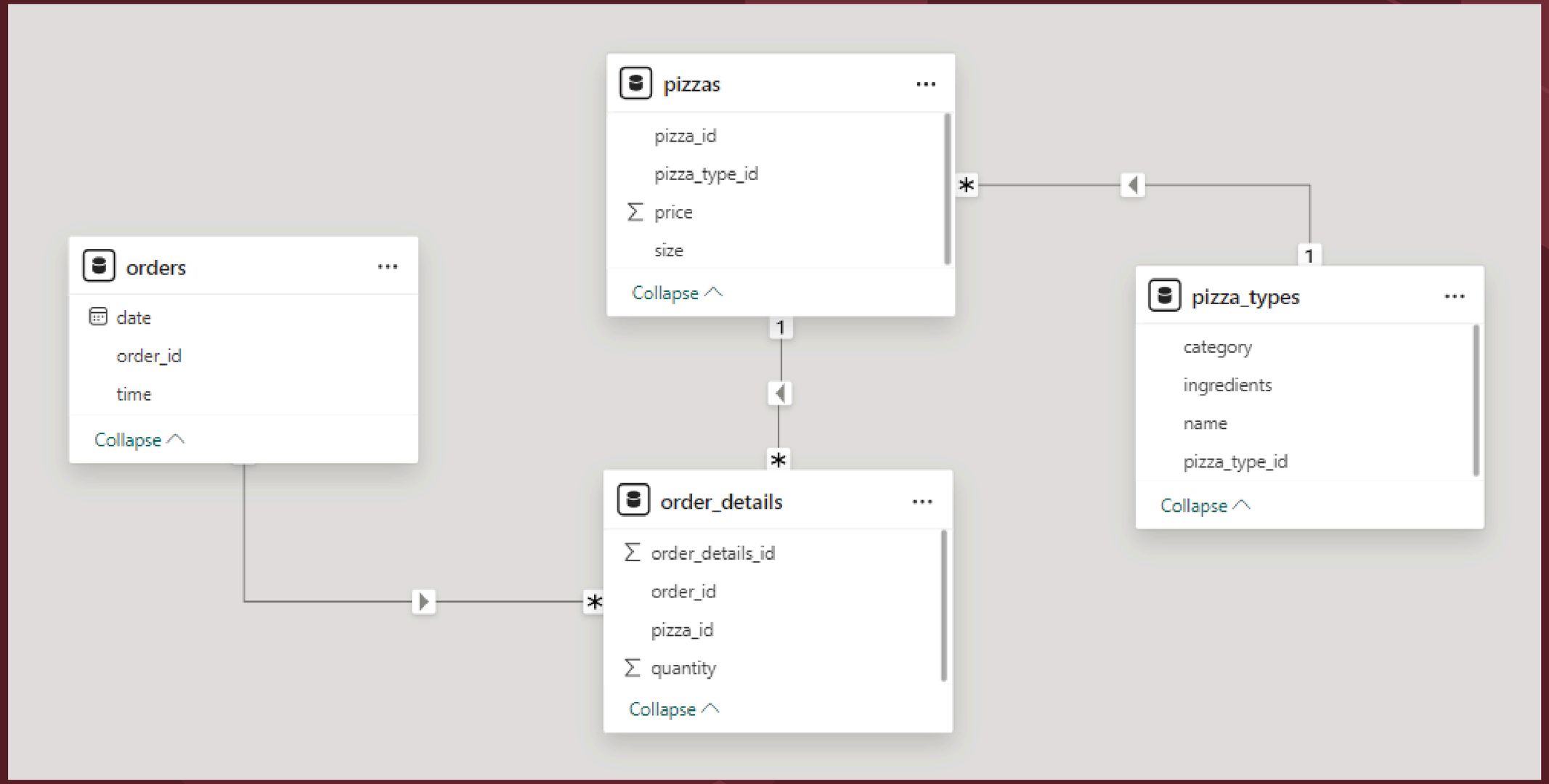
PIZZA SALES SQL PROJECT

Introduction

Hello! Myself Karan Kumar, Reg No: 2021UGME026, a fourth year undergraduate student at National Institute of Technology, Jamshedpur in Mechanical Engineering branch.

