



PIZZA STORE

HOME

ABOUT

MORE

# SQL Project On Pizza Store



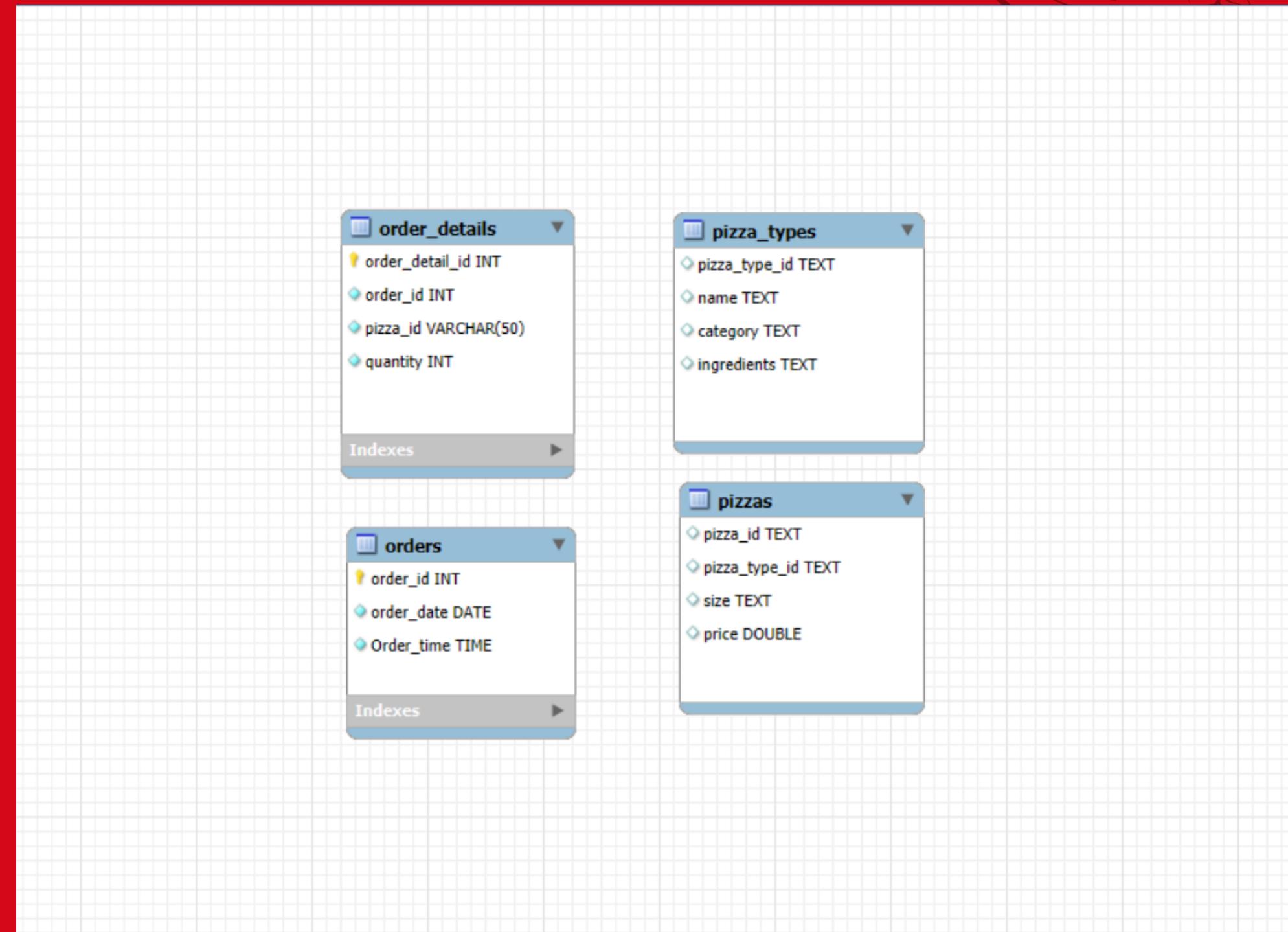
# Introduction

- Purpose: Hands-on project to practice and strengthen SQL skills
- Data Used: 4 tables – orders, order\_details, pizzas, pizza\_types
- Goal: Analyze sales performance and customer ordering patterns
- Approach:
- Created MySQL database and schema
- Applied joins, aggregations, subqueries, and window functions
- Key Insights Extracted:
- Total sales and order volume
- Revenue by pizza type, category, and size
- Peak order hours and quantity trends
- Daily cumulative revenue
- Top 3 best-selling pizzas in each category



# EER Diagram

- The database consists of 4 interconnected tables: orders, order\_details, pizzas, and pizza\_types, structured for efficient sales analysis.
- Relationships are established using primary and foreign keys to link orders with pizzas and their types for detailed revenue and trend insights.





