



Project's Aim

This SQL project on sales pizza reports aims to leverage data analysis techniques to extract valuable insights from a database, enabling stakeholders to make informed decisions and drive business growth in the competitive pizza industry.

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(o.Quantity * p.price), 2) AS Total_sales
```

FROM

```
orders_details o
```

JOIN

```
pizzas p ON p.pizza_id = o.pizza_id;
```

Result Grid

Total_sales
817860.05

Identify the highest-priced pizza

```
SELECT pt.name, p.price  
FROM pizza_types pt  
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id  
ORDER BY p.price DESC  
LIMIT 1;
```

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

SELECT

Quantity, COUNT(order_id_details)

FROM

orders_details

GROUP BY Quantity;

Quantity	COUNT(order_id_details)
1	47693
2	903
3	21
4	3

Identify the most common pizza size ordered

```
SELECT  
    p.size, COUNT(o.order_id_details) AS order_count  
FROM  
    pizzas p  
        JOIN  
    orders_details o ON p.pizza_id = o.Pizza_id  
GROUP BY p.size  
ORDER BY order_count DESC;
```

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

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

SELECT

pt.name, SUM(o.Quantity) AS Total_pizza_order

FROM

pizza_types pt

JOIN

pizzas p ON pt.pizza_type_id = p.pizza_type_id

JOIN

orders_details o ON o.Pizza_id = p.pizza_id

GROUP BY name

ORDER BY Total_pizza_order DESC

LIMIT 5;

name	Total_pizza_Types
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

pt.category, SUM(o.Quantity) AS Total_pizza_order

FROM

pizza_types pt

JOIN

pizzas p ON pt.pizza_type_id = p.pizza_type_id

JOIN

orders_details o ON o.Pizza_id = p.pizza_id

GROUP BY category

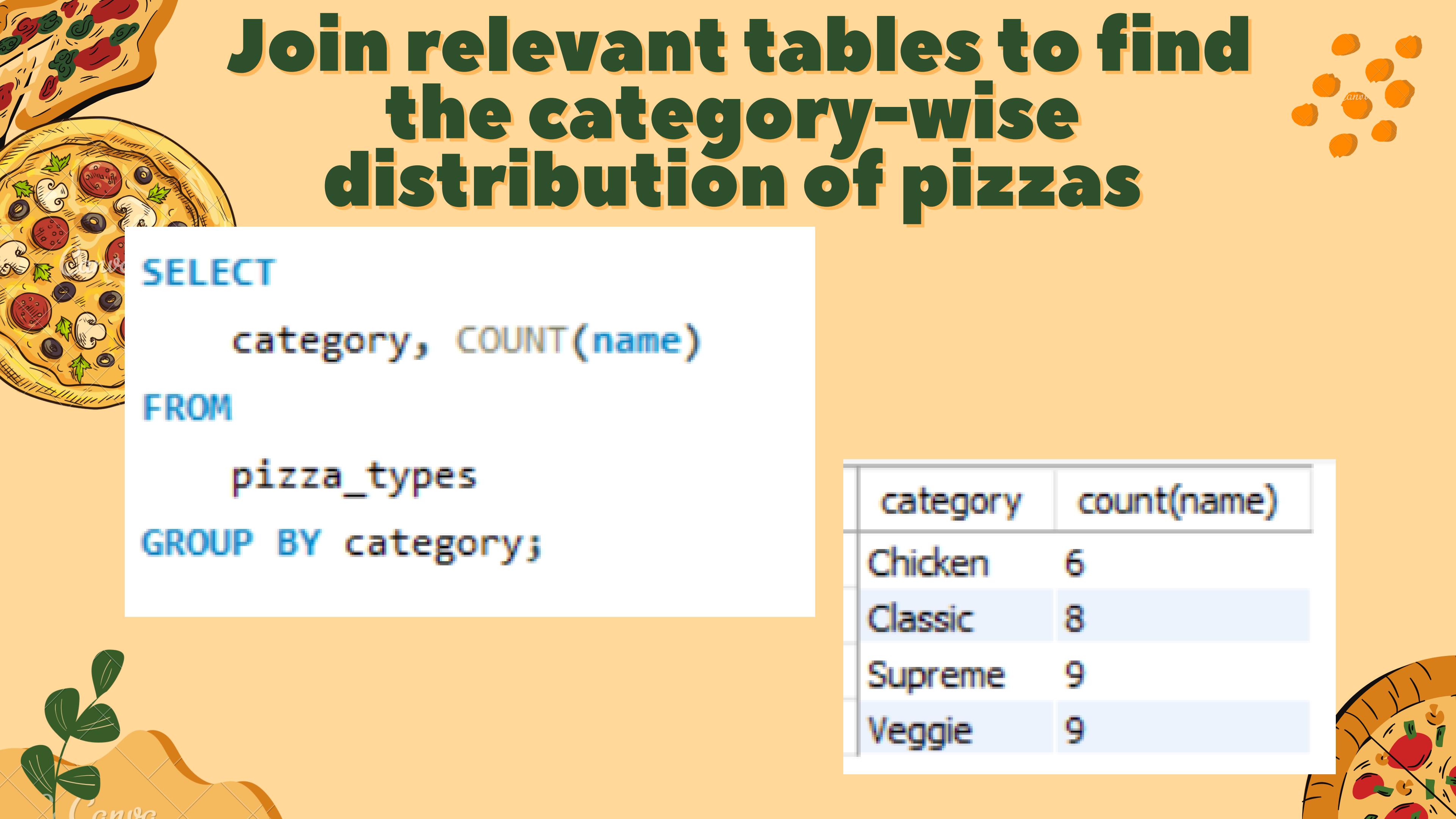
ORDER BY Total_pizza_order DESC;

category	Total_pizza_order
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time)  
order by count(order_id) desc;
```

HOUR(order_time)	COUNT(order_id)
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1



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(count_Qty)) AS Per_Day
```

FROM

```
(SELECT
```

```
o.order_date, SUM(od.Quantity) AS Count_Qty
```

FROM

```
orders o
```

```
JOIN orders_details od ON o.order_id = od.order_id
```

```
GROUP BY o.order_date) AS order_quantity;
```

	Per_Day
▶	138

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

SELECT

```
pt.name, SUM(o.Quantity * p.price) AS revenue
```

FROM

```
pizza_types pt
```

JOIN

```
pizzas p ON pt.pizza_type_id = p.pizza_type_id
```

JOIN

```
orders_details o ON o.Pizza_id = p.pizza_id
```

GROUP BY pt.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 pt.category, round(SUM(o.Quantity * p.price) / (SELECT  
ROUND(SUM(o.Quantity * p.price), 2) AS Total_sales  
FROM  
orders_details o  
JOIN  
pizzas p ON p.pizza_id = o.pizza_id)*100,2) as revenue  
FROM  
pizza_types pt  
JOIN  
pizzas p ON pt.pizza_type_id = p.pizza_type_id  
JOIN  
orders_details o ON o.Pizza_id = p.pizza_id  
GROUP BY pt.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 cul_revenue
from
(select o.order_date ,round(sum(od.Quantity*p.price),2) as revenue
from orders o join orders_details od
on o.order_id = od.order_id join pizzas p
on od.Pizza_id = p.pizza_id
group by o.order_date) as sales limit 20;
```

order_date	cul_revenue
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65