

# ANALYSIS OF PIZZA SALES DATA USING SQL



# INTRODUCTION

This project analyzes pizza sales data to uncover key metrics and insights. By examining order quantities, revenue, popular pizza types, and distribution patterns, we aim to understand customer preferences and optimize business strategies. The following slides present our findings and recommendations based on this comprehensive data analysis.

Retrieve the total number of orders placed

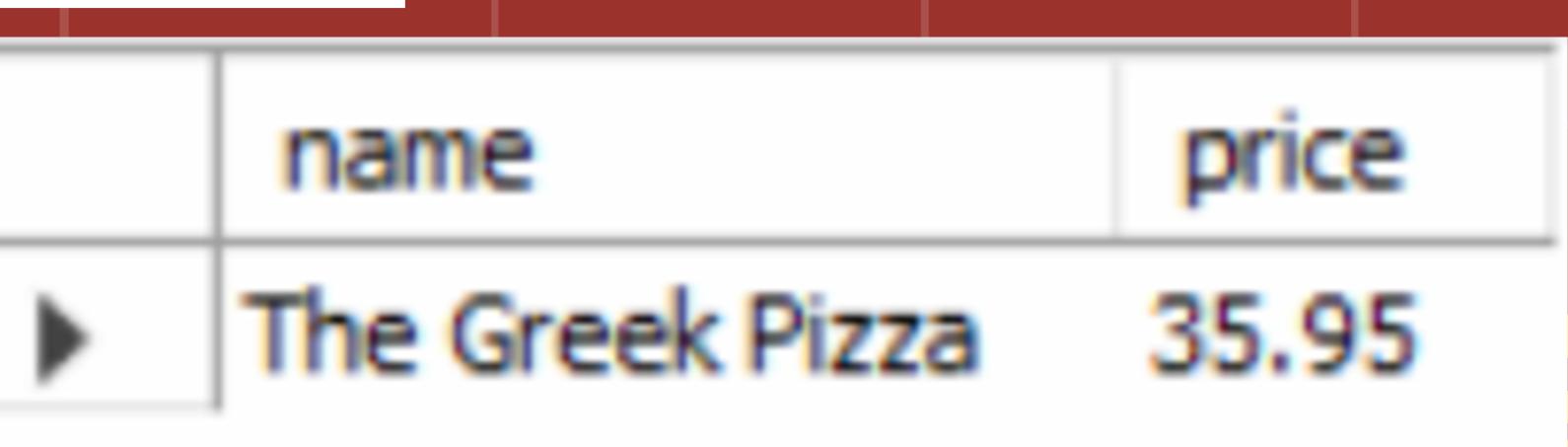
```
SELECT  
    COUNT(order_id) AS total_number_of_orders  
FROM  
    orders;
```

