PIZZA HUT SALES

using SQL

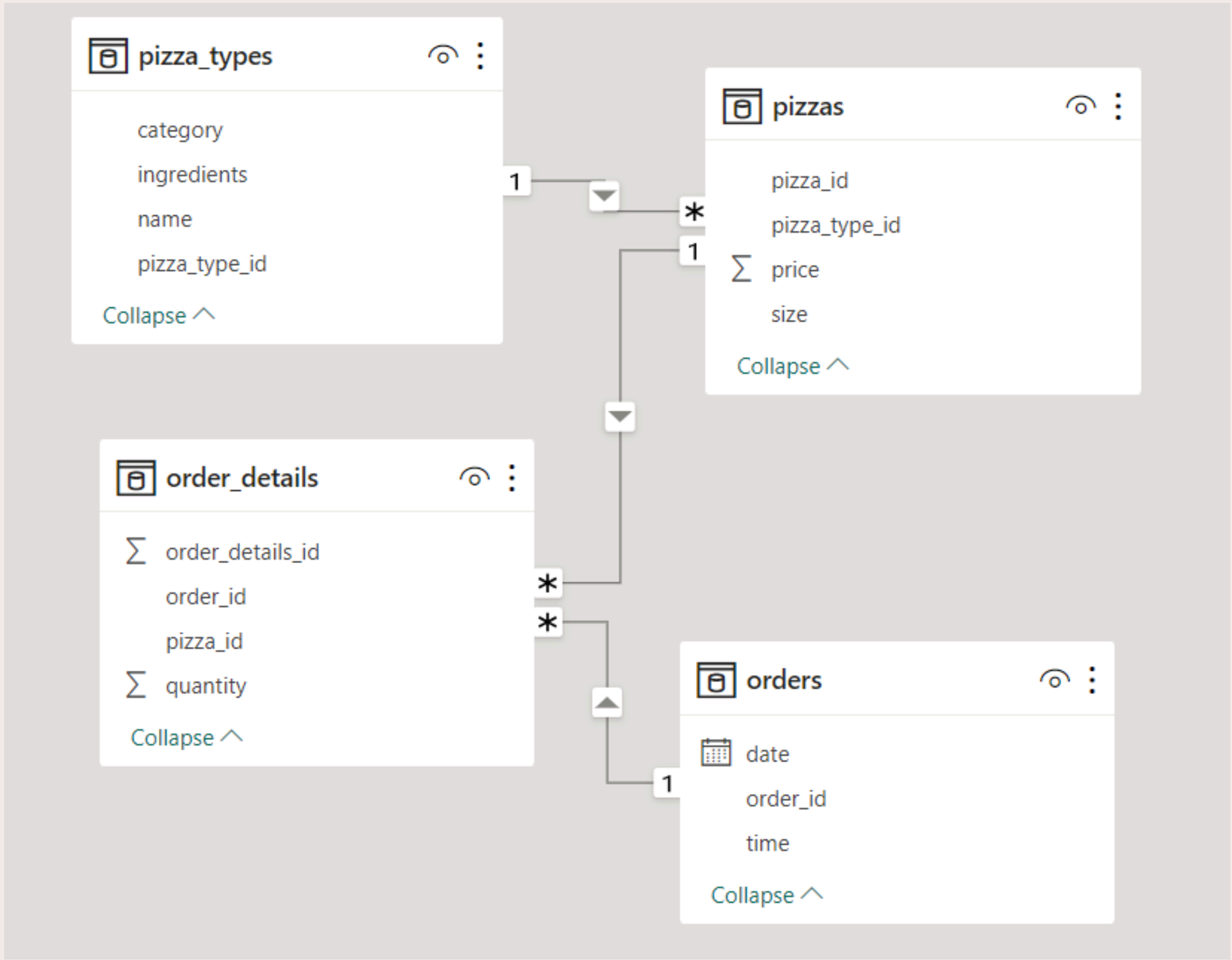
Hello!!!

My Name is Sanika Jadhav and in this project i have utilized SQL queries to solve some questions related to pizza sales.

