

# **SQL PROJECT**

# **PIZZA SALES**

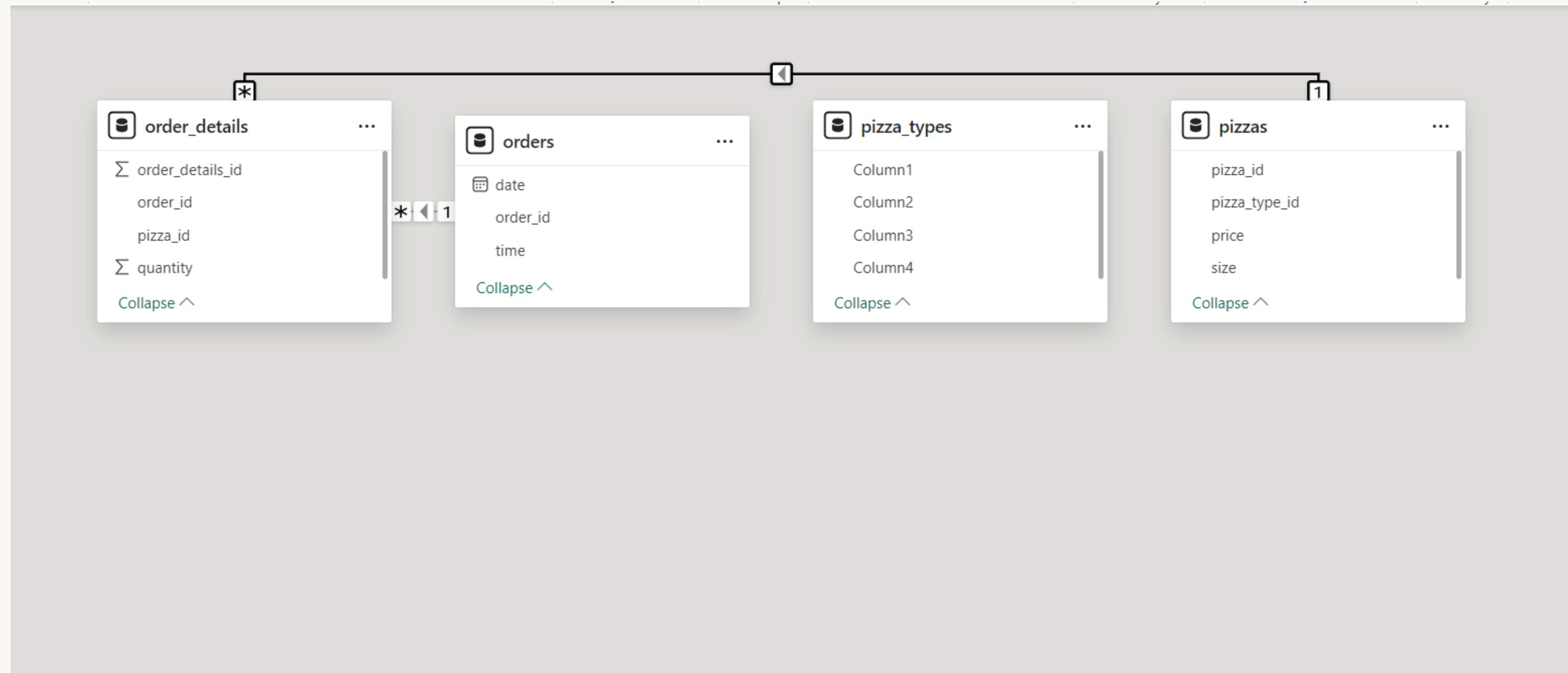


# INTRODUCTION

My name is Shivam Thakral. In this project I have utilized SQL queries to solve questions that were related to pizza sales.



# DATA MODEL OF ALL TABLES



## ALL QUESTIONS

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

### SQL QUERY

```
2      I  
3 • select count(order_id) as total_orders from orders;
```

### OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:

	total_orders
▶	21350

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

## SQL QUERY

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

## OUTPUT

Result Grid	
	total_sales
▶	817860.05

# IDENTIFY THE HIGHEST-PRICED PIZZA.

## SQL QUERY

```
3 • SELECT
4   pizza_types.name, pizzas.price
5 FROM
6   pizza_types
7     JOIN
8   pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 ORDER BY pizzas.price DESC
10 LIMIT 1;
```

## OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	name	price		
▶	The Greek Pizza	35.95		

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SQL QUERY 2

```
3• select pizzas.size, count(order_details.order_details_id) as  
4 from pizzas join order_details  
5 on pizzas.pizza_id = order_details.pizza_id  
6 group by pizzas.size order by order_count desc ;  
7
```

OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	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.

## SQL QUERY

```
4 • SELECT
5     pizza_types.name, SUM(order_details.quantity) AS quantity
6 FROM
7     pizza_types
8         JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10        JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC
14 LIMIT 5;
```

## OUTPUT

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaian 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.

### SQL QUERY

```
4 • select pizza_types.category,  
5   sum(order_details.quantity) as quantity  
6   from pizza_types join pizzas  
7   on pizza_types.pizza_type_id = pizzas.pizza_type_id  
8   join order_details  
9   on order_details.pizza_id = pizzas.pizza_id  
10  group by pizza_types.category order by quantity desc;
```

### OUTPUT

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

## SQL QUERY

```
2  
3 • SELECT  
4     HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5 FROM  
6     orders  
7 GROUP BY HOUR(order_time);
```

## OUTPUT

	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	

## JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

### SQL QUERY

```
1 -- Join relevant tables to find the
2 -- category-wise distribution of pizzas.
3
4 select category , count(name) from pizza_types
5 group by category;
```

### OUTPUT

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

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

## SQL QUERY

```
3 • select pizza_types.name,  
4   sum(order_details.quantity * pizzas.price) as revenue  
5   from pizza_types join pizzas  
6   on pizzas.pizza_type_id = pizza_types.pizza_type_id  
7   join order_details  
8   on order_details.pizza_id = pizzas.pizza_id  
9   group by pizza_types.name order by revenue desc limit 3;
```

## OUTPUT

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

## SQL QUERY

```
3 • select order_date,  
4   sum(revenue) over(order by order_date) as cum_revenue  
5   from  
6   (select orders.order_date,  
7     sum(order_details.quantity * pizzas.price) as revenue  
8     from order_details join pizzas  
9     on order_details.pizza_id = pizzas.pizza_id  
10    join orders  
11    on orders.order_id = order_details.order_id  
12   group by orders.order_date) as sales;
```

## OUTPUT

	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.

### SQL QUERY

```
4 • select category, name, revenue,  
5   rank() over(partition by category order by revenue desc) as rn  
6   from  
7   (select pizza_types.category, pizza_types.name,  
8     sum((order_details.quantity) * pizzas.price) as revenue  
9   from pizza_types join pizzas  
10  on pizza_types.pizza_type_id = pizzas.pizza_type_id  
11  join order_details  
12  on order_details.pizza_id = pizzas.pizza_id  
13  group by pizza_types.category, pizza_types.name) as a ;
```

### OUTPUT

	category	name	revenue	rn
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken Pizza	42768	2
3	Chicken	The California Chicken Pizza	41409.5	3
4	Chicken	The Southwest Chicken Pizza	34705.75	4
5	Chicken	The Chicken Alfredo Pizza	16900.25	5
6	Chicken	The Chicken Pesto Pizza	16701.75	6

**THANK YOU**