

SQL PROJECT

PIZZA HOT SALES

Comprehensive Data Insights & Business Trends

* TOOL: SQL, DBMS, EXCEL

BALRAM SHAH

DATA SCIENCE AND ANALYSIS ENTHUSIASTS

11/20/2025



INTRODUCTION

INTRODUCTION

This SQL project analyzes pizza sales data to identify the top-selling pizzas, revenue patterns, customer preferences, and peak ordering times.

The insights help the business improve marketing, manage inventory better, and increase overall sales performance.

OBJECTIVES

- Identify top-selling pizzas.
- Analyze revenue trends.
- Understand time-based ordering patterns.
- Provide actionable business recommendations.

DATASET-OVERVIEW

Tables Used:

- **orders** – order timestamps
- **order_details** – pizza quantity per order
- **pizzas** – pizza sizes, price
- **pizza_types** – flavors, categories



PROBLEM STATEMENT

Basic Analysis

- Total number of orders.
- Total revenue generated.
- Highest-priced pizza.
- Most common pizza size.
- Top 5 most ordered pizza types by quantity.

Intermediate Analysis

- Category-wise total quantity.
- Orders distribution by hour.
- Category distribution.
- Avg pizzas ordered per day.
- Top 3 pizza types by revenue.

Advance Analysis

- % revenue contribution of each pizza type.
- Cumulative revenue over time.
- Category-wise top 3 pizzas by revenue.



DATABASE SCHEMA (ER DIAGRAM SUMMARY)

ERD Explanation

1. pizza_types → pizzas

- Each pizza type can have multiple sizes.

2. pizzas → order_details

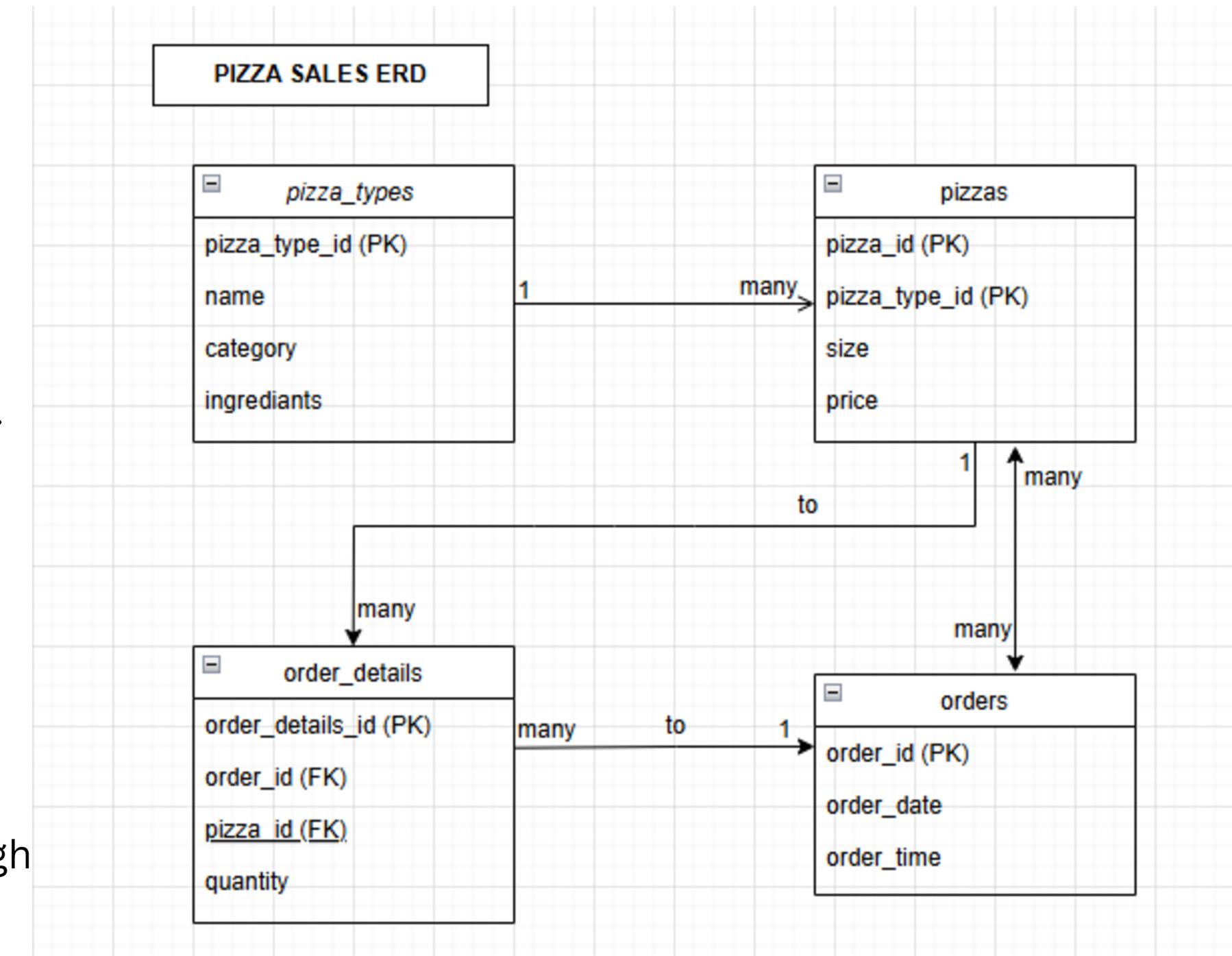
- A single pizza can be ordered by many customers.

