

# SQL Project

## On Pizza Sales Data Analysis


By Abhishek Gosavi

Guide by Ayushi Jain (WsCubeTech)





# Hello!



My name is Abhishek, I am an aspiring Data Analyst, passionate about extracting insights from data and making data-driven decisions. I have hands-on experience with SQL, Python, Power BI, and MS Excel and have worked on various data analysis projects. This project focuses on analyzing pizza sales data using SQL to derive meaningful business insights. The dataset contains information about orders, pizzas, order details and sales transactions, allowing us to explore various patterns and trends in pizza sales.

I structured my analysis into three levels: Basic, Intermediate and Advanced, covering a range of SQL queries from simple aggregations to complex joins and revenue analysis.



# Questions

## Basic:

- 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.





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

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	total_order
▶	21350

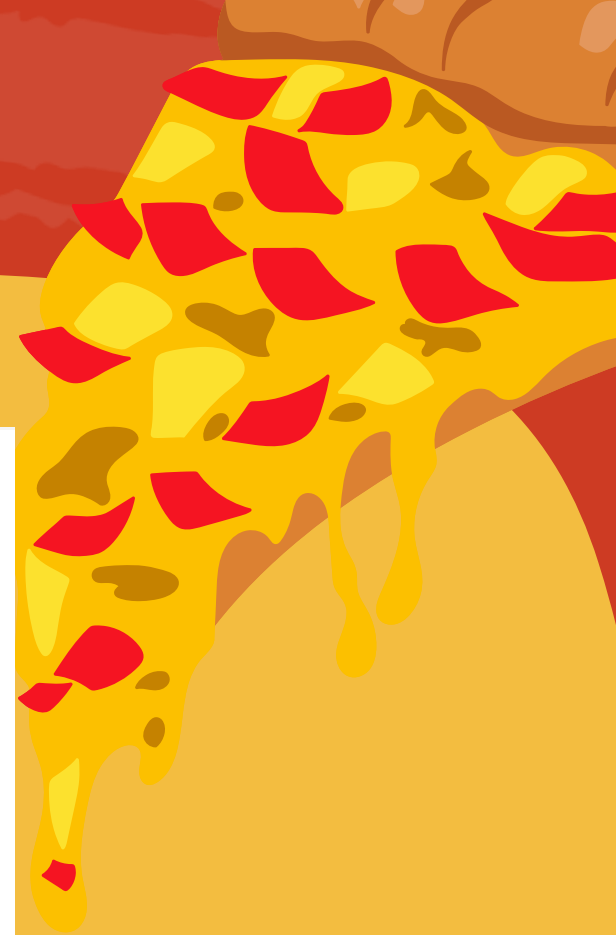




```
1  -- 2 Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4      round(SUM(order_details.quantity * pizzas.price),2) AS total_revenue
5  FROM
6      order_details
7      JOIN
8      pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	total_revenue
▶	817860.05







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

Result Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	name	price				
▶	The Greek Pizza	35.95				





```
1  # 4 Identify the most common pizza size ordered.
2
3  • SELECT
4      size,
5      COUNT(order_details.order_details_id) AS most_ordered_size
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY most_ordered_size DESC
12 LIMIT 1;
```

Result Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:	Fetch rows:
	size	most_ordered_size				
▶	L	18526				



1 # 5 List the top 5 most ordered pizza types along with their quantities.  
2  
3 • SELECT  
4 pizza\_types.category, SUM(order\_details.quantity) AS quant  
5 FROM  
6 pizza\_types  
7 JOIN  
8 pizzas ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id  
9 JOIN  
10 order\_details ON pizzas.pizza\_id = order\_details.pizza\_id  
11 GROUP BY category  
12 ORDER BY quant DESC;






Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	category	quant
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050









```
1  -- 6 Join the necessary tables to find the total quantity of each pizza category ordered.
2
3  • SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9      JOIN
10     order_details ON pizzas.pizza_id = order_details.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY quantity DESC
13 LIMIT 5;
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	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

```
1  -- 7 Determine the distribution of orders by hour of the day.
2
3  • SELECT
4      HOUR(order_time), COUNT(orders.order_id) AS orders
5  FROM
6      orders
7  GROUP BY HOUR(order_time)
8  ORDER BY orders DESC;
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	HOUR(order_time)	orders
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009



