# Pizza
# Sales report

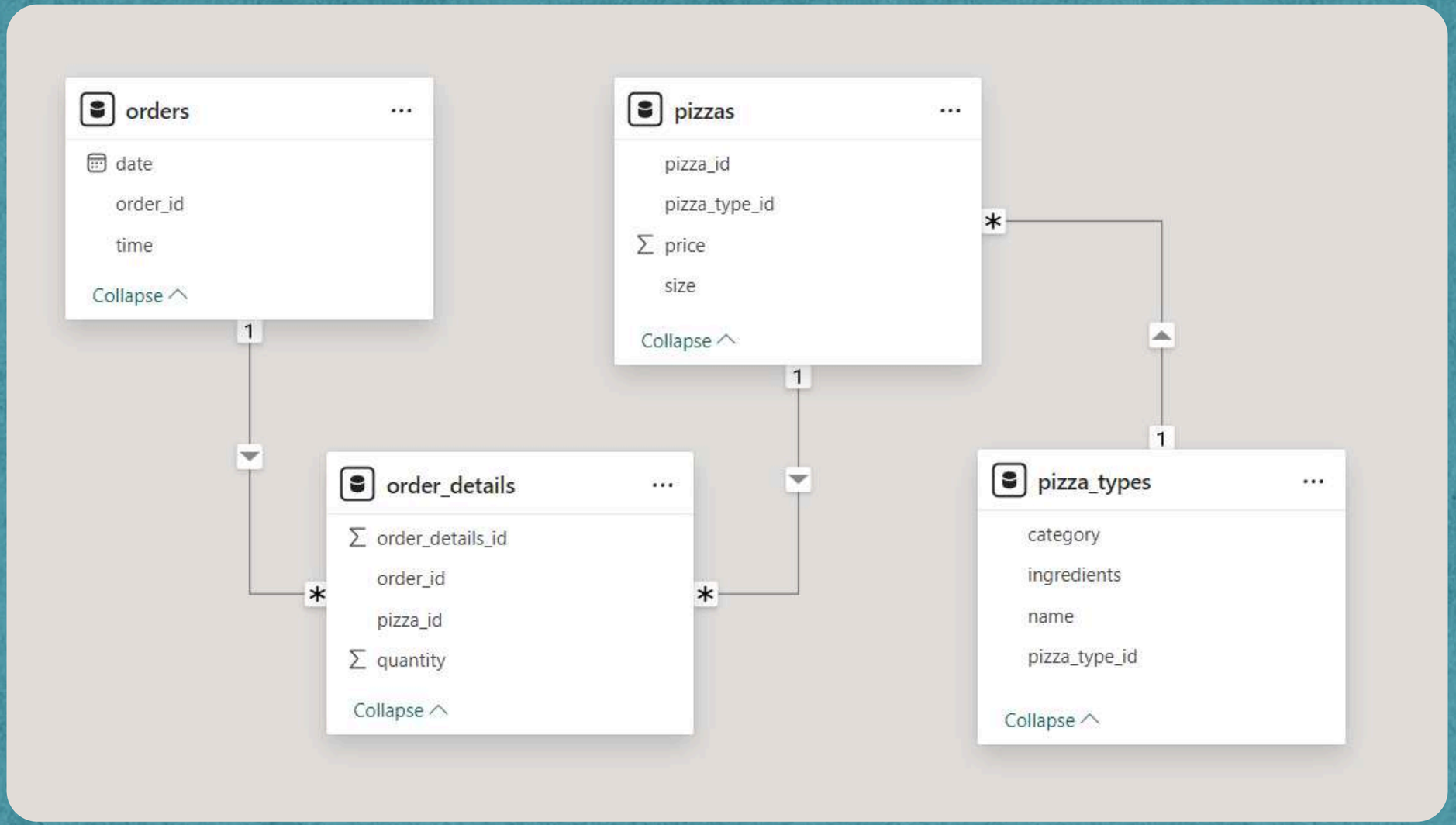
*Bhaskar Sinha*

2024

# Introduction

Welcome to my comprehensive SQL project focused on analyzing pizza sales data. This project utilizes SQL to extract valuable insights from a pizza sales database, addressing a range of business questions from basic to advanced levels. By querying the database, we can uncover trends, understand customer preferences, and identify key factors driving sales performance.

