Pizza Sql Project

About project :

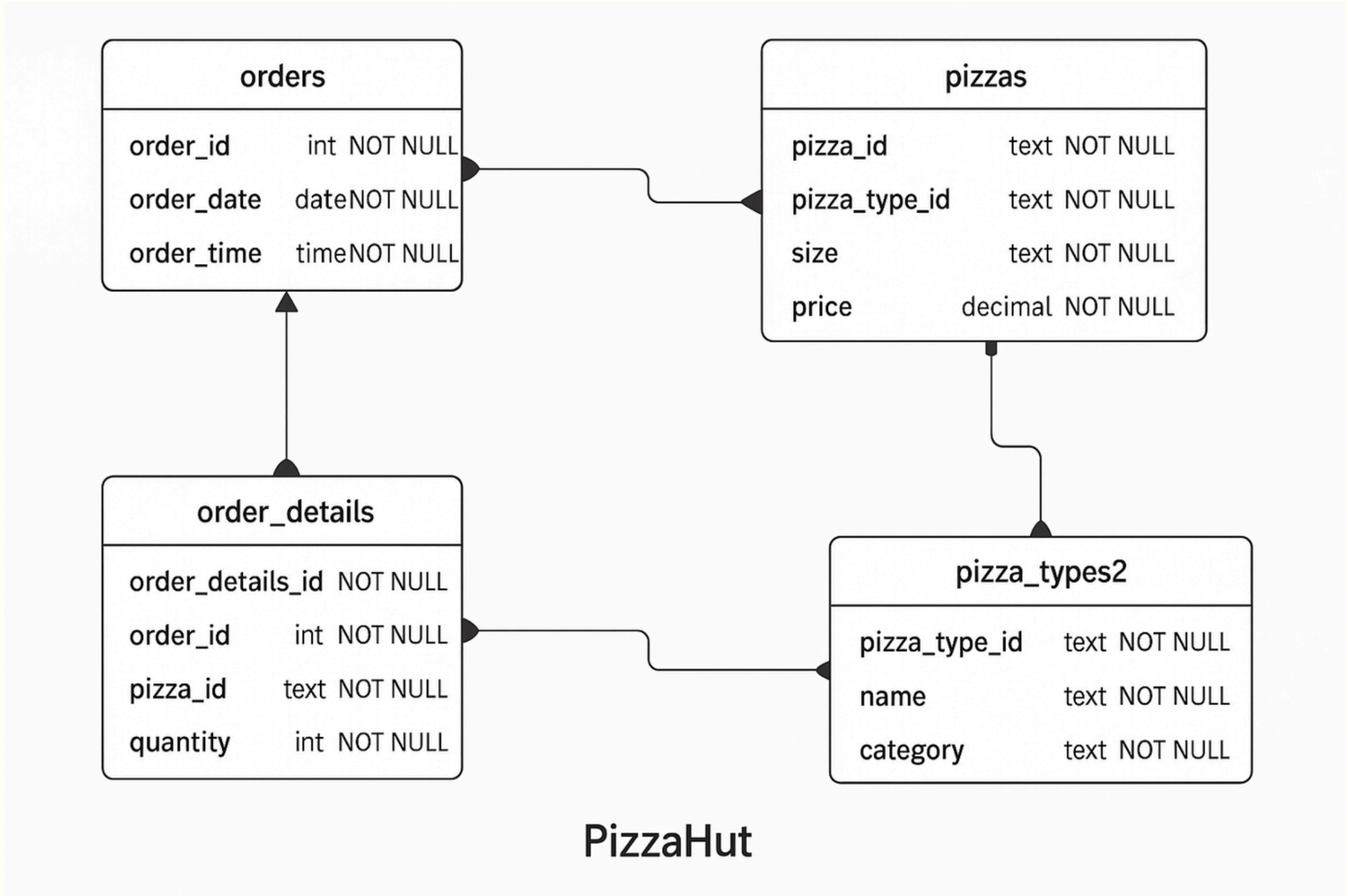
This project focuses on the design and analysis of a relational database or a pizza delivery service—PizzaHut. The objective is to build a structured database system that efficiently stores, manages, and retrieves business-critical information such as orders, pizza details, and sales performance.