total_number_of_orders
21350



## Identify the highest-priced pizza

```
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
```

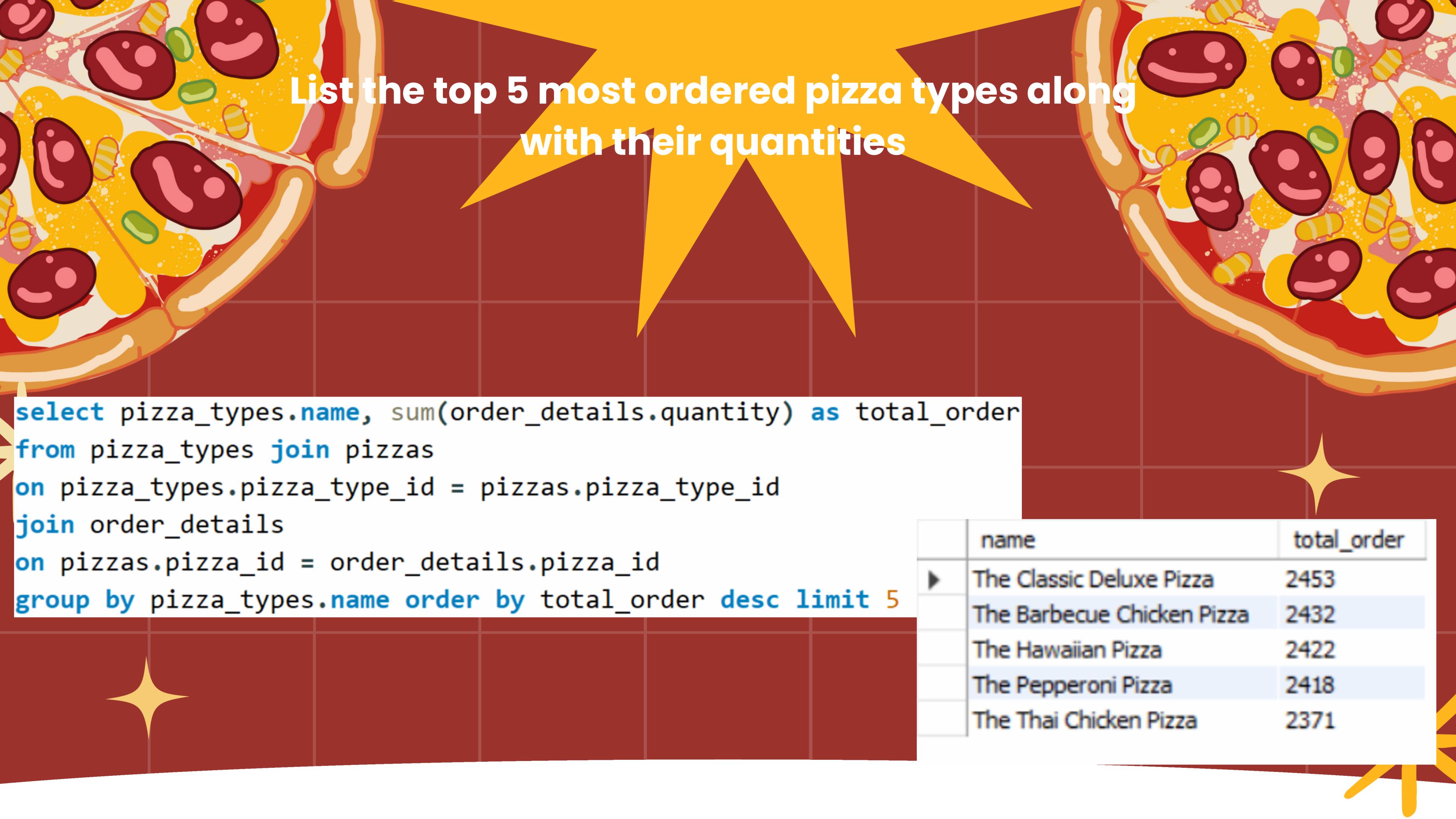


name	price
The Greek Pizza	35.95

# Identify the most common pizza size ordered

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

size	total_count
L	18526
M	15385
S	14137
XL	544
XXL	28



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

```
select pizza_types.name, sum(order_details.quantity) as total_order
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.name order by total_order desc limit 5
```

	name	total_order
▶	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 total_quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category order by total_quantity desc
```

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

# Determine the distribution of orders by hour of the day

```
select hour(time) as hour, count(order_id) as total_order  
from orders  
group by hour order by hour asc
```

hour	total_order
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 relevant tables to find the category-wise distribution of pizzas



