❖ SQL Project

A Project on Pizza Hut Sales Analysis

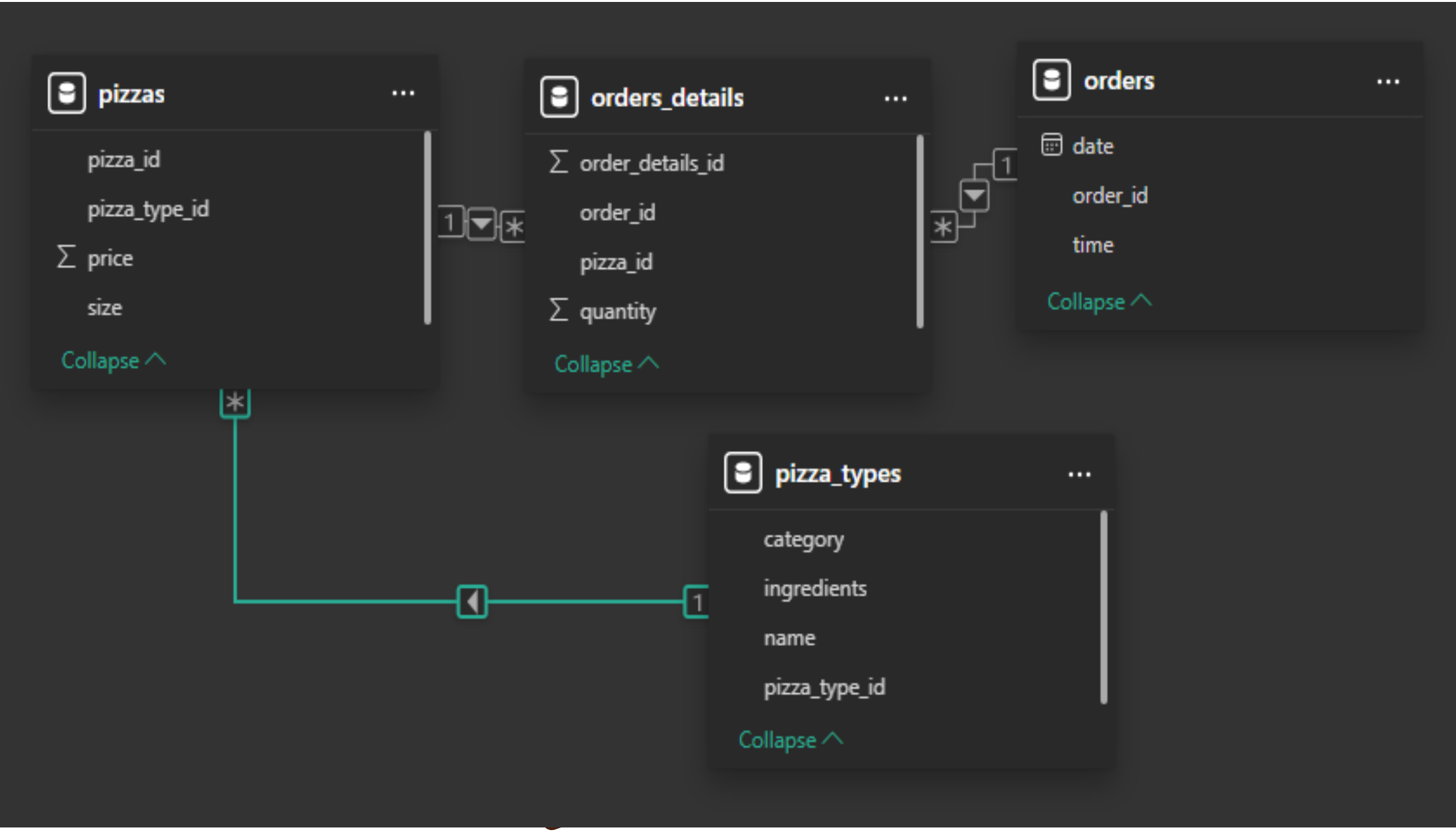
https://www.linkedin.com/in/ganesh-jodtale-5642422ba/

Ganesh Jodtale

PROJECT SUMMARY

This SQL project analyzes pizza sales data to identify the top-performing pizza types across different categories based on revenue. Multiple tables (orders, pizzas, pizza_types, and orders_details) were joined using foreign key relationships. Revenue was calculated by multiplying quantity and price, then aggregated by pizza type and category. Using window functions like RANK(), the top 3 most ordered pizzas per category were extracted. The insights can support business decisions such as menu optimization and targeted promotional strategies.

