



All  
About  
Pizza!

# SQL PROJECT ON PIZZA SALES

# INTRO TO PROJECT

Hello, my name is Majid, and in this project, I have utilised SQL queries to solve questions that were related to pizza sales. This project is an SQL-based analysis of pizza sales data designed to answer key business questions. It involved writing queries to extract insights on sales performance, such as identifying the most popular pizza categories, analyzing daily and monthly order trends, and calculating important metrics like total revenue and average order value.



## Retrieve the total number of orders placed

```
1  -- Retrieve the total number of orders placed.  
2  
3 • SELECT  
4      COUNT(order_id) AS total_orders  
5 FROM  
6     orders;  
7  
8
```

Result Grid	
	total_orders
	21350

# Calculate the total revenue generated from pizza sales

```
22 •   SELECT
23   -     ROUND(SUM(order_details.quantity * pizzas.price),
24           2) AS total_revenue
25   FROM
26       order_details
27   JOIN
28       pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid			F
total_revenue			
817860.05			

## Identify the highest-priced pizza

```
37 •   SELECT
38       pizza_types.name, pizzas.price
39   FROM
40       pizza_types
41           JOIN
42       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
43   ORDER BY pizzas.price DESC
44   LIMIT 1;
```

Result Grid Filter Rows:  Search

name	price
The Greek Pizza	35.95

## Identify the most common pizza size ordered

```
50 •   SELECT
51       pizzas.size, COUNT(order_details.quantity) AS order_count
52   FROM
53       pizzas
54       JOIN
55       order_details ON pizzas.pizza_id = order_details.pizza_id
56   GROUP BY pizzas.size
57   ORDER BY order_count DESC;
```

Result Grid  

size	order_count
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
66   FROM
67     pizza_types
68       JOIN
69         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
70           JOIN
71             order_details ON order_details.pizza_id = pizzas.pizza_id
72     GROUP BY pizza_types.name
73     ORDER BY quantity DESC
74     LIMIT 5;
```

Result Grid Filter Rows:  Search

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

## 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;
```

Result Grid

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_count  
FROM  
    orders  
GROUP BY hour;
```

Result Grid 

	hour	order_count
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

Result 39

# Find the category-wise distribution of pizzas

```
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

	category	total_quantity
	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

**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, 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 order_quantity;
```

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 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 revenue DESC
LIMIT 3;
```

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 category 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_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 Grid



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
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;
```

Result Grid Filter Rows:

	order_date	cum_revenue
	2015-01-01	2713.8500000000004
	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
...	...	...

Result 45

# 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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

Result Grid		Filter Rows:	Search	Export:
	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 Onion Ring Pizza	24681.25		

THANK  
YOU

