

MySQL Project: Zomato Database

Step 1: Database Schema Design

Create a database named `zomato`:

```
CREATE DATABASE zomato;  
USE zomato;
```

Step 2: Creating Tables

1. Users Table

- Stores information about the users of the app.

```
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  
);
```

```
[mysql> select * from users;  
+-----+-----+-----+-----+-----+  
| user_id | username      | email              | phone      | address      |  
+-----+-----+-----+-----+-----+  
| 101 | Jignesh Borse | Jignesh@gmail.com | 9373305566 | Vadgaon      |  
| 102 | Prajwal Mohite | Prajwal@gmail.com | 9876543210 | Kothrud      |  
| 103 | Gaurav Pawar  | Gaurav@gmail.com  | 9988776655 | Baner        |  
| 104 | Rohit Savale  | Rohit@gmail.com   | 9123456789 | Hinjewadi    |  
| 105 | Anubhaw Mishra | Anubhaw@gmail.com | 7894561230 | Wakad        |  
| 106 | Mitesh Shinde | Mitesh@gmail.com  | 8569741230 | Aundh        |  
| 107 | Aman Kumar    | Aman@gmail.com    | 9871234560 | Pimpri       |  
| 108 | Yash Baad     | Yash@gmail.com    | 9356789123 | Chinchwad    |  
| 109 | Zayed Bakoda  | Zayed@gmail.com   | 9876540987 | Vadgaon      |  
| 110 | Jay Borse     | Jay@gmail.com     | 9373305567 | Viman Nagar  |  
+-----+-----+-----+-----+-----+  
10 rows in set (0.00 sec)
```

2. Restaurants Table

- Stores details of the restaurants listed on the platform.

```
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)  
);
```

```
mysql> select * from restaurants;
```

restaurant_id	name	location	rating	cuisine	contact
1	The Food Court	Kothrud	4.5	Indian	9876543210
2	Spice Symphony	Baner	4.2	Chinese	9988776655
3	Urban Tandoor	Hinjewadi	4.8	Mughlai	9123456789
4	Wakad Bistro	Wakad	4.1	Continental	7894561230
5	Aroma Dine	Aundh	3.9	Italian	8569741230
6	Biryani Junction	Pimpri	4.7	Indian	9871234560
7	Chaat Chowk	Chinchwad	4.3	Street Food	9356789123
8	Vadgaon Delights	Vadgaon	4	South Indian	9876540987
9	Noodle Nest	Viman Nagar	4.4	Chinese	9373305567
10	Taco Time	Hadapsar	4.6	Mexican	9765432109

```
10 rows in set (0.00 sec)
```

3. Menu_Items Table

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

```
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  
);
```

```
[mysql> select * from Menu_Items;  
+-----+-----+-----+-----+-----+  
| item_id | restaurant_id | item_name | price | description |  
+-----+-----+-----+-----+-----+  
| 1 | 1 | Butter Chicken | 350.00 | Rich chicken curry. |  
| 2 | 2 | Hakka Noodles | 200.00 | Stir-fried noodles. |  
| 3 | 3 | Paneer Tikka | 280.00 | Grilled paneer cubes. |  
| 4 | 4 | Grilled Chicken Salad | 300.00 | Fresh chicken salad. |  
| 5 | 5 | Margherita Pizza | 450.00 | Classic cheese pizza. |  
| 6 | 6 | Hyderabadi Biryani | 400.00 | Flavored biryani rice. |  
| 7 | 7 | Pani Puri | 50.00 | Crispy and tangy snack. |  
| 8 | 8 | Masala Dosa | 120.00 | Crispy rice crepe. |  
| 9 | 9 | Chicken Manchurian | 320.00 | Chicken in tangy sauce. |  
| 10 | 10 | Beef Tacos | 250.00 | Spicy Mexican tacos. |  
+-----+-----+-----+-----+-----+  
10 rows in set (0.01 sec)
```

4. Orders Table

- Manages information about customer orders.

```
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  
);
```

```
[mysql> select * from orders;
```

order_id	user_id	restaurant_id	order_date	status	total_price
1	101	1	2025-01-01 12:30:00	Completed	750.00
2	102	2	2025-01-02 18:45:00	Pending	400.00
3	103	3	2025-01-03 14:20:00	Cancelled	500.00
4	104	4	2025-01-04 19:00:00	Completed	850.00
5	105	5	2025-01-05 13:10:00	Pending	450.00
6	106	6	2025-01-06 20:15:00	Completed	1200.00
7	107	7	2025-01-07 16:30:00	Completed	300.00
8	108	8	2025-01-08 11:45:00	Cancelled	200.00
9	109	9	2025-01-09 17:20:00	Pending	700.00
10	110	10	2025-01-10 15:00:00	Completed	950.00

```
10 rows in set (0.00 sec)
```