```
select category, count(name) as 'category-wise distribution'  
from pizza_types  
group by category
```



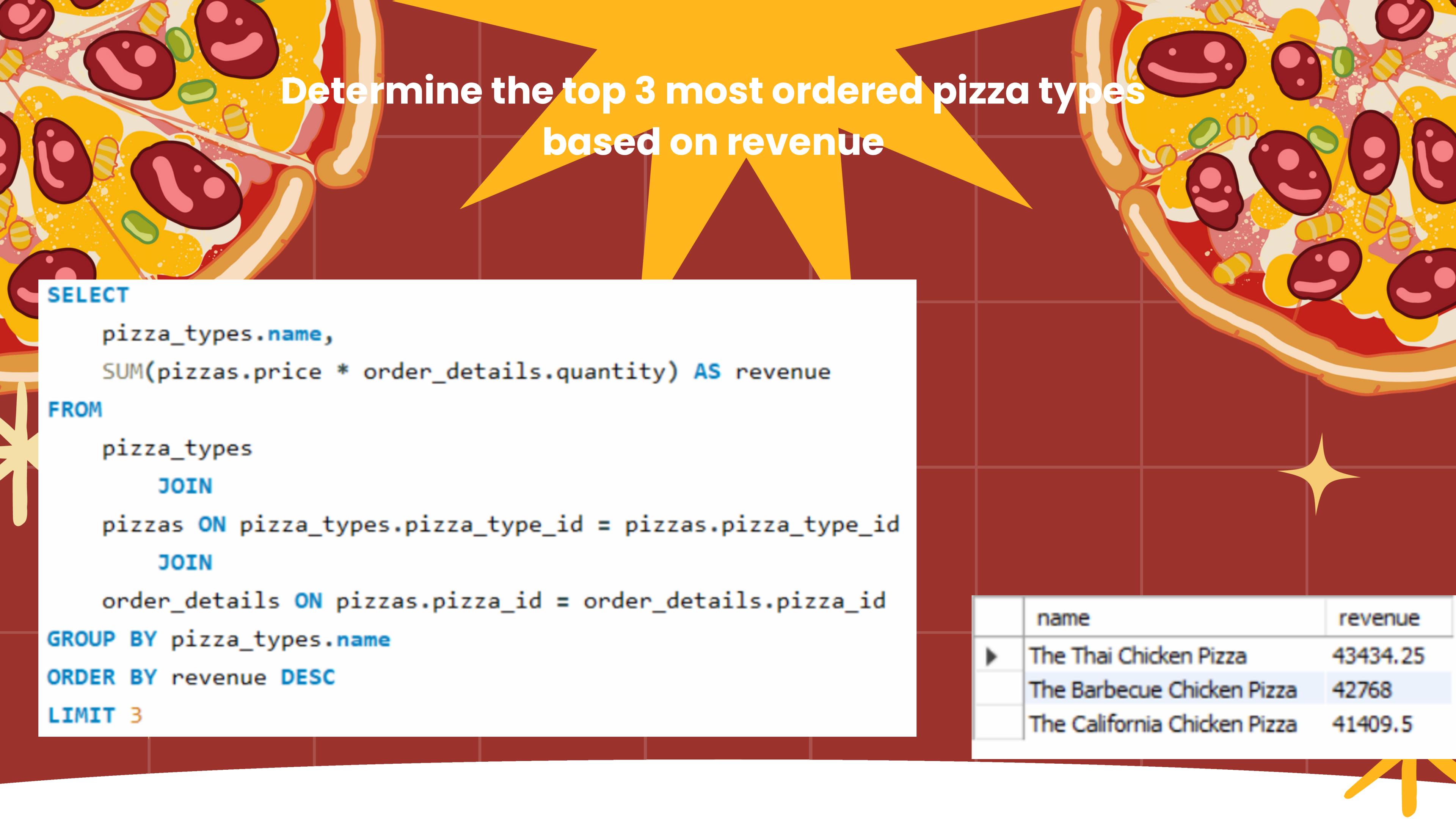
category	category-wise distribution
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```
SELECT  
    ROUND(AVG(total_quantity), 0) AS 'average number of pizzas ordered per day'  
FROM  
(SELECT  
    orders.date, SUM(order_details.quantity) AS total_quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.date) AS wtc;
```

average number of pizzas ordered per day

▶ 138



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

```
SELECT  
    pizza_types.name,  
    SUM(pizzas.price * order_details.quantity) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.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(pizzas.price * order_details.quantity) / (SELECT
        ROUND(SUM(pizzas.price * order_details.quantity),
        2) AS total_revenue
    )
    FROM
        pizzas
        JOIN
            order_details ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS percentage
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
            order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY percentage DESC
```

	category	percentage
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# Analyze the cumulative revenue generated over time

```
select date, sum(total_rev) over (order by date) as cum_rev from  
(select orders.date, sum(pizzas.price * order_details.quantity) as total_rev  
from orders join order_details  
on orders.order_id = order_details.order_id  
join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
group by orders.date) as wtc limit 5
```

	date	cum_rev
▶	2015-01-01	2713.850000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

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

```
select category, name, total_rev from
(select category, name, total_rev, rank() over(partition by category order by total_rev desc) as rn from
(select pizza_types.name, pizza_types.category, sum(pizzas.price * order_details.quantity) as total_rev
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, pizza_types.category) as a) as b
where rn <=3
```

category	name	total_rev
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

# THANK YOU

Thank you for your attention and participation. If you have any further questions, please feel free to contact me at

[abubakrshaik96@gmail.com](mailto:abubakrshaik96@gmail.com)

I appreciate your time and look forward to any feedback or discussions.