```
1  -- 8 Join relevant tables to find the category-wise distribution of pizzas.
2
3  • SELECT
4      category, COUNT(name)
5  FROM
6      pizza_types
7  GROUP BY category;
8
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





```
1  -- 9 Group the orders by date and calculate the average number of pizzas ordered per day.
2
3  •  SELECT
4      ROUND(AVG(quantity), 2)
5  FROM
6      (SELECT
7          orders.order_date, SUM(order_details.quantity) AS quantity
8      FROM
9          orders
10         JOIN order_details ON orders.order_id = order_details.order_id
11        GROUP BY orders.order_date) AS order_quantity;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:




ROUND(AVG(quantity), 2)

▶ 138.47



```
1  -- 10 Determine the top 3 most ordered pizza types based on revenue.
2
3  • SELECT
4      pizza_types.name,
5      SUM(order_details.quantity * pizzas.price) AS revenue
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
15
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5







```
1  -- 11 Calculate the percentage contribution of each pizza type to total revenue.
2
3  • SELECT
4      pizza_types.category,
5      ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
6          ROUND(SUM(pizzas.price * order_details.quantity),
7              2) AS total_sale
8          FROM
9              order_details
10             JOIN
11                 pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
12          2) AS revenue
13  FROM
14      pizza_types
15      JOIN
16      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
17      JOIN
18      order_details ON pizzas.pizza_id = order_details.pizza_id
19  GROUP BY pizza_types.category
20  ORDER BY revenue DESC;
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68







```
1  -- 12 Analyze the cumulative revenue generated over time.
2
3  • select order_date,
4     round(sum(revenue)over(order by order_date),2) as cum_revenue
5  from
6  (select orders.order_date, sum(order_details.quantity*pizzas.price) as revenue
7   from order_details join
8   pizzas on order_details.pizza_id = pizzas.pizza_id join
9   orders on order_details.order_id= orders.order_id group by orders.order_date)as sales;
```




Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	order_date	cum_revenue
▶	2015-01-01	2713.85
	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
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4

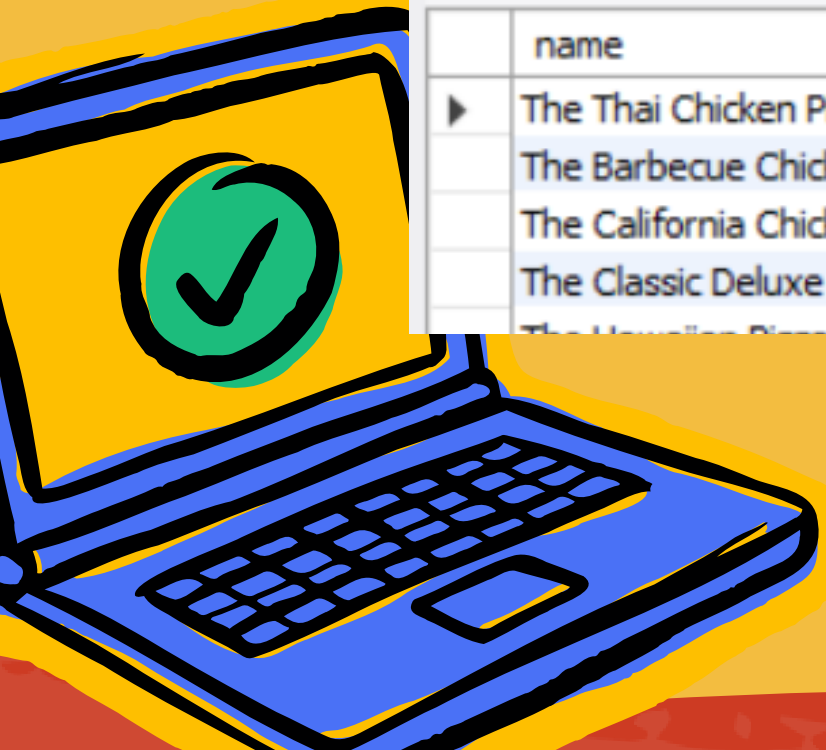
Result 1 x



```
1  -- 13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3  • select name,revenue from
4  (select category,name,revenue, rank() over(partition by category order by revenue desc)as rn from
5  (select pizza_types.category,pizza_types.name, sum((order_details.quantity)*(pizzas.price))as revenue from pizza_types join
6  pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id join order_details on pizzas.pizza_id=order_details.pizza_id
7  group by pizza_types.category,pizza_types.name)as sales)as b
8  where rn <=3;
```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	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	33373.25





# Thank You!



## Email

abhipgjain@gmail.com



## Website

[linkedin](#)



## Phone Number

9158789698



[GitHub](#)