5. Order_Items Table

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

```
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  
);
```

```
[mysql> select * from order_items;
```

order_item_id	order_id	item_id	quantity	price
1	1	1	2	700.00
2	2	2	1	200.00
3	3	3	3	840.00
4	4	4	2	600.00
5	5	5	1	450.00
6	6	6	2	800.00
7	7	7	4	200.00
8	8	8	1	120.00
9	9	9	2	640.00
10	10	10	3	750.00

```
10 rows in set (0.00 sec)
```

6. Reviews Table

- Stores reviews from users for restaurants.

```
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  
);
```

```
mysql> select * from reviews;
```

review_id	user_id	restaurant_id	rating	comments	review_date
1	101	1	5	Amazing food, will definitely come back!	2025-01-26 23:29:18
2	102	2	4	Delicious noodles, but could use a bit more spice.	2025-01-26 23:29:18
3	103	3	3	Good, but not as flavorful as expected.	2025-01-26 23:29:18
4	104	4	4	Great grilled chicken, but the service was slow.	2025-01-26 23:29:18
5	105	5	5	Best pizza ever, loved the crust and toppings!	2025-01-26 23:29:18
6	106	6	4	Authentic biryani, very tasty and fulfilling.	2025-01-26 23:29:18
7	107	7	2	The pani puri was too soggy and lacked flavor.	2025-01-26 23:29:18
8	108	8	5	Perfect dosa, crispy and perfectly spiced.	2025-01-26 23:29:18
9	109	9	4	Great chicken manchurian, loved the tangy sauce.	2025-01-26 23:29:18
10	110	10	5	Best tacos in Pune! Fresh and full of flavor.	2025-01-26 23:29:18

```
10 rows in set (0.00 sec)
```

Step 3: Inserting Sample Data

Add sample data to the tables to test functionality.

Users Table

```
INSERT INTO Users (username, email, phone, address) VALUES
('Jignesh Borse', 'Jignesh@gmail.com', '9373305566', 'Vadgaon'),
('Prajwal Mohite', 'Prajwal@gmail.com', '9373305567', 'Kothrud'),
('Gaurav Pawar', 'Gaurav@gmail.com', '9373305568', 'Baner'),
('Rohit Savale', 'Rohit@gmail.com', '9373305569', 'Viman Nagar'),
('Anubhaw Mishra', 'Anubhaw@gmail.com', '9373305570', 'Hinjewadi'),
('Mitesh Shinde', 'Mitesh@gmail.com', '9373305571', 'Pimpri'),
('Aman Kumar', 'Aman@gmail.com', '9373305572', 'Wakad'),
('Yash Baad', 'Yash@gmail.com', '9373305573', 'Hadapsar'),
('Zayed Bakoda', 'Zayed@gmail.com', '9373305574', 'Aundh'),
('Shivam Yadav', 'Shivam@gmail.com', '9373305575', 'Kharadi');
```

Restaurants Table

```
INSERT INTO Restaurants (name, location, rating, cuisine, contact) VALUES
('The Pizza Place', 'Vadgaon', 4.5, 'Italian', '9373305566'),
('Sushi House', 'Kothrud', 4.0, 'Japanese', '9373305567'),
('Tandoori Nights', 'Baner', 3.5, 'Indian', '9373305568'),
('Grill Master', 'Viman Nagar', 4.2, 'Continental', '9373305569'),
('Pizza Corner', 'Hinjewadi', 5.0, 'Italian', '9373305570'),
('Biryani Paradise', 'Pimpri', 4.3, 'Indian', '9373305571'),
('Chaat Junction', 'Wakad', 3.8, 'Street Food', '9373305572'),
('Dosa Delight', 'Hadapsar', 4.7, 'South Indian', '9373305573'),
('The Chinese Kitchen', 'Aundh', 4.1, 'Chinese', '9373305574'),
('Taco Bell', 'Kharadi', 4.6, 'Mexican', '9373305575');
```

