

# PIZZA-HUB ANALYSIS USING SQL



# INTRODUCTION



This project involves exploratory data analysis of an imaginary restaurant, **PizzaHub**, using **SQL**. It focuses on uncovering business insights from sales and order data to support **data-driven decisions**.

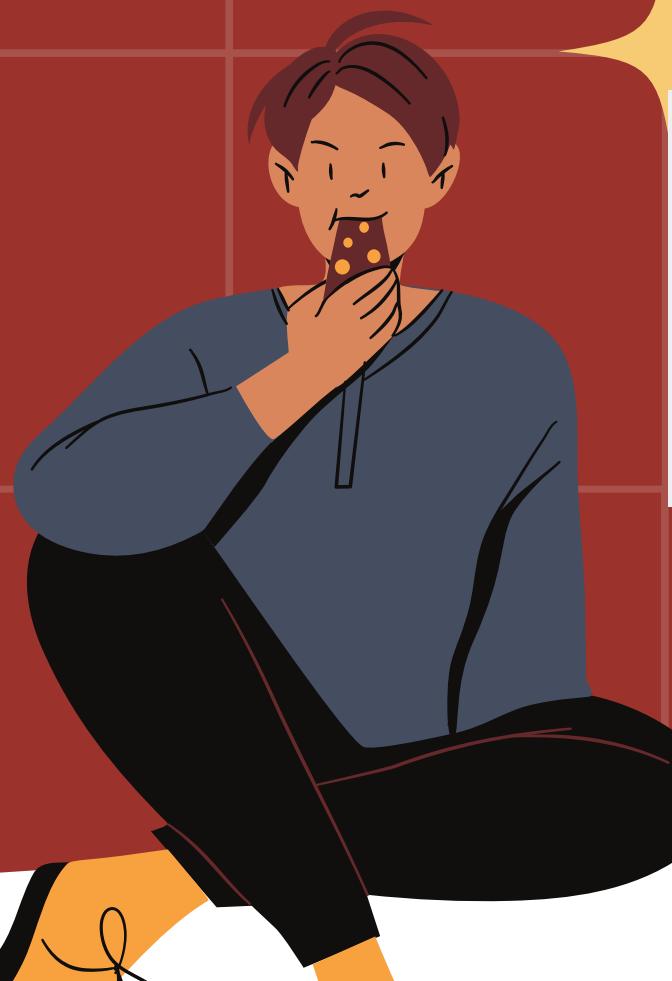
## Skills Showcased:

- SQL Joins & Aggregations
- Window Functions
- Date-Time Functions
- Data Cleaning & Transformation
- Analytical Thinking

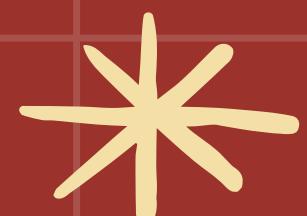
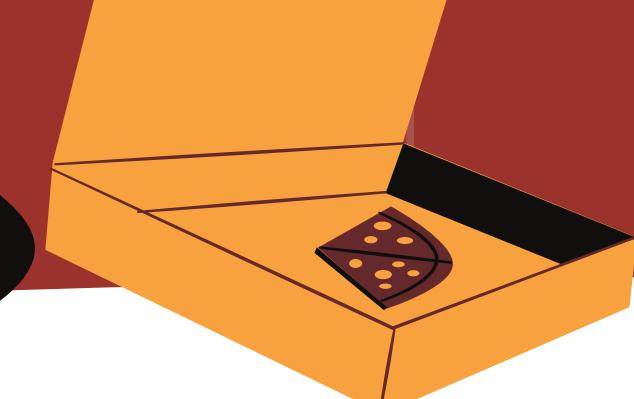
# PROBLEM STATEMENTS

What is the total count of all orders?

```
SELECT  
    COUNT(order_id) AS orders_count  
FROM  
    orders;
```



