MySQL Project: Zomato Database

Step 1: Database Schema Design

Step 2: Creating Tables

1. Users Table
   -Stores information about the users of the app.

2. Restaurants Table

   -Stores details of the restaurants listed on the platform.

```
mysql> CREATE DATABASE zomato;
Query OK, 1 row affected (0.00 sec)

mysql> USE zomato;
Database changed
mysql> CREATE TABLE Users (
    ->     user_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     username VARCHAR(50) NOT NULL,
    ->     email VARCHAR(100) NOT NULL UNIQUE,
    ->     phone VARCHAR(15),
    ->     address TEXT
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Restaurants (
    ->     restaurant_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     name VARCHAR(100) NOT NULL,
    ->     location VARCHAR(100),
    ->     rating FLOAT CHECK (rating >= 0 AND rating <= 5),
    ->     cuisine VARCHAR(50),
    ->     contact VARCHAR(15)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

3. Menu_Items Table
   -Stores information about the menu items offered by each restaurant.

4. Orders Table

   -Manages information about customer orders.

```
mysql> CREATE TABLE Menu_Items (
    ->     item_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     restaurant_id INT,
    ->     item_name VARCHAR(100) NOT NULL,
    ->     price DECIMAL(8, 2) NOT NULL,
    ->     description TEXT,
    ->     FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE Orders (
    ->     order_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     user_id INT,
    ->     restaurant_id INT,
    ->     order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->     status ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',
    ->     total_price DECIMAL(8, 2),
    ->     FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    ->     FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)
```

4.Order_Items Table

-Manages individual items in an order(**many-to-many relationship** between orders and Menu_Items).

```
mysql> CREATE TABLE Order_Items (
    ->     order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     order_id INT,
    ->     item_id INT,
    ->     quantity INT DEFAULT 1,
    ->     price DECIMAL(8, 2),
    ->     FOREIGN KEY (order_id) REFERENCES Orders(order_id) ON DELETE CASCADE,
    ->     FOREIGN KEY (item_id) REFERENCES Menu_Items(item_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.03 sec)
```

-- 6. Reviews Table

-Stores reviews from users for restaurants.

```
mysql> CREATE TABLE Reviews (
    ->     review_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     user_id INT,
    ->     restaurant_id INT,
    ->     rating INT CHECK (rating >= 0 AND rating <= 5),
    ->     comments TEXT,
    ->     review_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->     FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE CASCADE,
    ->     FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.04 sec)
```

Step 3: Inserting Sample Data

Add sample data to the tables to test functionality.

Users Table

```
mysql> INSERT INTO Users (username, email, phone, address) VALUES
    -> ('Vasudha', 'vasudha@example.com', '1234512345', '101 Tech St'),
    -> ('Neha', 'neha@example.com', '5432154321', '202 Tech Ave'),
    -> ('Mahi', 'mahi@example.com', '9876598765', '303 Developer Rd'),
    -> ('Jeeva', 'jeeva@example.com', '6543265432', '404 Code Ln'),
    -> ('Sahana', 'sahana@example.com', '3214321432', '505 Debug Blvd');
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from Users;
+---------+----------+---------------------+------------+------------------+
| user_id | username | email               | phone      | address          |
+---------+----------+---------------------+------------+------------------+
|       1 | Vasudha  | vasudha@example.com | 1234512345 | 101 Tech St      |
|       2 | Neha     | neha@example.com    | 5432154321 | 202 Tech Ave     |
|       3 | Mahi     | mahi@example.com    | 9876598765 | 303 Developer Rd |
|       4 | Jeeva    | jeeva@example.com   | 6543265432 | 404 Code Ln      |
|       5 | Sahana   | sahana@example.com  | 3214321432 | 505 Debug Blvd   |
+---------+----------+---------------------+------------+------------------+
5 rows in set (0.00 sec)
```

Restaurants Table

```
mysql> INSERT INTO Restaurants (name, location, rating, cuisine, contact) VALUES
    -> ('Pizza Palace', 'Downtown', 4.5, 'Italian', '1112223333'),
    -> ('Sushi World', 'Uptown', 4.8, 'Japanese', '2223334444'),
    -> ('Burger Hub', 'Midtown', 4.2, 'American', '3334445555'),
    -> ('Curry House', 'Oldtown', 4.6, 'Indian', '4445556666'),
    -> ('Taco Land', 'Southside', 4.3, 'Mexican', '5556667777');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from Restaurants;
+---------------+-------------+-----------+--------+----------+------------+
| restaurant_id | name        | location  | rating | cuisine  | contact    |
+---------------+-------------+-----------+--------+----------+------------+
|             1 | Pizza Palace | Downtown  |    4.5 | Italian  | 1112223333 |
|             2 | Sushi World  | Uptown    |    4.8 | Japanese | 2223334444 |
|             3 | Burger Hub   | Midtown   |    4.2 | American | 3334445555 |
|             4 | Curry House  | Oldtown   |    4.6 | Indian   | 4445556666 |
|             5 | Taco Land    | Southside |    4.3 | Mexican  | 5556667777 |
+---------------+-------------+-----------+--------+----------+------------+
5 rows in set (0.00 sec)
```

Menu_Items Table

