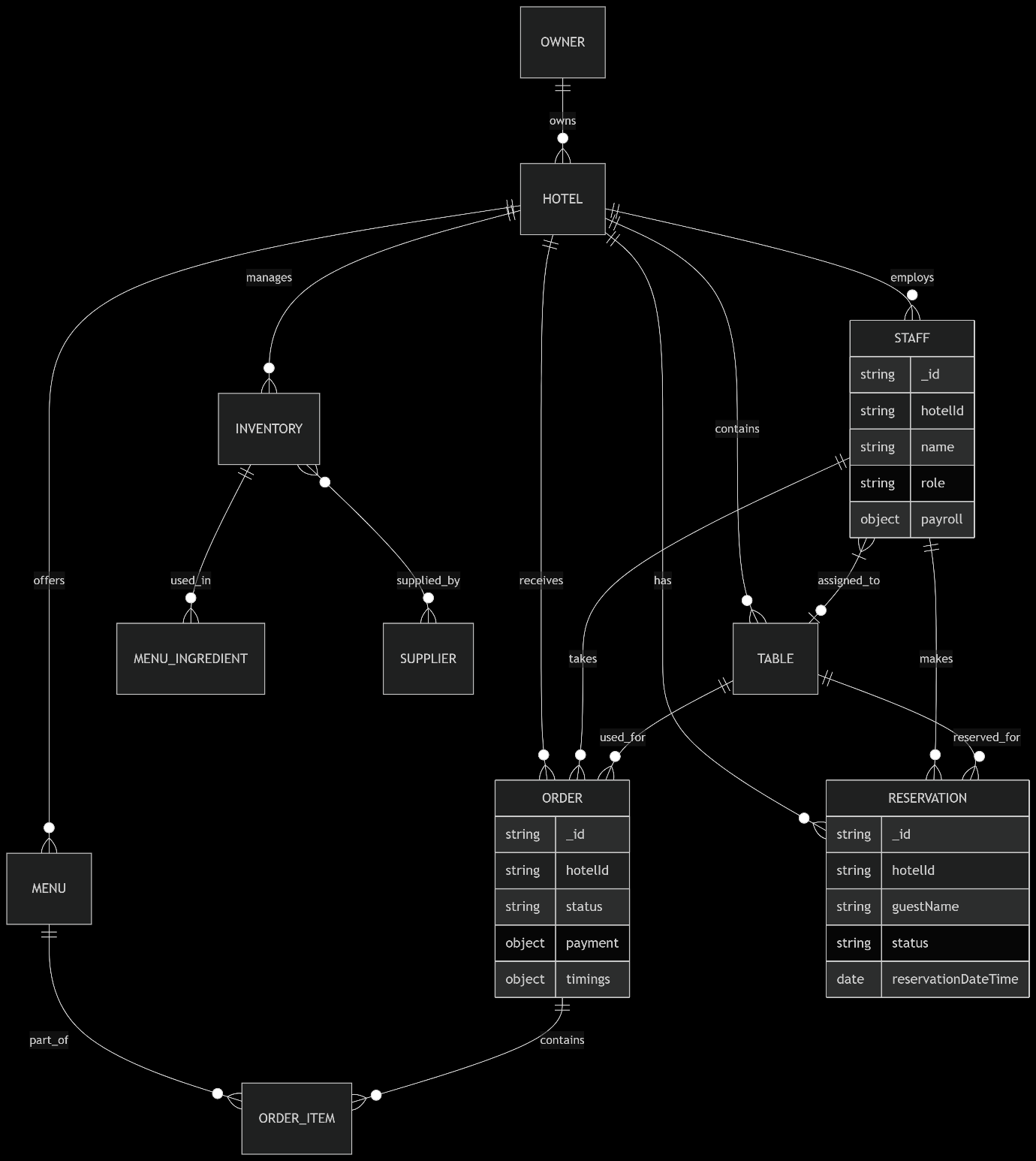
**Hotel Management System - Database Schema & Design Documentation**

**1. Entity-Relationship Diagram**



**2. Consolidated Data Schema Overview**

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Primary Key** | **Key Attributes (Simplified)** | **Foreign Key(s)** |
| **Hotel** | Entity | Primary Key | Key Attributes (Simplified) |
| **Owner** | Hotel | \_id | name, address, contact, settings, taxConfig |
| **Staff** | Owner | \_id | name, contact, address, status |
| **Table** | Staff | \_id | name, role, contact, hotelId, payroll, shift |
| **Inventory** | Table | \_id | tableNumber, capacity, status, section |
| **Menu** | Inventory | \_id | name, type, category, stockQuantity, reorderThreshold |
| **Reservation** | Menu | \_id | name, category, price, isAvailable, ingredients[] |
| **Order** | Reservation | \_id | guestName, guestContact, reservationDateTime, status, numberOfGuests |

