

# SOL PROJECT ON PIZZA SALE



# INTRODUCTION

Hello, my name is Manish Bhusari, and in this project, I have utilized SQL queries to solve questions from beginners to advanced level that were related to pizza sales.

# 1. Retrieving the total number of orders placed.

```
select count(order_id)as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350



## 2. Calculating 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 ;
```

Result Grid	
	total_sales
▶	817860.05



### 3. Identifying 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 Row

	name	price
	The Greek Pizza	35.95



#### 4. Identifying the most common pizza size ordered.

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

	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(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;
```

Result Grid		Filter Rows:
	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



6. 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



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

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

	HOUR(order_time)	COUNT(order_id)
11	1231	
12	2520	
13	2455	
14	1472	
15	1468	
16	1920	
17	2336	



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

```
select category, count(name) from pizza_types  
group by category;
```

Result Grid | Filter Rows

	category	count(name)
▶	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), 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



10. 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
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5



# 11. 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



## 12. Analyze 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(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 ;
```

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



### 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 rn
from
(select pizza_types.category,pizza_types.name,
sum((orders_details.quantity)* pizzas.price) 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,pizza_types.name) as a) as b
where rn <= 3 ;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5





# THANK YOU

“Have fun making your own pizza  
and enjoy every bite”