```
mysql> INSERT INTO Menu_Items (restaurant_id, item_name, price, description) VALUES
    -> (1, 'Margherita Pizza', 9.99, 'Classic cheese and tomato pizza'),
    -> (1, 'Pepperoni Pizza', 12.99, 'Pepperoni pizza with mozzarella'),
    -> (2, 'Salmon Sushi', 15.99, 'Fresh salmon nigiri sushi'),
    -> (2, 'Tuna Sushi', 14.99, 'Fresh tuna nigiri sushi'),
    -> (3, 'Cheeseburger', 8.99, 'Juicy beef burger with cheese');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from Menu_Items;
+---------+---------------+------------------+-------+---------------------------------+
| item_id | restaurant_id | item_name        | price | description                     |
+---------+---------------+------------------+-------+---------------------------------+
|       1 |             1 | Margherita Pizza |  9.99 | Classic cheese and tomato pizza |
|       2 |             1 | Pepperoni Pizza  | 12.99 | Pepperoni pizza with mozzarella |
|       3 |             2 | Salmon Sushi     | 15.99 | Fresh salmon nigiri sushi       |
|       4 |             2 | Tuna Sushi       | 14.99 | Fresh tuna nigiri sushi         |
|       5 |             3 | Cheeseburger     |  8.99 | Juicy beef burger with cheese   |
+---------+---------------+------------------+-------+---------------------------------+
5 rows in set (0.00 sec)
```

Orders Table

```
mysql> INSERT INTO Orders (user_id, restaurant_id, total_price, status) VALUES
    -> (1, 1, 22.98, 'Completed'),
    -> (2, 2, 30.98, 'Pending'),
    -> (3, 3, 8.99, 'Completed'),
    -> (4, 4, 50.00, 'Pending'),
    -> (5, 5, 15.00, 'Cancelled');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> desc Orders;
+---------------+-----------------------------------------+------+-----+-------------------+----------------+
| Field         | Type                                    | Null | Key | Default           | Extra          |
+---------------+-----------------------------------------+------+-----+-------------------+----------------+
| order_id      | int(11)                                 | NO   | PRI | NULL              | auto_increment |
| user_id       | int(11)                                 | YES  | MUL | NULL              |                |
| restaurant_id | int(11)                                 | YES  | MUL | NULL              |                |
| order_date    | datetime                                | YES  |     | CURRENT_TIMESTAMP |                |
| status        | enum('Pending','Completed','Cancelled') | YES  |     | Pending           |                |
| total_price   | decimal(8,2)                            | YES  |     | NULL              |                |
+---------------+-----------------------------------------+------+-----+-------------------+----------------+
6 rows in set (0.00 sec)
```

```
mysql> select * from Orders;
+----------+---------+---------------+---------------------+-----------+-------------+
| order_id | user_id | restaurant_id | order_date          | status    | total_price |
+----------+---------+---------------+---------------------+-----------+-------------+
|        1 |       1 |             1 | 2025-01-24 20:58:14 | Completed |       22.98 |
|        2 |       2 |             2 | 2025-01-24 20:58:14 | Pending   |       30.98 |
|        3 |       3 |             3 | 2025-01-24 20:58:14 | Completed |        8.99 |
|        4 |       4 |             4 | 2025-01-24 20:58:14 | Pending   |       50.00 |
|        5 |       5 |             5 | 2025-01-24 20:58:14 | Cancelled |       15.00 |
+----------+---------+---------------+---------------------+-----------+-------------+
5 rows in set (0.00 sec)
```

Reviews Table

