:**

Breaking down the most profitable pizzas by category gives a detailed view of which pizzas are driving success in each segment.

QUERY

```
select TOP 3 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
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5


THANK
YOU