

Where Every Slice is a Taste of Perfection

WELCOME TO PIZZA RESTO



HELLO EVERYONE

I am **NEHA TIWARI** Here I
giving summary of my
project of SQL

In today's competitive food industry, understanding sales performance is crucial for any business, especially in the fast-paced world of pizza restaurants. This project focuses on analyzing pizza sales data using Structured Query Language (SQL) to derive meaningful insights and assist in strategic decision-making.



QUESTIONS ARE

BASIC

Retrieve the total number 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.



- **INTERMEDIATE**

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.

- **ADVANCED**

Calculate the percentage contribution of each pizza type to total revenue.

Analyze the cumulative revenue generated over time.

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

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

OUTPUT

Result Grid | Filter

	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
```

OUTPUT

	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
```

OUTPUT

Result Grid | Filter F

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size, count(order_details.order_details_id) as order_counts  
from pizzas join order_details  
on pizzas.pizza_id= order_details.pizza_id  
group by pizzas.size order by order_counts desc
```

OUTPUT

Result Grid | Filter

	size	order_counts
•	L	18526
	M	15385
	S	14137
	XL	544
	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;
```

OUTPUT

	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(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;
```

OUTPUT

	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_counts from orders  
group by hour(order_time);
```

OUTPUT

hour	order_counts
11	1231
12	2520
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

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

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

OUTPUT

	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(quatity),0) from  
  (select orders.order_date, sum(order_details.quatity) as quatity  
   from orders join order_details  
   on orders.order_id =order_details.order_id  
   group by orders.order_date) as order_quatity;
```

OUTPUT

	round(avg(quatity),0)
▶	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;
```

OUTPUT

	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
```

OUTPUT

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

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(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
```

OUTPUT

order_date	cum_revenue
2015-01-01	2713.850000000004
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

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
```

OUTPUT

Result Grid		
	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	32265.7000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

*THANK you
FOR ATTENTION*