**3. Data Flow & Core Business Processes**

The system follows a logical flow through these operational stages:

3.1 Setup & Management Phase

* **OWNER** entities own one or more **HOTEL** entities
* Each **HOTEL** employs multiple **STAFF** members
* Hotels manage **TABLES**, **INVENTORY**, and **MENU** items

3.2 Reservation & Seating Process

// Example: Creating a Reservation

{

"hotelId": "hotel\_123",

"tableId": "table\_10",

"guestName": "Jane Doe",

"madeBy": { "staffId": "staff\_reception\_456", ... },

"status": "confirmed",

"reservationDateTime": { "$date": "2023-10-27T19:30:00Z" }

}

3.3 Ordering & Preparation Process

// Example: Item in an Order

"items": [

{

"menuId": "menu\_pasta\_alfredo", // Links to Menu

"name": "Fettuccine Alfredo",

"price": { "$numberDouble": "16.99" },

"qty": 2

}

]

3.4 Inventory Management Process

// Example: Menu Item consuming Inventory

"ingredients": [

{ "name": "Fettuccine", "quantity": "200g", "supplierId": "supplier\_pasta\_co" },

{ "name": "Heavy Cream", "quantity": "100ml", "supplierId": "supplier\_dairy\_co" }

]

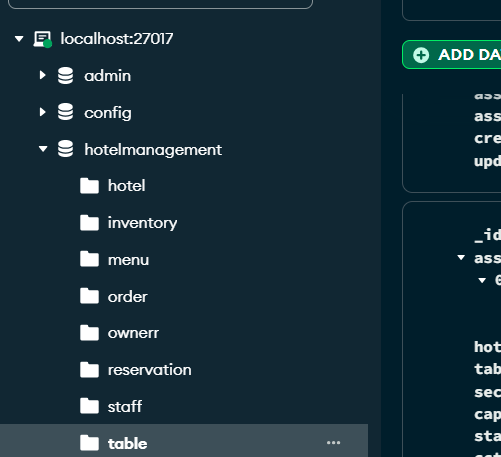
3.5 Payment & Completion Process

* Orders are marked as served then completed
* Payments are processed
* Table status returns to available

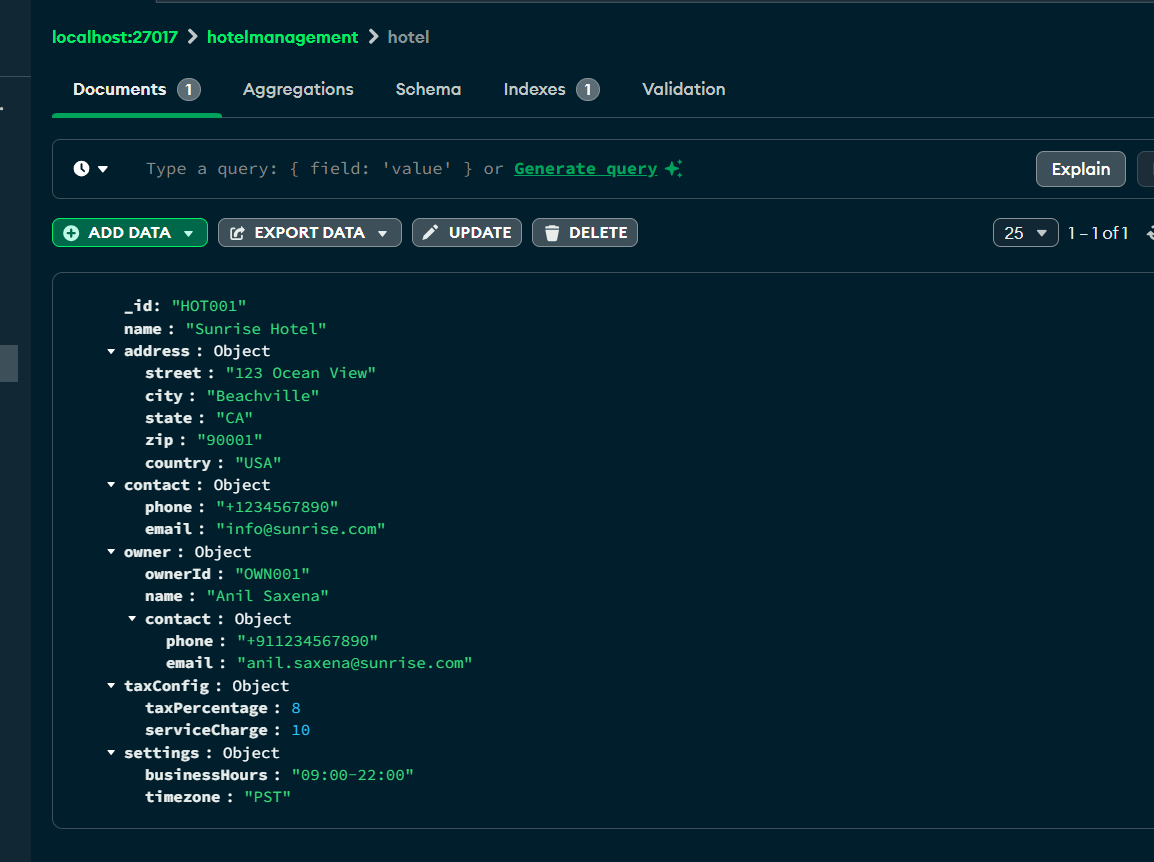
**4. Key Relationships and Dependencies**

