



PIZZA RESTAURANT

# PIZZA SALES PROJECT USING SQL + PYTHON

Md Aftab Uddin



# INTRODUCTION

## PIZZA SALES DATA ANALYSIS PROJECT

In this project, I analyzed a fictional pizza restaurant dataset to derive meaningful insights into sales performance, customer preferences, and operational efficiency. Using Python with SQL queries in Jupyter Notebook, I explored structured data stored in a relational database and focused on the following aspects:

- Extracting and summarizing key metrics such as total orders and overall revenue.
- Identifying best-selling pizza types, sizes, and categories to understand customer choices.
- Exploring sales trends over time to identify patterns in ordering behaviour.
- Utilizing advanced SQL techniques, including joins, window functions, and aggregations, to analyze revenue contributions, category-wise performance, and cumulative sales trends.



# PROBLEMS

## Basics

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

## Intermediate

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

## Advanced

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



## Retrieve The Total Number Of Orders Placed

```
query = """ select count(order_id) from orders """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
print(df)
```

21350

## Calculate The Total Revenue Generated From Pizza Sales.

```
query = """ select round(sum(price * order_details.quantity),2) from pizzas
join order_details
on order_details.pizza_id = pizzas.pizza_id """
cur.execute(query)
data = cur.fetchall()
data[0][0]
```

817860.05

## Find The Second Highest Price Of A Pizza

```
query = """ select max(pizzas.price) as sec_max from pizzas
where pizzas.price < (select max(pizzas.price) from pizzas)"""
cur.execute(query);
data = cur.fetchall();
df = pd.DataFrame(data);
```

25.5



## Identify The Highest-Priced Pizza.

```
query = """ select pizza_types.name, price from pizzas
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc limit 1 """
cur.execute(query)
data = cur.fetchall()
data[0][0]
```

'The Greek Pizza'

## Identify The Most Common Pizza Size Ordered.

```
query = """ select size, count(quantity) as count from pizzas
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by count desc """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

	pizza size	ordered num
	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28





## List The Top 5 Most Ordered Pizza Types Along With Their Quantities.

```
query = """ select pizza_types.name, sum(order_details.quantity) as 'ordered' 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 ordered desc limit 5 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ['pizza name', 'qty ord'])
```



pizza name	qty ord
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

```
query = """ select pizza_types.category, sum(order_details.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 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

category	quantity	order
Classic	14888	
Veggie	11649	
Supreme	11987	
Chicken	11050	

# Determine The Distribution Of Orders By Hour Of The Day

```
query = """ select hour(orders.time), count(orders.order_id) as order_count from orders
group by hour(orders.time) """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
print(df)
```

order/hours	quantity	order
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	



# Join Relevant Tables To Find The Category-Wise Distribution Of Pizzas.

```
query = """ select pizza_types.category, count(pizza_types.name), round(sum(pizzas.price),2)
from pizza_types
join pizzas
group by pizza_types.category """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

pizza category	quantity pizza	Total Price
Veggie	864	14204.7
Supreme	864	14204.7
Classic	768	12626.4
Chicken	576	9469.8

# Group The Orders By Date & Calculate The Average Number Of Pizzas Ordered Per Day

```
query = """ select round(avg(quantity),2) from
(select orders.date, count(order_details.quantity) as quantity from orders
join order_details
on order_details.order_id = orders.order_id
group by orders.date) as a """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

135.81

# Determine The Top 3 Most Ordered Pizza Types Based On Revenue.

```
query = """ select pizza_types.name, round(sum(pizzas.price * order_details.quantity),2)
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 5 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

pizza name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50

# Analyze The Cumulative Revenue Generated Over Time.

```
query = """ select dates, round(cumulative_revenue, 2) from
(select dates, sum(revenue) over(order by dates) as cumulative_revenue from
(select orders.date as dates, sum(order_details.quantity * pizzas.price) as revenue from pizzas
join order_details
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date ) as sales
) as cumulative_sales """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

