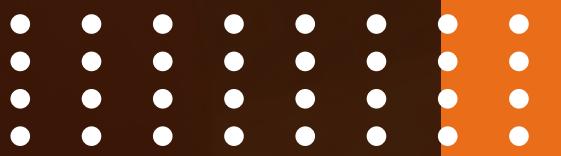


PIZZA SALES ANALYSIS

Using SQL

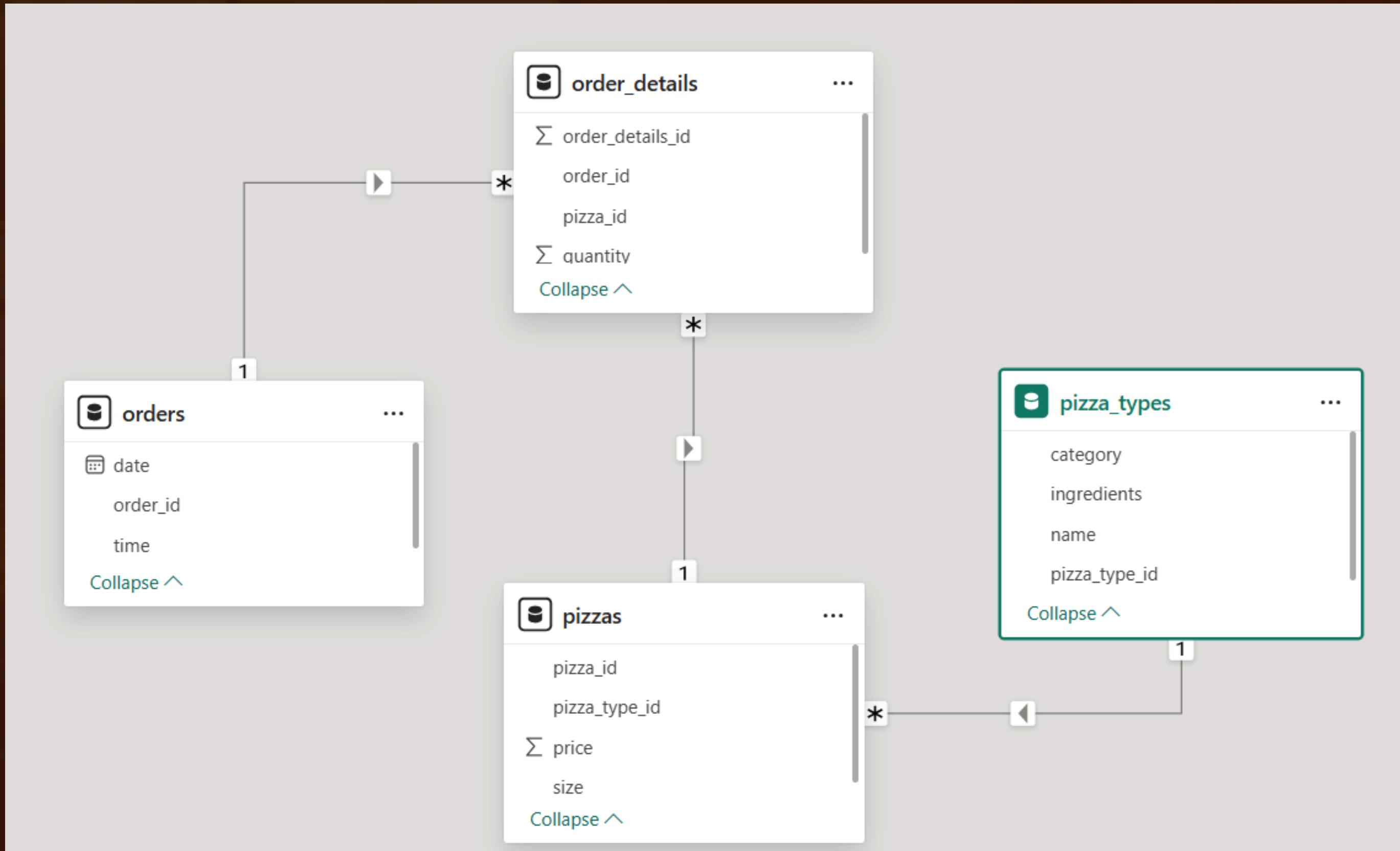


OBJECTIVE



- 1. Analyze sales patterns and revenue trends to optimize pricing and identify peak business hours for efficient staff scheduling.**
- 2. Evaluate menu performance by tracking best-selling and underperforming pizza types to guide inventory management decisions.**
- 3. Create data-driven insights for menu optimization and marketing strategies based on customer ordering preferences and product performance.**

SCHEMA



Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



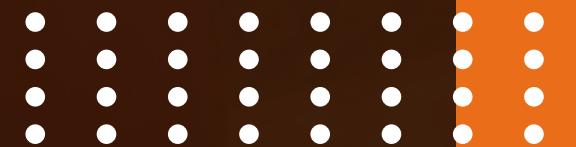
```
SELECT  
    COUNT(order_id) AS Total_Orders  
FROM  
    orders;
```

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



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

Q3. IDENTIFY THE HIGHEST-PRICED PIZZA.



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

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



```
SELECT  
    pz.size, COUNT(od.quantity) AS count_orders  
FROM  
    order_details AS od  
        JOIN  
    pizzas AS pz ON od.pizza_id = pz.pizza_id  
GROUP BY pz.size  
ORDER BY count_orders DESC  
;
```

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



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

Q6. THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



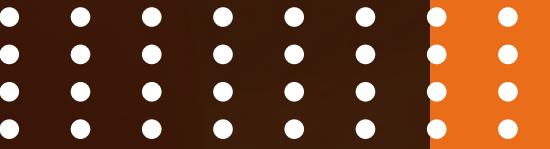
```
SELECT
    pizza_types.category, SUM(order_details.quantity) AS Quantity
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY Quantity DESC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

Q7. 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  
;
```

Q8.GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



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

Q9. 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  
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;
```

Q10. 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((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 a ) as b
where rn<=3;
```