SQL PROJECT:
PIZZA SALES INSIGHTS

INTRODUCTION

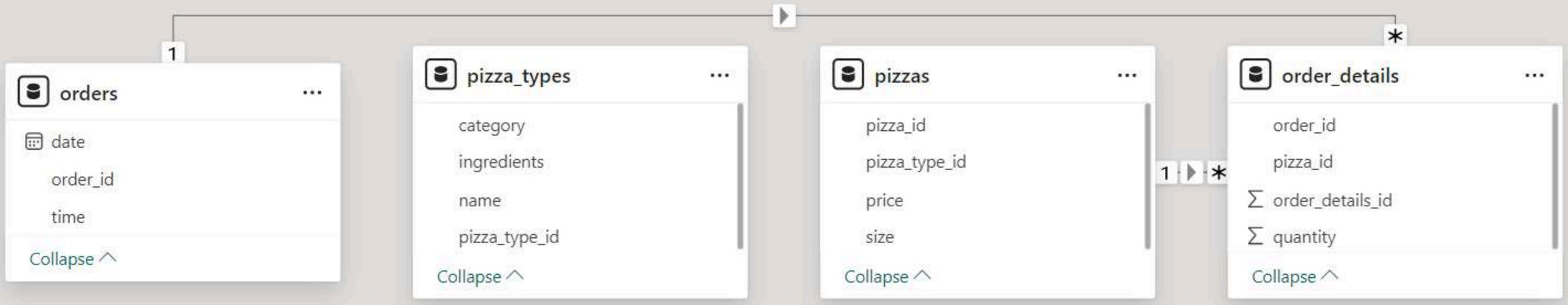
Hello, I'm Nikita Sharma, and in this project, I used SQL Queries to analyze pizza sales data and uncover valuable business insights. Working with datasets like Orders, Order Details, Pizzas, and Pizza Types, I explored:

- Total orders, revenue, and top-selling pizzas.
- Highest-priced pizza, most popular size, and top 5 pizza types.
- Hourly order trends, category distribution, and daily averages.
- Top 3 pizza types by revenue and their sales contributions over time.

This project highlights how SQL turns raw data into actionable insights, helping businesses optimize pricing, inventory, and menu strategies.

t

DATA FRAMEWORK FOR PIZZA ANALYSIS



Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

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;
```

Identify the highest-priced pizza.

```sql
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;
```

Identify the most common pizza size ordered.

```sql
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;
```

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

```sql
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;
```

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC;
```

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);
```

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

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

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

```sql
SELECT
    ROUND(AVG(total_pizzas))
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS total_pizzas
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS total_orders;
```

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) 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;
```

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON order_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
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

Analyze the cumulative revenue generated over time.

```sql
SELECT order_date,
SUM(revenue) over(order by order_date) as cum_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;
```

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

```sql
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(order_details.quantity * pizzas.price) 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 category) as revenue
where rn <= 3;
```

THANK YOU