Menu_Items Table

```
INSERT INTO Menu_Items (restaurant_id, item_name, price, description)
VALUES
(1, 'Butter Chicken', 350.00, 'Rich chicken curry'),
(2, 'Hakka Noodles', 200.00, 'Stir-fried noodles'),
(3, 'Paneer Tikka', 280.00, 'Grilled paneer cubes'),
(4, 'Grilled Chicken Salad', 300.00, 'Fresh chicken salad'),
(5, 'Margherita Pizza', 450.00, 'Classic cheese pizza'),
(6, 'Hyderabadi Biryani', 400.00, 'Flavored biryani rice'),
(7, 'Pani Puri', 50.00, 'Crispy and tangy snack'),
(8, 'Masala Dosa', 120.00, 'Crispy rice crepe'),
(9, 'Chicken Manchurian', 320.00, 'Chicken in tangy sauce'),
(10, 'Beef Tacos', 250.00, 'Spicy Mexican tacos');
```

Orders Table

```
INSERT INTO Orders (user_id, restaurant_id, order_date, status,
total_price) VALUES
(1, 1, '2025-01-01', 'Completed', 750.00),
(2, 2, '2025-01-02', 'Pending', 400.00),
(3, 3, '2025-01-03', 'Cancelled', 500.00),
(4, 4, '2025-01-04', 'Completed', 850.00),
(5, 5, '2025-01-05', 'Pending', 450.00),
(6, 6, '2025-01-06', 'Completed', 1200.00),
(7, 7, '2025-01-07', 'Completed', 300.00),
(8, 8, '2025-01-08', 'Cancelled', 200.00),
(9, 9, '2025-01-09', 'Pending', 700.00),
(10, 10, '2025-01-10', 'Completed', 950.00);
```

Order_Items Table

```
INSERT INTO Order_Items (order_id, item_id, quantity, price) VALUES
(1, 1, 2, 700.00),
(2, 2, 1, 200.00),
(3, 3, 3, 840.00),
(4, 4, 2, 600.00),
(5, 5, 1, 450.00),
(6, 6, 2, 800.00),
(7, 7, 4, 200.00),
(8, 8, 1, 120.00),
(9, 9, 2, 640.00),
(10, 10, 3, 750.00);
```

Reviews Table

```
INSERT INTO Order_Items (order_id, item_id, quantity, price) VALUES
(1, 1, 2, 700.00),
(2, 2, 1, 200.00),
(3, 3, 3, 840.00),
(4, 4, 2, 600.00),
(5, 5, 1, 450.00),
(6, 6, 2, 800.00),
(7, 7, 4, 200.00),
(8, 8, 1, 120.00),
(9, 9, 2, 640.00),
(10, 10, 3, 750.00);
```

Step 4: Writing Queries

1. Get all restaurants and their average rating:

```
SELECT name, location, AVG(rating) AS average_rating
FROM Restaurants
JOIN Reviews ON Restaurants.restaurant_id = Reviews.restaurant_id
GROUP BY Restaurants.restaurant_id;
```

restaurant_name	average_rating
The Food Court	5.0000
Spice Symphony	4.0000
Urban Tandoor	3.0000
Wakad Bistro	4.0000
Aroma Dine	5.0000
Biryani Junction	4.0000
Chaat Chowk	2.0000
Vadgaon Delights	5.0000
Noodle Nest	4.0000
Taco Time	5.0000

10 rows in set (0.00 sec)

2. Find all menu items of a restaurant:

```
SELECT item_name, price, description
FROM Menu_Items
WHERE restaurant_id = 1;
```

item_name	price	description
Margherita Pizza	450.00	Classic cheese pizza.

1 row in set (0.00 sec)

3. Retrieve all orders by a user:

```
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	order_date	status	total_price	restaurant_name
1	2025-01-01 12:30:00	Completed	750.00	The Food Court

1 row in set (0.00 sec)

4. Display all reviews for a restaurant:

```
SELECT Users.username, Reviews.rating, Reviews.comments  
FROM Reviews  
JOIN Users ON Reviews.user_id = Users.user_id  
WHERE restaurant_id = 2;
```

username	restaurant_name	rating	comments	review_date
Jay Borse	Taco Time	5	Best tacos in Pune! Fresh and full of flavor.	2025-01-26 23:29:18

1 row in set (0.00 sec)

5. Calculate total earnings for each restaurant:

```
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;
```

restaurant_name	total_earnings
The Food Court	1400.00
Spice Symphony	200.00
Urban Tandoor	2520.00
Wakad Bistro	1200.00
Aroma Dine	450.00
Biryani Junction	1600.00
Chaat Chowk	800.00
Vadgaon Delights	120.00
Noodle Nest	1280.00
Taco Time	2250.00

10 rows in set (0.00 sec)