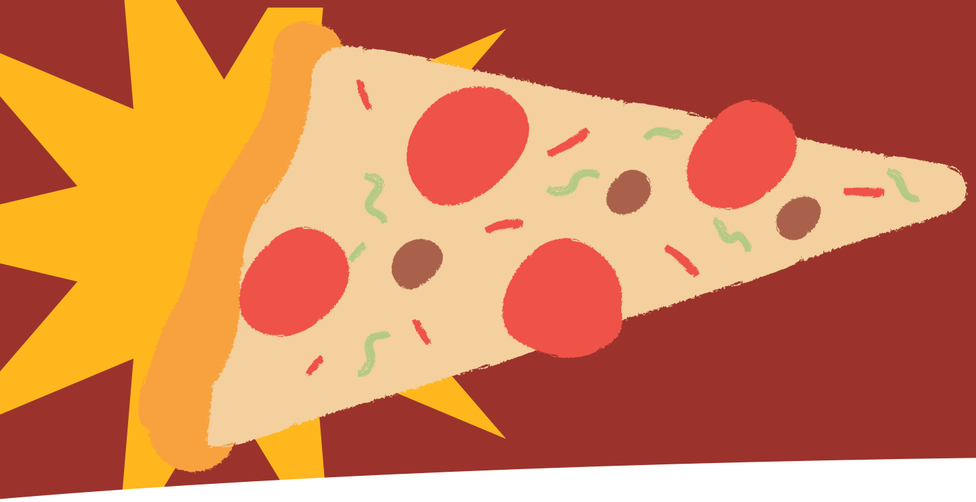
PIZZA SALES DATA ANALYSIS USING SQL

DESCRIPTION

This SQL project presents an analysis of a Pizza store database, providing insights into the store's business expansion. The study of this data will enhance our understanding of the store's growth trajectory in the food industry.