Using SQL, we developed queries to extract insights into customer behavior, product performance, revenue trends, and operational patterns. The project not only highlights database design and relational integrity but also demonstrates how data can drive business decisions through analysis of ordering trends, product demand, and revenue generation.

This end-to-end solution provides a comprehensive view of PizzaHut's operations and serves as a foundation for data-driven decision-making in the food retail sector.

PIZZA SQL

SCHEMA



PIZZA SQL

```sql
#1.Retrieve the total number of orders placed.

select count(order_id) as total_orders from orders;
```

```sql
#2.Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_Revenue
FROM
    order_details
        LEFT JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

```sql
#3.Identify the highest-priced pizza.

SELECT
    pizza_types2.name, pizzas.price
FROM
    pizza_types2
        LEFT JOIN
    pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

```sql
#4.Identify the most common pizza size ordered.

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
LIMIT 1;
```

```sql
#1.Join the necessary tables to find the total quantity of each pizza category ordered.

SELECT
    pizza_types2.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizzas
        LEFT JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        LEFT JOIN
    pizza_types2 ON pizzas.pizza_type_id = pizza_types2.pizza_type_id
GROUP BY pizza_types2.category
ORDER BY quantity DESC
```

```sql
#2.Determine the distribution of orders by hour of the day.

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

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

select pizza_types2.category,count(name)as Total from pizza_types2
group by pizza_types2.category
```

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

SELECT
    ROUND(AVG(quantity), 0) as AVG
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    LEFT JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_Quantity;
```

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

SELECT
    pizza_types2.name,
    SUM(order_details.quantity * pizzas.price) AS Revenue
FROM
    pizzas
        LEFT JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        LEFT JOIN
    pizza_types2 ON pizzas.pizza_type_id = pizza_types2.pizza_type_id
GROUP BY pizza_types2.name
ORDER BY Revenue DESC
LIMIT 3;
```

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

SELECT
    pizza_types2.category,
    round((SUM(order_details.quantity * pizzas.price)/(SELECT
    SUM(order_details.quantity * pizzas.price)
             AS Total_sales
FROM
    order_details
        LEFT JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id ))* 100,2) as Revenue
FROM
    order_details
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        LEFT JOIN
    pizza_types2 ON pizzas.pizza_type_id = pizza_types2.pizza_type_id
GROUP BY pizza_types2.category
order by Revenue desc
```

```sql
#2.Analyze the cumulative revenue generated over time.

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
        LEFT JOIN
    orders ON order_details.order_id = orders.order_id
        LEFT JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.order_date)as sales;
```

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

select category,name,revenue,rank() over(partition by category order by revenue desc) as Rn
from
 (SELECT
    pizza_types2.category,
    pizza_types2.name,
    SUM(order_details.quantity * pizzas.price) AS Revenue
FROM
    pizza_types2
        JOIN
    pizzas ON pizza_types2.pizza_type_id = pizzas.pizza_type_id
        LEFT JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types2.category, pizza_types2.name) as sales;
```

KEY FINDINGS

1. **Total Orders:**
   - A clear count of all orders placed provides insight into overall business volume.
2. **Revenue Generation:**
   - Total revenue is driven by the combination of pizza quantity and individual pizza prices, indicating strong sales performance.

1. **Best Performing Products:**
   - Highest Priced Pizza: Identified to understand premium offerings.
   - Top 3 Pizza Types by Revenue: Shows the best-selling, most profitable items.
   - Top 5 Pizza Types by Quantity: Highlights customer preferences.
2. **Customer Preferences:**
   - Most Common Pizza Size: Useful for inventory and packaging planning.
   - Category-wise Sales: Helps assess popularity of pizza categories (e.g., classic, veggie, meat).
3. **Time-based Insights:**
   - Hourly Order Distribution: Helps optimize staffing and operations during peak hours.
   - Daily Average Pizzas Ordered: Shows customer engagement over time.
   - Cumulative Revenue Over Time: Tracks business growth trajectory.
4. **Revenue Contribution:**
   - By Category: Percentage contribution shows which category drives more profit.
   - Top 3 by Category: Reveals top items within each category for targeted marketing.