3. orders → order_details

- Each order can contain multiple pizzas.

4. orders ↔ pizzas (Many-to-Many)

- Many pizzas in an order
- Many orders can include the same pizza
- This many-to-many relationship is created through order_details.



ANALYSIS :

1 - Retrieve the total number of orders placed.

```
3 • SELECT  
4     COUNT(order_id) AS total_order  
5 FROM  
6     orders;
```

Result Grid	
	total_order
▶	21350

2 - Calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_revenue
▶	817860.05

3 - Identify the highest-priced pizza.

```
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 3;
```

	name	price
▶	The Greek Pizza	35.95
▶	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65

4- Identify the most common pizza size ordered.

```
3 •  SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_size
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_size DESC
12 LIMIT 1;
```

	size	order_size
▶	L	18526

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

```
3 •   SELECT
4     pizza_types.name,
5       SUM(order_details.quantity) AS order_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 order_quantity DESC
14 LIMIT 5;
```

	name	order_quantity
▶	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.

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

	total_quantity	pizza_category
▶	14888	Classic
	11987	Supreme
	11649	Veggie
	11050	Chicken

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

```
3 •   SELECT  
4       HOUR(order_time), COUNT(order_id) as count  
5   FROM  
6       orders  
7   GROUP BY HOUR(order_time) order by count;  
8
```

HOUR(order_time)	count
9	1
10	8
23	28
22	663
21	1198
11	1231
15	1468
14	1472
20	1642
16	1920
19	2009
17	2336
18	2399
13	2455
12	2520

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

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

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

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

```
3 •   SELECT
4     ROUND(AVG(pizza_ordered), 0) AS average_pizza_order
5   FROM
6     (SELECT
7       orders.order_date,
8         SUM(order_details.quantity) AS pizza_ordered
9     FROM
10    orders
11   JOIN order_details ON orders.order_id = order_details.order_id
12   GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	average_pizza_order
▶	138

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

```
3 •   SELECT
4     pizza_types.name,
5     ROUND(SUM(order_details.quantity * pizzas.price),
6           2) AS revenue
7   FROM
8     pizza_types
9     JOIN
10    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11     JOIN
12    order_details ON order_details.pizza_id = pizzas.pizza_id
13   GROUP BY pizza_types.name
14   ORDER BY revenue DESC
15   LIMIT 3;
```

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

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

```
3 •   SELECT
4     pizza_types.category,
5     ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6             ROUND(SUM(order_details.quantity * pizzas.price),
7             2) AS revenue
8         FROM
9             order_details
10            JOIN
11                pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
12            2) AS total_revenue
13    FROM
14        pizza_types
15            JOIN
16                pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
17            JOIN
18                order_details ON order_details.pizza_id = pizzas.pizza_id
19    GROUP BY category
20    ORDER BY total_revenue DESC;
```

	category	total_revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12 -Analyze the cumulative revenue generated over time.

