

# HELLO!

**"My name is Reema. In this SQL project on pizza sales, I used queries to analyze data, answer key questions, and gain valuable insights into customer preferences, sales trends, and inventory management."**



### **Q:Basic:**

***Retrieve the total number of orders placed.***

***Calculate the total revenue generated from pizza sales.***

***Identify the highest-priced pizza.***

***Identify the most common pizza size ordered.***

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

### **Q:Intermediate:**

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

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

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

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

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

### **Q:Advanced:**

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

***Analyze the cumulative revenue generated over time.***

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



*Retrieve the total number of orders placed.?*

```
SELECT
```

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

```
FROM
```

```
orders;
```

total\_orders

21350

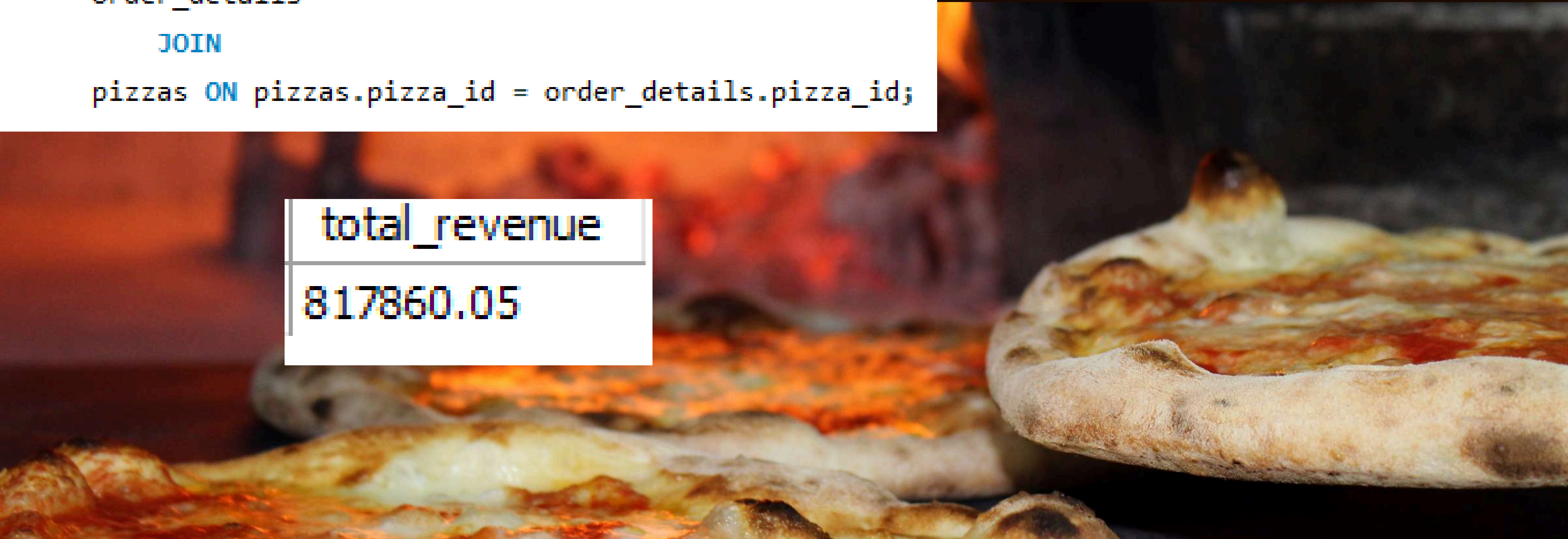




*Calculate the total revenue generated from  
pizza sales?*

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

total_revenue
817860.05





*Identify the highest-priced pizza?.*

```
SELECT  
    MAX(price)  
FROM  
    pizzas;
```

	MAX(price)
	35.95





*Identify the most common pizza size ordered?*

```
select quantity, count(order_id) from order_details group by quantity;  
SELECT  
    pizzas.size, COUNT(order_details.order_details_id)  
FROM  
    pizzas  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size limit 1;
```

size	COUNT(order_details.order_details_id)
M	15385



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

```
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;
```



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?*

```
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;
```

category	quantity
Chicken	11050
Classic	14888
Supreme	11987
Veggie	11649



*Determine the distribution of orders by hour of the day?*

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

hour	orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1100





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

```
• SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



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

