

# SQL PROJECT ON PIZZA SALES



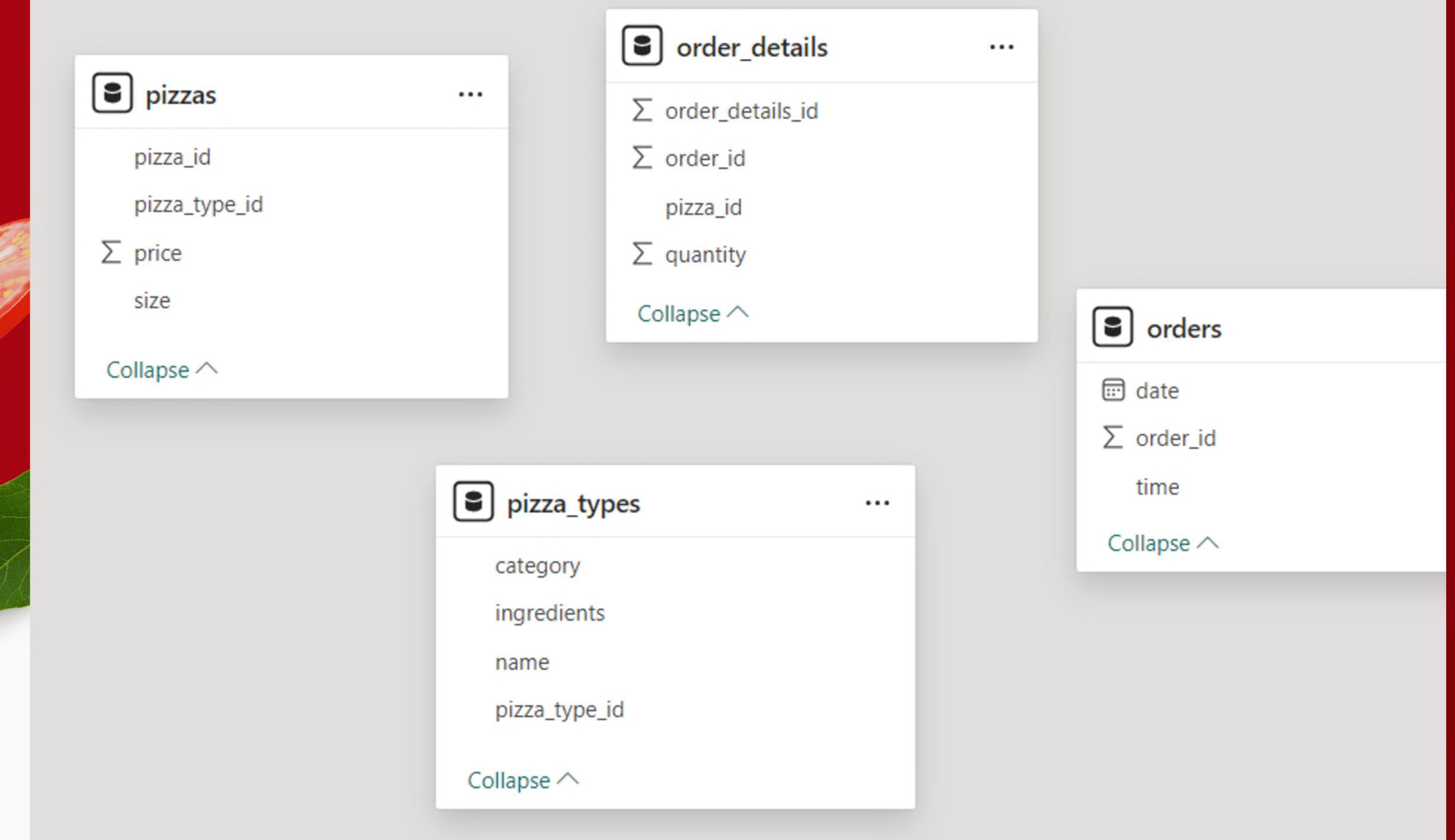
# HELLO,

My name is Kishor Shetty B. In this project, I have utilized SQL queries to analyze and solve various questions related to pizza sales.





# SUMMARY OF DATASETS



- **Pizza Types** : Includes details like **pizza\_type\_id**, **name**, **category**, and **ingredients** for each pizza type.
- **Orders** : Contains order-related information, such as **order\_id**, **order\_date**, and **order\_time**.
- **Order Details** : Tracks individual order items with **order\_details\_id**, **order\_id**, **pizza\_id**, and **quantity**.
- **Pizzas** : Links pizzas to their types with **pizza\_id**, **pizza\_type\_id**, **size**, and **price**.

# QUESTIONS

## Basic:

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

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

## Intermediate:

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

Determine the distribution of orders by hour of the day.

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

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

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

## Advanced:

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

Analyze the cumulative revenue generated over time.

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

**Retrieve the total number of orders placed.**

```
select count(order_id) as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

```
3 •  SELECT  
4   ROUND(SUM(order_details.quantity * pizzas.price),  
5         2) AS total_sales  
6  
FROM  
7   order_details  
8   JOIN  
9   pizzas ON order_details.pizza_id = pizzas.pizza_id;  
10
```

Result Grid	
	total_sales
▶	817860.05

# Identify the highest-priced pizza.

```
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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

## Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id)
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(order_details.order_details_id) DESC
LIMIT 1;
```

Result Grid		
	size	COUNT(order_details.order_details_id)
▶	L	18526

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

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

	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.

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

Result Grid | Filter Rows:

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
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
	9	1

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

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

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

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

Result Grid | Filter Rows:

	Avg_pizzas_ordered_per_day
▶	138

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

```
select pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue  
from order_details join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
join pizza_types  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
group by pizza_types.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

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

```
• select pizza_types.category, round(sum(order_details.quantity * pizzas.price) /
  (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
      2) AS total_sales
  FROM
    order_details
    JOIN
      pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,2) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_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

## Analyze the cumulative revenue generated over time.

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

	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
	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
	2015-01-13	29831.300000000003
	2015-01-14	32358.700000000004
	2015-01-15	34343.50000000001
	2015-01-16	36027.65000000001

## 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 rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<= 3 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



# THANK YOU!