```
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 order_details.order_id = orders.order_id  
12      group by orders.order_date) as daily_sales;
```

	order_date	cum_revenue
1	2015-01-25	56631.40000000001
2	2015-01-26	58515.80000000001
3	2015-01-27	61043.85000000001
4	2015-01-28	63059.85000000001
5	2015-01-29	65105.150000000016
6	2015-01-30	67375.45000000001
7	2015-01-31	69793.30000000002
8	2015-02-01	72982.50000000001
9	2015-02-02	75311.10000000002
10	2015-02-03	77925.90000000002
11	2015-02-04	80159.80000000002
12	2015-02-05	82375.60000000002
13	2015-02-06	84885.55000000002
14	2015-02-07	87123.20000000001
15	2015-02-08	89158.20000000001
16	2015-02-09	91353.55000000002
17	2015-02-10	93410.05000000002
18	2015-02-11	95870.05000000002
19	2015-02-12	98028.85000000002
20	2015-02-13	100783.35000000002
21	2015-02-14	103102.50000000001
22	2015-02-15	105243.75000000001
23	2015-02-16	107212.55000000002
24	2015-02-17	109334.45000000001
25	2015-02-18	111077.20000000002

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

```
3 • select category, name, revenue
4
5   from
6   (select category, name, revenue,
7    rank() over (partition by category order by revenue) as cr
8
9   from
10  (select pizza_types.category, pizza_types.name,
11  sum(order_details.quantity * pizzas.price) as revenue
12
13  from pizza_types join pizzas
14  on pizza_types.pizza_type_id = pizzas.pizza_type_id
15  join order_details
16  on order_details.pizza_id = pizzas.pizza_id
17
18  group by pizza_types.category, pizza_types.name) as cat_name_revenue) as name_revenue
19  where cr <=3 ;
--
```

	category	name	revenue
▶	Chicken	The Chicken Pesto Pizza	16701.75
	Chicken	The Chicken Alfredo Pizza	16900.25
	Chicken	The Southwest Chicken Pizza	34705.75
	Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
	Classic	The Big Meat Pizza	22968
	Classic	The Napolitana Pizza	24087
	Supreme	The Brie Carre Pizza	11588.49999999999
	Supreme	The Spinach Supreme Pizza	15277.75
	Supreme	The Calabrese Pizza	15934.25
	Veggie	The Green Garden Pizza	13955.75
	Veggie	The Mediterranean Pizza	15360.5
	Veggie	The Spinach Pesto Pizza	15596

BUSINESS INSIGHTS

- Total Orders Are Very High (21,350 orders).
- Total Revenue is Excellent (₹817,860.05).
- The Greek Pizza is the Most Expensive (₹35.95).
- Large Size (L) is the Most Ordered (18,526 orders).
- Top 5 Most Ordered Pizzas Are Mostly Classic & Chicken Based Examples: Classic Deluxe, Barbecue Chicken, Hawaiian, Pepperoni, Thai Chicken.
- Classic Category Has the Highest Quantity Sold (14,888).
- Peak Ordering Hours Are Between 12 PM – 2 PM and 5 PM – 8 PM.
- Average Pizzas Ordered per Day = 138 pizzas/day.
- Top 3 Revenue-Generating Pizzas: Thai Chicken Pizza – ₹43,345, Barbecue Chicken Pizza – ₹42,768, California Chicken Pizza – ₹41,409.
- Category-Wise Highest Revenue: Classic = highest revenue (₹26.91%), Supreme = ₹25.46%, Chicken = ₹23.96%, Veggie = lowest (₹23.68%)
- And Cumulative Revenue Shows Steady Growth.



BUSINESS RECOMMENDATIONS

- Increase Stock for Classic & Chicken Pizzas.
- Focus Promotions on Top Revenue Pizzas Example: Thai Chicken Pizza, Barbecue Chicken Pizza.
- Offer Combo Deals for Large Size Pizzas.
- Increase Staff & Kitchen Preparation During Peak Hours.
- Improve or Rebrand Veggie Category.
- Use Dynamic Pricing During High-Demand Times.
- Promote Medium & Large Group Packs.
- Track Slow-Moving Pizzas.



THANK YOU FOR WATCHING

If you have any queries or feedback,
I'd love to hear from you.

BALRAM SHAH

DATA SCIENCE AND ANALYSIS ENTHUSIASTS

11 / 20 / 2025