DATA SCHEMA



BASIC QUESTIONS

Retrieve the total number of orders placed.

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

Result Grid

	total_orders
▶	21350

Ganesh Jodtale

Calculate the total revenue generated from pizza sales.

```sql
-- i.e. Sum of qty x price
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;
```

Result Grid

	Total Revenue
▶	817860.05

Ganesh Jodtale

Identify the highest-priced pizza.

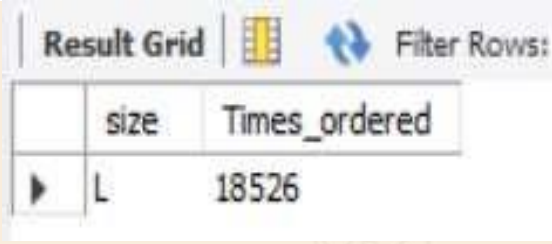
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        LEFT JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

Ganesh Jodtale

Identify the most common pizza size ordered.

```sql
-- i.e. we need order_detail_id and size
SELECT
    size, COUNT(order_details_id) AS Times_ordered
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY (size)
ORDER BY Times_ordered DESC
LIMIT 1;
```



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

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

Result Grid | Filter Rows:

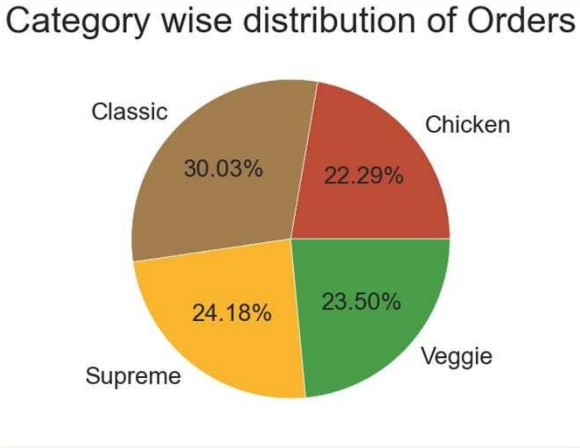
name	total_qty
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Ganesh Jodtale

ADVANCED QUESTIONS

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

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



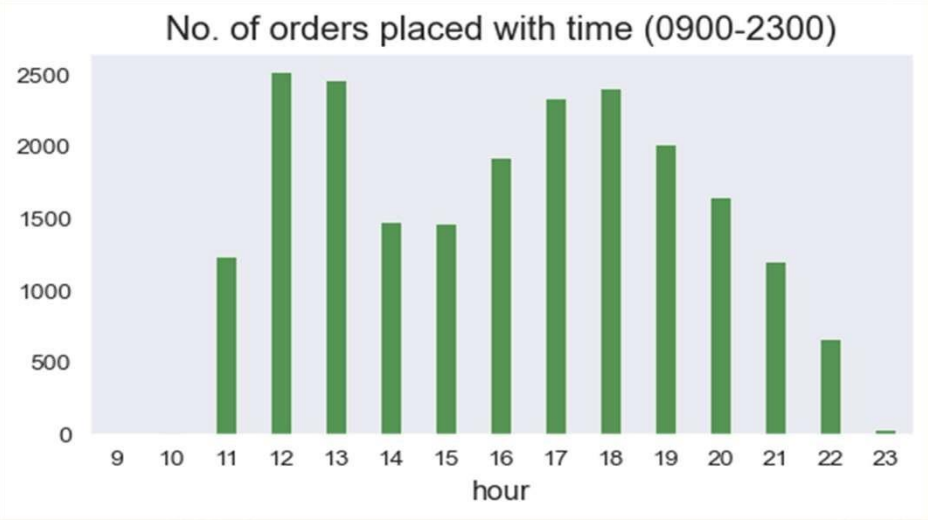
Result Grid | Filter Rows:

category	total_qty
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Ganesh Jodtale

