📌 **Database Schema: foodorder**

The database comprises the following collections, along with their respective schemas:

✅ 1. Users Collection

```
{
  "_id": ObjectId,
  "user_id": String, // Unique
  "name": String,
  "email": String,
  "phone": String,
  "address": [
    {
      "address_id": String,
      "house_no": String,
      "street": String,
      "city": String,
      "state": String,
      "pin_code": String,
      "landmark": String
    }
  ],
  "created_at": Date
}
```

✅ 2. Restaurants Collection

```
{
  "_id": ObjectId,
  "restaurant_id": String, // Unique
  "name": String,
  "location": {
    "type": String, // "Point"
    "coordinates": [Double, Double] // [Longitude, Latitude]
  },
  "address": {
    "street": String,
    "city": String,
    "state": String,
    "pin_code": String
  },
  "cuisines": [String],
  "rating": Double,
  "is_open": Boolean,
  "timings": String,
  "created_at": Date
}
```

✅ 3. FoodItems Collection

```
{
  "_id": ObjectId,
  "food_id": String, // Unique
  "restaurant_id": String, // Reference Restaurants.restaurant_id
  "name": String,
  "description": String,
  "price": Double,
  "category": String,
  "is_veg": Boolean,
  "available": Boolean
}
```

✅ 4. Orders Collection

```
{
  "_id": ObjectId,
  "order_id": String, // Unique
  "user_id": String, // Reference Users.user_id
  "restaurant_id": String, // Reference Restaurants.restaurant_id
  "food_items": [
    {
      "food_id": String, // Reference FoodItems.food_id
      "quantity": Int,
      "price": Double
    }
  ],
  "total_amount": Double,
  "order_status": String, // ["Pending", "Accepted", "Preparing", "Picked Up", "Delivered",
"Cancelled"]
  "payment": {
    "method": String, // ["Credit Card", "Debit Card", "UPI", "COD", etc.]
    "status": String // ["Paid", "Unpaid", "Failed", "Refunded"]
  },
  "delivery_address_id": String, // Reference Users.address.address_id
  "ordered_at": Date,
  "delivered_at": Date,
  "delivery_person_id": String // Reference DeliveryPersons.delivery_person_id
}
```

✅ 5. DeliveryPersons Collection

```
{
  "_id": ObjectId,
  "delivery_person_id": String, // Unique
  "name": String,
  "phone": String,
  "vehicle": {
    "type": String, // ["Bike", "Scooter", "Car", etc.]
```

```
   "registration_number": String
  },
  "current_location": {
   "type": String, // "Point"
   "coordinates": [Double, Double] // [Longitude, Latitude]
  },
  "availability_status": String, // ["Available", "On-Delivery", "Offline"]
  "created_at": Date
}
```

✅ 6. Reviews Collection

```
{
  "_id": ObjectId,
  "review_id": String, // Unique
  "order_id": String, // Reference Orders.order_id
  "user_id": String, // Reference Users.user_id
  "restaurant_id": String, // Reference Restaurants.restaurant_id
  "rating": Int, // Typically between 1 to 5
  "comment": String,
  "reviewed_at": Date
}
```

✅ 7. Payments Collection

```
{
  "_id": ObjectId,
  "payment_id": String, // Unique
  "order_id": String, // Reference Orders.order_id
  "user_id": String, // Reference Users.user_id
  "amount": Double,
  "method": String, // ["Credit Card", "Debit Card", "UPI", "COD"]
  "transaction_id": String,
```

"status": String, // ["Completed", "Pending", "Failed", "Refunded"]

"paid_at": Date

}

🛠️ **Indexes for Performance Optimization**

The following indexes improve database performance:

// Geospatial indexing

Restaurants: { location: "2dsphere" }

DeliveryPersons: { current_location: "2dsphere" }

// Unique indexing

Users: { user_id: 1 }, unique

Restaurants: { restaurant_id: 1 }, unique

FoodItems: { food_id: 1 }, unique

Orders: { order_id: 1 }, unique

DeliveryPersons: { delivery_person_id: 1 }, unique

Reviews: { review_id: 1 }, unique

Payments: { payment_id: 1 }, unique

🚩 **Relationships (References)**

Here are the relationships between collections:

Collection	**Field**	**References Collection.field**
FoodItems	restaurant_id	Restaurants.restaurant_id
Orders	user_id	Users.user_id
Orders	restaurant_id	Restaurants.restaurant_id
Orders	food_items.food_id	FoodItems.food_id
Orders	delivery_address_id	Users.address.address_id
Orders	delivery_person_id	DeliveryPersons.delivery_person_id
Reviews	order_id	Orders.order_id

Collection	Field	References Collection.field
Reviews	user_id	Users.user_id
Reviews	restaurant_id	Restaurants.restaurant_id
Payments	order_id	Orders.order_id
Payments	user_id	Users.user_id

📌 **Sample commands to create the schema:**

```
use swiggy_db;


db.createCollection("Users");

db.createCollection("Restaurants");

db.createCollection("FoodItems");

db.createCollection("Orders");

db.createCollection("DeliveryPersons");

db.createCollection("Reviews");

db.createCollection("Payments");


// Creating indexes

db.Restaurants.createIndex({location: "2dsphere"});

db.DeliveryPersons.createIndex({current_location: "2dsphere"});


db.Users.createIndex({user_id: 1}, {unique: true});

db.Restaurants.createIndex({restaurant_id: 1}, {unique: true});

db.FoodItems.createIndex({food_id: 1}, {unique: true});

db.Orders.createIndex({order_id: 1}, {unique: true});

db.DeliveryPersons.createIndex({delivery_person_id: 1}, {unique: true});

db.Reviews.createIndex({review_id: 1}, {unique: true});

db.Payments.createIndex({payment_id: 1}, {unique: true});
```

🚩 **Summary:**

- This schema provides clarity and ensures efficient data management.
- Collections are designed for scalability, maintaining clear relationships using references.
- Indexing significantly optimizes geospatial queries and improves search/query performance.