Pizza Sales SQL Project

Presented by:
Anusua Biswas

Brief Overview of the Project

The Pizza Sales SQL project involves designing a database to store and analyze data related to pizza orders, including order details, pizza types, and sales information. By leveraging SQL queries, the project aims to gain insights into sales patterns and customer preferences.

An Analysis of Pizza Orders and Sales

Objectives of the Project

- Create a robust database schema for pizza sales data.
- Collect and store comprehensive data on orders and pizzas.
- Analyze the data to identify trends and insights.
- Generate actionable recommendations for improving sales and operations.

Importance of Analyzing Pizza Sales Data

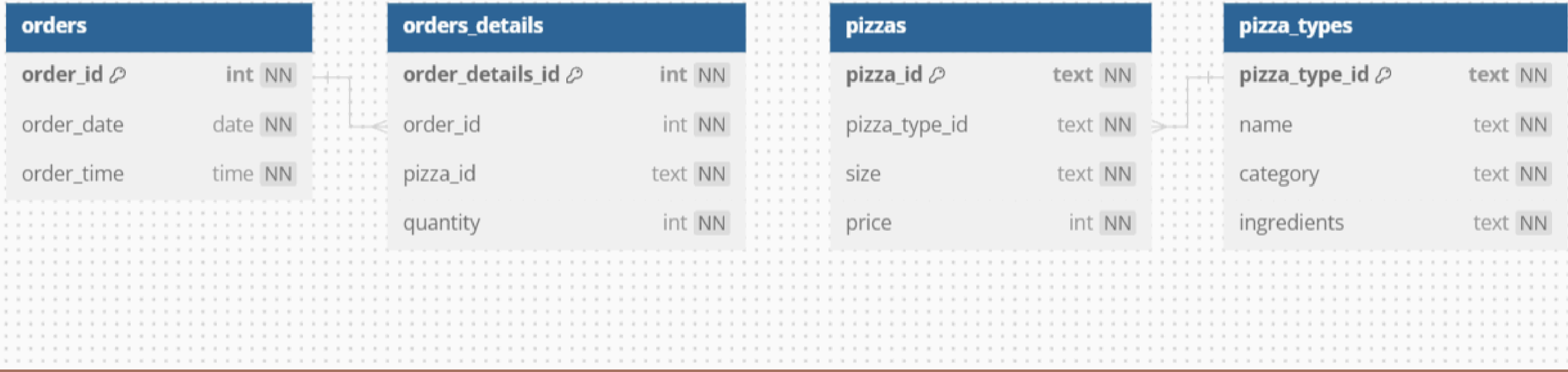
- **Identify best-selling pizzas and peak order times.**
- **Understand customer preferences and behavior.**
- **Optimize inventory management and reduce waste.**
- **Increase revenue through targeted marketing strategies.**

Description of the Database Schema

The Pizza Sales database schema consists of four main tables:

1. Orders
    - Columns: order_id (primary key), order_date, order_time
    - Purpose: Stores basic order information.
2. Order_Details
    - Columns: order_details_id (primary key), order_id (foreign key), pizza_id, quantity
    - Purpose: Contains details of each order, linking to the orders table.
3. Pizzas
    - Columns: pizza_id (primary key), pizza_type_id (foreign key), size, price
    - Purpose: Holds information about individual pizzas.
4. Pizza_Types
    - Columns: pizza_type_id (primary key), name, category, ingredients
    - Purpose: Describes different types of pizzas.

