Home About Contact

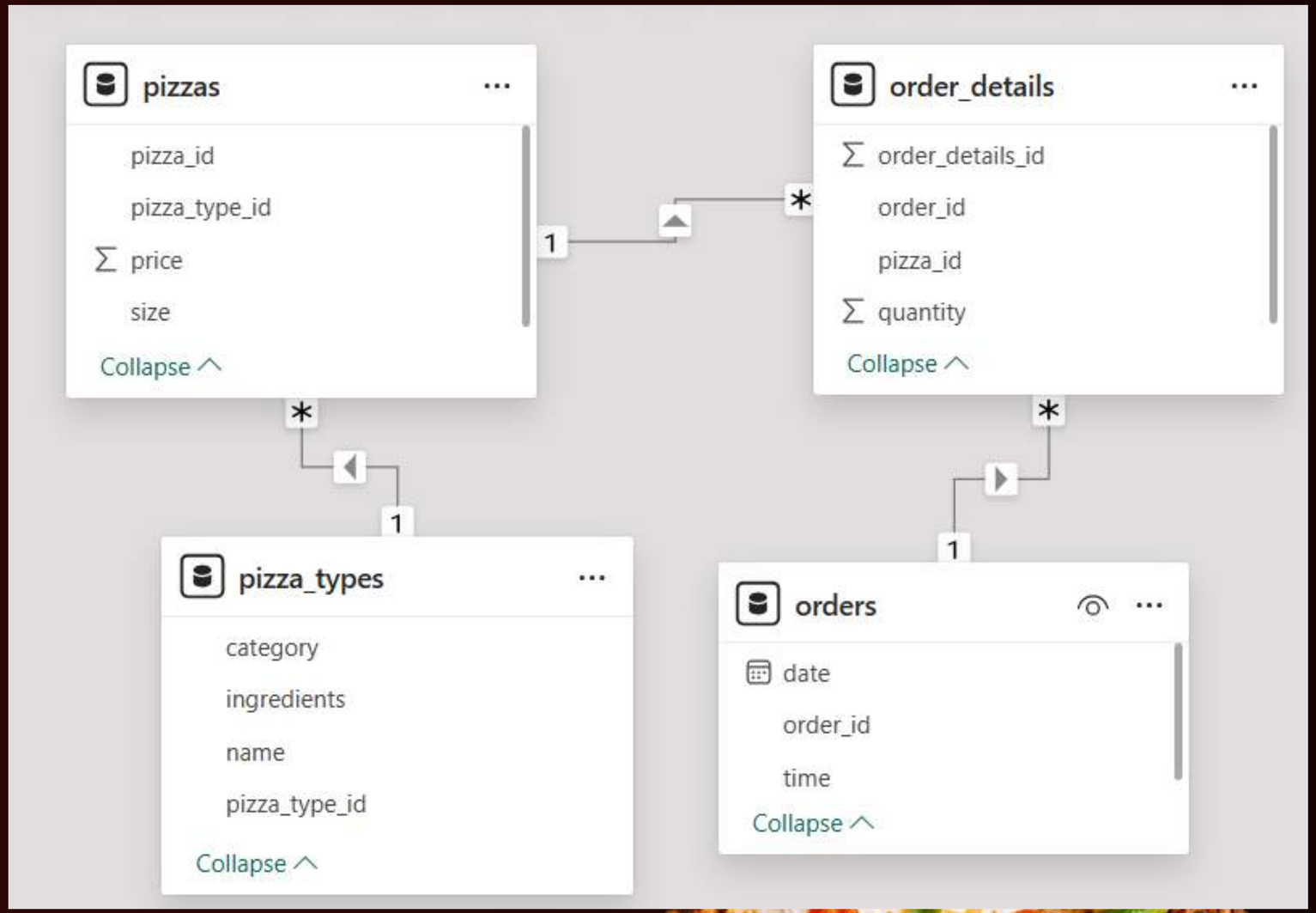
PIZZA SALES DATA ANALYSIS PROJECT

BY SAKSHI GUPTA

SHODWE
Pizza Resto

- **Project Objective:** Quantitative analysis of pizza sales performance
- **Methodology:** SQL query development across relational database
- **Scope:** 15+ analytical questions across 4 data tables
- **Tools:** MySQL 8.0

DATA ARCHITECTURE



ANALYSIS QUERIES

-- Retrieve the total number of orders placed.

```
use pizza;
select count(order_id) as total_order
from orders;
```

Result Grid

	total_orders
▶	14762

-- Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
    order_details AS od
        LEFT JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id;
```

	total_revenue
▶	20654.5

ANALYSIS QUERIES

-- Identify the most common pizza size ordered.