```
SELECT
    ROUND(AVG(quantity), 0)
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;
```

`ROUND(AVG(quantity), 0)`

138

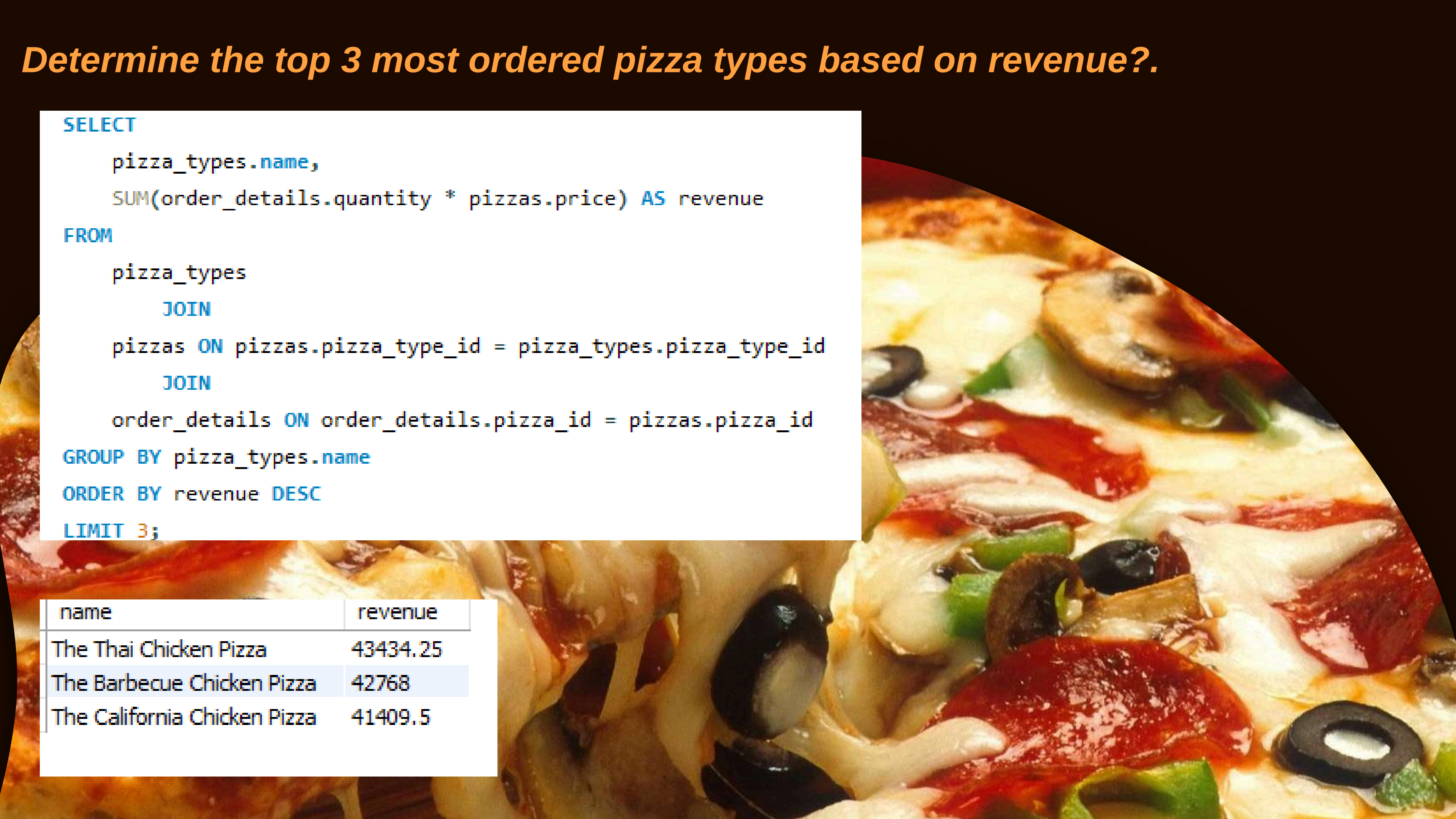




*Determine the top 3 most ordered pizza types based on revenue?.*

```
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;
```

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?