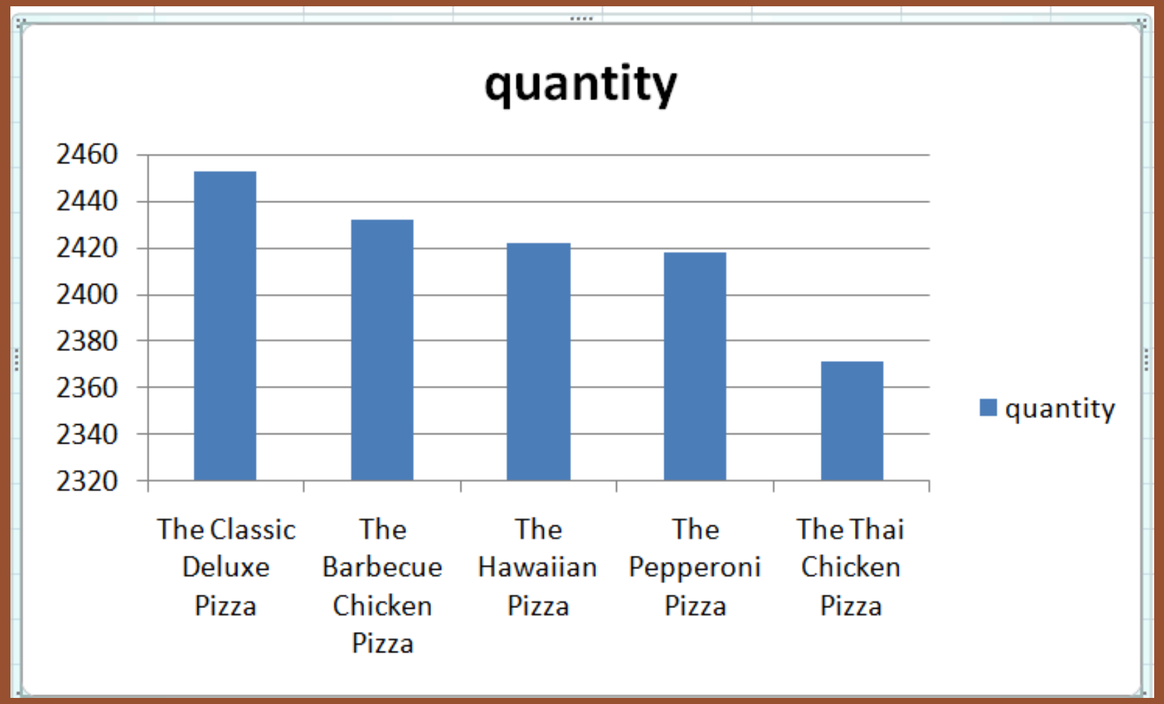
Entity-Relationship Diagram



1. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
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;
```

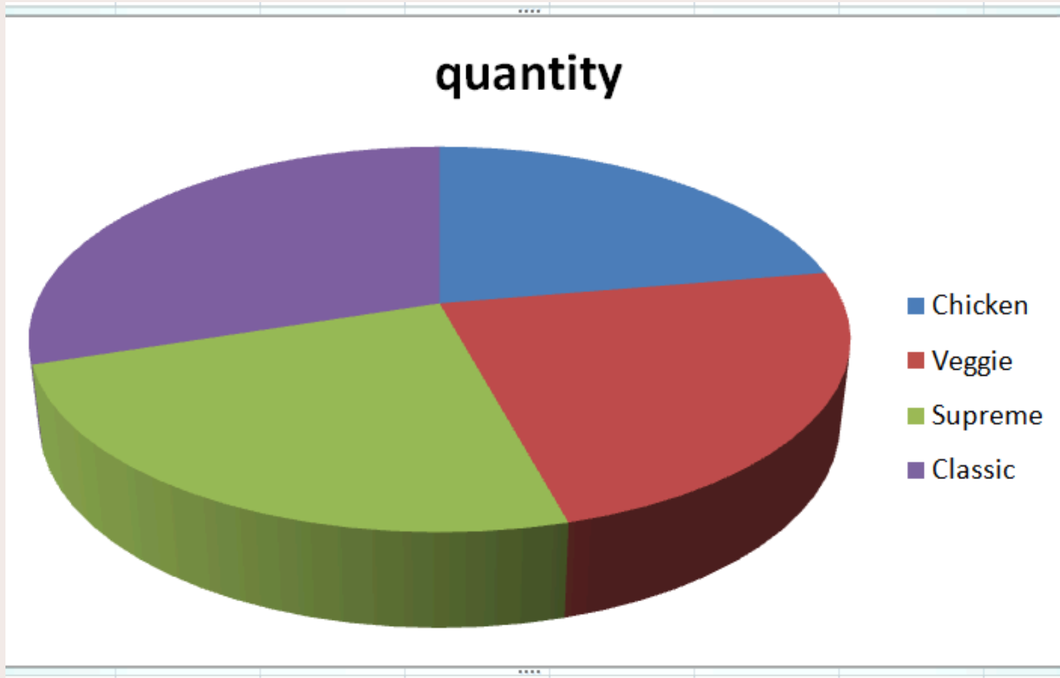
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



2. 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;
```

