PIZZA SALES ANALYSIS USING MYSQL

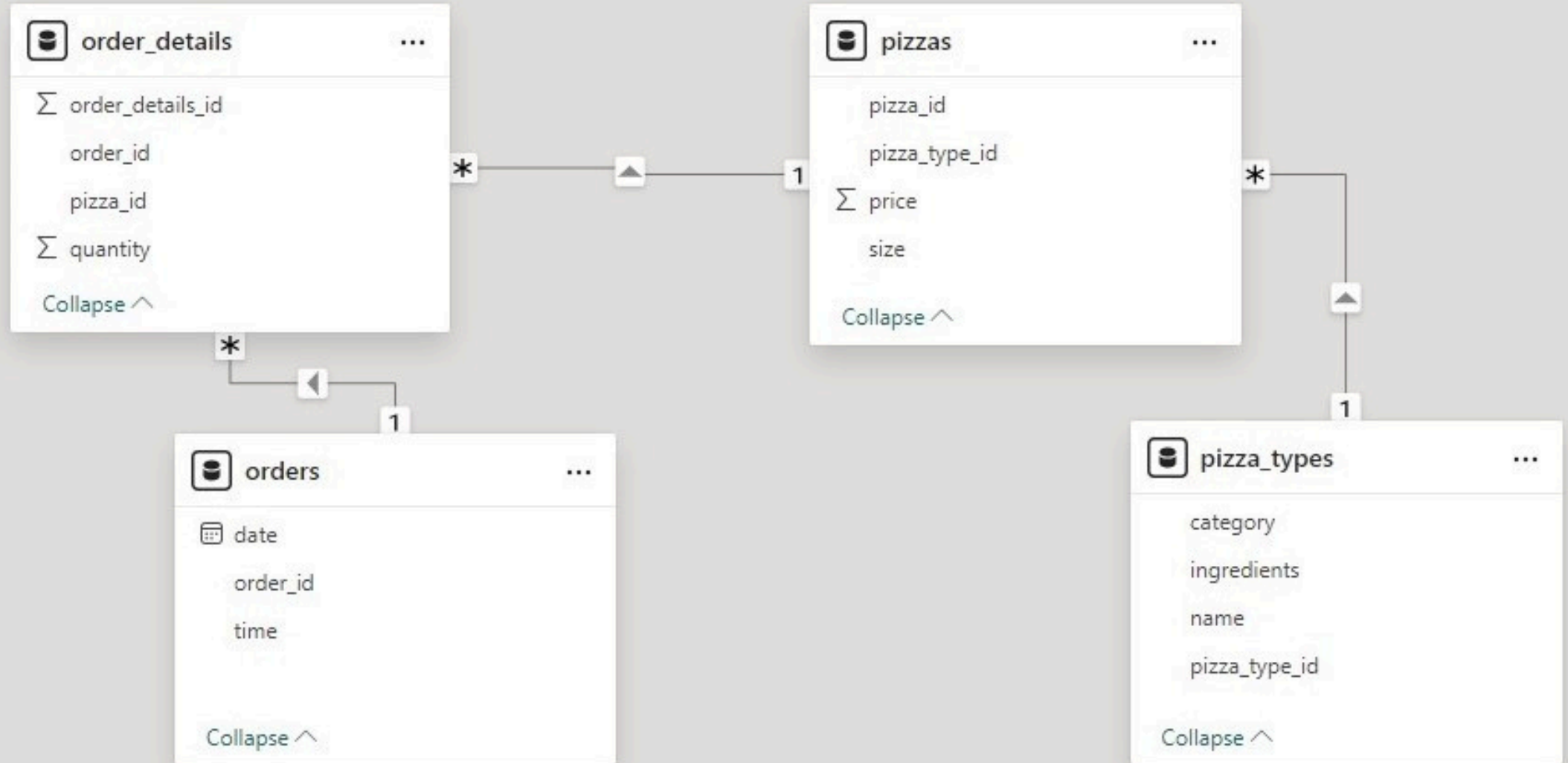# HELLO, EVERYONE!

MY NAME IS SHUBHAM KUMAR, AND I'M EXCITED TO SHARE WITH YOU MY DATA ANALYSIS PROJECT ON PIZZA SALES, USING SQL QUERIES. AND SOLVE DIFFERENT QUESTIONS RELATED TO PIZZA SALES. THIS PROJECT MARKS A SIGNIFICANT MILESTONE IN MY JOURNEY AS A DATA SCIENTIST, AND I'M THRILLED TO PRESENT MY FINDINGS AND INSIGHTS.

DATA MODELLING

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS Total_of_orders
FROM
    pizzahut.orders;
```

Result Grid

Total_of_orders
21350

Calculate the total revenue generated from pizza sales.

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

Result Grid

Total_revenue
817860.05

Identify the highest-priced pizza.

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

	name	price
▶	The Greek Pizza	35.95

Result Grid | Filter

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS Orders_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY orders_count DESC
LIMIT 1;
```

	size	Orders_count
▶	L	18526

Result Grid

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

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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
    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 orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_quantity DESC;
```

category	Total_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			Filter Rows:
	hour	order_count	
▶	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	

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

```sql
SELECT
    category, COUNT(name) AS No_of_Pizza
FROM
    pizza_types
GROUP BY category;
```

category	No_of_Pizza
Chicken	6
Classic	8
Supreme	9
Veggie	9

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
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

Avg_Pizza_Ordered_Per_day
▶ 138

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

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.quantity * pizzas.price),
        0) 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.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	Revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

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(pizzas.price * orders_details.quantity),
                                2) AS Total_revenue
            FROM
                pizzas
                    JOIN
                orders_details ON pizzas.pizza_id = orders_details.pizza_id) * 100,
        2) AS Percentage_contribution
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 Percentage_contribution DESC;
```

Result Grid | | Filter Rows:

category	Percentage_contribution
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

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

order_date	cum_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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7

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 R_n
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 R_n <=3;
```

Result Grid		
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 FOR REVIEWING MY PIZZA SALES ANALYSIS PROJECT.

KEY INSIGHTS INCLUDE:

- **TOTAL ORDERS AND REVENUE GENERATED**
- **HIGHEST-PRICED PIZZA AND MOST COMMON PIZZA SIZE**
- **TOP 5 MOST ORDERED PIZZAS**
- **QUANTITY OF EACH PIZZA CATEGORY ORDERED**
- **ORDERS DISTRIBUTION BY HOUR**
- **AVERAGE PIZZAS ORDERED PER DAY**
- **TOP 3 PIZZAS BASED ON REVENUE**
- **REVENUE CONTRIBUTIONS OF EACH PIZZA TYPE**
- **CUMULATIVE REVENUE OVER TIME**
- **TOP 3 PIZZA CATEGORY WISE**

THIS PROJECT HAS ENHANCED MY DATA ANALYSIS SKILLS AND PROVIDED VALUABLE INSIGHTS INTO PIZZA SALES TRENDS. I APPRECIATE YOUR FEEDBACK AND LOOK FORWARD TO APPLYING THESE SKILLS IN FUTURE PROJECTS.
THANK YOU!