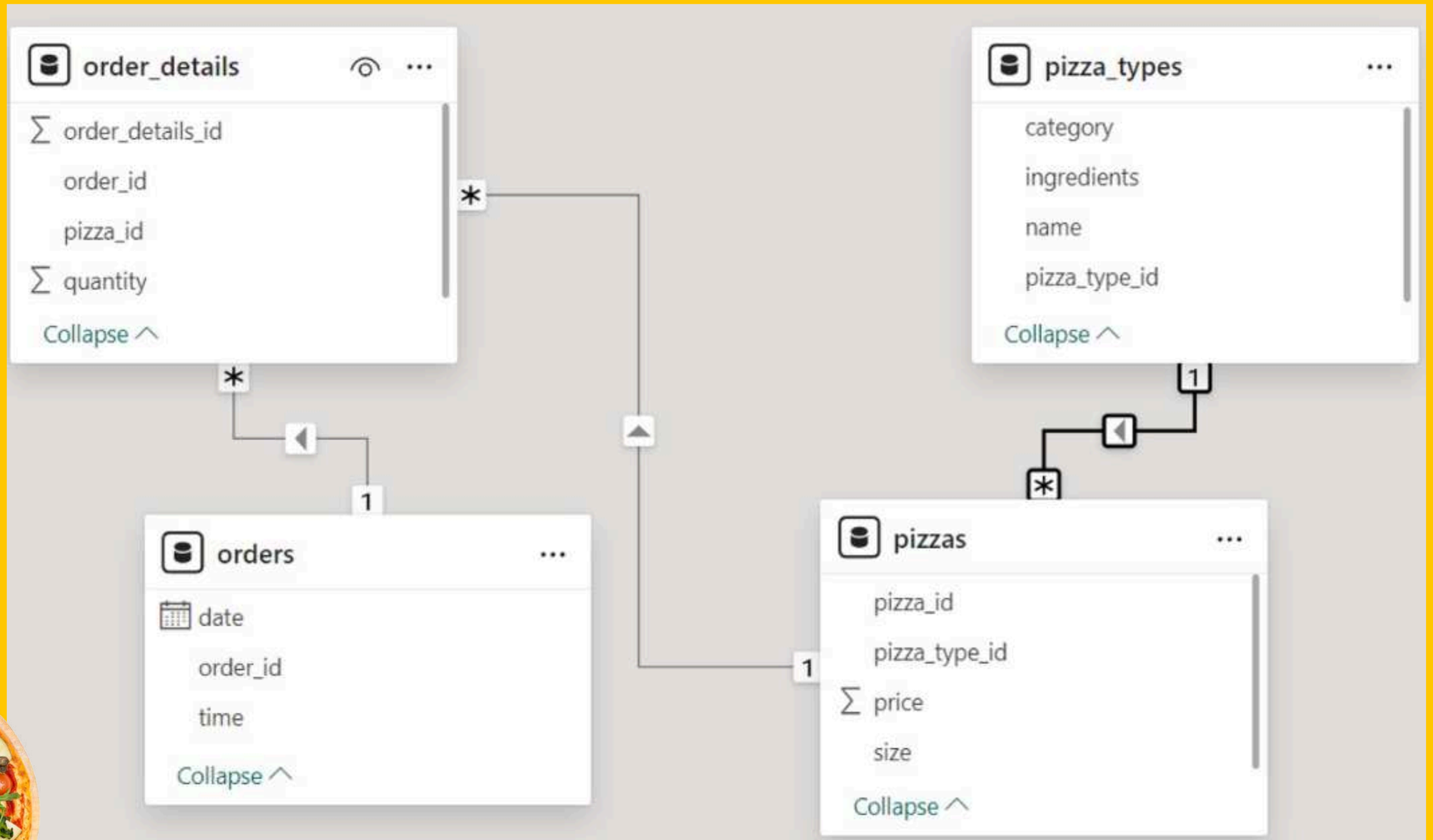
SQL PROJECT-
ON PIZZA SALES
ANALYSIS

In this project I have utilized SQL to solve business queries related to Pizza Sales.

DATABASE SCHEMA OVERVIEW



BASIS QUESTIONS

Q1. Retrive the total number of orders placed

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

Result Grid

	Total_Orders
▶	21350

Q2. Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS Total_Revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid

	Total_Revenue
▶	817860.05

Q3. Identify the highest-priced pizza.

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

	Pizza_Name	price
▶	The Greek Pizza	35.95

Q4. Identify the highest-priced pizza.

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS total_orders
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_orders DESC
```

	size	total_orders
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

Q5. List the top 5 most ordered pizza types along with their quantities.

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

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

INTERMEDIATE QUESTIONS

Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

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

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS Hours, COUNT(order_id) AS Orders
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY orders DESC;
```

Hours	Orders
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472

Q8. Group the orders by date and calculate the average number of pizzas ordered per day.

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

	average_perday
▶	138

Q9. Determine the top 3 most ordered pizza types based on revenue.

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

pizza_type_id	name	revenue
thai_ckn	The Thai Chicken Pizza	43434.25
bbq_ckn	The Barbecue Chicken Pizza	42768
cali_ckn	The California Chicken Pizza	41409.5

ADVANCED QUESTIONS

Q10. Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(orders_details.quantity * pizzas.price),
                            0) AS total_sales
            FROM
                orders_details
                    JOIN
                pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
        2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q11. Analyze the cumulative revenue generated over time.

```sql
select order_date,
round(sum(revenue) over(order by order_date),0) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity*pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on  orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9864

Q12.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 ranking
from
(select pizza_types.category, pizza_types.name,
sum(orders_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by  pizza_types.category, pizza_types.name) as a) as b
where ranking <=3;
```

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

THANK YOU