



SQL PROJECT ON PIZZA SALES





About Project,

The Pizza Sales project utilizes SQL to generate a comprehensive sales insight report, analyzing customer orders, sales trends, and product performance. This report provides actionable insights, helping the pizza business optimize inventory, identify top-selling items, and understand customer preferences, ultimately driving better decision-making and increased profitability.





Hello,

My Name is **Machhindra Shimpi**

Data enthusiast with a passion for uncovering insights, I'm excited to apply my SQL skills to analyze pizza sales data.

Linkdin: <https://www.linkedin.com/in/machhindra-shimpi-61340a247>

GitHub: https://github.com/Machhindra1945/Pizza_sales_Report

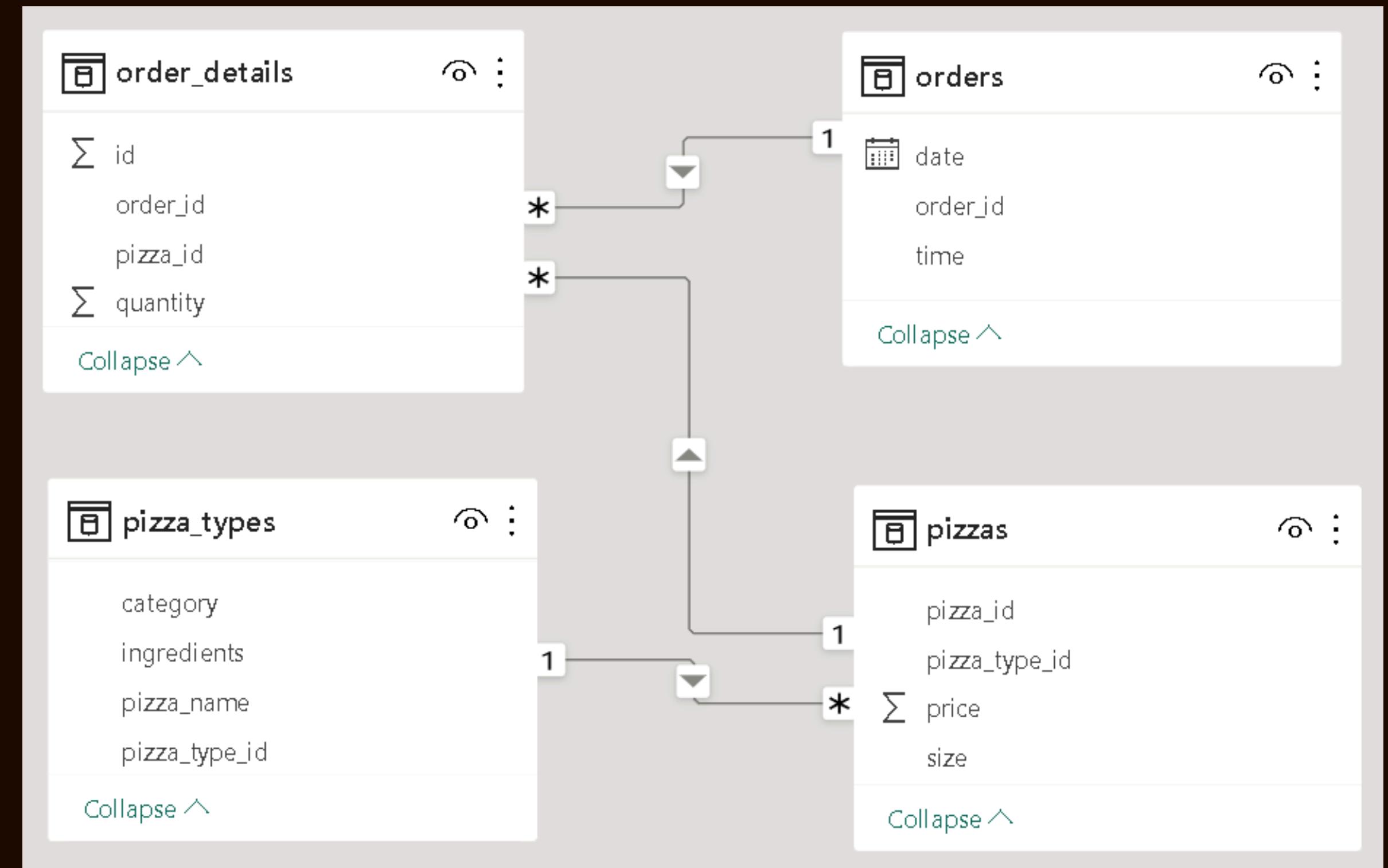
Mobile No: 7397944586





DATA MODEL

Here we can see the Data Model of pizza sales data. those we get from client.