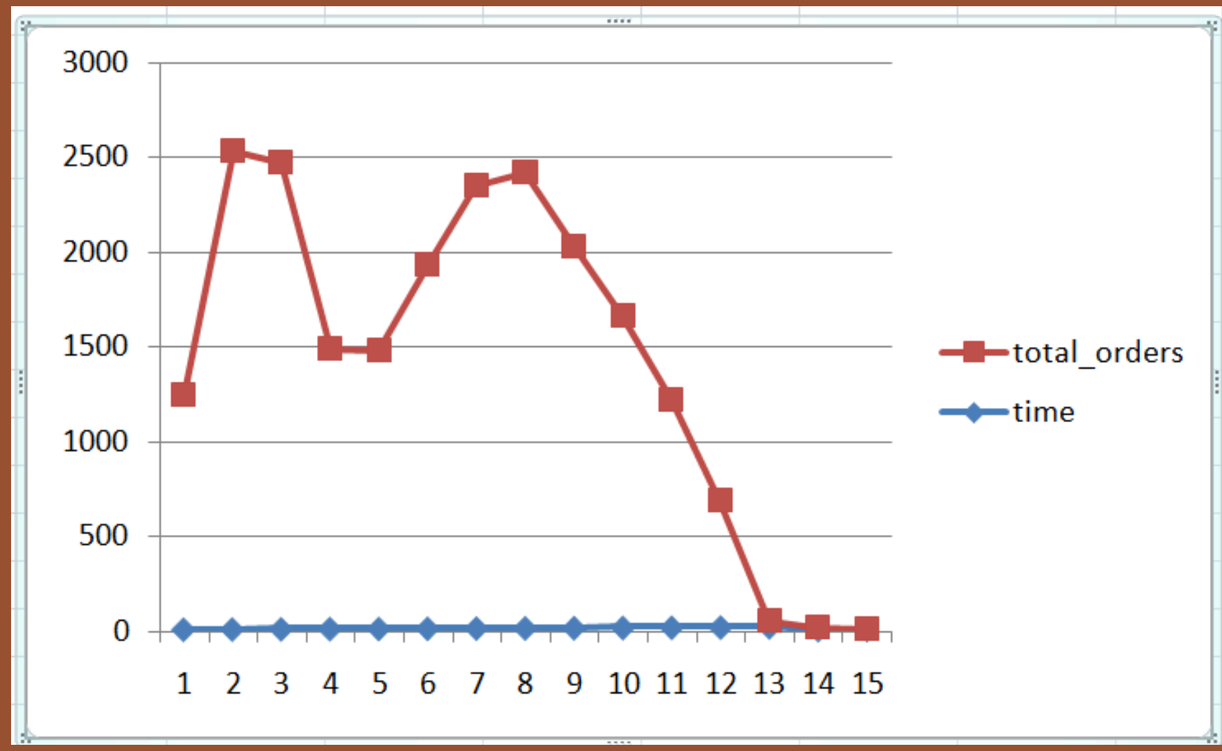
category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888



3. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS time,
    COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY time;
```

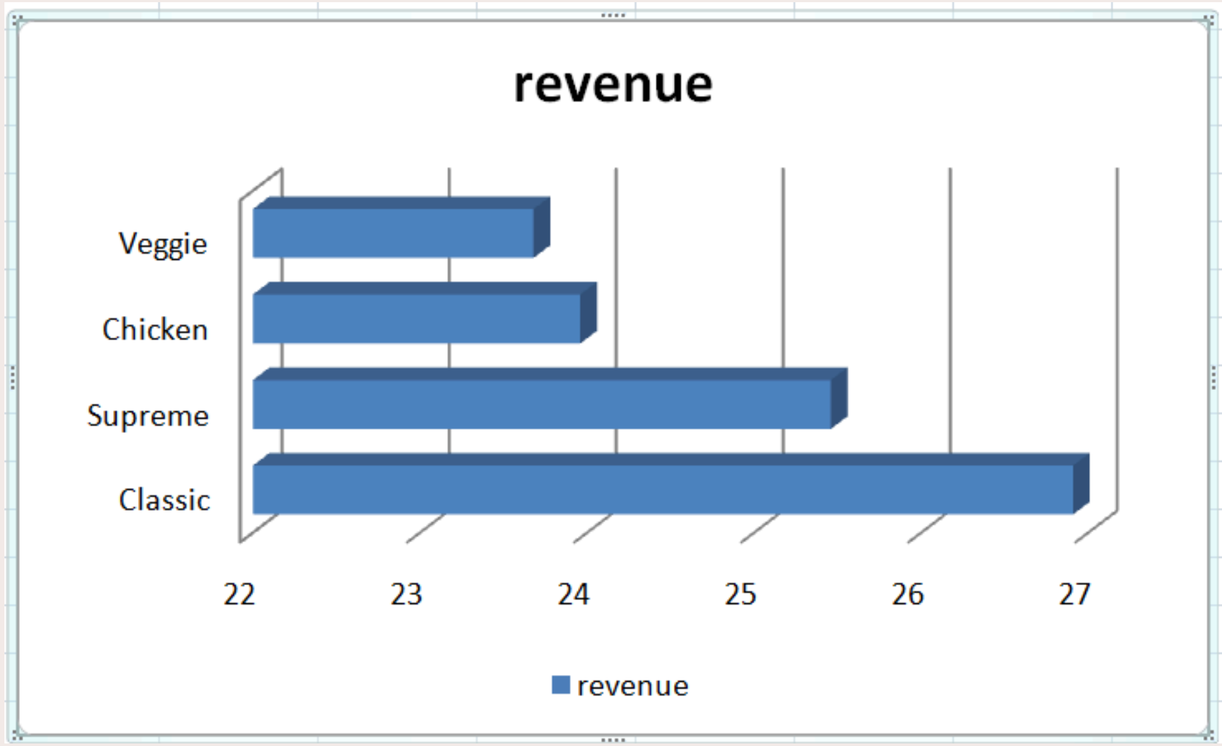
time	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336



4. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    orders_details
                        JOIN
                    pizzas ON orders_details.pizza_id = pizzas.pizza_id)) * 100,
            2) 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 DESC;
```

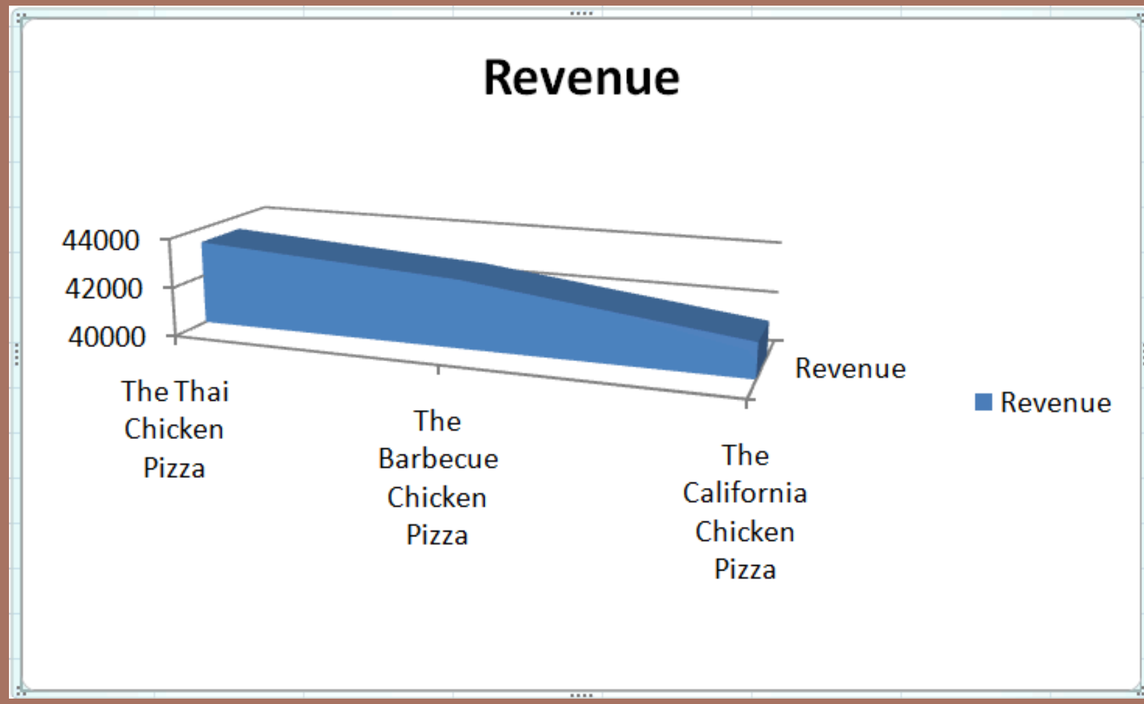
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



5. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    round(SUM(orders_details.quantity * pizzas.price),2) 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.name
ORDER BY SUM(orders_details.quantity * pizzas.price) DESC
LIMIT 3;
```

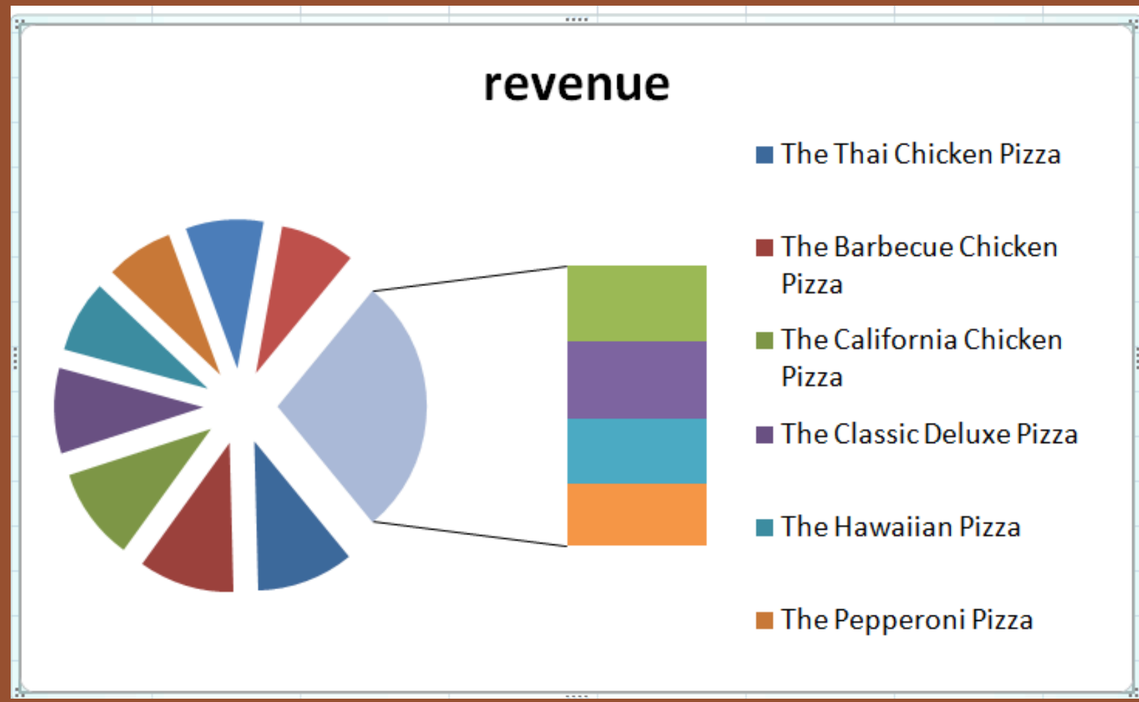
name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



6. 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 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



Conclusion

Summary of the Project :

The Pizza Sales SQL project involved designing a database to store and analyze data on pizza orders, details, types, and sales. Through SQL queries, we gained valuable insights into sales patterns and customer preferences.

Key Takeaways :

- Identified top-selling pizzas and peak order times.
- Analyzed customer preferences for different pizza types and sizes.
- Revealed trends in sales over time, aiding in inventory and marketing strategies.

Recommendations for Pizza Hut Based on Findings :

- Focus marketing efforts on popular pizzas and peak times.
- Adjust inventory levels based on sales trends to reduce waste.
- Consider introducing new pizza types based on customer preferences and ingredient popularity.


Thank
You!