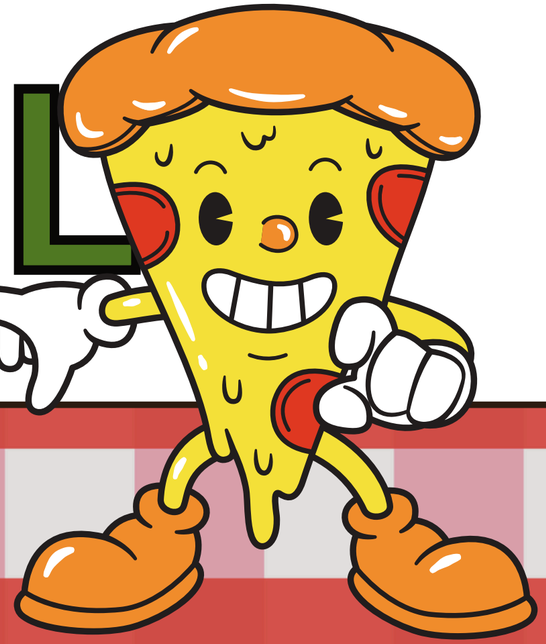
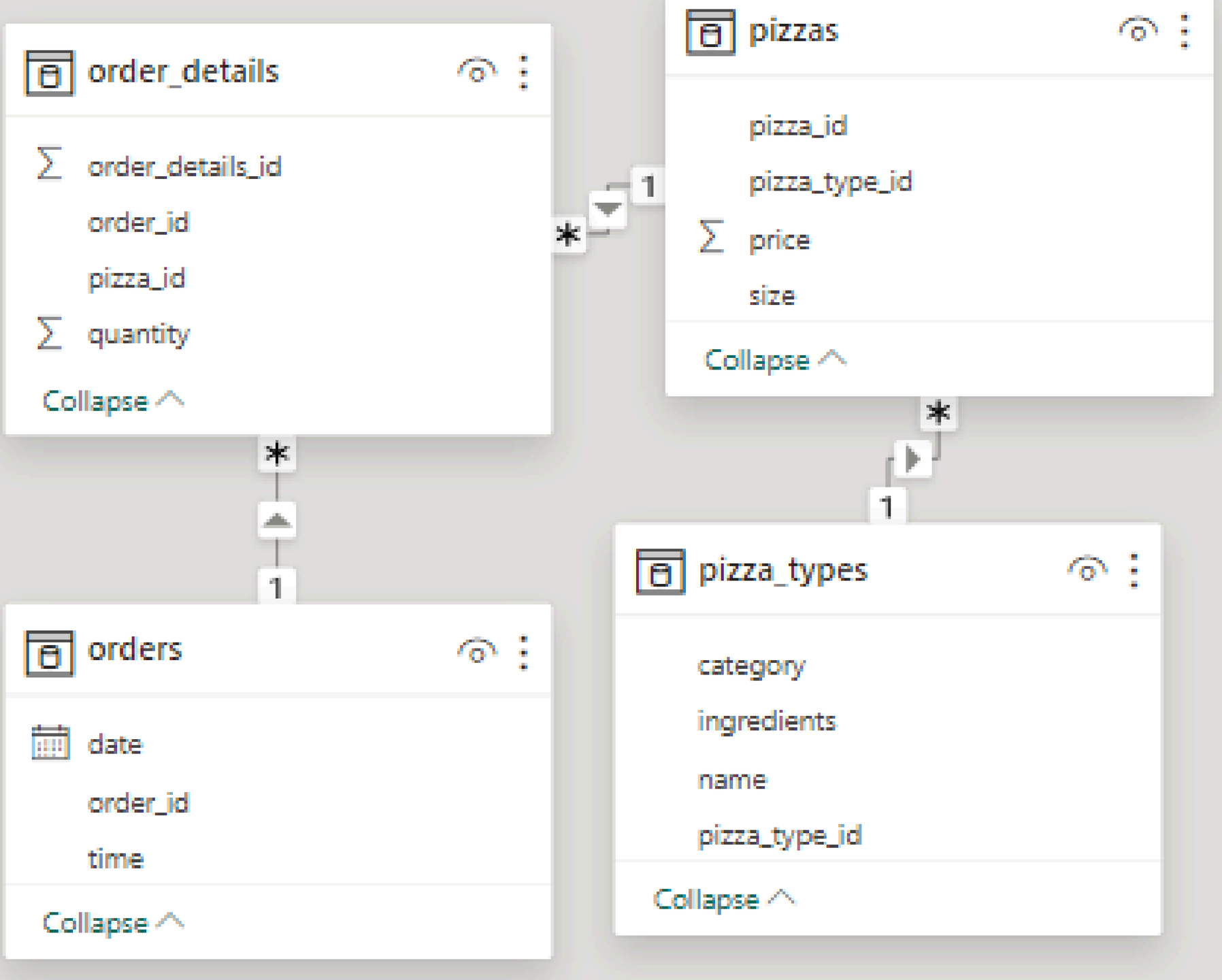
PIZZA SALE PROJECT -SQL



