

Delicious Pizza for Everyone!

# SQL PROJECT ON PIZZA SALES



# Hello!

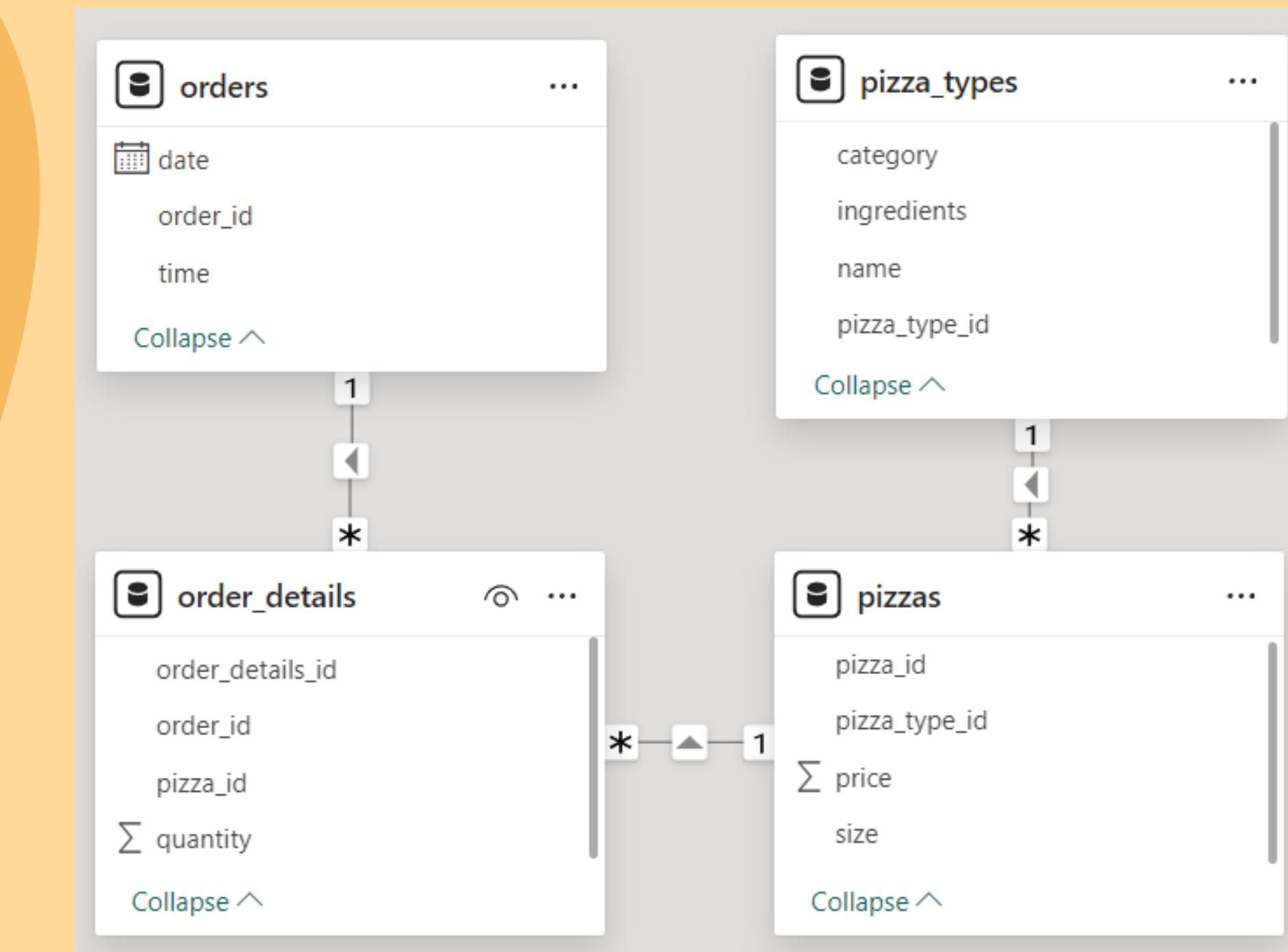
My name is Shailendra Kolare and in this project I have utilized **SQL** queries to solve questions that were related to Pizza Sales.

# Introduction

Welcome to our **SQL** project focusing on pizza sales analysis! In today's dynamic food industry, understanding customer preferences, sales trends, and operational efficiency is crucial for success. Through this project, we delve into a comprehensive analysis of pizza sales data, aiming to uncover valuable insights that can inform strategic decision-making for our pizzeria.



