

*PIZZAHUT*

**SQL Project: Data Analysis  
for Pizza Sales**

**By Ananya Roy**

# Index

1. Introduction
2. Basic Analysis
3. Intermediate Analysis
4. Advanced Analysis
5. Conclusion

---

# 1. Introduction

## Project Objective

The goal of this project is to analyze pizza sales data using SQL to extract actionable insights. By querying the dataset, I aim to:

- Assess Sales Performance:
  - Measure total orders and revenue.
- Identify Key Metrics:
  - Find the highest-priced items, most common sizes, and top-selling pizzas.
- Analyze Trends:
  - Examine order distribution by time and daily averages.
- Revenue Insights:
  - Determine revenue contributions of different pizza types and categories.

Data is sourced from [GitHub Repository: Pizza Sales SQL](<https://github.com/Ayushi0214/pizza-sales---SQL>). This analysis will support strategic decisions on menu offerings, pricing, and promotions.




## 2. Basic Analysis

---

## 2.1 Total Number of Orders Placed

```
1  -- total number of orders places
2  •  SELECT
3      COUNT(order_id) AS total_count
4  FROM
5      orders;
```

Result Grid 	
	total_count
▶	21350

## 2.2 Total Revenue from Pizza Sales

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  •  SELECT
4      ROUND(SUM(p.price * od.quantity), 2) AS total_revenue
5  FROM
6      order_details AS od
7      JOIN
8      pizzas AS p ON p.pizza_id = od.pizza_id;
```

Result Grid		Filter
	total_revenue	
▶	817860.05	

## 2.3 Highest-Priced Pizza

```
1  -- Identify the highest-priced pizza.
2
3  •  SELECT
4      pizza_types.name, pizzas.price
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9  ORDER BY pizzas.price DESC
10 LIMIT 1;
```

Result Grid			Filter Row
	name	price	
▶	The Greek Pizza	35.95	

## 2.4 Most Common Pizza Size Ordered

```
1  -- types of quantities used
2
3  •  select od.quantity, count(od.order_details_id) as count
4     from order_Details as od
5     group by od.quantity;
```

Result Grid			Filter Rows:	
	quantity	count		
▶	1	47693		
	2	903		
	3	21		
	4	3		



## 2.5 Top 5 Most Ordered Pizza Types

```
1  -- List the top 5 most ordered pizza types along with their quantities
2
3  •  select pt.name, p.pizza_id, sum(od.quantity) as count
4      from order_details as od
5      join pizzas as p on p.pizza_id=od.pizza_id
6      join pizza_types as pt on p.pizza_type_id=pt.pizza_type_id
7      group by pt.name, p.pizza_id
8      order by count desc
9      limit 5;
```

Result Grid			
Filter Rows:			
	name	pizza_id	count
▶	The Big Meat Pizza	big_meat_s	1914
	The Thai Chicken Pizza	thai_chn_l	1410
	The Five Cheese Pizza	five_cheese_l	1409
	The Four Cheese Pizza	four_cheese_l	1316
	The Classic Deluxe Pizza	classic_dlx_m	1181



### 3. Intermediate Analysis

—

## 3.1 Total Quantity of Each Pizza Category Ordered

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered
2
3  •  select pt.category,
4      sum(od.quantity) as quantity
5  from pizza_types pt
6  join pizzas p on p.pizza_type_id=pt.pizza_type_id
7  join order_details od on p.pizza_id=od.pizza_id
8  group by pt.category
9  order by quantity desc;
```

Result Grid			Filter Rows:
	category	quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

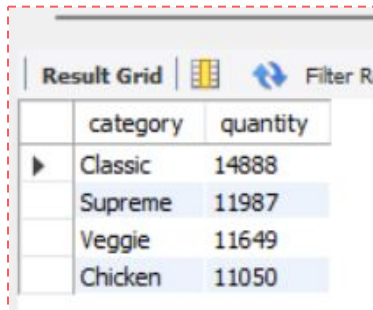
## 3.2 Distribution of Orders by Hour of the Day

```
1  -- Determine the distribution of orders by hour of the day.  
2  
3  •  select hour(order_time) as hour, count(hour(order_time)) as count from orders  
4  group by hour(order_time)  
5  order by hour(order_time);
```

Result Grid		
	hour	count
▶	9	1
	10	8
	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