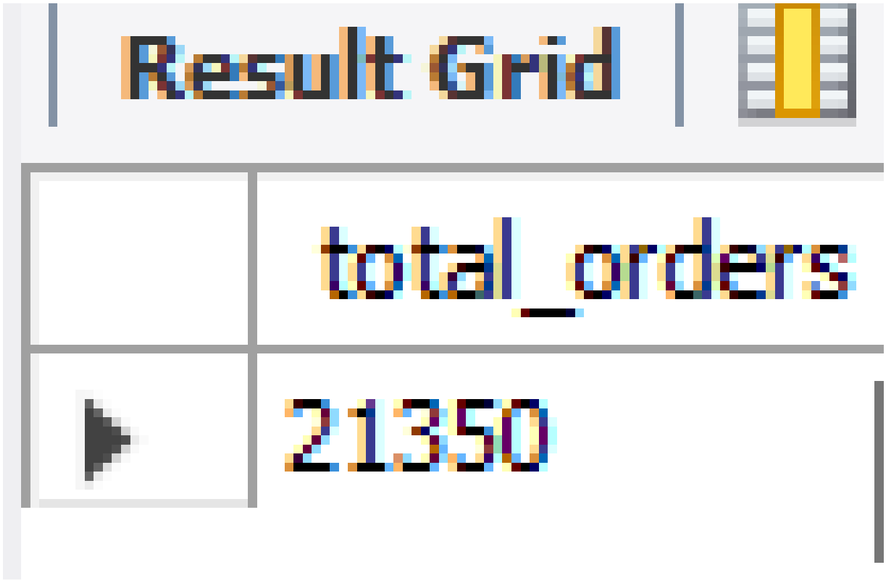
Hello Everyone

My Name is Priyanshu Patni and in this projects i have utilized sql query to solve questions that was related to pizza's sales .


order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
orders
date
order_id
time
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
1
*
*
1
*
1

-- 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.

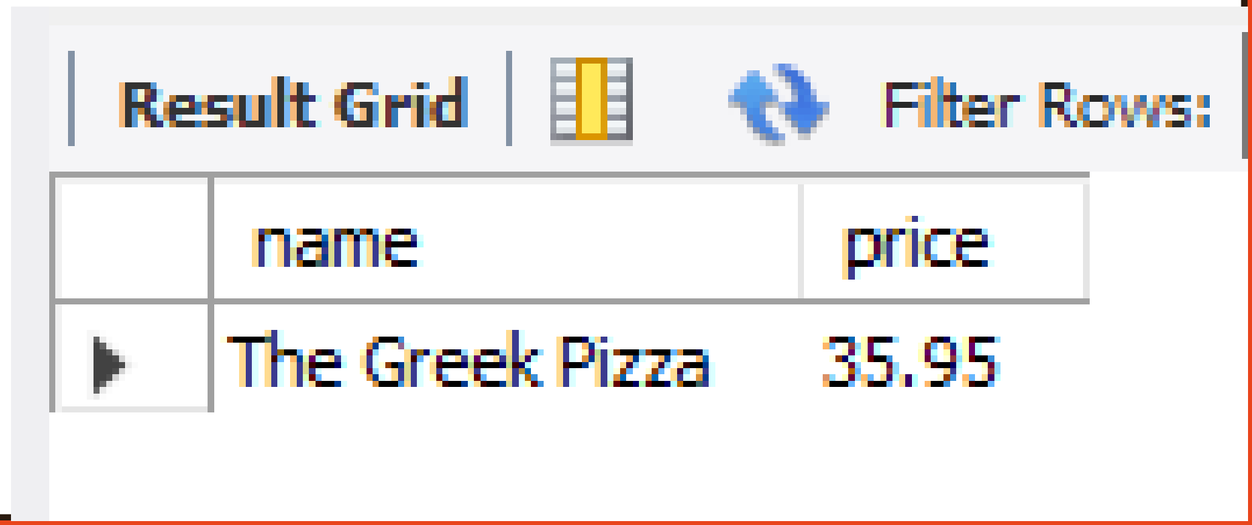
```sql
SELECT
    ROUND(SUM(p.price * o.quantity), 2) AS total_sales
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id;
```

Result Grid

	total_sales
▶	817860.05

-- Identify the highest-priced pizza.

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



	name	price
▶	The Greek Pizza	35.95

-- Identify the most common pizza size ordered.

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

Result Grid | Filte

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

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

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

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

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

```sql
SELECT
    pt.category, SUM(o.quantity) AS quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS o ON o.pizza_id = p.pizza_id
GROUP BY pt.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(order_time) AS hour, COUNT(order_id)
    AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filte

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

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

```sql
SELECT
    pt.category, COUNT(pt.pizza_type_id) AS count
FROM
    pizza_types AS pt
GROUP BY pt.category
ORDER BY count DESC;
```

Result Grid | Filter

category	count
Supreme	9
Veggie	9
Classic	8
Chicken	6

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

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

	avg_pizza_ordered_per_day
▶	138

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

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

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

Result Grid | Filter Rows:

category	revenue_in_percent
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

-- Analyze the cumulative revenue generated over time.

```sql
select order_date,round(sum(revenue)
over(order by order_date),2) as cum_revenue
from
(select o.order_date,sum(p.price*od.quantity)as revenue
from orders as o join order_details as od
on o.order_id=od.order_id join pizzas as p
on p.pizza_id=od.pizza_id
group by o.order_date)as sales
```

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

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

```sql
select category,name,revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pt.category,pt.name ,sum(p.price*od.quantity)as revenue
from pizzas as p join order_details as od
on od.pizza_id=p.pizza_id join pizza_types as pt
on pt.pizza_type_id=p.pizza_type_id
group by pt.category,pt.name)as a) as b
where rn<=3;
```

Result Grid | Filter Rows:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Cla	
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70(

The California Chicken Pizza

THANKYOU