

# PIZZA

ORDER NOW

DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT ·  
**25%**



# **WELCOME OUR PIZZA**

Hello, my name is Neha Kumari in this project i have utilized SQL query to solve questions that is related to pizza sales.





## OBJECTIVES

- Retrieve the total no. 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.
- 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.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyse the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types.
- based on revenue for each pizza category.

# Retrieve the total no. of orders placed

```
SELECT  
    *  
  
FROM  
    orders;  
  
SELECT  
    COUNT(order_id) AS total_orders  
  
FROM  
    orders;
```

Result Grid	
	total_orders
	21350



# Calculate the total revenue generated from pizza sales.

```
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;
```

Result Grid	
	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;
```

Result Grid | Filter

	name	price
▶	The Greek Pizza	35.95



# Identify the most common pizza size ordered.

```
SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity;

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;
```

	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28



# 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;
```

Result Grid | Filter Rows:

	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	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;
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# 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);
```

Result Grid |

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198



# 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(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;
```

Result Grid	
	avg_pizza_ordered_per_day
▶	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;
```

Result Grid		
	category	revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68



# Analyse 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
▶	2015-01-01	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



# 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;
```

Result Grid		Filter Rows:
	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.5
10	The Four Cheese Pizza	32265.70000000065
11	The Mexicana Pizza	26780.75





**THANK YOU**