This project provides a detailed analysis of pizza sales, utilizing data extracted through SQL queries. The goal is to gain insights into the sales performance across various dimensions such as time periods, product categories, etc. By analyzing trends in pizza orders, we aim to identify key drivers of sales growth, popular products and customer preferences. The findings from this project will support data-driven decision-making, enabling the business to optimize inventory management, marketing strategies and overall operational efficiency.

Model View of the Tables



Q1. Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) as Total_Orders
FROM
    orders;
```

Result Grid

	Total_Orders
▶	29396

Q 2. Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(p.price * od.quantity), 2) AS Total_Revenue
FROM
    pizzas p
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id;
```

Result Grid

	Total_Revenue
▶	817860.05

Q 3. Identify the highest-priced pizza.

```sql
SELECT
    pt.name, p.price
FROM
    pizzas p
        INNER JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Result Grid | Filter R

	name	price
▶	The Greek Pizza	35.95

Q4. Identify the most common pizza size ordered.

```sql
SELECT
    p.size, COUNT(od.order_details_id) AS order_count
FROM
    order_details od
        INNER JOIN
    pizzas p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid

	size	order_count
▶	L	18526

Q5. List the top 5 most ordered pizza types along with their quantities.

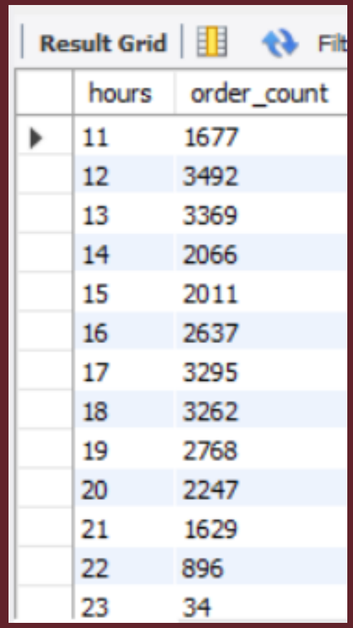
```sql
SELECT
    pt.name, SUM(od.quantity) AS quantity
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```

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

Q6. Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(o.time) AS hours, COUNT(order_id) order_count
FROM
    orders o
GROUP BY hours;
```



Result Grid

hours	order_count
11	1677
12	3492
13	3369
14	2066
15	2011
16	2637
17	3295
18	3262
19	2768
20	2247
21	1629
22	896
23	34

Q7. Find the category-wise distribution of pizzas.

```sql
SELECT
    pt.category, COUNT(pt.category) as count
FROM
    pizza_types pt
GROUP BY category;
```

Result Grid

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Q8. Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(q), 2) average_pizzas_ordered
FROM
    (SELECT
        o.date, SUM(od.quantity) AS q
    FROM
        orders o
    INNER JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.date) AS order_quantity;
```

Result Grid | Filter R

	average_pizzas_ordered
▶	190.63

Q9. Determine the top 3 most ordered pizza types based on revenue.

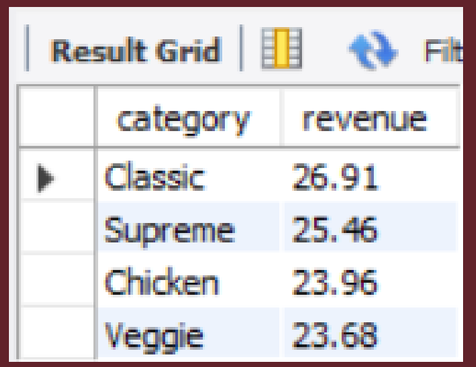
```sql
SELECT
    pt.name, SUM(quantity * price) AS revenue
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

Q 10. Calculate the percentage contribution of each pizza category to total revenue.

```
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                    ROUND(SUM(od.quantity * p.price), 2)
                FROM
                    order_details od
                        INNER JOIN
                    pizzas p ON od.pizza_id = p.pizza_id) * 100,
            2) AS revenue
FROM
    order_details od
        INNER JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        INNER JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category
ORDER BY revenue DESC;
```



Result Grid | Filt

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

THANK YOU!