

# SQL PROJECT

## PIZZASS SALES

- Analyzing Pizza Sales with SQL: A Database Approach
- SQL Database for Tracking Pizza Sales and Customer Data
- Pizza Sales Insights through SQL Querying
- Building a SQL Database for Pizza Restaurant Sales Analytics



# INTRODUCTION ABOUT THE PROJECT

The purpose of this project is to create a database system that efficiently tracks and analyzes the sales of a pizza restaurant.

The goal is to design an SQL-based solution to store and retrieve important data, such as orders, customers, menu items, and sales transactions.

By organizing and analyzing this data, the system can help the restaurant better understand its sales performance, customer preferences, and inventory management

In this project, the key components include: total revenue generated

most ordered pizzas , quantity based top 3 pizza,  
percentage included by revenue ,name, category etc

# PIZZA SALES QUERIES

## 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.

Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_order  
FROM  
    pizzass.orders;
```

Result Grid	
	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales

```
SELECT  
    round(SUM(orders_details.quantity * pizzas.price),2) as total_revenue  
FROM  
    orders_details  
        JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid	
	total_revenue
▶	817860.05

# Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC;
```

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Spinach Supreme Pizza	20.75
	The Barbecue Chicken Pizza	20.75
	The California Chicken Pizza	20.75
	The Spicy Italian Pizza	20.75
	The Chicken Alfredo Pizza	20.75
	The Chicken Pesto Pizza	20.75
	The Italian Supreme Pizza	20.75
	The Southwest Chicken Pizza	20.75
	The Prosciutto and Arugula...	20.75
	The Pepper Salami Pizza	20.75

# Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
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(orders_details.quantity) AS quantity
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
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter 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

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

```
SELECT  
    pizza_types.category,  
    SUM(orders_details.quantity) AS quantity  
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.category  
ORDER BY quantity DESC;
```

	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 orders_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

Result Grid | Filter

	hour	orders_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

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

```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid		
	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
select round(avg(quantity),0) from
(select orders.order_date, sum(orders_details.quantity) as quantity
 from orders join orders_details
on orders.order_id = orders_details.order_id
group by orders.order_date) as order_quantity ;
```

	round(avg(quantity),0)
▶	138

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

```
select pizza_types.name,  
sum(orders_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id= pizza_types.pizza_type_id  
join orders_details  
on orders_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 type to total revenue

```
select pizza_types.category,  
round(sum(orders_details.quantity * pizzas.price) /(SELECT  
    round(SUM(orders_details.quantity * pizzas.price),2) as total_revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id)*100,2) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id= pizza_types.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

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

```
select pizza_types.name,  
       round(sum(orders_details.quantity * pizzas.price) / (SELECT  
                                         round(SUM(orders_details.quantity * pizzas.price), 2) as total_revenue  
FROM  
       orders_details  
       JOIN  
         pizzas ON pizzas.pizza_id = orders_details.pizza_id)*100, 2) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id= pizza_types.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name order by revenue desc;
```

	name	revenue
▶	The Thai Chicken Pizza	5.31
	The Barbecue Chicken Pizza	5.23
	The California Chicken Pizza	5.06
	The Classic Deluxe Pizza	4.67
	The Spicy Italian Pizza	4.26
	The Southwest Chicken Pizza	4.24
	The Italian Supreme Pizza	4.09
	The Hawaiian Pizza	3.95
	The Four Cheese Pizza	3.95
	The Sicilian Pizza	3.78
	The Pepperoni Pizza	3.69
	The Greek Pizza	3.48
	The Mexicana Pizza	3.27
	The Five Cheese Pizza	3.19

# Analyze the cumulative revenue generated over time.

```
select order_date,  
       round(sum(revenue) over (order by order_date),2)as cum_revenue  
  from  
(select orders.order_date,  
        round( SUM(orders_details.quantity * pizzas.price),2) as revenue  
   FROM orders_details JOIN pizzas  
      ON pizzas.pizza_id = orders_details.pizza_id  
  join orders  
    on orders.order_id = orders_details.order_id  
 group by orders.order_date) as sales;
```

	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
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7

# 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 rnk  
from  
(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.category,pizza_types.name)as a) as b  
where rnk<= 3;
```

	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 Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

# CONCLUSION

The Pizza Sales Analysis System successfully demonstrates how SQL can be used to manage and analyze sales data for a pizza restaurant. By organizing data into relevant tables—such as total revenue generated, top 3 pizzas, cumulative revenue by category, name and order specifics—the system provides a structured way to store and retrieve information essential for business operations.

Through the use of SQL queries, we were able to derive valuable insights, such as identifying the best-selling pizzas, understanding customer preferences, and calculating sales revenue. Additionally, this project highlights the importance of an efficient database structure for tracking sales over time, ensuring that the restaurant can optimize inventory, manage resources, and make informed business decisions.

The system could be further enhanced by adding features like customer loyalty programs, real-time sales tracking, and integration with payment systems or delivery services. Overall, this project serves as a strong foundation for building a comprehensive business intelligence solution for a pizza restaurant, ensuring better customer service and more effective business management.