```
select count(od.quantity),p.size
from order_details as od
left join pizzas as p
on od.pizza_id=p.pizza_id
group by p.size
order by count(p.size) desc;
```

count(od.quantity)	size
486	L
390	M
330	S
13	XL

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

```
select sum(od.quantity) as quantity ,pt.name
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.name
order by quantity desc
limit 5 ;
```

quantity	name
72	The Pepperoni Pizza
64	The Thai Chicken Pizza
61	The Classic Deluxe Pizza
59	The Hawaiian Pizza
58	The Italian Supreme Pizza

ANALYSIS QUERIES

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

```sql
select pt.category, sum(od.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 od
on od.pizza_id=p.pizza_id
group by pt.category
order by quantity desc;
```

category	quantity
Classic	376
Supreme	302
Veggie	297
Chicken	272

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

```sql
select category ,count(pizza_type_id)
from pizza_types
group by category;
```

category	count(pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9

ANALYSIS QUERIES

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

```sql
select count(order_id), hour(order_time)
from orders
group by hour(order_time)
order by count(order_id) desc ;
```

count(order_id)	hour(order_time)
1721	12
1675	13
1661	18
1631	17
1374	19
1332	16

-- Identify the highest-priced pizza.

```sql
select pt.name,p.price
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id=p.pizza_type_id
order by price desc
limit 1;
```

name	price
The Greek Pizza	35.95

ANALYSIS QUERIES

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

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

average_pizza_ordered
139

ANALYSIS QUERIES

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

```sql
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue,
    pt.name,
    p.pizza_id
FROM
    order_details AS od
        LEFT JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types AS pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name , p.pizza_id
ORDER BY total_revenue DESC
LIMIT 3
;
```

total_revenue	name	pizza_id
851	The Five Cheese Pizza	five_cheese_l
809.25	The Thai Chicken Pizza	thai_ckn_l
726.25	The Spicy Italian Pizza	spicy_ital_l

ANALYSIS QUERIES

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

```sql
select pt.category, round(sum(od.quantity*p.price)/(SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
FROM
    order_details AS od
        LEFT JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id)*100,2) 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;
```

category	revenue
Classic	26.62
Veggie	24.31
Supreme	25.5
Chicken	23.57

ANALYSIS QUERIES

-- 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 revenue_rank
from
(select pt.category, pt.name,sum(od.quantity*p.price) as revenue
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,pt.name
order by revenue desc) as a) as b
where revenue_rank <= 3  ;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	1188
Chicken	The Barbecue Chicken Pizza	1022.75
Chicken	The California Chicken Pizza	1002.75
Classic	The Classic Deluxe Pizza	967
Classic	The Pepperoni Pizza	902.75
Classic	The Hawaiian Pizza	791.5
Supreme	The Italian Supreme Pizza	1030.25
Supreme	The Spicy Italian Pizza	999.25
Supreme	The Sicilian Pizza	873.25
Veggie	The Five Cheese Pizza	851
Veggie	The Four Cheese Pizza	810.40...
Veggie	The Mexicana Pizza	621

ANALYSIS QUERIES

-- Analyze the cumulative revenue generated over time.

```
select order_date,
sum(revenue) over (order by order_date )
from
   (SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS revenue, o.order_date
FROM
    order_details AS od
        LEFT JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
    join orders as o
    on o.order_id=od.order_id
    group by o.order_date) as sales ;
```

order_date	sum(revenue) over (order by order_date)
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	20654.5