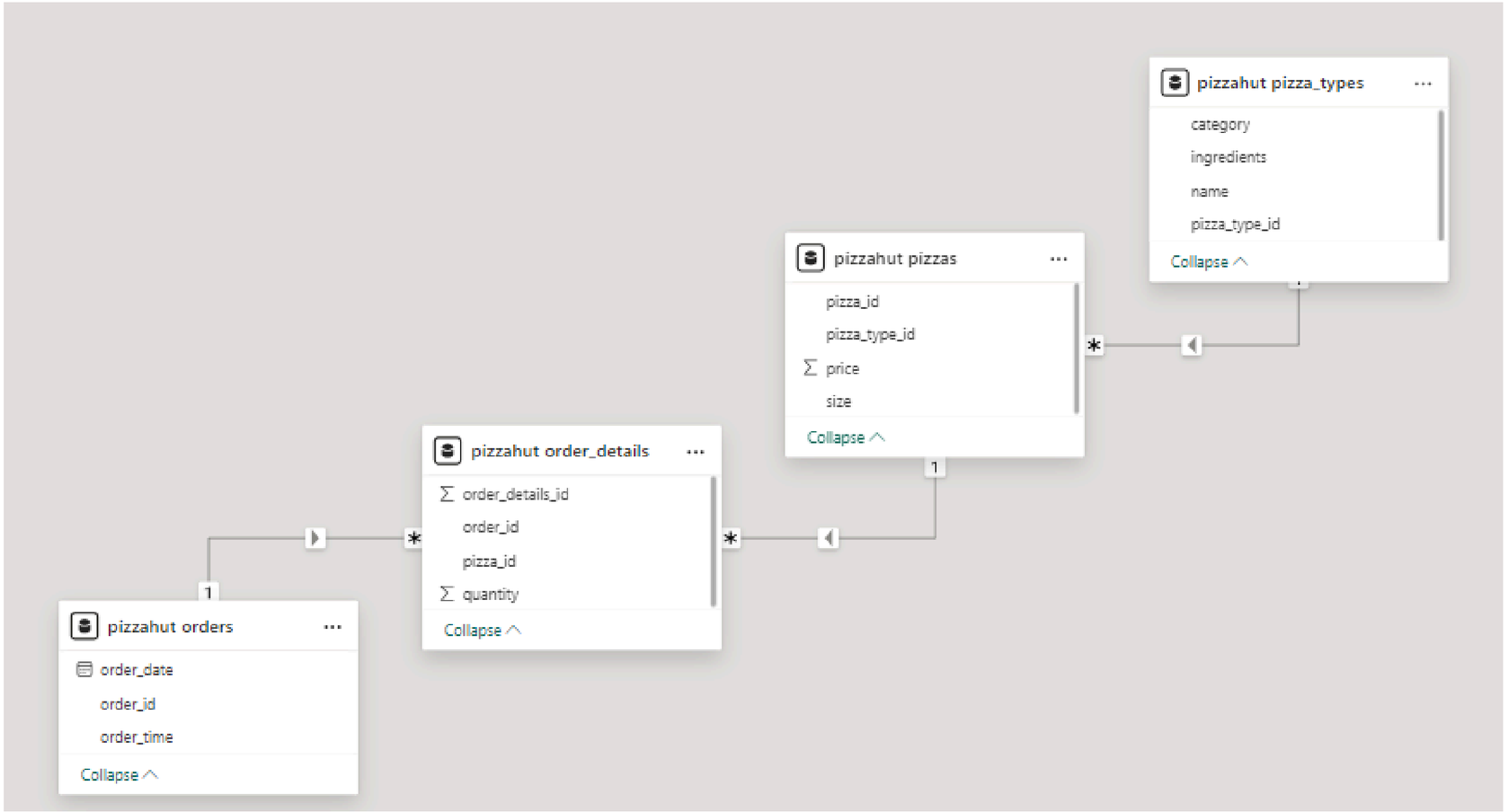
PIZZA SALES ANALYSIS

-YAGNIK GOSAI

Objective:

To Answer The Questions Related To Pizza Sales

Schema:


pizzahut pizza_types
category
ingredients
name
pizza_type_id
Collapse
pizzahut pizzas
pizza_id
pizza_type_id
price
size
Collapse
pizzahut order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizzahut orders
order_date
order_id
order_time
Collapse
1
*
1
*
1
*

Procudure:

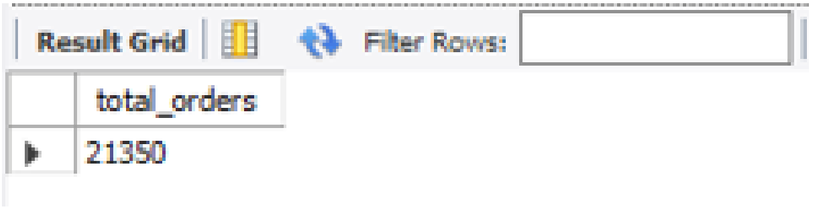
1.)Created The Database

2.)Imported The csv Files Into Database as Tables

3.)Performed Data Querying

1.)Retrieve the total number of orders placed.

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



2._Calculate the total revenue generated from pizza sales.