## 3.3 Category-Wise Distribution of Pizzas

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered
2  |
3  •  select pt.category,
4     sum(od.quantity) as quantity
5     from pizza_types pt
6     join pizzas p on p.pizza_type_id=pt.pizza_type_id
7     join order_details od on p.pizza_id=od.pizza_id
8     group by pt.category
9     order by quantity desc;
```

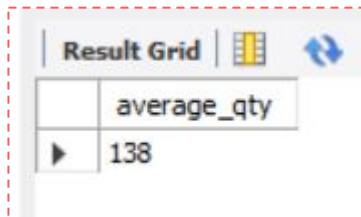


The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of the SQL query, showing four rows of data. The first row is 'Classic' with a quantity of 14888. The second row is 'Supreme' with a quantity of 11987. The third row is 'Veggie' with a quantity of 11649. The fourth row is 'Chicken' with a quantity of 11050. The table has a header with 'category' and 'quantity' columns. There are also icons for 'Filter' and 'Refresh' in the top right corner of the grid.

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## 3.4 Average Number of Pizzas Ordered per Day

```
1      -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3  •   select round(avg(order_quantity.quantity_ordered_that_day),0) as average_qty
4      from
5      (SELECT o.order_date, sum(od.quantity) as quantity_ordered_that_day
6       FROM orders o
7       join order_details od
8       on o.order_id = od.order_id
9       group by o.order_date) as order_quantity;
```





The screenshot shows a database interface with a 'Result Grid' tab. The grid has two columns: 'average\_qty' and a value '138'. The value '138' is highlighted with a mouse cursor. The entire screenshot is enclosed in a red dashed border.

	average_qty
▶	138



## 3.5 Top 3 Most Ordered Pizza Types by Revenue

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3  •  SELECT pt.name, round(sum(p.price*od.quantity),2) as revenue
4     FROM pizzahut.pizza_types pt
5     join pizzas p on p.pizza_type_id=pt.pizza_type_id
6     join order_details od on p.pizza_id=od.pizza_id
7     group by pt.name
8     order by revenue desc
9     limit 3;
```

Result Grid   Filter Rows: <input type="text"/>		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



## 4. Advanced Analysis

*PIZZA*  
*Sales analysis*





## 4.1 Percentage Contribution of Each Pizza Type to Total Revenue

```
1      -- Calculate the percentage contribution of each pizza type to total revenue
2
3  ● ○ SELECT category, round(sum(od.quantity*p.price)*100/
4  ○ (select sum(od.quantity*p.price)
5    from order_details od
6    join pizzas p on od.pizza_id=p.pizza_id
7    ),2)
8  as percent_revenue FROM pizza_types pt
9  join pizzas p on p.pizza_type_id=pt.pizza_type_id
10 join order_details od on od.pizza_id=p.pizza_id
11 group by category;
```

Result Grid			Filter Rows:
	category	percent_revenue	
▶	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

## 4.2 Cumulative Revenue Generated Over Time

```
1  -- Analyze the cumulative revenue generated over time
2
3  • select order_date, revenue, round(sum(revenue) over(order by order_date),2)
4    as cumulative_revenue from
5  (select o.order_date, round(sum(od.quantity*p.price),2) as revenue from orders o
6   join order_details od on od.order_id=o.order_id
7   join pizzas p on p.pizza_id=od.pizza_id
8   group by o.order_date) as sales
```

Result Grid   Filter Rows: <input type="text"/>			
	order_date	revenue	cumulative_revenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55

\*\*\* Only first few rows of the solution is provided as the solution is big

## 4.3 Top 3 Most Ordered Pizza Types by Revenue for Each Pizza Category

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3  •  select * from
4  (select category, name, revenue, rank() over(partition by category order by revenue desc) as C_rank
5  from
6  (SELECT pt.category,pt.name, round(sum(p.price*od.quantity),2) as revenue
7  FROM pizzahut.pizza_types pt
8  join pizzas p on p.pizza_type_id=pt.pizza_type_id
9  join order_details od on p.pizza_id=od.pizza_id
10 group by pt.category,pt.name) as a)as b
11 where C_rank<=3;
```

Result Grid

Filter Rows:

Export:

	category	name	revenue	C_rank
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.7	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

19



## 5. Insights and Recommendations

Based on the analysis, actionable insights and recommendations are provided to the pizza restaurant chains. These insights can help in making informed decisions to improve sales, marketing strategies, and overall business performance.

---





## 6. Conclusion

### Future Work -

1. **Visualizations:** Enhancing the analysis by presenting findings through visualizations using Python libraries such as Matplotlib and Seaborn.
2. **Integration with BI Tools:** Integrate the data with Business Intelligence (BI) tools for interactive dashboards and real-time insights.

---

# Contact

Ananya Roy

Jadavpur University

[ananyadav.rnj900@gmail.com](mailto:ananyadav.rnj900@gmail.com)

[www.linkedin.com/in/ananya-roy-ju](https://www.linkedin.com/in/ananya-roy-ju)