```
mysql> INSERT INTO Reviews (user_id, restaurant_id, rating, comments) VALUES
    -> (1, 1, 5, 'Amazing pizza! Highly recommend.'),
    -> (2, 2, 4, 'Great sushi, but a bit pricey.'),
    -> (3, 3, 3, 'Burger was average.'),
    -> (4, 4, 5, 'Fantastic curry!'),
    -> (5, 5, 4, 'Loved the tacos!');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> desc Reviews;
+---------------+----------+------+-----+-------------------+----------------+
| Field         | Type     | Null | Key | Default           | Extra          |
+---------------+----------+------+-----+-------------------+----------------+
| review_id     | int(11)  | NO   | PRI | NULL              | auto_increment |
| user_id       | int(11)  | YES  | MUL | NULL              |                |
| restaurant_id | int(11)  | YES  | MUL | NULL              |                |
| rating        | int(11)  | YES  |     | NULL              |                |
| comments      | text     | YES  |     | NULL              |                |
| review_date   | datetime | YES  |     | CURRENT_TIMESTAMP |                |
+---------------+----------+------+-----+-------------------+----------------+
5 rows in set (0.00 sec)

mysql> select * from Reviews;
+-----------+---------+---------------+--------+---------------------------------+---------------------+
| review_id | user_id | restaurant_id | rating | comments                        | review_date         |
+-----------+---------+---------------+--------+---------------------------------+---------------------+
|         1 |       1 |             1 |      5 | Amazing pizza! Highly recommend. | 2025-01-24 21:00:56 |
|         2 |       2 |             2 |      4 | Great sushi, but a bit pricey.   | 2025-01-24 21:00:56 |
|         3 |       3 |             3 |      3 | Burger was average.              | 2025-01-24 21:00:56 |
|         4 |       4 |             4 |      5 | Fantastic curry!                 | 2025-01-24 21:00:56 |
|         5 |       5 |             5 |      4 | Loved the tacos!                 | 2025-01-24 21:00:56 |
+-----------+---------+---------------+--------+---------------------------------+---------------------+
5 rows in set (0.00 sec)
```

Step 4: Writing Queries

1. Get all restaurants and their average rating

```
mysql> SELECT Restaurants.name, Restaurants.location, AVG(Reviews.rating) AS average_rating
    -> FROM Restaurants
    -> JOIN Reviews ON Restaurants.restaurant_id = Reviews.restaurant_id
    -> GROUP BY Restaurants.restaurant_id;
+--------------+-----------+----------------+
| name         | location  | average_rating |
+--------------+-----------+----------------+
| Pizza Palace | Downtown  |         5.0000 |
| Sushi World  | Uptown    |         4.0000 |
| Burger Hub   | Midtown   |         3.0000 |
| Curry House  | Oldtown   |         5.0000 |
| Taco Land    | Southside |         4.0000 |
+--------------+-----------+----------------+
5 rows in set (0.01 sec)
```

2. Find all menu items of a restaurant

```
mysql> SELECT item_name, price, description
    -> FROM Menu_Items
    -> WHERE restaurant_id = 1;
+-------------------+-------+---------------------------------+
| item_name         | price | description                     |
+-------------------+-------+---------------------------------+
| Margherita Pizza  |  9.99 | Classic cheese and tomato pizza |
| Pepperoni Pizza   | 12.99 | Pepperoni pizza with mozzarella |
+-------------------+-------+---------------------------------+
2 rows in set (0.00 sec)
```

3. Retrieve all orders by a user

```
mysql> SELECT Orders.order_id, Restaurants.name AS restaurant_name, Orders.total_price, Orders.status
    -> FROM Orders
    -> JOIN Restaurants ON Orders.restaurant_id = Restaurants.restaurant_id
    -> WHERE Orders.user_id = 1;
+----------+-----------------+-------------+-----------+
| order_id | restaurant_name | total_price | status    |
+----------+-----------------+-------------+-----------+
|        1 | Pizza Palace    |       22.98 | Completed |
+----------+-----------------+-------------+-----------+
1 row in set (0.01 sec)
```

4. Display all reviews for a restaurant

```
mysql> SELECT Users.username, Reviews.rating, Reviews.comments
    -> FROM Reviews
    -> JOIN Users ON Reviews.user_id = Users.user_id
    -> WHERE restaurant_id = 2;
+----------+--------+--------------------------------+
| username | rating | comments                       |
+----------+--------+--------------------------------+
| Neha     |      4 | Great sushi, but a bit pricey. |
+----------+--------+--------------------------------+
1 row in set (0.00 sec)
```

5.Calculate total earnings for each restaurant

```
mysql> SELECT Restaurants.name, SUM(Orders.total_price) AS total_earnings
    -> FROM Orders
    -> JOIN Restaurants ON Orders.restaurant_id = Restaurants.restaurant_id
    -> WHERE Orders.status = 'Completed'
    -> GROUP BY Restaurants.restaurant_id;
+--------------+----------------+
| name         | total_earnings |
+--------------+----------------+
| Pizza Palace |          22.98 |
| Burger Hub   |           8.99 |
+--------------+----------------+
2 rows in set (0.00 sec)
```