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 BY hour ASC;
```



Result Grid | Filter

hour	order_count
9	1
10	8
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

Ganesh Jodtale

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

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

Result Grid | Filter Rows:

	ROUND(AVG(quantity), 0)
▶	138

Ganesh Jodtale

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

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

Result Grid | Filter Rows:

name	tot_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Ganesh Jodtale

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

```sql
-- Revenue% by Category

CREATE VIEW v3 AS
    (SELECT
        category, SUM(quantity * price) AS tot_revenue
    FROM
        order_details
            JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id
            JOIN
        pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    GROUP BY category
    ORDER BY tot_revenue DESC);
```

```sql
-- select * from v3

SELECT
    category,
    ROUND(tot_revenue * 100 / (SELECT
                    SUM(tot_revenue)
                FROM
                    v3),
            2) AS 'Revenue%'
FROM
    v3
GROUP BY category
ORDER BY 'Revenue%' DESC;
```

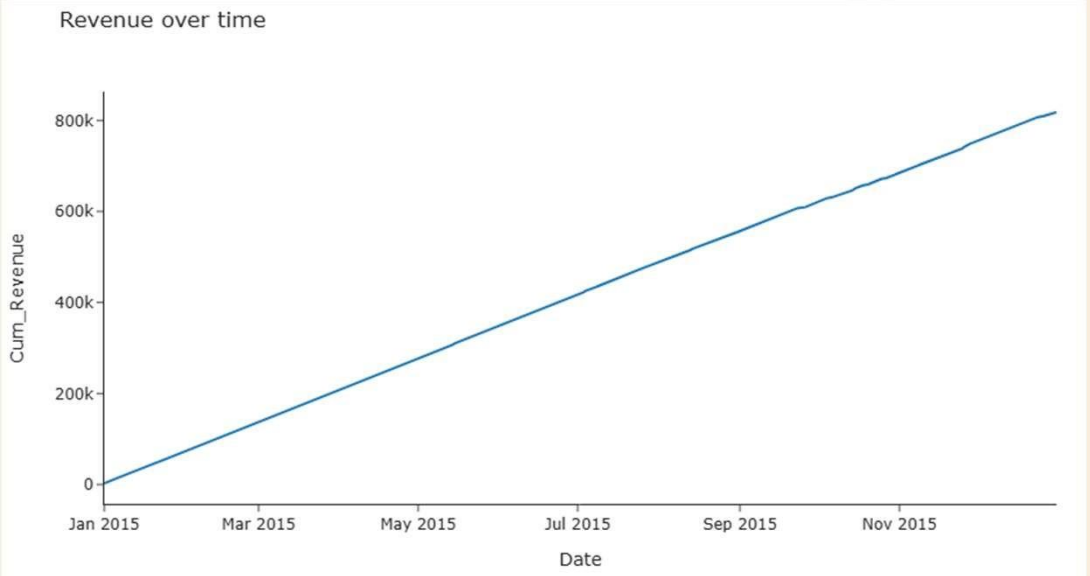
Result Grid | Filter Rows:

category	Revenue%
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Ganesh Jodtale

Analyze the cumulative revenue generated over time.

```sql
-- CUMULATIVE SUM
select order_date, round(sum(revenue) over(order by order_date),2) as cum_revenue
from (select order_date, round(sum(quantity*price),2) as revenue
from order_details join orders
on order_details.order_id =orders.order_id
join pizzas on pizzas.pizza_id = order_details.pizza_id
group by order_date order by 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
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6

Ganesh Jodtale

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

```sql
-- RANK
select category, name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as class from
(select category, name, sum(quantity* price) as revenue from
order_details join pizzas on order_details.pizza_id =pizzas.pizza_id
join pizza_types on pizza_types.pizza_type_id =pizzas.pizza_type_id
group by category,  name) as a) as b where class <=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 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	32265.70
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Ganesh Jodtale

THANKS

ganeshjodtale19@gmail.com

https://github.com/Ganesh-Jodtale

https://www.linkedin.com/in/ganesh-jodtale-5642422ba/