



PIZZA

PIZZA SALES





PIZZA SALES

SQL PROJECT ON PIZZA SALES

I'm Omkar Naik, I made a project on Pizza Sales Analysis using SQL. In this project, I worked with a pizza sales dataset and utilized various SQL queries to extract meaningful insights. I solved business-related questions such as total revenue, top-selling pizzas, average order value, best-performing days, and more. This project helped me strengthen my skills in data filtering, aggregation, joins, and date functions in SQL.





QUESTIONS

Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

CODE :-

```
SELECT  
    COUNT(order_id) AS total_order_placed  
FROM  
    orders
```

OUTPUT:-

	total_order_placed
▶	21350

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

CODE :-

```
select round(sum( orders_details.quantity * pizzas.price ),2) as total_revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id
```

OUTPUT:-

	total_revenue
▶	817860.05

3.IDENTIFY THE HIGHEST-PRICED PIZZA.

CODE :-

```
select pizza_types.name , pizzas.price  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by price desc limit 1
```

OUTPUT:-

	name	price
▶	The Greek Pizza	35.95

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

CODE :-

```
SELECT pizzas.size, COUNT(orders_details.order_details_id) AS count
FROM pizzas
JOIN orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY count DESC
LIMIT 1
```

OUTPUT:-

	size	count
▶	L	18526

5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

CODE :-

```
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
```

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

6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

CODE :-

```
SELECT  
    pizza_types.category, SUM(orders_details.quantity) 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
```

OUTPUT:-

	category	quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

2. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

CODE :-

```
SELECT  
    HOUR(orders.order_time) AS time,  
    COUNT(orders.order_id) AS orders,  
    pizza_types.name  
FROM  
    orders  
    JOIN  
    orders_details ON orders.order_id = orders_details.order_id  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id  
    JOIN  
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
GROUP BY time , pizza_types.name  
ORDER BY orders DESC
```

OUTPUT:-

	time	orders	name
▶	13	337	The Classic Deluxe Pizza
	12	335	The Pepperoni Pizza
	12	332	The Classic Deluxe Pizza
	12	313	The California Chicken Pizza
	12	312	The Barbecue Chicken Pizza
	12	310	The Hawaiian Pizza
	13	305	The Hawaiian Pizza

8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

CODE :-

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

OUTPUT:-

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

S. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

CODE :-

```
SELECT  
    ROUND(AVG(quantity), 0) AS average  
FROM  
    (SELECT  
        orders.order_date AS date,  
        SUM(orders_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY date) AS order_quatity;
```

OUTPUT:-

	average
▶	138

10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

CODE :-

```
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
        0) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```

OUTPUT:-

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

CODE :-

```
SELECT
    pizza_types.category AS pizza_type,
    ROUND(SUM(pizzas.price * orders_details.quantity) / (SELECT
        SUM(orders_details.quantity * pizzas.price)
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
    2) AS revenue_percentage
FROM
    pizza_types
JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage DESC;
```

OUTPUT:-

	pizza_type	revenue_percentage
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

CODE :-

```
select order_date , round(sum(revenue) over(order by order_date),0) as cum_revenue  
from  
(select orders.order_date , sum(orders_details.quantity * pizzas.price) as revenue  
from orders join orders_details  
on orders.order_id = orders_details.order_id  
join pizzas  
on pizzas.pizza_id = orders_details.pizza_id  
group by orders.order_date) as sales ;
```

OUTPUT:-

order_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9864
2015-01-05	11930

13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

CODE :-

```
select category,  
       name , revenue, raank from  
(select category , name , revenue, rank() over(partition by category order by revenue desc) as raank  
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 raank <= 3;
```

OUTPUT:-

category	name	revenue	raank
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3



PIZZA

THANK YOU!

