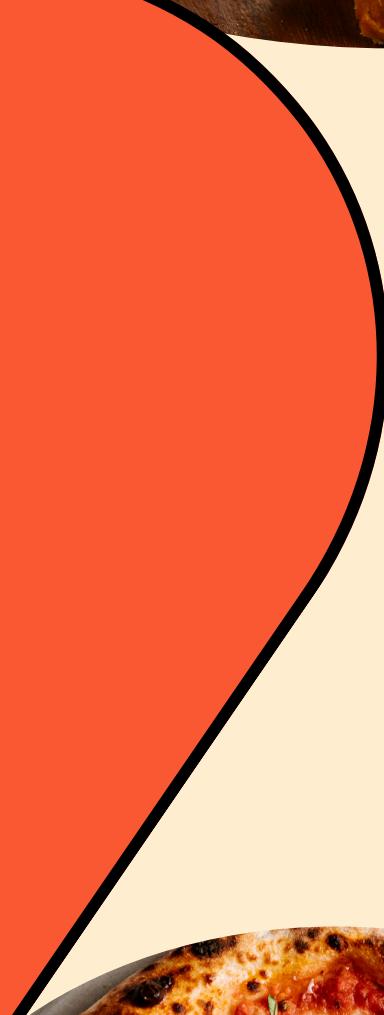


**-PIZZA\_SALES-**

# **ANALYSES**

**SQL PROJECT**





Analyzed pizza sales data using SQL by working with multiple relational tables (orders, order\_details, pizzas, pizza\_type). The project covered basic to advanced analysis, including total orders and revenue, best-selling pizzas and sizes, category-wise demand, hourly order trends, and average daily sales. Advanced insights included revenue contribution by pizza type, cumulative revenue analysis, and identifying top revenue-generating pizzas across categories. This project demonstrates strong SQL skills and business-focused data analysis.

# TABLES AND THEIR SCHEMAS

Table:order\_details

Columns:  
order\_details\_id  
int PK  
order\_id  
int  
pizza\_id  
varchar(50)  
quantity  
int

Table:orders

Columns:  
order\_id  
int PK  
order\_date  
date  
order\_time  
time

Table:pizza\_types

Columns:  
pizza\_type\_id  
text  
name  
text  
category  
text  
ingredients  
text

Table:pizzas

Columns:  
pizza\_id  
text  
pizza\_type\_id  
text  
size  
text  
price  
double

-- Retrieve the total number 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_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05

-- Identify the highest-priced pizza

```
SELECT
    pizza_types.name, pizzas.price AS highest_priced_pizza
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	name	highest_priced_pizza
▶	The Greek Pizza	35.95

-- Identify the most common pizza size ordered

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 limit 1;

Result Grid | Filter

	size	order_count
▶	L	18526

-- 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;

	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 total\_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;

Result Grid | Filter Rows:

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	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;
```

	hour	order_count
	9	1
▶	10	8
	23	28
	22	663
	21	1198
	11	1231
	15	1468
	14	1472
	20	1642
	16	1920
	19	2009
	17	2336
	18	2399
	13	2455
	12	2520



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

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

Result Grid | Filter Rows:

	category	pizza_distribution
▶	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 AS date,
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS quantity_orderd;
```

Result Grid | Filter Rows:

	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 total_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.name
ORDER BY total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

	name	total_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  
        SUM(order_details.quantity * pizzas.price) AS total_revenue FROM  
        order_details JOIN  
        pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100, 2) AS revenue_percentage  
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_percentage DESC;
```

Result Grid | Filter Rows:

	category	revenue_percentage
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



-- Analyze the cumulative revenue generated over time

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

	order_date	cum_revenue
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9864
	2015-01-05	11930
	2015-01-06	14358
	2015-01-07	16561
	2015-01-08	19399
	2015-01-09	21526
	2015-01-10	23990
	2015-01-11	25863
	2015-01-12	27782
	2015-01-13	29831



```

select category,name, revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as pizza_rank
from
(select
pizza_types.category,pizza_types.name,round(sum(order_details.quantity*pizzas.price),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,pizza_types.name)as a)as b
where pizza_rank<=3;

```

Result Grid | Filter Rows: | Export

	category	name	revenue
	Veggie	The Four Cheese Pizza	32265.7
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5



# **THANK YOU**

## **DO LIKE AND COMMENT**

