

Where Every Slice is a Taste of Perfection

PIZZA SALES

SQL
PROJECT

PROJECT





ABOUT MY PROJECT



PIZZA SALES PROJECT

I'm Chandan Patel, and this is my Pizza Sales Analysis project using SQL. The database includes four main tables: orders, order_details, pizzas, and pizza_types. I used SQL to analyze sales data and answer real business questions like:

- Which pizzas sell the most?
- What are the peak order times?
- Which sizes and categories perform best?

This project helped me strengthen my skills in data analysis, joins, aggregations, and reporting using SQL for real-world use cases.

QUESTION 1

RETRIVE THE TOTAL NUMBER OF ORDER PLACED

```
SELECT count(Order_id) as total_orders from orders;
```

Result



Result Grid	
	total_orders
▶	21350

QUESTION 2

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result

Result Grid	
	total_revenue
▶	24121.6



QUESTION 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

Result Grid			Filter Rows
	NAME	price	
▶	The Greek Pizza	35.95	



QUESTION 4

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

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

Result

Result Grid			Filter
	size	order_count	
▶	L	569	



QUESTION 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
ORDER BY quantity DESC
LIMIT 5;
```



Result

Result Grid			Filter Rows:
	name	quantity	
▶	The Pepperoni Pizza	85	
	The Classic Deluxe Pizza	73	
	The Thai Chicken Pizza	70	
	The Hawaiian Pizza	66	
	The Italian Supreme Pizza	65	



QUESTION 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result

Result Grid		
	category	quantity
▶	Classic	434
	Supreme	359
	Veggie	355
	Chicken	307



QUESTION 7

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

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

Result

Result Grid		
	hour	count
▶	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



QUESTION 8

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

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

Result

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



QUESTION 9

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

```
SELECT
    ROUND(AVG(quantity), 0) AS average_pizaas_perDay
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 orders_quantity;
```

Result

Result Grid		Filter Row
	average_pizaas_perDay	
▶	132	



QUESTION 10

DETERMINE THE TOP 3 MOST PIZZA TYPES ORDERED BASED ON THE 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;
```

Result

	name	Revenue
▶	The Thai Chicken Pizza	1296.5
	The Barbecue Chicken Pizza	1160.75
	The Italian Supreme Pizza	1158.75



QUESTION 11

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),  
           1) AS total_revenue  
       FROM  
         order_details  
       JOIN  
         pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,2) 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

Result Grid			Filter
	category	Revenue	
▶	Classic	26.44	
	Supreme	25.99	
	Veggie	24.83	
	Chicken	22.74	



QUESTION 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(order_details.quantity * pizzas.price) as Revenue from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_Id = order_details.order_Id group by orders.order_date) as sales;
```

Result

Result Grid			Filter Rows:
	order_date	Cum_Revenue	
▶	2015-01-01	2713.85000000000004	
	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	



QUESTION 13

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON THE 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 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, pizza_types.name
) as a) as b where rn <= 3 ;
```

Result



Result Grid			Filter Rows:
	name	revenue	
▶	The Thai Chicken Pizza	1296.5	
	The Barbecue Chicken Pizza	1160.75	
	The California Chicken Pizza	1132	
	The Classic Deluxe Pizza	1148.5	
	The Pepperoni Pizza	1084.5	
	The Greek Pizza	947.5	
	The Italian Supreme Pizza	1158.75	
	The Spicy Italian Pizza	1127.75	
	The Sicilian Pizza	954.75	

	The Greek Pizza	947.5
	The Italian Supreme Pizza	1158.75
	The Spicy Italian Pizza	1127.75
	The Sicilian Pizza	954.75
	The Five Cheese Pizza	980.5
	The Four Cheese Pizza	947.60...
	The Mexicana Pizza	806.75

PIZZ SALES

THANK YOU
FOR ATTENTION

SQL PROJECT