# Entity-Relationship Diagram



# Questions Solved

**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.

```sql
SELECT
    COUNT(order_id) AS "Total Orders"
FROM
    orders;
```

| | Total Orders |
|---|---|
| ▶ | 21350 |

# Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS 'Total  Revenue ($)'
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

| | Total Revenue ($) |
|---|---|
| ▶ | 817860.05 |

# Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name AS 'Name of the Pizza',
    pizzas.price AS 'Price ($)'
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

| | Name of the Pizza | Price ($) |
|---|---|---|
| ▶ | The Greek Pizza | 35.95 |

# Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size AS 'Size of Pizza',
    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
LIMIT 1;
```

| | Size of Pizza | Order Count |
|---|---|---|
| ▶ | M | 15385 |

# List the top 5 most ordered pizzas along with their quantities.

```sql
SELECT
    pizza_types.name AS 'Name of the Pizza',
    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 of the Pizza | 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.

```sql
SELECT
    pizza_types.category AS 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 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.

```sql
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 |
| 20 | 1642 |
| 21 | 1198 |
| 22 | 663 |
| 23 | 28 |
| 10 | 8 |
| 9 | 1 |

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

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

| category | count |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

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

```sql
SELECT
    ROUND(AVG(Quantity), 0) as "Average no. of orders per day"
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;
```

| Average no. of orders per day |
| --- |
| 138 |

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

```sql
SELECT
    pizza_types.name AS 'Name of the Pizza',
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS 'Revenue ($)'
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY 'Revenue ($)' DESC
LIMIT 3;
```

| Name of the Pizza | Revenue ($) |
|---|---|
| The Hawaiian Pizza | 32273 |
| The Classic Deluxe Pizza | 38180 |
| The Five Cheese Pizza | 26066 |

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

```sql
SELECT
    pizza_types.category AS 'Category of Pizza',
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS 'Total  Revenue ($)'
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS 'Contribution in Revenue (%)'
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;
```

| Category of Pizza | Contribution in Revenue (%) |
|---|---|
| Classic | 26.91 |
| Veggie | 23.68 |
| Supreme | 25.46 |
| Chicken | 23.96 |

# Analyze the cumulative revenue generated over time.

```sql
SELECT
        order_date as 'Order Date',
            round(sum(revenue) over(order by order_date), 2) as 'Cumulative Revenue'
                from
    (SELECT
            orders.order_date,
                SUM(order_details.quantity * pizzas.price) AS revenue
        FROM
            order_details
                JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
                JOIN
        orders ON orders.order_id = order_details.order_id
        GROUP BY orders.order_date) AS sales;
```

| Order Date | Cumulative 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 |

# Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
SELECT Category,
       name 'Top 3 Pizzas under specified category respectively',
       Revenue as 'Revenue ($)'
           FROM
(SELECT
        category AS 'Category',
        name,
        revenue, RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
           FROM
(SELECT
    pizza_types.category,
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
           2) AS revenue
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 , pizza_types.name) as a) as b
WHERE rn <= 3;
```

| Category | Top 3 Pizzas under specified category respectively | Revenue ($) |
|---|---|---|
| Chicken | The Thai Chicken Pizza | 43434.25 |
| Chicken | The Barbecue Chicken Pizza | 42768 |
| Chicken | The California Chicken Pizza | 41409.5 |
| Classic | The Classic Deluxe Pizza | 38180.5 |
| Classic | The Hawaiian Pizza | 32273.25 |
| Classic | The Pepperoni Pizza | 30161.75 |
| Supreme | The Spicy Italian Pizza | 34831.25 |
| Supreme | The Italian Supreme Pizza | 33476.75 |
| Supreme | The Sicilian Pizza | 30940.5 |
| Veggie | The Four Cheese Pizza | 32265.7 |
| Veggie | The Mexicana Pizza | 26780.75 |
| Veggie | The Five Cheese Pizza | 26066.5 |

# Contact

7908857188

22bsm012@iiitdmj.ac.in

# Thank you!

*Bhaskar Sinha*

2024