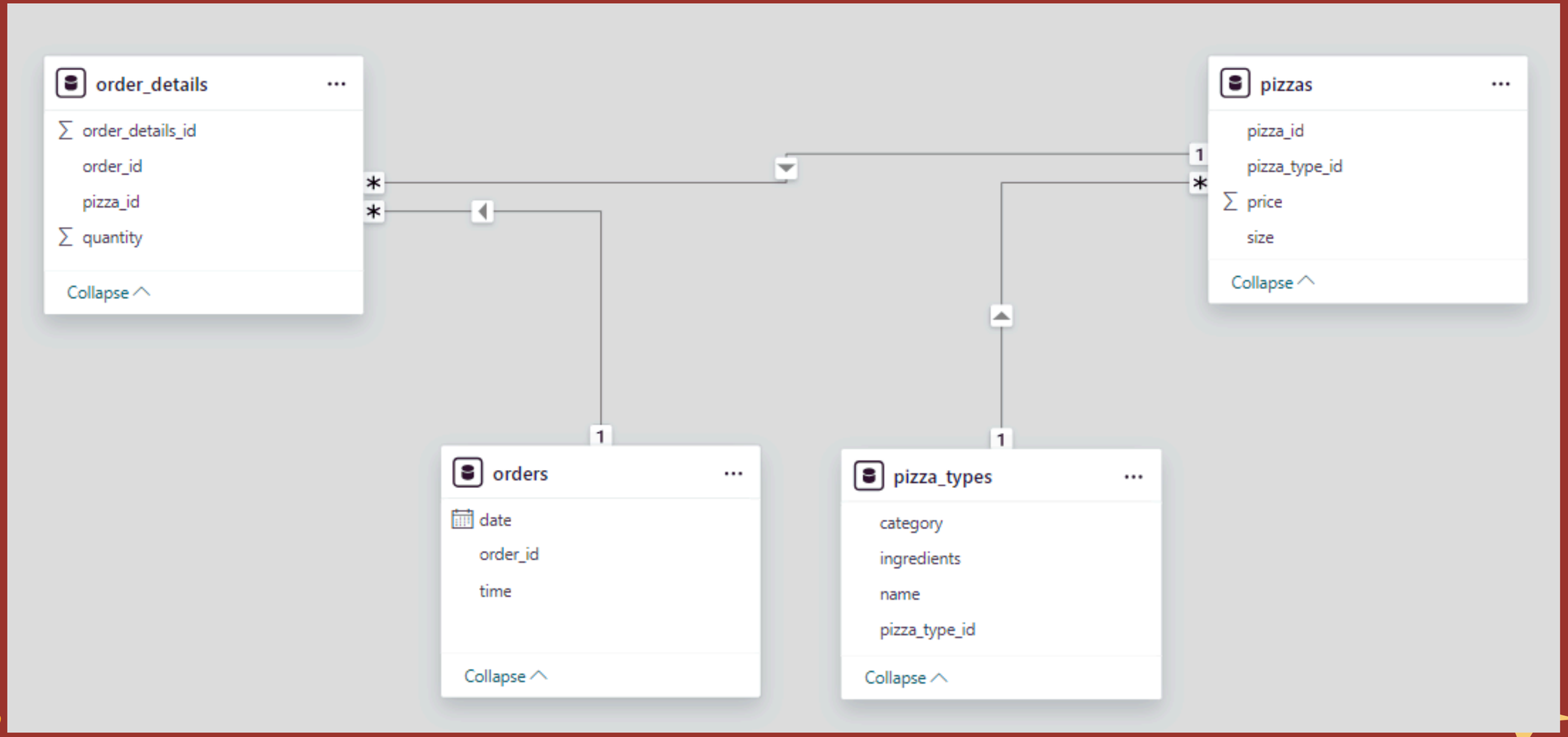


DATABASE AND TOOLS

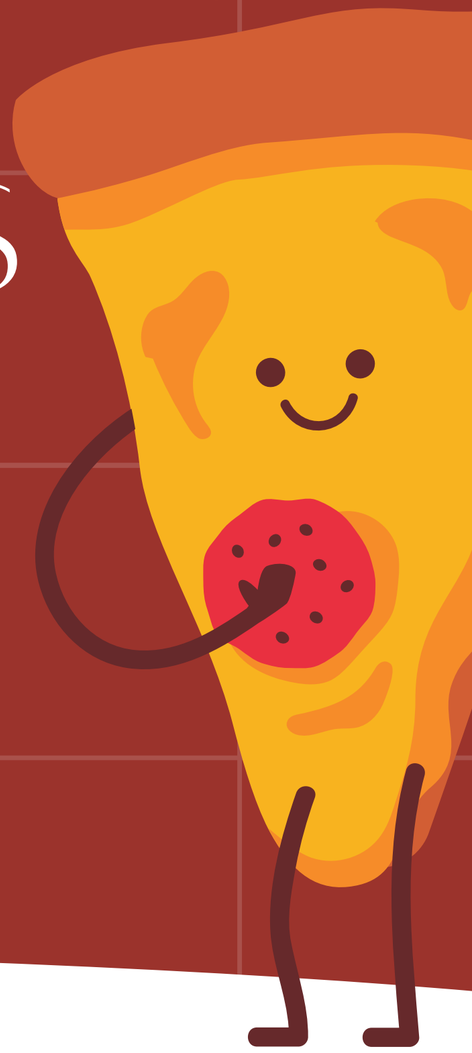
- **MySQL**
- **MySQL Workbench**

This project is a comprehensive SQL-based analysis tool for monitoring pizza sales, utilizing the robust MySQL database management system to store and manage sales data efficiently.

SCHEMA


order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
orders
date
order_id
time
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
1
*
1
*
1
1

A SET OF SQL QUESTIONS TAILORED TO THE PROJECT'S INSIGHTS.



LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS most_ordered_pizza
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered_pizza DESC
LIMIT 5;
```

name	most_ordered_piz
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select hour(order_time) as Hours , count( order_id) as order_distribution
from orders
group by Hours ;
```

Hours	order_distribution
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

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity)) AS avg_pizza_ordered
FROM
    (SELECT
        orders.order_date AS dates,
            SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY dates) AS order_quantity;
```

	avg_pizza_ordered
▶	138

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

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price)) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
round(SUM(order_details.quantity * pizzas.price)) as revenue
from order_details JOIN pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9863
2015-01-05	11929
2015-01-06	14358
2015-01-07	16560
2015-01-08	19398
2015-01-09	21525
2015-01-10	23989
2015-01-11	25861
2015-01-12	27780
2015-01-13	29830
2015-01-14	32357
2015-01-15	34342


For a concise overview of my project, including datasets and a PDF with questions, visit my GitHub page. The link is provided for full details.
https://github.com/PallaviJaiswal40/Pizza_Sales_SQL_Project


THANK YOU