



# Pizza Sales Analysis using SQL

by DURGA BUDHATHOKI





# HELLO EVERYONE

My name is Durga Budhathoki, and I have developed a SQL project in which I utilized SQL Server in Visual Studio Code to analyze pizza sales data. This analysis is based on four datasets that provide valuable insights into various aspects of pizza sales performance. To uncover these insights, I addressed 13 key questions, which helped in understanding sales trends, customer behavior, and overall business performance.



# ALL 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.

## Advanced:

- 1) Calculate the percentage contribution of each pizza type to total revenue.
- 2) Analyze the cumulative revenue generated over time.
- 3) Determine the top 3 most ordered pizza types based on revenue for each pizza category.

## Intermediate:

- 1) Join the necessary tables to find the total quantity of each pizza category ordered.
- 2) Determine the distribution of orders by hour of the day.
- 3) Join relevant tables to find the category-wise distribution of pizzas.
- 4) Group the orders by date and calculate the average number of pizzas ordered per day.
- 5) Determine the top 3 most ordered pizza types based on revenue.

# Retrieve the total number of orders placed.

```
Select  
|   Count(*) as total_number_orders  
from orders;
```



## RESULTS

	total_number_orders
1	21350

# Calculate the total revenue generated from pizza sales.

```
SELECT
    SUM(p.price * od.quantity) as revenue
FROM
    pizzas p
JOIN orders_details od ON p.pizza_id = od.pizza_id;
```



RESULTS	
	revenue
1	827450

# Identify the highest-priced pizza.

```
SELECT
    top(1)
    name,
    price
FROM
    pizzas p
JOIN pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY price DESC ;
```



## RESULTS

	name	price
1	The Greek Pizza	36

# Identify the most common pizza size ordered.

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



## RESULTS

	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

# List the top 5 most ordered pizza types along with their quantities.

```
SELECT
    top(5)
    pt.name,
    sum(od.quantity) as total_quantity
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER by total_quantity DESC;
```



RESULTS		
	name	total_quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

# Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
    pt.category,
    sum(od.quantity) as total_quantity
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY total_quantity DESC ;
```



## RESULTS

	category	total_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

```
SELECT  
    DatePart(HOUR, time) as hour,  
    count(order_id) as order_count  
FROM orders  
GROUP BY DatePart(HOUR, time)  
ORDER BY [hour];
```

RESULTS		
	hour	order_count
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

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

```
SELECT  
    category,  
    count(*) as Count  
FROM  
    pizza_types  
GROUP BY category  
ORDER BY count DESC;
```



RESULTS		
	category	Count
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6



# Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    AVG(total_quantity) as avg_pizzas_order_per_day
FROM
    (SELECT
        date,
        sum(quantity) as total_quantity
    FROM
        orders_details od
    JOIN orders o ON od.order_id = o.order_id
    GROUP BY date) AS order_quantity
```



RESULTS	
	avg_pizzas_order_per_day
1	138

# Determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    top (3)
    pt.name,
    SUM(od.quantity * p.price) as revenue
FROM pizza_types pt
JOIN pizzas p on pt.pizza_type_id = p.pizza_type_id
JOIN orders_details od on od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
```



RESULTS		
	name	revenue
1	The Thai Chicken Pizza	44027
2	The Barbecue Chicken Pizza	43376
3	The California Chicken Pizza	42002

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

```
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) /
        (SELECT SUM(od2.quantity * p2.price)
            FROM orders_details od2
            JOIN pizzas p2 ON od2.pizza_id = p2.pizza_id) * 100, 2) AS revenue_percentage
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue_percentage DESC;
```



## RESULTS

	category	revenue_percentage
1	Classic	26.960000
2	Supreme	25.510000
3	Chicken	24.010000
4	Veggie	23.530000

# Analyze the cumulative revenue generated over time.

```
SELECT
    date,
    SUM(revenue) OVER(order by date) as cum_revenue
FROM
    (SELECT
        orders.date,
        SUM (pizzas.price * orders_details.quantity) as revenue
    FROM
        orders_details
    JOIN orders ON orders_details.order_id=orders.order_id
    JOIN pizzas ON orders_details.pizza_id = pizzas.pizza_id
    GROUP BY date) as sales
```

RESULTS		
	date	cum_revenue
1	2015-01-01	2746
2	2015-01-02	5512
3	2015-01-03	8203
4	2015-01-04	9983
5	2015-01-05	12075
6	2015-01-06	14532
7	2015-01-07	16761
8	2015-01-08	19628
9	2015-01-09	21777
10	2015-01-10	24270
11	2015-01-11	26161
12	2015-01-12	28105
13	2015-01-13	30182
14	2015-01-14	32742
15	2015-01-15	34752
16	2015-01-16	37371
17	2015-01-17	39459



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

```
WITH a AS (
    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.name, pizza_types.category
),
rankData AS(
    SELECT
        category,
        name,
        revenue,
        rank() OVER(partition by category order by revenue desc) as RN
    FROM a
)
SELECT * FROM rankData
where RN <=3;
```



RESULTS				
	category	name	revenue	RN
1	Chicken	The Thai Chicken Pizza	44027	1
2	Chicken	The Barbecue Chicken Pizza	43376	2
3	Chicken	The California Chicken Pizza	42002	3
4	Classic	The Classic Deluxe Pizza	38417	1
5	Classic	The Hawaiian Pizza	33122	2
6	Classic	The Pepperoni Pizza	30637	3
7	Supreme	The Spicy Italian Pizza	35516	1
8	Supreme	The Italian Supreme Pizza	34232	2
9	Supreme	The Sicilian Pizza	30456	3
10	Veggie	The Four Cheese Pizza	32478	1
11	Veggie	The Five Cheese Pizza	26771	2
12	Veggie	The Mexicana Pizza	26564	3

**THANK YOU  
FOR ATTENTION**