Result Grid	
	orders_count
▶	21350



# Determine the total revenue from pizza sales.

```
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),  
        2) AS revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```



Result Grid	
	revenue
▶	817860.05



# Find the most expensive 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;

Result Grid | Filter Rows:

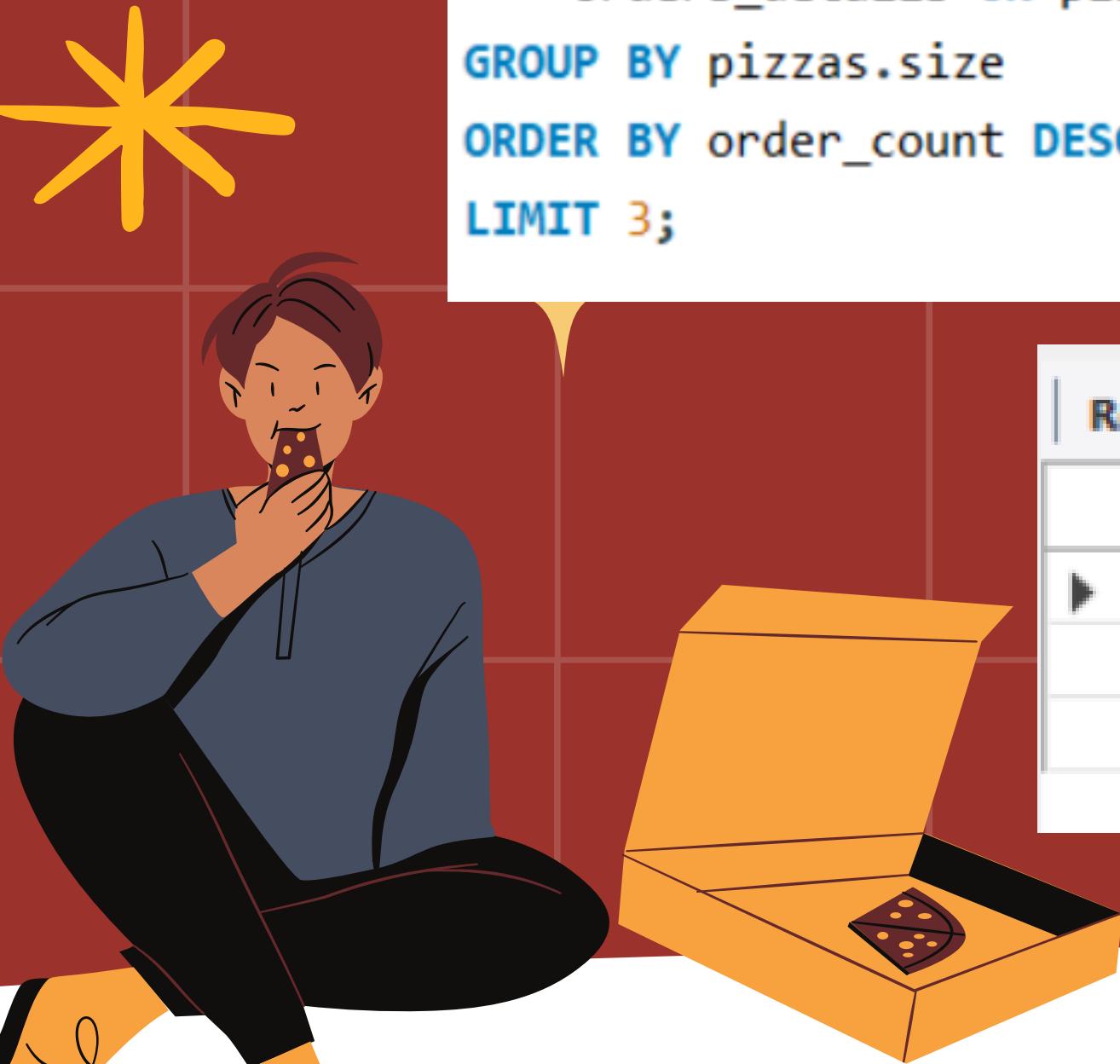
	name	price
▶	The Greek Pizza	35.95



# Determine the most frequently ordered pizza size.

