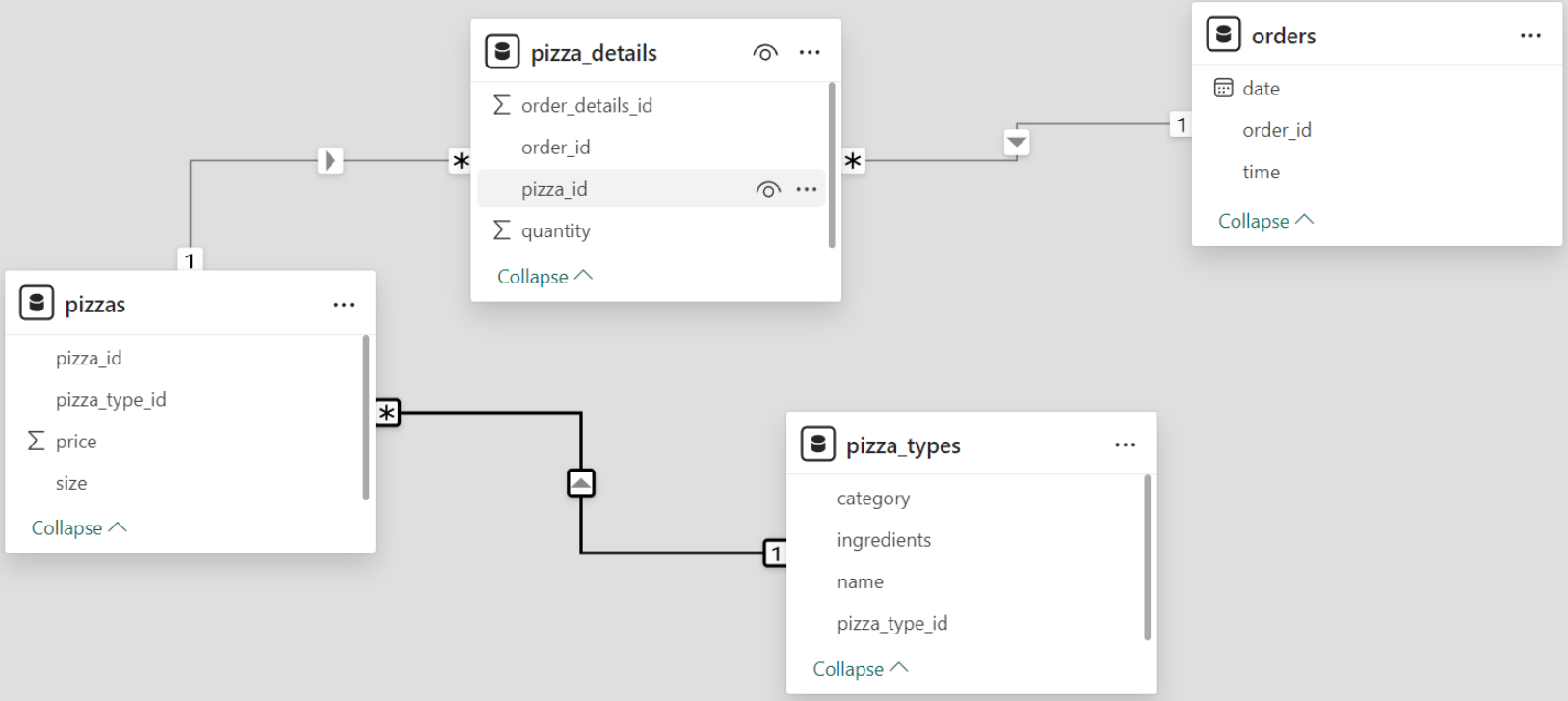
Pizzeria

Pizza Restaurant Data Querying in MySQL

Data Brief

- In the competitive landscape of the food industry, data-driven decision-making is paramount for enhancing operational efficiency, customer satisfaction, and overall profitability. This project focuses on leveraging MySQL to analyze and extract valuable insights from a comprehensive dataset of a pizza restaurant.
- The dataset encompasses various critical aspects of the restaurant's operations, including the names, prices, and categories of pizzas, along with detailed order information such as quantities ordered, order IDs, pizza IDs, and pizza type IDs.

Data Model


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

Objective

The primary objective of this project is to utilize MySQL for querying and analyzing the data to uncover actionable insights that can help the restaurant optimize its operations. The specific goals include:

1. Identifying Top Ordered Pizzas:

Determine which pizzas are the most popular among customers based on order frequency.

2. Revenue Analysis:

Analyze which pizzas contribute the most to the restaurant's revenue on a daily and hourly basis. Understand revenue distribution across different pizza categories.

