



**SHODWE**  
Pizza Resto

[Home](#)

[About](#)

[Contact](#)

# PIZZA SALES

● WHERE EVERY SLICE TELLS A STORY







**SHODWE**  
Pizza Resto

[Home](#)

[About](#)

[Contact](#)

# HELLO !

I am Anshu Prasad, and I worked on a SQL-based project focused on analyzing sales data from a pizza restaurant. Through the use of complex queries, I examined sales trends and revenue generation. This project highlights my proficiency in SQL, data analysis, and my ability to address real-world business challenges.



## 1) Retrieve the total number of orders placed

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

Result Grid	
	Total_Order
▶	21350

## 2) Calculate the total revenue generated from pizza sales



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

Result Grid	
	total_sales
▶	817860



### 3) 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;
```

Result Grid     Filter Rows		
	name	price
▶	The Greek Pizza	35.95

## 4) Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

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



5) 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
```



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
The California Chicken Pizza	2370
The Sicilian Pizza	1938
The Spicy Italian Pizza	1924
The Southwest Chicken Pizza	1917
The Big Meat Pizza	1914
The Four Cheese Pizza	1902
The Italian Supreme Pizza	1884






# SHODWE

Pizza Resto

[Home](#)[About](#)[Contact](#)

6) 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 category
ORDER BY quantity DESC;
```



	category	Quantity
▶	Chicken	11050
	Classic	14888
	Supreme	11987
	Veggie	11649



## 7) Determine the distribution of orders by hour of the day

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



hour	order_count
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1






**SHODWE**

Pizza Resto


[Home](#)[About](#)[Contact](#)

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

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



	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





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


```
SELECT  
    ROUND(AVG(quantity)) as avg_pizza_ordered_per_day  
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 order_date) AS Order_quantity;
```

Result Grid		Filter Rows
	avg_pizza_ordered_per_day	
▶	138	



10) 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



## 11) Calculate the percentage contribution of each pizza type to total revenue

```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price)) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 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.category
ORDER BY revenue DESC;
```

Result Grid    Filter Rows: 		
	category	revenue
▶	Classic	26.905961900569793
	Supreme	25.456312816374222
	Chicken	23.955139021348398
	Veggie	23.682592375223507



## 12) Analyze the cumulative revenue generated over time.

```
select order_date,  
sum(revenue) over (order by order_date) as cum_reve  
FROM  
(select orders.order_date,  
sum(order_details.quantity * pizzas.price) as reve  
from order_details Join pizzas  
ON order_details.pizza_id = pizzas.pizza_id  
Join orders  
ON orders.order_id = order_details.order_id
```

Result Grid			Filter Rows:
	order_date	cum_revenue	
▶	2015-01-01	2713.8500000000004	
	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	21526.4	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	





# SHODWE

Pizza Resto

[Home](#)[About](#)[Contact](#)

13) 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 ran
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 ran <=3;
```

Result Grid	Filter Rows:
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
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.700000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



# THANK YOU

## FOR ATTENTION

● 2025 PIZZA RESTO PRESENTATION