# Database



# 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.

**SELECT**

    ROUND(SUM(order\_details.quantity \* pizzas.price),  
          2) **AS** Total\_Sales

**FROM**

    order\_details

**JOIN**

    pizzas **ON** pizzas.pizza\_id = order\_details.pizza\_id;

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 pizzas.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) AS Order_Count  
FROM  
    pizzas  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY Order_Count DESC;
```

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

# 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 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.category

**ORDER BY** Quantity **DESC;**

	category	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);
```

	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



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



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

Result Grid | Filter Row

	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_Pizza_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	
	Avg_Pizza_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  
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_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 pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS Revenue_Percentage
FROM
    pizza_types
    JOIN
        pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY Revenue_Percentage DESC;
```

	category	Revenue_Percentage
▶	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 cum_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) 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



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

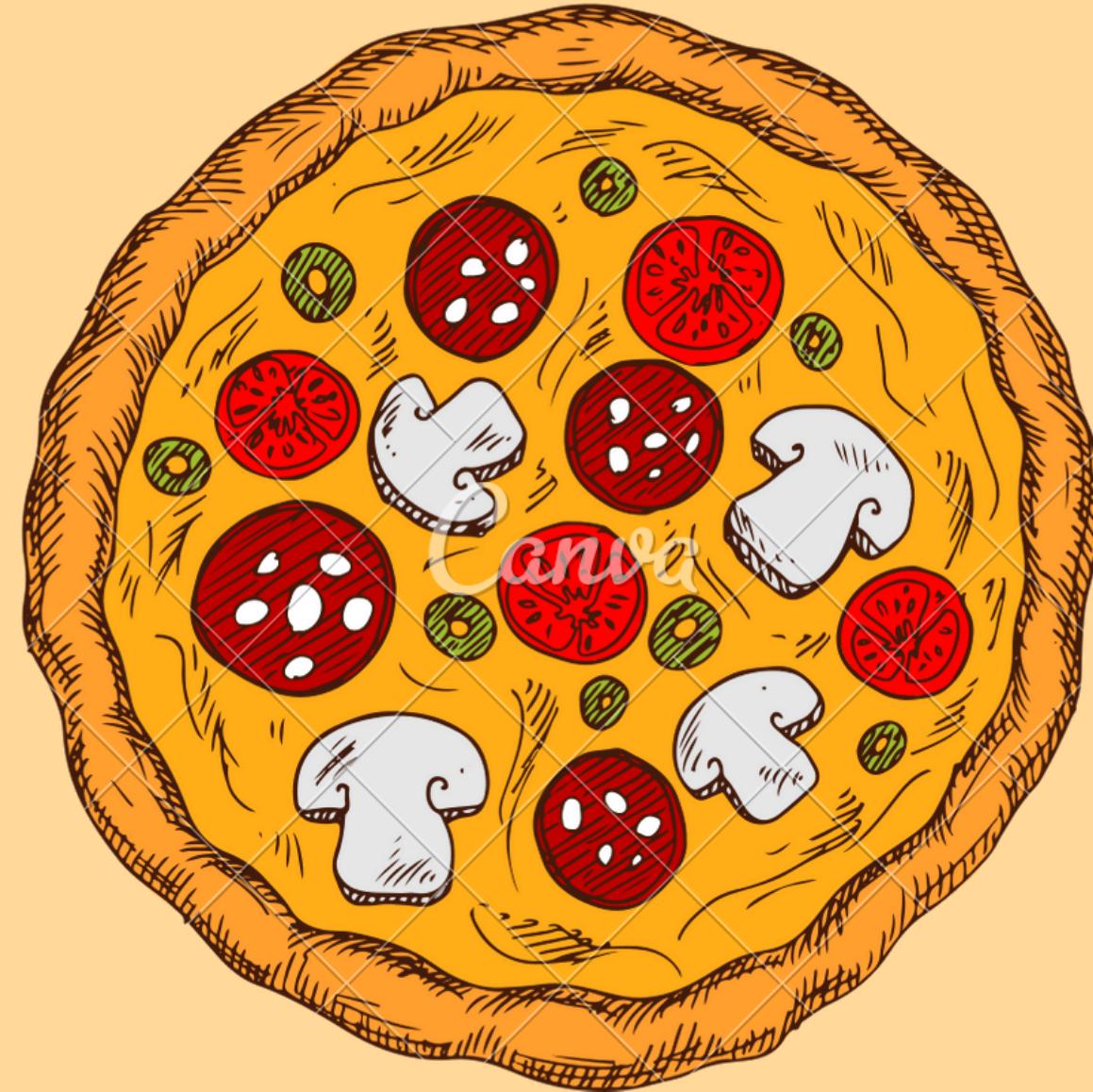


```
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name, sum((order_details.quantity)* pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
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;
```

Result Grid | Filter Rows:

	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
	The Pepperoni Pizza	30161.75





**THANK  
YOU**

**[www.linkedin.com/in/shailendra-kolare-027a8b134](https://www.linkedin.com/in/shailendra-kolare-027a8b134)**