```
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  
LIMIT 3;
```

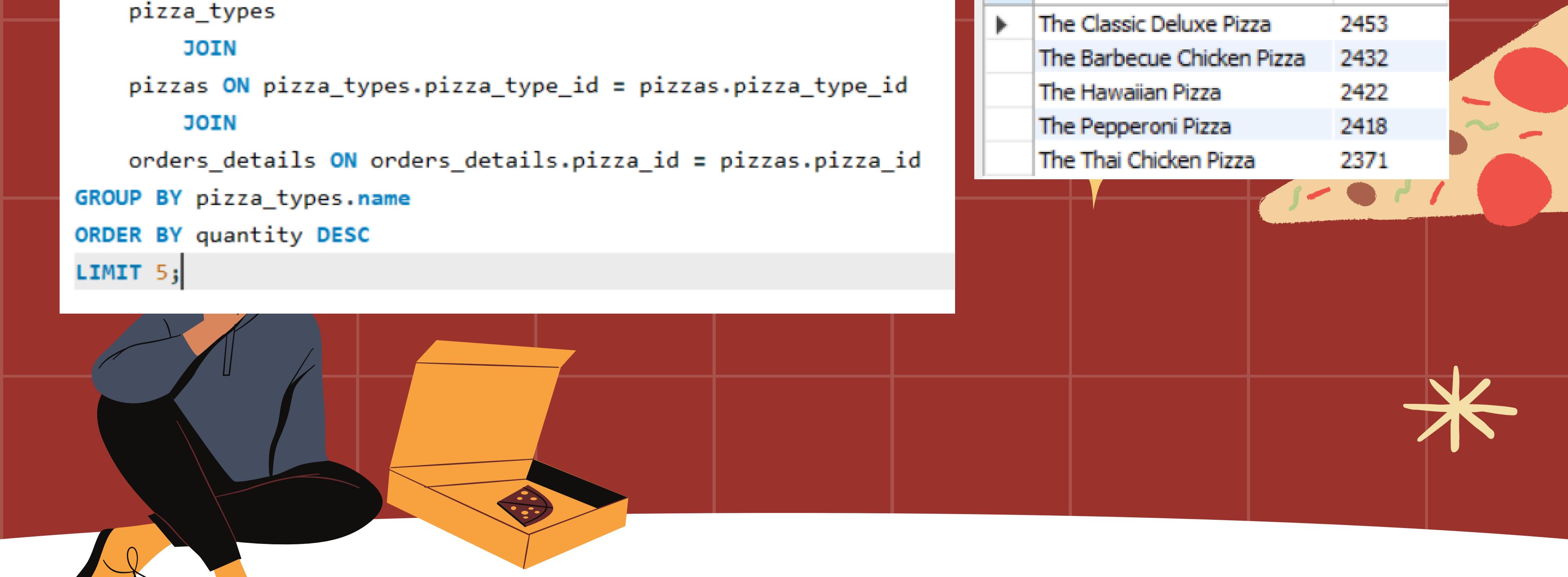
	size	order_count
▶	L	18526
	M	15385
	S	14137



# Show the top 5 most popular pizza types and the number of times each was ordered.

```
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;
```

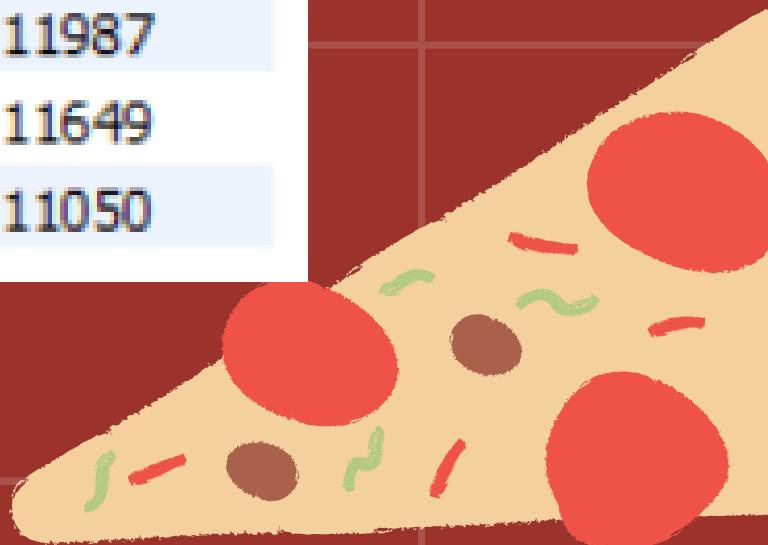
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



# Combine the relevant tables to determine the total quantity ordered for each pizza category.

