Pizza Resto

About Contact

SQL PROJECT

- PIZZA SALES

Pizza Resto

ABOUT ME

Hello, my name is **Khushi.** I graduated with a Bachelor of Computer Applications (BCA) from Kurukshetra University (KUK) and have completed a specialized course in Data Science and Artificial Intelligence from Ducat.

I am passionate about data science, SQL, and Python, with hands-on experience in projects such as:

- Google Play Store Analysis – exploring app trends, ratings, and user engagement
- Flight Price Prediction – applying feature engineering and machine learning forecasting
- Pizza Sales SQL Project – analyzing sales data to uncover insights on customer preferences, revenue trends, and performance metrics

My strengths include:

- SQL schema design and troubleshooting
- Python for data analysis
- Clear communication and documentation
- Structured problem-solving with practical examples

Beyond academics, I apply my data-driven mindset to personal projects, including health tracking and calorie analysis, treating wellness as a structured experiment. I enjoy sharing my learning journey on platforms like LinkedIn to connect with others and grow professionally.

My short-term goal is to master SQL and data science while building a strong portfolio on GitHub and LinkedIn. Long-term, I aspire to contribute to impactful projects that combine technology and real-world problem solving.

INTRODUCTION

In this project, I explored pizza sales data using SQL as the primary tool for analysis. The goal was to answer business-driven questions such as identifying top-selling pizza categories, calculating daily and cumulative revenue, and ranking pizzas based on performance.

To achieve this, I utilized MySQL Workbench as the environment for writing and executing queries. The project involved:

- Data integration: Joining multiple tables such as orders, order_details, pizzas, and pizza_types to build meaningful relationships.
- Revenue analysis: Using aggregate functions like SUM() to calculate total and category-wise sales.
- Ranking and comparison: Applying window functions (RANK(), ROW_NUMBER()) to identify best-selling pizzas within each category.
- Trend tracking: Leveraging cumulative sums with OVER(ORDER BY ...) to monitor revenue growth over time.
- SQL Workbench utilization: Executing queries, debugging errors, and visualizing results in a structured manner.

This project demonstrates how structured query language (SQL) can be applied to real-world datasets to uncover insights, support decision-making, and highlight sales trends in the food industry. By combining schema design, query optimization, and analytical functions, the pizza sales dataset becomes a powerful resource for understanding customer preferences and business performance.

Pizza Resto

ABOUT DATABASE

Database Description

The Pizza Sales Database is designed to capture and analyze customer orders, product details, and sales performance. It consists of four interconnected tables that together provide a comprehensive view of transactions and product information.

Tables Overview

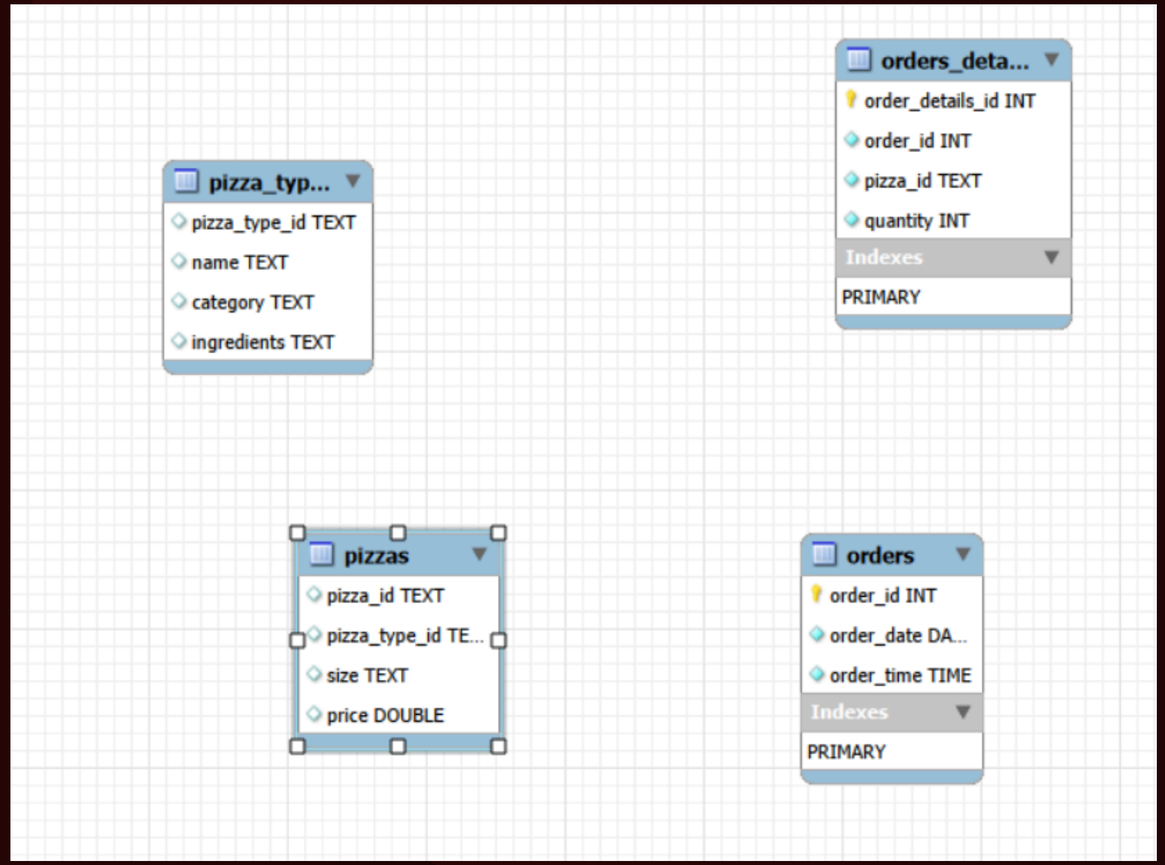
- Orders:
  - Attributes: order_id, order_date, order_time
  - Purpose: Stores information about each customer order, including the unique identifier, the date of purchase, and the time of transaction.
- Order_Details:
  - Attributes: order_detail_id, order_id, pizza_id, quantity
  - Purpose: Represents the line items of each order. It links specific pizzas to an order and records the quantity purchased.
- Pizza_Types:
  - Attributes: pizza_type_id, name, category, ingredients
  - Purpose: Contains descriptive information about each pizza type, including its name, category (e.g., vegetarian, non-vegetarian), and list of ingredients.
- Pizzas:
  - Attributes: pizza_id, pizza_type_id, size, prize
  - Purpose: Defines the available pizzas by linking them to their type, specifying the size (small, medium, large, etc.), and recording the price.

