

# Delicio PIZZA

## Pizza Sales

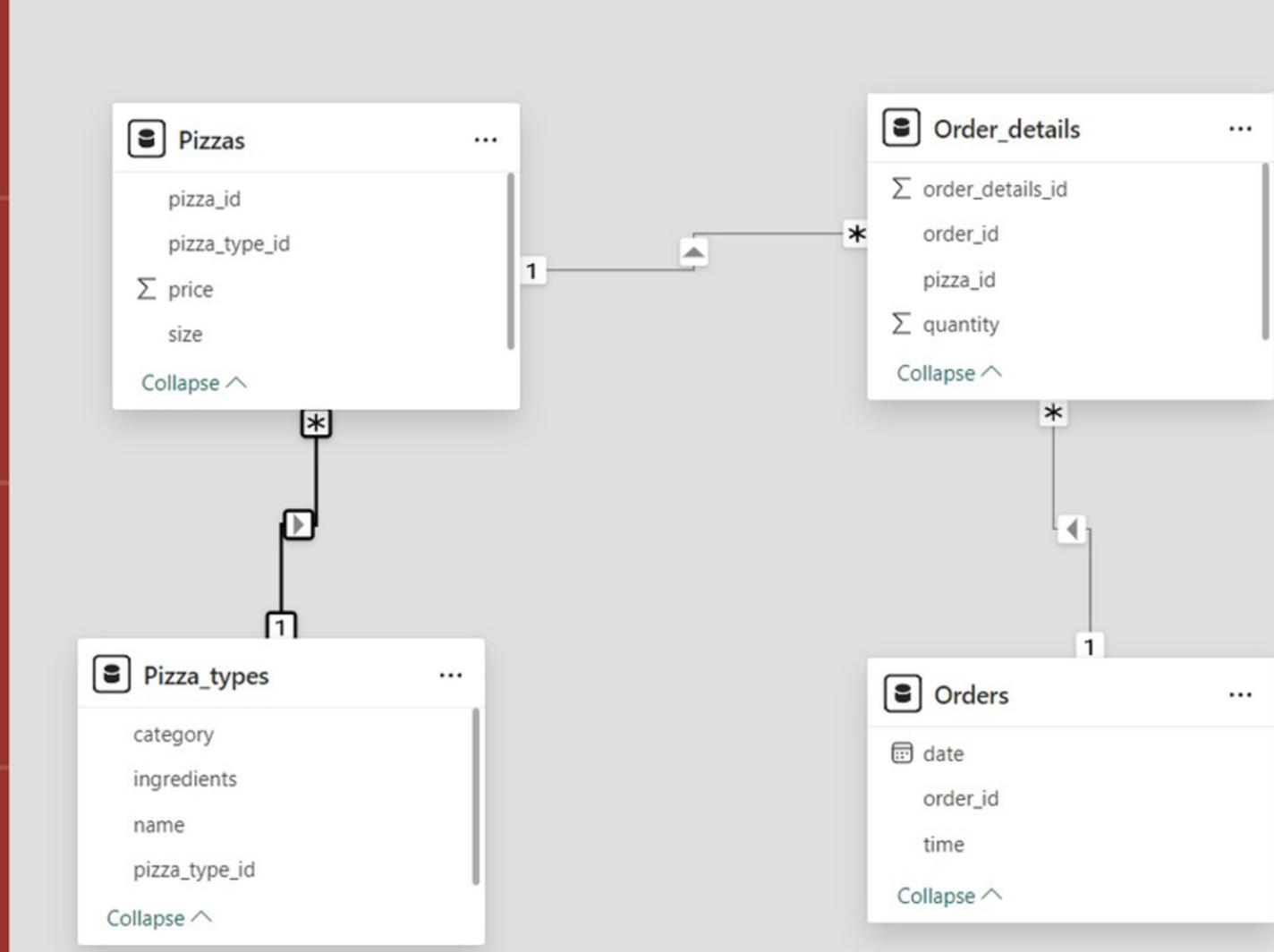
### Analysis



**Excited to share a recent project where I dived deep into the world of pizza sales, leveraging Excel and SQL to extract valuable insights. 🚀 Here's a brief overview of what I accomplished:**



# DATABASE SCHEMA



# 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(orders_details.Quantity * pizzas.price),  
        2) AS Total_Sales  
FROM  
    orders_details  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_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 Row

	name	price
▶	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

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

Result Grid | Filter

size	order_count
L	18526
M	15385

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

```
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.

SELECT

```
    pizza_types.category,  
    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.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 hours, COUNT(order_id)  
FROM  
    orders AS order_count  
GROUP BY HOUR(order_time);
```

	hours	count(order_id)
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

**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)  
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
round(avg(Quantity),0)		
▶	138	Resets all sort

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

SELECT

```
    pizza_types.name,  
    SUM(orders_details.Quantity * pizzas.price) AS revenue
```

FROM

```
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.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	43434.25
	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(orders_details.Quantity * pizzas.price) /(SELECT
        ROUND(SUM(orders_details.Quantity * pizzas.price),
        2) 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 pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.Pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid

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 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	2713.850000000000
	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

# 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((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 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
	The Spicy Italian Pizza	34831.25

**Excited about the outcomes and  
looking forward to applying these  
strategies in future projects!**

**This structure provides a  
comprehensive overview while  
focusing on actionable insights to  
drive sales growth and improve  
customer satisfaction.**