



SQL SALES ANALYSIS PROJECT



BY HAYA PARVEEN



INTRODUCTION



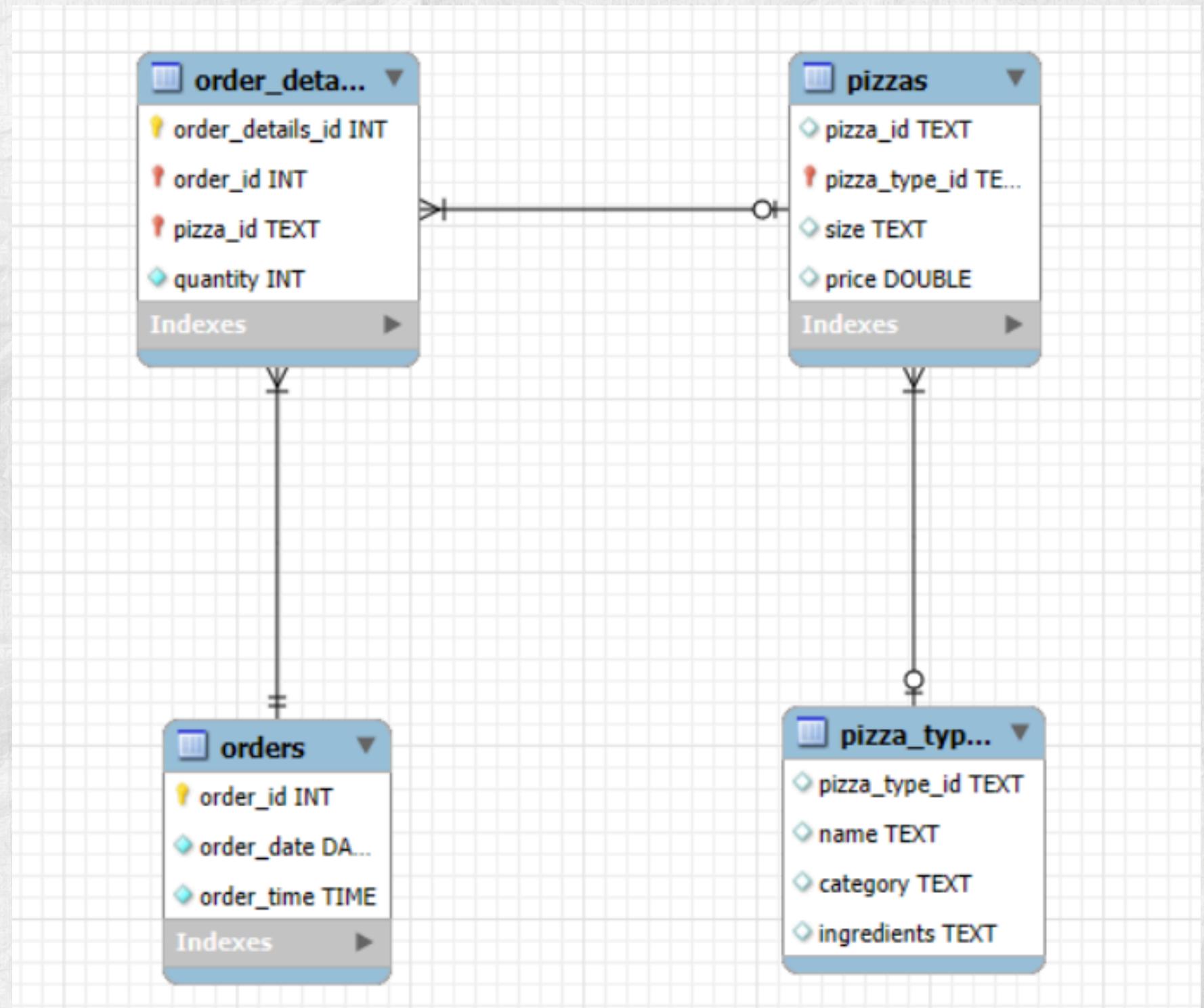
Hi, I'm Haya Parveen, a beginner Data Analyst passionate about learning how to work with real data.