Introduction

Pizza sales data, commonly stored in a relational database, typically includes information about orders, order dates and times, categories, pizza types, names, and prices. Using SQL (Structured Query Language), businesses can efficiently manage, query, and analyze this data to gain insights and improve their operations.

Database Schema



APPLICATIONS

Using SQL in Daily Life

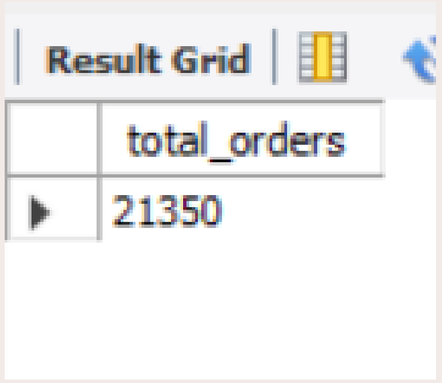
SQL is a powerful tool for managing pizza sales data in everyday business operations. Here are some practical applications:

- Sales Analysis: Businesses can analyze sales patterns to identify peak ordering times, popular pizza types, and sales trends.
- Customer Insights: SQL queries can help businesses understand customer preferences and behavior, allowing for targeted marketing campaigns.

By leveraging SQL, pizza businesses can make data-driven decisions to enhance customer satisfaction, streamline operations, and boost sales.

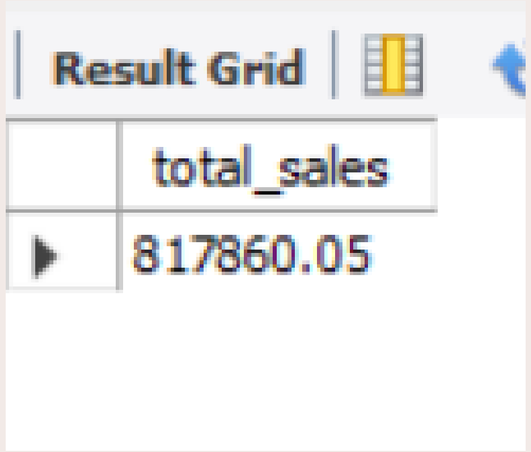
Retrieve the total number of orders placed.

```sql
select count(order_id) as total_orders from pizzahut.orders;
```



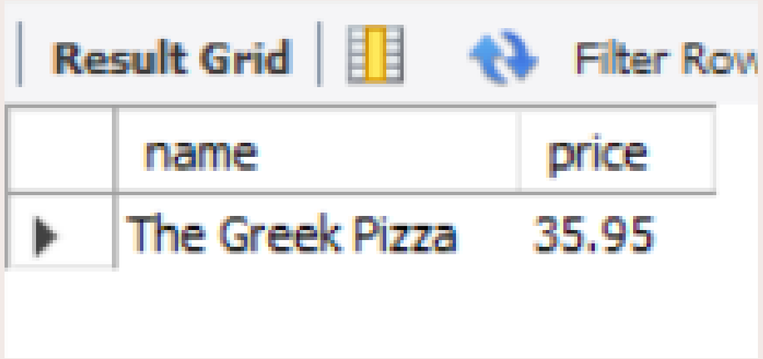
Calculate the total revenue generated from pizza sales.

```
SELECT
    round(SUM(pizzahut.order_details.quantity * pizzahut.pizzas.price), 2) AS total_sales
FROM
    pizzahut.order_details
        JOIN
    pizzahut.pizzas ON pizzas.pizza_id = order_details.pizza_id
```



Identify the highest-priced pizza.

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



	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_detailsid
FROM
    pizzahut.order_details
        JOIN
    pizzahut.pizzas ON pizzahut.order_details.pizza_id = pizzahut.pizzas.pizza_id
GROUP BY size
ORDER BY order_detailsid DESC;
```

Result Grid | Filte

size	order_detailsid
L	18526
M	15385
S	14137
XL	544
XXL	28

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

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

Result Grid | Filter Rows:

name	quantity
The Barbecue Chicken Pizza	2432
The Classic Deluxe Pizza	2453
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
    pizzahut.pizza_types.category,
    SUM(pizzahut.order_details.quantity) AS quantity
FROM
    pizzahut.pizza_types
        JOIN
    pizzahut.pizzas ON pizzahut.pizza_types.pizza_type_id = pizzahut.pizzas.pizza_type_id
        JOIN
    pizzahut.order_details ON pizzahut.order_details.pizza_id = pizzahut.pizzas.pizza_id
GROUP BY pizzahut.pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filte

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(pizzahut.orders.order_time) AS hourorder,
    COUNT(order_id) AS idorder
FROM
    pizzahut.orders
GROUP BY hourorder;
```

