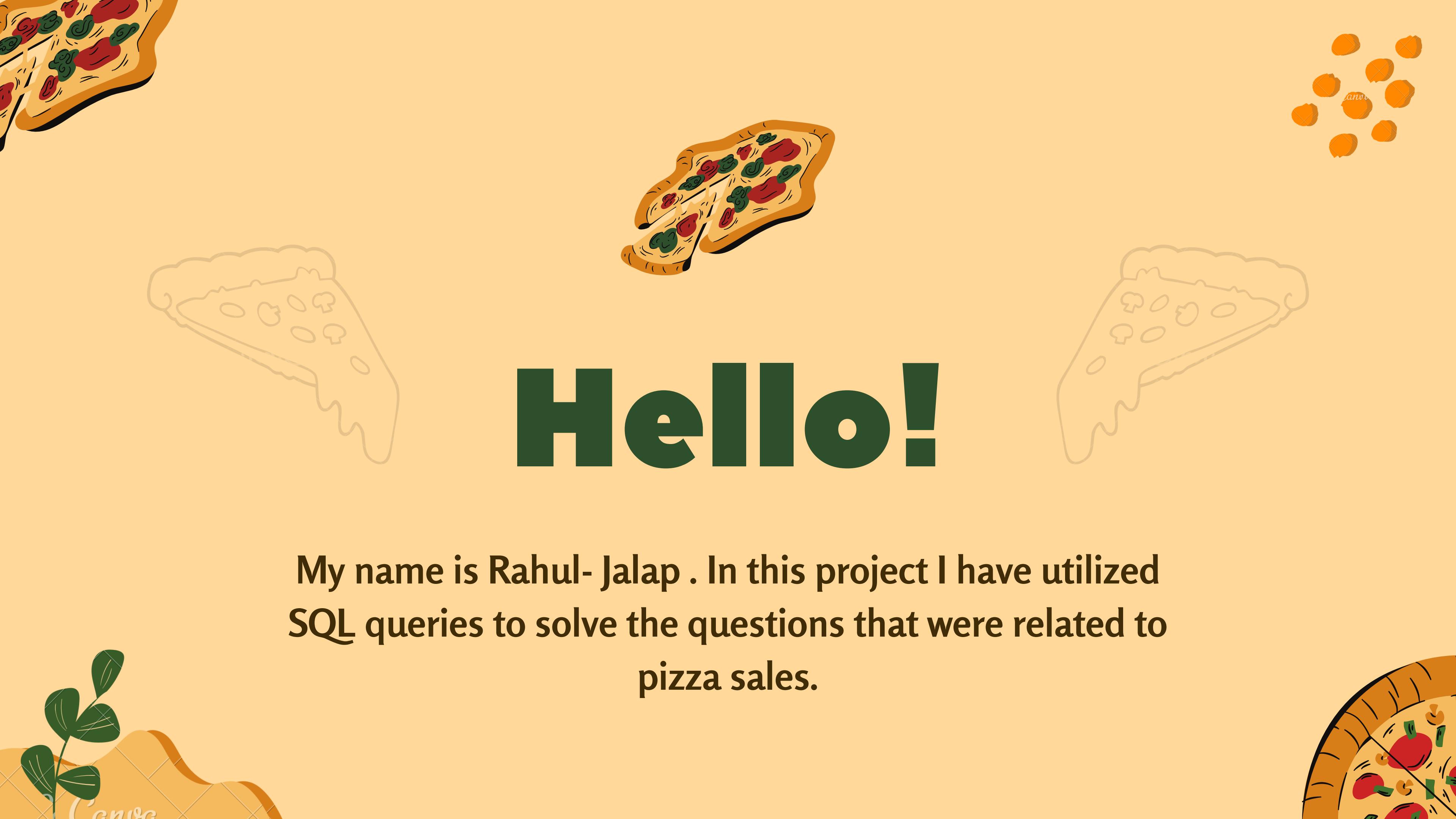


Delicious and Fresh Pizza for Everyone!

PIZZA





Hello!

My name is Rahul- Jalap . In this project I have utilized SQL queries to solve the questions that were related to pizza sales.

A cartoon illustration of a chef with a mustache, wearing a red apron and a white chef's hat, standing behind a counter and holding a pizza slice.