In this project, I followed a guided SQL walkthrough to analyze a pizza sales dataset. I started by importing raw CSV files into a database, created four related tables, and then wrote SQL queries step-by-step to explore the data.

I analyzed total orders, revenue, top-selling pizzas, popular sizes, and customer ordering trends. This helped me understand how to connect tables using joins, use aggregate functions, and write queries to answer real business questions.

The goal was to build my foundation in SQL by working on a complete project - from loading data to generating insights.

SCHEMAS



< BACK

BASIC QUERIES



1. Retrieve the total number of orders placed



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

	Total_number_of_orders
▶	21350

2. Calculate the total revenue generated from pizza sales

```
3 • SELECT  
4     ROUND(SUM(o.quantity * p.price), 2) Total_revenue  
5 FROM  
6   order_details o  
7   JOIN  
8     pizzas p ON o.pizza_id = p.pizza_id;
```

	Total_revenue
▶	817860.05

3. Identify the highest-priced pizza.

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

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```
2
3 • SELECT
4     p.size, COUNT(o.order_details_id) order_count
5 FROM
6     order_details o
7     JOIN
8     pizzas p ON o.pizza_id = p.pizza_id
9 GROUP BY p.size
10 ORDER BY COUNT(o.order_details_id) DESC;
11
```

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



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

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

INTERMEDIATE QUERIES



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

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

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
3 • SELECT
4     category, COUNT(name) pizza_count
5 FROM
6     pizza_types
7 GROUP BY category;
```

	category	pizza_count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

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

	ROUND(AVG(quantity),0)
▶	138

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

```
3 • SELECT
4     pt.name, SUM(o.quantity * p.price) Total_revenue
5 FROM
6     pizza_types pt
7     JOIN
8     pizzas p ON pt.pizza_type_id = p.pizza_type_id
9     JOIN
10    order_details o ON p.pizza_id = o.pizza_id
11   GROUP BY pt.name
12  ORDER BY Total_revenue DESC
13 LIMIT 3;
```

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

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



```
3 • SELECT  
4     EXTRACT(HOUR FROM order_time) Hour,  
5     COUNT(order_id) order_count  
6 FROM  
7     orders  
8 GROUP BY EXTRACT(HOUR FROM 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
	22	663
	23	28
	10	8
	9	1

ADVANCE QUERIES



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

```
3 • SELECT pt.category,
4     ROUND(SUM(od.quantity * p.price) /
5         (SELECT ROUND(SUM(od.quantity * p.price), 2) total_sales
6          FROM order_details od
7            JOIN pizzas p
8              ON od.pizza_id = p.pizza_id) * 100, 2) revenue
9    FROM pizza_types pt
10   JOIN pizzas p
11     ON pt.pizza_type_id = p.pizza_type_id
12   JOIN order_details od ON od.pizza_id = p.pizza_id
13 GROUP BY pt.category
14 ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

2. Analyze the cumulative revenue generated over time

```
3 • SELECT order_date,
4     SUM(revenue) OVER(ORDER BY order_date) cum_revenue
5     FROM
6     (SELECT o.order_date,
7         SUM(od.quantity * p.price) revenue
8         FROM order_details od
9           JOIN pizzas p
10          ON od.pizza_id=p.pizza_id
11        JOIN orders o
12          ON o.order_id=od.order_id
13        GROUP BY o.order_date) 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

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

```
4 •  SELECT name, revenue FROM
5   (SELECT category, name, revenue,
6    RANK() OVER(PARTITION BY category ORDER BY REVENUE DESC) rn
7   FROM
8     (SELECT pt.category,pt.name,
9      SUM(od.quantity * p.price) revenue
10     FROM pizza_types pt
11     JOIN pizzas p
12     ON pt.pizza_type_id=p.pizza_type_id
13     JOIN order_details od
14     ON od.pizza_id=p.pizza_id
15     GROUP BY pt.category,pt.name) A) B
16 WHERE rn<=3;
```

	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
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.7000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



THANK YOU!

This project helped me understand how data analysis drives business decisions.
I look forward to building more real-world projects using SQL and Power BI.

 [GitHub](#)
 [LinkedIn](#)