```
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

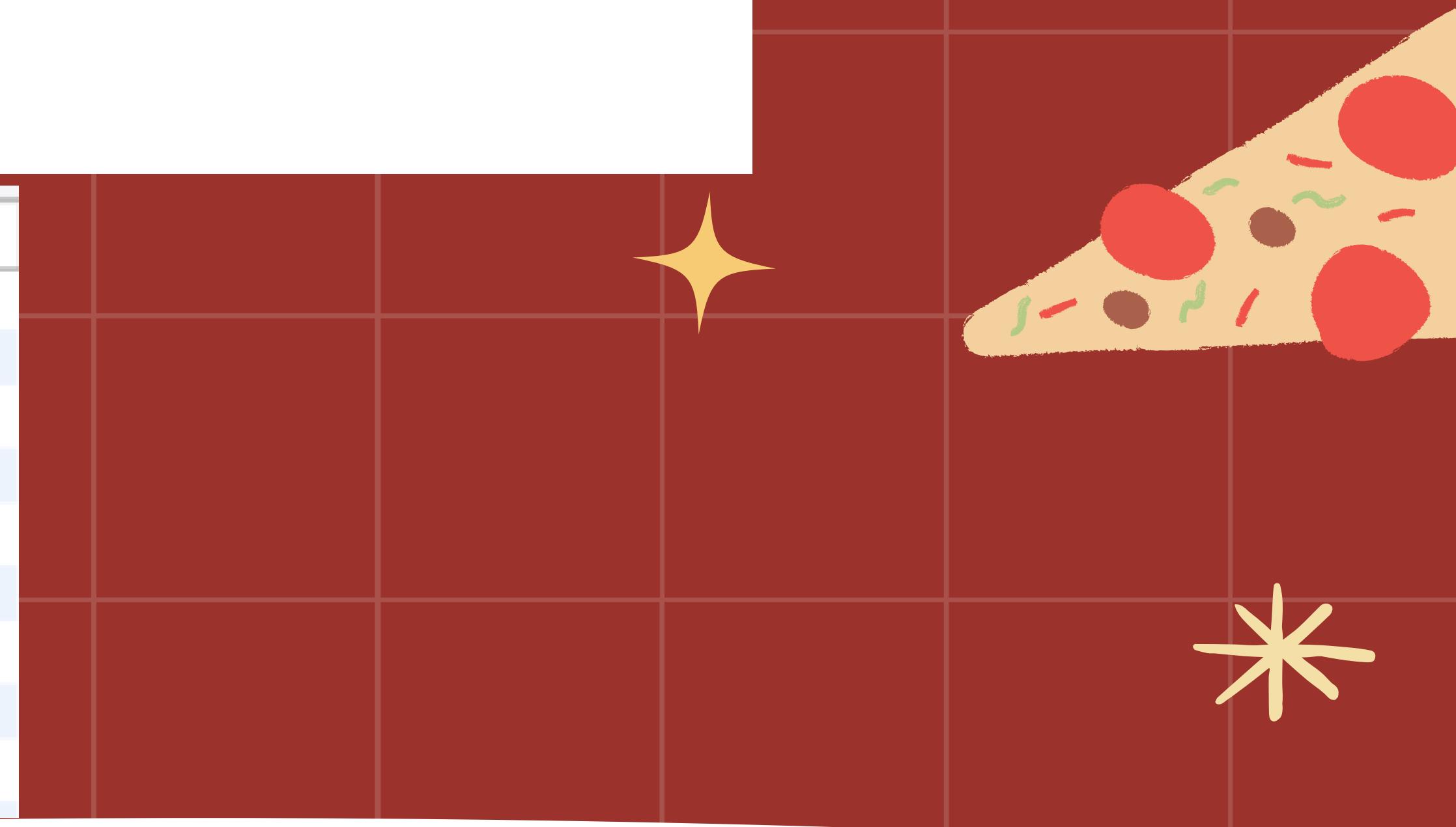


# Examine how orders are distributed across different hours in a day.

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



	hour	order_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 pizzas.

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

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



# Aggregate the orders by date and compute the daily average of pizzas ordered.

```
SELECT  
    ROUND(AVG(quantity), 0) AS average_pizza_ordered  
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 total_quantity;
```



	average_pizza_ordered
▶	138



# Identify the top three pizza types with the highest revenue from orders.

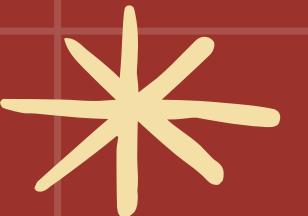
```
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
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,
    (SUM(orders_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_details.quantity * pizzas.price),
            2))
    FROM
        orders_details
        JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100 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
ORDER BY revenue;
```

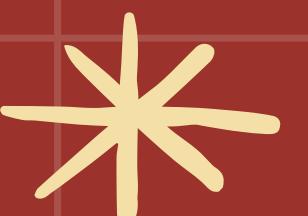
	category	revenue
▶	Veggie	23.682590927384577
	Chicken	23.955137556847287
	Supreme	25.45631126009862
	Classic	26.90596025566967



# 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,  
        SUM(orders_details.quantity * pizzas.price) AS revenue  
    FROM  
        orders_details  
    JOIN  
        pizzas ON orders_details.pizza_id = pizzas.pizza_id  
    JOIN  
        orders ON orders.order_id = orders_details.order_id  
    GROUP BY  
        orders.order_date  
) AS sales;
```

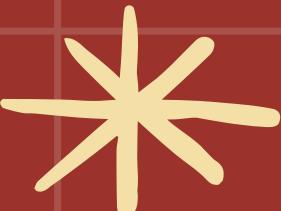
	order_date	cum_revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5



# Identify the top three pizza types with the highest revenue within each pizza category

```
SELECT
    name, revenue
FROM (
    SELECT
        category, name, revenue,
        RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
    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 rn <= 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



# CONCLUSION AND KEY LEARNINGS

HANDS-ON EXPERIENCE WITH  
RELATIONAL DATABASES AND  
SCHEMA DESIGN.

MASTERY IN WRITING CLEAN,  
EFFICIENT SQL QUERIES.

ABILITY TO TRANSFORM BUSINESS  
NEEDS INTO ANALYTICAL SOLUTIONS.