Relationships

- Orders ⇔ Order_Details: Connected through order_id, allowing each order to have multiple pizzas.
- Order_Details ⇔ Pizzas: Linked via pizza_id, ensuring each detail corresponds to a specific pizza.
- Pizzas ⇔ Pizza_Types: Connected through pizza_type_id, associating each pizza with its broader category and ingredients.

Pizza Resto

EER MODEL



1.
Pizza Resto

RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

Home About Contact

```
select count(order_id) as total_orders from orders;
```

Result Grid

total_orders
21350

2.
Pizza Resto

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES


```sql
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid

	total_sales
▶	817860.05

3.
Pizza Resto

IDENTIFY THE HIGHEST PRICED PIZZA

```sql
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;
```

Result Grid | Filter Rows

name	price
The Greek Pizza	35.95

4.
Pizza Resto

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid

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

5.
Pizza Resto

list the top 5 most ordered pizza types along with their quantities

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

$30

OUTPUT

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

6.
Pizza Resto

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.Pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

OUTPUT

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7.
Pizza Resto

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY (order_time);
```

OUTPUT

HOUR(order_time)	order_count
11	2
11	1
12	1
12	3
12	1
12	1
12	1
12	1
12	1
12	1
13	2
13	1

8.

join relevent tables to find the category Wize distribution of pizzas

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

9.

Group the orders by date and calculate the avg number of pizzas ordered per day

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza
▶	138

10.

Determine the top 3 most ordered pizza types based on revenue

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

OUTPUT

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 of total revenue.

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

category	revenue
Classic	26.9059602566967
Supreme	25.4563126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

12. Analyze the cumulative revenue generated over time

```sql
SELECT
    sales.order_date,
    SUM(sales.revenue) OVER (ORDER BY sales.order_date) AS cum_revenue
FROM (
    SELECT
        orders.order_date,
        SUM(orders_details.quantity * pizzas.price) AS revenue
    FROM orders_details
    JOIN pizzas
        ON orders_details.pizza_id = pizzas.pizza_id
    JOIN orders
        ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date
) AS sales;
```

Result Grid | Filter Rows:

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
2015-01-10	23990.350000000002
2015-01-11	25862.65

13.

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

```
select name,revenue from (select category,name,revenue,rank() over(partition by category order by revenue  desc) as rn
from
    (select pizza_types.category,pizza_types.name,
           sum((orders_details.quantity)*pizzas.price)as revenue
from
     Pizza_types
     join
     pizzas
     on pizza_types.pizza_type_id = pizzas.pizza_type_id
 join
 orders_details
 on  orders_details.pizza_id=pizzas.pizza_id
 group by pizza_types.category,pizza_types.name)as a) as b
 where rn<=3;
```

OUTPUT

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	32265.70000000065
The Mexicana Pizza	26780.75

THANK YOU