```
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;
```



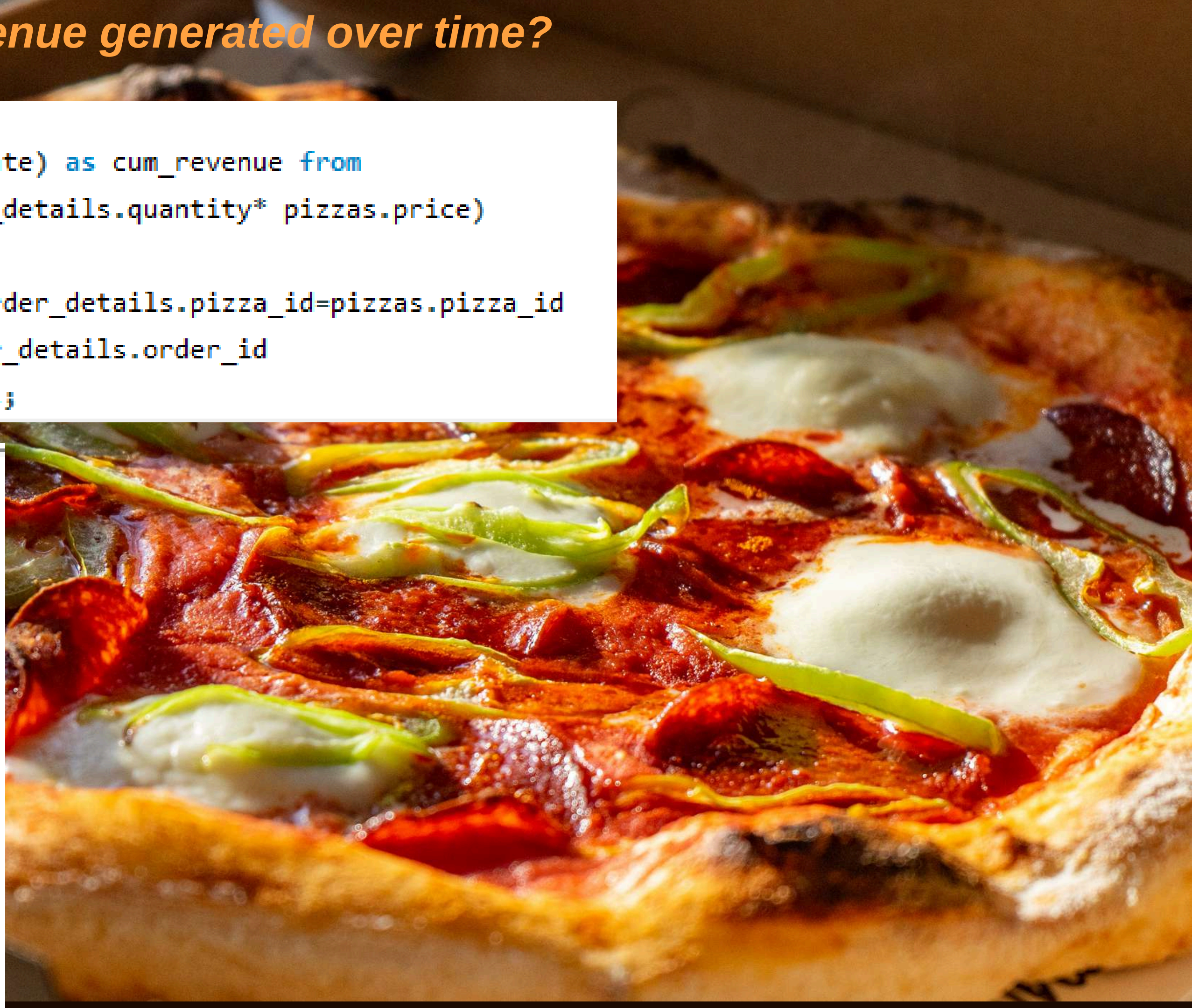
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



# Analyze the cumulative revenue generated over time?

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

order_date	cum_revenue
2015-01-28	63059.850000000001
2015-01-29	65105.1500000000016
2015-01-30	67375.450000000001
2015-01-31	69793.300000000002
2015-02-01	72982.500000000001
2015-02-02	75311.100000000002
2015-02-03	77925.900000000002
2015-02-04	80159.800000000002
2015-02-05	82375.600000000002
2015-02-06	84885.550000000002





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

```
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;
```



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



THANK YOU!