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
    pizzas p
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
```

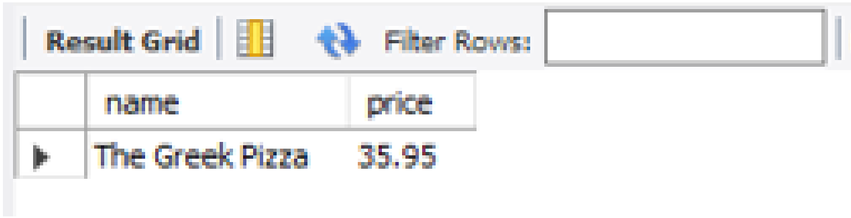
Result Grid | Filter Rows:

	total_revenue
▶	817860.05

3.)Identify the highest-priced pizza.

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

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



	name	price
▶	The Greek Pizza	35.95

4.)Identify the most common pizza size ordered.

```sql
-- Identify the most common pizza size ordered.

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

Result Grid | Filter Rows:

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

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

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

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

Result Grid | Filter Rows:

name	Quantities_Ordered
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

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

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

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7.)Determine the distribution of orders by hour of the day

```sql
-- Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows:

hour	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

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

SELECT
    category, COUNT(pizza_type_id) as quantity
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

category	quantity
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

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

with cte as(
select o.order_date , sum(od.quantity) as quantity from order_details od
join orders o on o.order_id=od.order_id
group by o.order_date)

select round(avg(quantity),0) as averageOrderPerDay from cte;
```

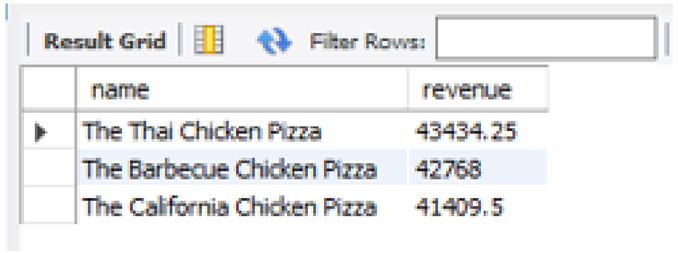
Result Grid | Filter Rows:

averageOrderPerDay
138

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

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

SELECT
    pt.name, SUM(p.price * od.quantity) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details 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

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

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pt.category,
    CONCAT(ROUND((SUM(p.price * od.quantity) / (SELECT
                            SUM(od.quantity * p.price)
                        FROM
                            pizzas p
                                JOIN
                            order_details od ON od.pizza_id = p.pizza_id)) * 100,
                    2),
            '%') AS revenue_percentage
FROM
    pizza_types pt
        JOIN
    pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue_percentage DESC;
```

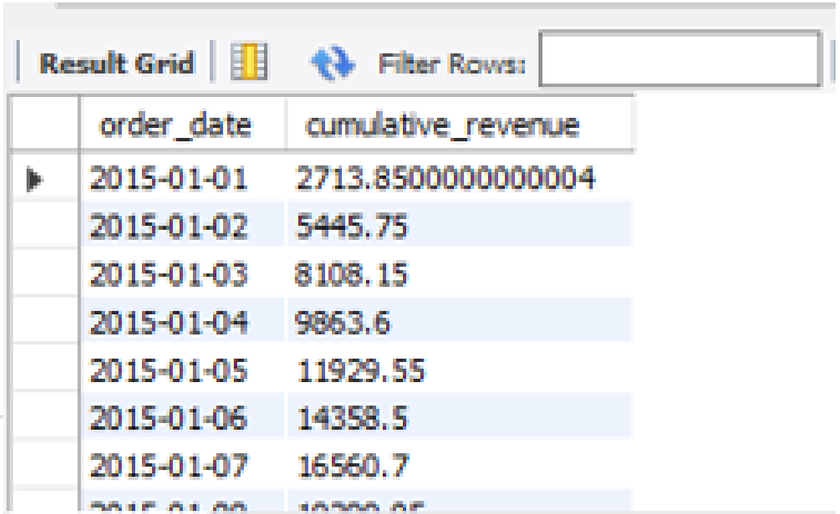
Result Grid | Filter Rows:

category	revenue_percentage
Classic	26.91%
Supreme	25.46%
Chicken	23.96%
Veggie	23.68%

12.)Analyze the cumulative revenue generated over time.

```sql
-- Analyze the cumulative revenue generated over time

SELECT order_date ,
       SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM
    (SELECT
        o.order_date, SUM(p.price * od.quantity) AS revenue
    FROM
        orders o
            JOIN
        order_details od ON od.order_id = o.order_id
            JOIN
        pizzas p ON p.pizza_id = od.pizza_id
    GROUP BY o.order_date) as table1;
```



Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-01-01	2713.8500000000004
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

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

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

SELECT category , name , revenue from
(SELECT category , name , revenue ,
       dense_rank() OVER(partition by category order by  revenue desc) AS rank_no
FROM
(SELECT
    pt.category, pt.name, SUM(p.price * od.quantity) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category , pt.name) AS a) AS b
WHERE rank_no <=3
```

	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 Classic Deluxe Pizza	38180.5
	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	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

THANK YOU !