

# PIZZAHUN T DATA ANALYSIS BY SQL

Presented by Ayush Ghayde

# ABSTRACT

The "Pizza Hunt" database project aims to improve management of pizza orders in a fictional establishment through a relational database management system (RDBMS).

Using SQL, it stores and manipulates data on pizzas, orders, types, and details across four main tables: orders, order\_details, pizza\_types, and pizzas. The project emphasizes the role of relational databases in boosting operational efficiency, enhancing customer service, and supporting data-driven decisions in the fast-food industry.

# INTRODUCTION

In today's fast-paced food service industry, efficient operations and data management are vital, especially for pizza establishments facing rising online ordering and delivery demands. This project centers on creating the "Pizza Hunt" database to support effective order processing and inventory management.

# GOALS

The SQL project aims to analyze pizza sales data through 10 targeted queries to:

- Understand customer preferences for popular pizza types, sizes, and categories.
- Evaluate sales performance by calculating total orders, revenue, and identifying highest-priced pizzas.
- Analyze trends by examining order distribution over time to reveal customer behavior patterns.

Provide actionable insights to inform business strategies, improve customer satisfaction, and optimize inventory management.

# METHODOLOGY

1. Data Collection:
  - Gather data from relevant tables (e.g., orders, pizzas).
2. Data Cleaning:
  - Remove duplicates and handle null values for consistency.
3. Database Design:
  - Understand relationships between tables to support queries.
4. Query Execution:
  - Run SQL queries to extract necessary data for analysis.
5. Data Analysis:
  - Analyze query results to identify trends and insights.

# Q1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
3 •   SELECT  
4       COUNT(order_id) AS total_orders  
5   FROM  
6       orders;
```

	total_orders
▶	21350

## Q2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

- **SELECT**

```
    ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
```

```
FROM
```

```
order_details
```

```
JOIN
```

```
pizzas ON pizzas.pizza_id = order_details.pizza_id
```

	total_sales
▶	817860.05

### Q3) 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 limit 1;
```

	name	price
▶	The Greek Pizza	35.95

## Q4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERD

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order count DESC;
```

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

# Q5) LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 5;
```

	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

## Q6) JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZAS CATEGORY ORDERD

```
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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## Q7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT  
    HOUR(order_time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2000

## Q8) JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS

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

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

# Q9) DETERMINE THE TOP 5 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 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;
```

	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 Spicy Italian Pizza	34831.25

## Q10) GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NO OF PIZZAS ORDERD PER DAY

```
SELECT  
    AVG(quantity)  
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	
	AVG(quantity)
▶	138.4749

# CONCLUSION

The SQL project analyzed pizza sales data by addressing 10 key queries, revealing insights such as:

- Dominant pizza types and sizes for targeted marketing.
- Revenue contributions of specific pizzas for inventory and pricing strategies.

Order distribution by time to optimize staffing and efficiency during peak hours.

# QNA SESSION

THANK  
YOU