Clinte requirement Question



1. Retrieve the total number of orders placed

Query Query History

```
1 select count(order_id) as Number_of_Orders from order_details
```

number_of_orders	bigint
	48620



2. Calculate the total revenue generated from pizza sales

Query Query History

```
1 ✓ select
2   round(sum( pizzas.price * order_details.quantity ), 2) as total_revenue
3   from pizzas join order_details
4     on pizzas.pizza_id = order_details.pizza_id
```

total_revenue	🔒
numeric	
817860.05	





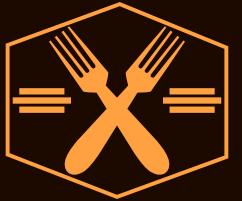
3. Identify the highest-priced pizza

Query Query History

```
1 ✓ select
2   pizza_types.pizza_name, pizzas.price
3   from pizzas join pizza_types
4   on pizzas.pizza_type_id = pizza_types.pizza_type_id
5   order by pizzas.price desc
6   limit 1
```

pizza_name	price
character varying (100)	numeric
The Greek Pizza	35.95





4. Identify the most common pizza size ordered

Query Query History

```
1 ✓ SELECT pizzas.size as Pizza_size,
2   count(order_details.quantity) as order_count
3   from pizzas join order_details
4     on pizzas.pizza_id = order_details.pizza_id
5   group by pizzas.size
6   order by order_count desc
7   limit 1
```

pizza_size	order_count
L	18526





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

Query Query History

```
1 ✓ SELECT pizza_types.pizza_name,  
2 sum(order_details.quantity) as quantity  
3 FROM pizza_types join pizzas  
4 on pizza_types.pizza_type_id = pizzas.pizza_type_id  
5 join order_details  
6 on order_details.pizza_id = pizzas.pizza_id  
7 group by pizza_types.pizza_name  
8 order by quantity desc  
9 LIMIT 5
```

pizza_name character varying (100)	quantity numeric
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371





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

Query Query History

```
1 ▾ SELECT pizza_types.category as Pizza_category,  
2   sum(order_details.quantity) as quantity  
3   FROM pizza_types join pizzas  
4   on pizza_types.pizza_type_id = pizzas.pizza_type_id  
5   join order_details  
6   on order_details.pizza_id = pizzas.pizza_id  
7   group by pizza_types.category  
8   order by quantity desc
```

pizza_category	quantity
character varying (50)	numeric
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050





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

Query Query History

```
1 ✓ select extract(hour from time) as order_hours,  
2   count(*) as order_count from orders  
3   group by order_hours  
4   order by order_hours
```

order_hours	order_count
numeric	bigint
9	1
10	8
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



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

Query Query History

```
1 ✓ select pizza_types.category,  
2 count(order_details.order_id) as order_count  
3 from pizza_types join pizzas  
4 on pizza_types.pizza_type_id = pizzas.pizza_type_id  
5 join order_details  
6 on pizzas.pizza_id = order_details.pizza_id  
7 group by pizza_types.category  
8 order by order_count desc
```

category	order_count
character varying (50)	bigint
Classic	14579
Supreme	11777
Veggie	11449
Chicken	10815



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

Query Query History

```
1 ✓ select pizza_types.pizza_name,
2     sum(pizzas.price * order_details.quantity) as total_revenue
3     from pizza_types join pizzas
4     on pizza_types.pizza_type_id = pizzas.pizza_type_id
5     join order_details
6     on pizzas.pizza_id = order_details.pizza_id
7     group by pizza_types.pizza_name
8     order by total_revenue desc
9     limit 3
```

pizza_name character varying (100)	total_revenue numeric
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50



10. Analyze the cumulative revenue generated over time



Query Query History

```
1 ✘ select date,  
2   sum(revenue) over(order by date) as cum_revenue  
3   from  
4     (select orders.date, sum(order_details.quantity * pizzas.price) as revenue  
5       from orders join order_details  
6       on orders.order_id = order_details.order_id  
7       join pizzas  
8       on order_details.pizza_id = pizzas.pizza_id  
9       group by orders.date) as sales  
10  limit 5
```

	date date	cum_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60
5	2015-01-05	11929.55



THANK YOU