# Retrieve the total number of orders placed.

```
27 •   SELECT count(order_id) as total_orders from orders;
```

---

**Result Grid** | Filter Rows:  Export: Wrap Cell Content:

	total_orders
▶	21350



# Calculate the total revenue generated from pizza sales.

```
32 •   SELECT
33     sum(order_details.quantity * pizzas.price) as total_sales
34     from order_details join pizzas
35     on pizzas.pizza_id = order_details.pizza_id
36
37
38
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:				
<table border="1"><thead><tr><th></th><th>total_sales</th></tr></thead><tbody><tr><td>▶</td><td>817860.049999993</td></tr></tbody></table>					total_sales	▶	817860.049999993	
	total_sales							
▶	817860.049999993							



# Identify the highest-priced pizza.

```
56 •   SELECT pizza_types.name, pizzas.price  
57     from pizza_types join pizzas  
58       on pizza_types.pizza_type_id = pizzas.pizza_type_id  
59     order by pizzas.price desc;
```

60

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

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Spinach Supreme Pizza	20.75
	The Barbecue Chicken Pizza	20.75
	The California Chicken Pizza	20.75



# Identify the most common pizza size ordered.

```
68 •    select pizzas.size, count(order_details.order_detail_id) as order_count  
69      from pizzas join order_details  
70        on pizzas.pizza_id = order_details.pizza_id  
71      group by pizzas.size order by order_count desc;  
72
```

---

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.

```
81 •   SELECT
82       pizza_types.name, SUM(order_details.quantity) AS Qty
83   FROM
84       pizza_types
85       JOIN
86           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
87       JOIN
88           order_details ON order_details.pizza_id = pizzas.pizza_id
89   GROUP BY pizza_types.name
90   ORDER BY Qty DESC
91   LIMIT 5;
```

---

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

	name	Qty
▶	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.

```
97 •   SELECT
98     pizza_types.category, SUM(order_details.quantity) AS QTY
99   FROM
100    pizza_types
101      JOIN
102      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
103      JOIN
104      order_details ON order_details.pizza_id = pizzas.pizza_id
105    GROUP BY pizza_types.category
106    ORDER BY QTY DESC;
```

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

	category	QTY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

```
110
111 •   SELECT
112      HOUR(Order_time), COUNT(order_id)
113  FROM
114    orders
115  GROUP BY HOUR(Order_time);
116
117
...
...
```

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

	hour(Order_time)	count(order_id)
▶	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



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

```
119  
120 •   SELECT category , count(name) from pizza_types  
121      GROUP BY category;  
122  
123  
124
```

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

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.

```
126 •   SELECT
127      ROUND(AVG(Qty), 0)
128  FROM
129  (SELECT
130      orders.order_date, SUM(order_details.quantity) AS Qty
131  FROM
132      orders
133  JOIN order_details ON orders.order_id = order_details.order_id
134  GROUP BY orders.order_date) AS order_quantity;
135
136
```

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

ROUND(avg(Qty),0)
138



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

```
139 •   SELECT
140     pizza_types.name,
141     SUM(order_details.quantity * pizzas.price) AS revenue
142   FROM
143     pizza_types
144       JOIN
145     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
146       JOIN
147     order_details ON order_details.pizza_id = pizzas.pizza_id
148   GROUP BY pizza_types.name
149   ORDER BY revenue DESC
150   LIMIT 3;
```

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

```
152
153 •   SELECT pizza_types.category,
154   ROUND(sum(order_details.quantity * pizzas.price)/ (SELECT
155     ROUND(sum(order_details.quantity * pizzas.price),2) AS total_sales
156   FROM order_details join pizzas
157   on pizzas.pizza_id = order_details.pizza_id)*100, 2) as revenue
158   from pizza_types join pizzas
159   on pizza_types.pizza_type_id = pizzas.pizza_type_id
160   join order_details
161   on order_details.pizza_id = pizzas.pizza_id
162   group by pizza_types.category order by revenue desc;
163
164
---
```

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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



# Analyze the cumulative revenue generated over time.

```
170 •  SELECT order_date,  
171      sum(revenue) over (order by order_date) as cum_revenue  
172      FROM  
173      (SELECT orders.order_date,  
174          sum(order_details.quantity * pizzas.price) as revenue  
175          from order_details join pizzas  
176          on order_details.pizza_id = pizzas.pizza_id  
177          join orders  
178          on orders.order_id = order_details.order_id  
179          GROUP BY orders.order_date) as sales;  
180
```

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

	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
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4



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

```
187 •   SELECT name, revenue FROM
188   (SELECT category , name , revenue,
189    RANK() OVER(partition by category order by revenue desc) as rn
190   from
191   (SELECT pizza_types.category, pizza_types.name,
192    sum(order_details.quantity * pizzas.price) as revenue
193    from pizza_types join pizzas
194    on pizza_types.pizza_type_id = pizzas.pizza_type_id
195    join order_details
196    on order_details.pizza_id = pizzas.pizza_id
197    group by pizza_types.category, pizza_types.name) as a) as b
198   WHERE rn <=3;
199
```

Result Grid	
	name
▶	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
	32265.70000000065
	The Mexicana Pizza
	26780.75
	The Five Cheese Pizza
	26066.5



PIZZA STORE

HOME

ABOUT

MORE



Thank  
you!

[www.reallygreatsite.com](http://www.reallygreatsite.com)