3. Category-Wise Distribution:

Examine the distribution of orders among different pizza categories to identify customer preferences and trends.

Total Revenue

The SQL Query:

```
select
round(sum(pizza_details.quantity * pizzas.price),2) as total_rev
from pizza_details join pizzas
on pizzas.pizza_id = pizza_details.pizza_id
```

Total_Revenue

$817860.05

Most Ordered Pizza Size

The SQL Query:

```
select quantity, count(order_id)
from pizza_details
group by quantity;
```

Most Ordered Pizza Size

PIZZAS	
Medium	15385
Large	18526
Small	14137
Extra Large	544
Extra Extra Large	28

Highest Priced Pizza

1

The Greek Pizza

$35.95

The SQL Query:

```
select pizza_types.name ,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

Total Quantity of Each Pizza Category Ordered

The SQL Query:

```
select pizza_types.category , sum(pizza_details.quantity) as quantity
from pizza_types join pizzas on
pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizza_details
on pizza_details.pizza_id = pizzas.pizza_id
group by category
order by quantity desc;
```

Total Quantity of Each Pizza Category Ordered

Classic

Total Quantity of Pizza Ordered in Classic Category:- 14888

Supreme

Total Quantity of Pizza Ordered in Supreme Category:- 11987

Veggie

Total Quantity of Pizza Ordered in Veggie Category:- 11649

Chicken

Total Quantity of Pizza Ordered in Chicken Category:- 11050

List of 5 most Ordered Pizza Types with their Quantities

The SQL Query:

```
select pizza_types.name, sum(pizza_details.quantity) as total_quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizza_details
on pizza_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by total_quantity desc limit 5 ;
```

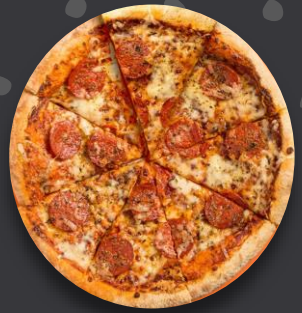
List of 5 most Ordered Pizza Types with their Quantities

- The Classic Deluxe Pizza: 2453

- The Hawaiian Pizza: 2422

- The Thai Chicken Pizza: 2371

- The Barbecue Chicken Pizza: 2432



- The Pepperoni Pizza: 2418

Distribution of Orders by Hour of the Day:

The SQL Query:

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

Result:

	hour(order_time)	count(order_id)
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

- With the analyzed data we can see that in the 16th hour of the dat most orders are received.

The Average Number of Pizzas Ordered Per Day

The SQL Query:

```
select round(avg(quantity_sum),0)as avg_order_per_day from
(select (orders.order_date) , sum(pizza_details.quantity) as quantity_sum
from orders join pizza_details
on orders.order_id = pizza_details.order_id
group by orders.order_date) as order_quantity;
```

The Average Number of Pizzas Ordered Per Day

Average number of Pizzas ordered:

138

The Percentage Contribution of Each Pizza Type to Total Revenue

The SQL Query:

```sql
SELECT
    pizza_types.category,
    ROUND((SUM(pizza_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(pizza_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    pizza_details
                        JOIN
                    pizzas ON pizzas.pizza_id = pizza_details.pizza_id)) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    pizza_details ON pizza_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

The Percentage Contribution of Each Pizza Type to Total Revenue

Classic 26.91%			Supreme 25.46%
Chicken 23.96%			Veggie 23.68%

Top 3 Most Ordered Pizza Types Based on Revenue

The SQL Query:

```sql
select pizza_types.name,
sum(pizza_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join pizza_details
on pizza_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc limit 3;
```

Top 3 Most Ordered Pizza Types Based on Revenue

The Chicken Pesto Pizza	The Barbecue Chicken Pizza	The California Chicken Pizza
43434.25	42768	41409.5

The Cumulative Revenue Generated Over Time

The SQL Query:

```sql
select order_date,
sum(revenue) over(order by order_date) as cumulative_rev
from
(select orders.order_date,
sum(pizza_details.quantity * pizzas.price) as revenue
from pizza_details join pizzas
on pizza_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = pizza_details.order_id
group by orders.order_date) as sales;
```

Result:

order_date	cumulative_rev
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
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

order_date	cumulative_rev
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.8500000001
2015-12-19	797083.05
2015-12-20	799187.9500000001
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

The Cumulative Revenue Generated Over Time

Cumulative Revenue over a Year

- The data given is of a whole year 2015 from January till December.
- Starting from order date *01-Jan-2015* the revenue showing is **2713.85** and its increasing cumulatively till *Dec 31st* which end up to be **817860.05**.

THANK YOU