Result Grid | Filt

hourorder	idorder
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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

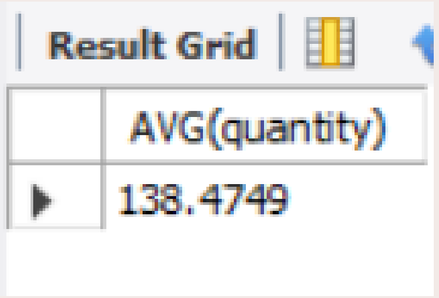
```
SELECT
    pizzahut.pizza_types.category,
    COUNT(pizzahut.pizza_types.name) AS name
FROM
    pizzahut.pizza_types
GROUP BY pizzahut.pizza_types.category;
```

Result Grid

category	name
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```
SELECT
    AVG(quantity)
FROM
    (SELECT
        pizzahut.orders.order_date AS orderdate,
            SUM(pizzahut.order_details.quantity) AS quantity
    FROM
        pizzahut.orders
    JOIN pizzahut.order_details ON pizzahut.orders.order_id = pizzahut.order_details.order_id
    GROUP BY pizzahut.orders.order_date) AS avgquantity;
```



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

```sql
SELECT
    pizzahut.pizza_types.name, SUM(quantity * price) AS revenue
FROM
    pizzahut.pizza_types
        JOIN
    pizzahut.pizzas ON pizzahut.pizza_types.pizza_type_id = pizzahut.pizzas.pizza_type_id
        JOIN
    order_details ON pizzahut.order_details.pizza_id = pizzahut.pizzas.pizza_id
GROUP BY pizzahut.pizza_types.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

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

```sql
SELECT
    pizzahut.pizza_types.category,
    ROUND(SUM(quantity * price) / (SELECT
                    ROUND(SUM(pizzahut.order_details.quantity * pizzahut.pizzas.price),
                                2) AS total_sales
                FROM
                    pizzahut.order_details
                        JOIN
                    pizzahut.pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS totalsales
FROM
    pizzahut.pizza_types
        JOIN
    pizzahut.pizzas ON pizzahut.pizza_types.pizza_type_id = pizzahut.pizzas.pizza_type_id
        JOIN
    order_details ON pizzahut.order_details.pizza_id = pizzahut.pizzas.pizza_id
GROUP BY pizzahut.pizza_types.category;
```

Result Grid

	category	totalsales
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

Analyze the cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over (order by order_date) as cum_revenue
from
(SELECT
    pizzahut.orders.order_date, round(SUM(quantity * price), 2) AS revenue
FROM
    pizzahut.orders
        JOIN
    pizzahut.order_details ON pizzahut.orders.order_id = pizzahut.order_details.order_id
        JOIN
    pizzahut.pizzas ON pizzahut.pizzas.pizza_id = pizzahut.order_details.pizza_id
GROUP BY pizzahut.orders.order_date) as totalsales;
```

Result Grid | Filter Rows:

order_date	cum_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
2015-01-08	19399.05
2015-01-09	21526.399999999998

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 rnk
 from
(SELECT
    pizzahut.pizza_types.category, pizzahut.pizza_types.name, sum(quantity * price) AS revenue
FROM
    pizzahut.pizza_types
        JOIN
    pizzahut.pizzas ON pizzahut.pizza_types.pizza_type_id = pizzahut.pizzas.pizza_type_id
        JOIN
    order_details ON pizzahut.order_details.pizza_id = pizzahut.pizzas.pizza_id
    group by pizzahut.pizza_types.category, pizzahut.pizza_types.name) as category) as b
    where rnk <= 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5


THANK
YOU