"Pizza Management System: SQL Implementation"

"Optimizing Pizza Operations with Structured Query Language"



Presented by :-
Rahul Jalap

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(orders_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

```
orders_details
```

```
JOIN
```

```
pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

total_sales

817860.05

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

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

```
sum(orders_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_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

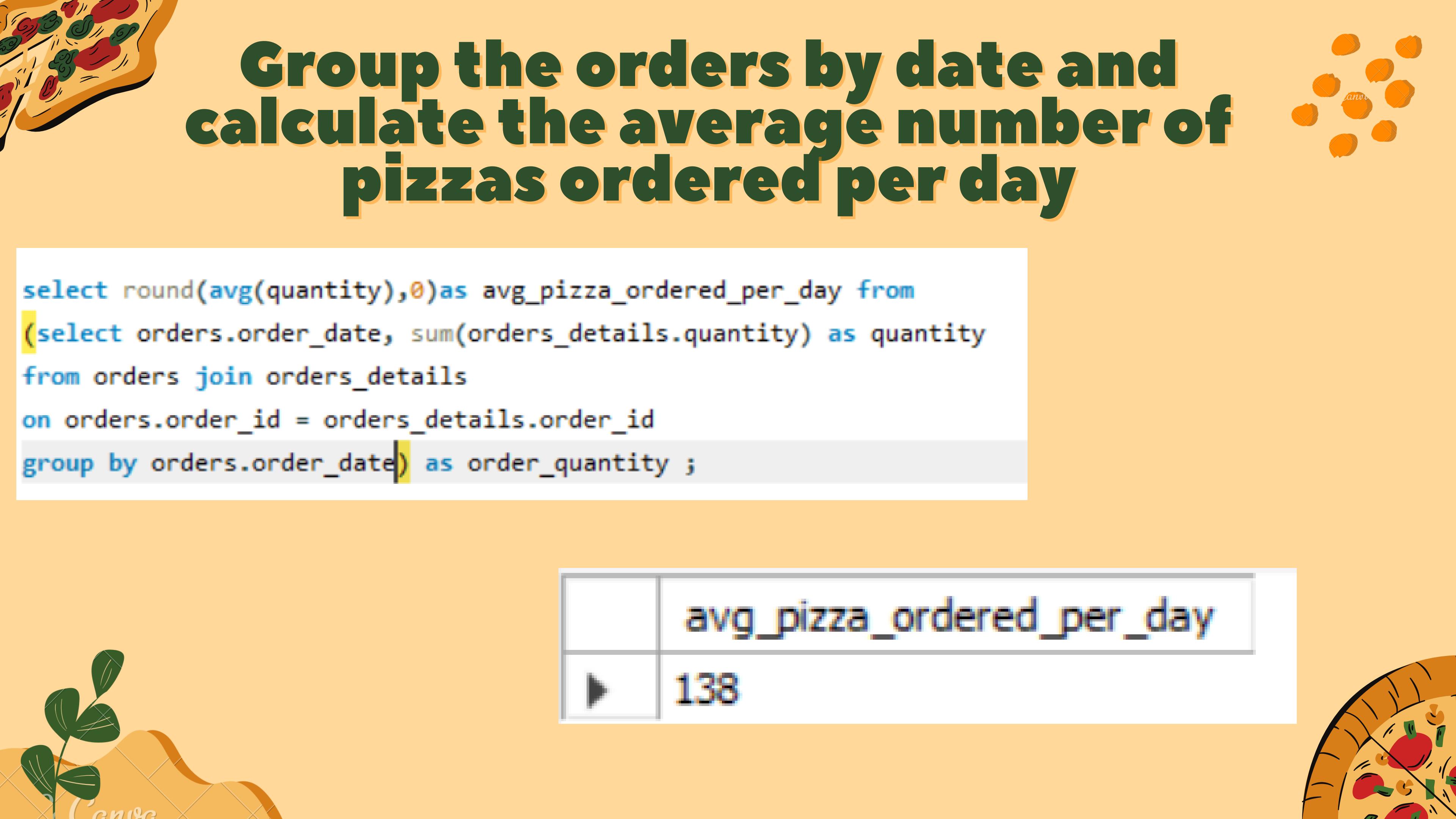
```
select pizza_types.category,  
sum(orders_details.quantity) as quantity  
  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by quantity desc;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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 average number of pizzas ordered per day

```
select round(avg(quantity),0)as avg_pizza_ordered_per_day from  
(select orders.order_date, sum(orders_details.quantity) as quantity  
from orders join orders_details  
on orders.order_id = orders_details.order_id  
group by orders.order_date) as order_quantity ;
```

	avg_pizza_ordered_per_day
▶	138



Determine the top 3 most ordered pizza types based on revenue



```
select pizza_types.name,  
sum(orders_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join orders_details  
on orders_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(orders_details.quantity * pizzas.price) / (SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS total_sales  
FROM  
    orders_details  
        JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id) )*100,2) as revenue  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_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 top 3 most cumulative revenue generated over time

```
select order_date,  
sum(revenue) over(order by order_date) as cum_revenue  
from  
(select orders.order_date,  
sum(orders_details.quantity * pizzas.price) as revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orders_details.order_id  
group by orders.order_date) as sales order by cum_revenue desc limit 3;
```

	order_date	cum_revenue
▶	2015-12-31	817860.05
	2015-12-30	814944.05
	2015-12-29	813606.25