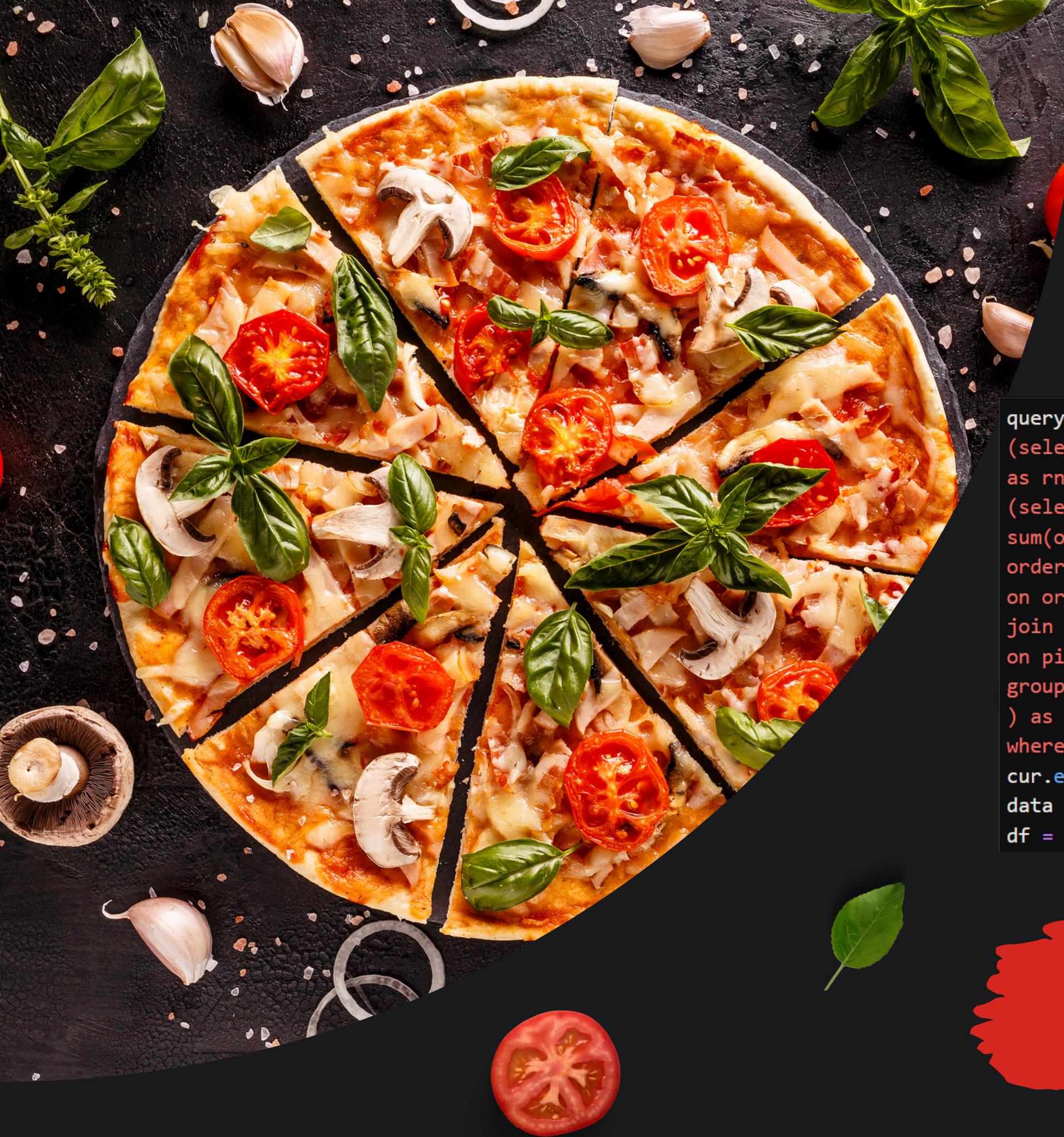
Date	Cummulative Revenue
01-01-2015	2713.85
01-02-2015	5903.05
01-03-2015	7501.60
01-04-2015	9678.45
01-05-2015	12250.40
01-06-2015	15318.15
01-07-2015	17549.65
01-08-2015	19990.20
01-09-2015	22343.05
01-10-2015	25545.20

# Calculate The Percentage Contribution Of Each Pizza Type To Total Revenue



```
query = """ select pizza_types.category,
    round(
        sum(pizzas.price * order_details.quantity) /
        (select sum(pizzas.price * order_details.quantity)
         from pizzas
         join order_details
         on order_details.pizza_id = pizzas.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 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns=['pizza category', 'rev percentage'])
```

pizza category	rev percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



## Determine The Top 3 Most Ordered Pizza Types Based On Revenue For Each Pizza Category.

```
query = """ select rn, pizza, revenue from
(select categories, pizza, revenue, rank() over(partition by categories order by revenue desc)
as rn from
(select pizza_types.category as categories, pizza_types.name as pizza,
sum(order_details.quantity * pizzas.price) as revenue from
order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by categories, pizza) as sales
) as top_sales
where rn<=3 """
cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data)
```

Thank You

Rank	Pizza Name	Revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50
1	The Classic Deluxe Pizza	38180.50
2	The Hawaiian Pizza	32273.25
3	The Pepperoni Pizza	30161.75
1	The Spicy Italian Pizza	34831.25
2	The Italian Supreme Pizza	33476.75
3	The Sicilian Pizza	30940.50
1	The Four Cheese Pizza	32265.70
2	The Mexicana Pizza	26780.75
3	The Five Cheese Pizza	26066.50

# PROJECT DATA ANALYSIS BY



Md Aftab Uddin





# PROJECT CONCLUSION

## PIZZA SALES DATA ANALYSIS

Through this project, I successfully analyzed the sales data of a fictional pizza restaurant, uncovering valuable insights into customer preferences, sales patterns, and business performance.

Thank you