|  |  |  |  |
| --- | --- | --- | --- |
| From Entity | To Entity | Relationship Description | Linking Field(s) |
| Staff | Hotel | Many-to-One: Many staff members work at one hotel. | staff.hotelId -> hotel.\_id |
| Table | Hotel | Many-to-One: Many tables belong to one hotel. | table.hotelId -> hotel.\_id |
| Order | Table | Many-to-One: An order is taken for a specific table. | order.tableId -> table.\_id |
| Order | Staff | Many-to-One: An order is taken by a specific staff member. | order.staffId -> staff.\_id |
| Order | Reservation | Many-to-One: An order can be linked to a reservation (optional). | order.reservationId -> reservation.\_id |
| Reservation | Staff | Many-to-One: A reservation is made by a staff member. | reservation.madeBy.staffId -> staff.\_id |
| Menu | Inventory | Many-to-Many: A menu item uses multiple inventory items. An inventory item is used in multiple menu items. | menu.ingredients[] ≈ inventory.name & inventory.\_id |
| From Entity | To Entity | Relationship Description | Linking Field(s) |

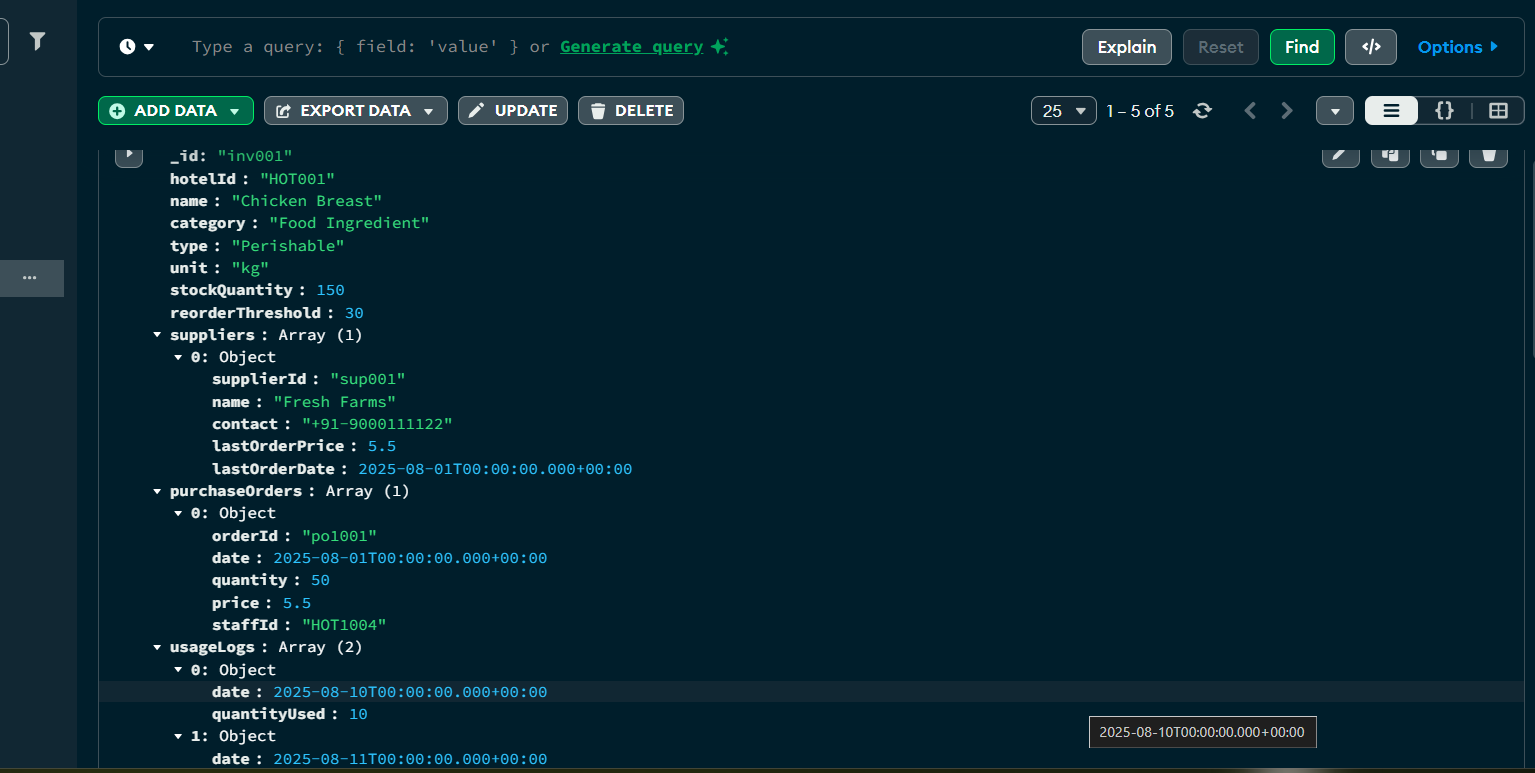
AS IT IS FOR MULTIPLE USER



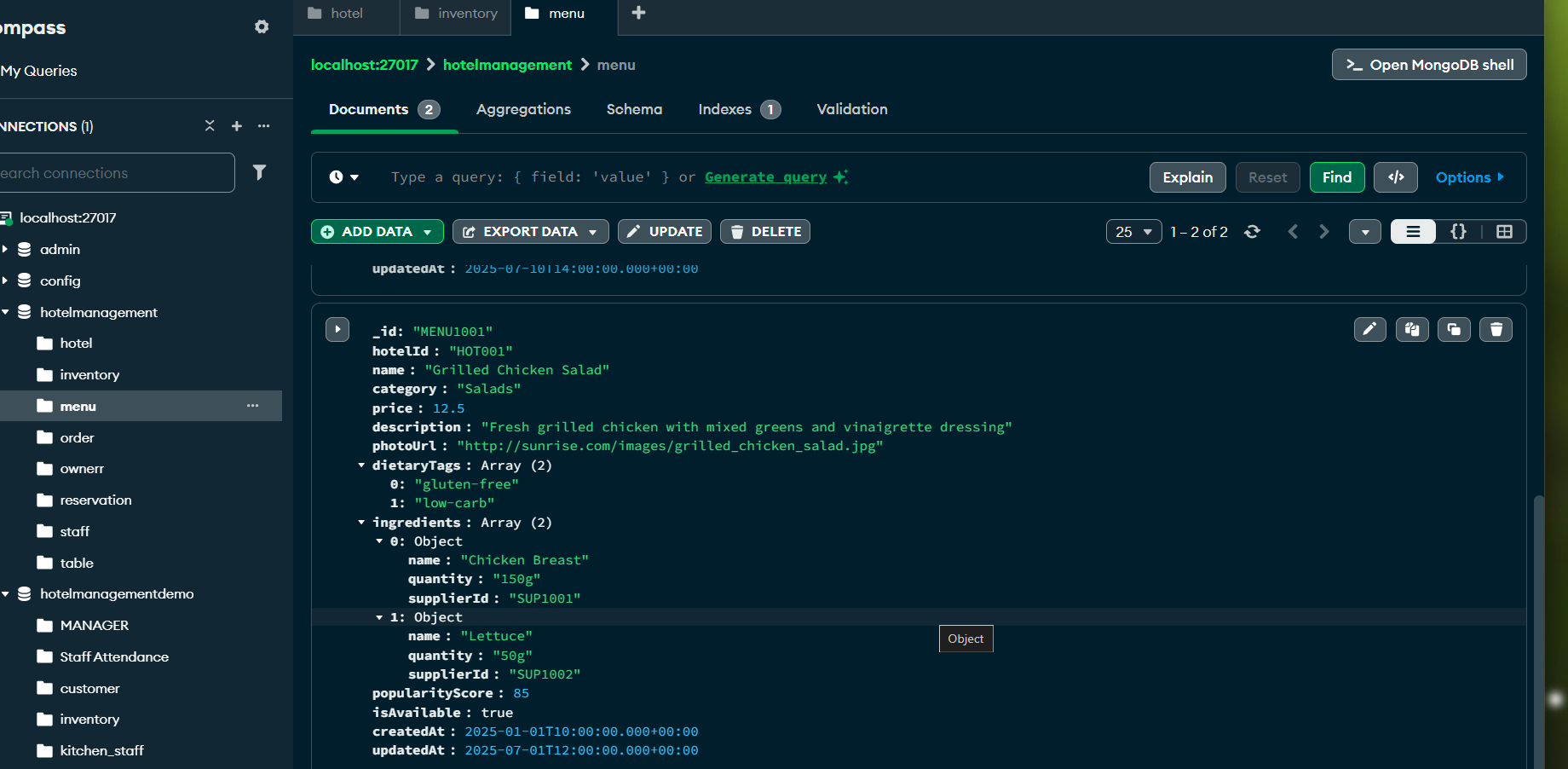
Hotel collection



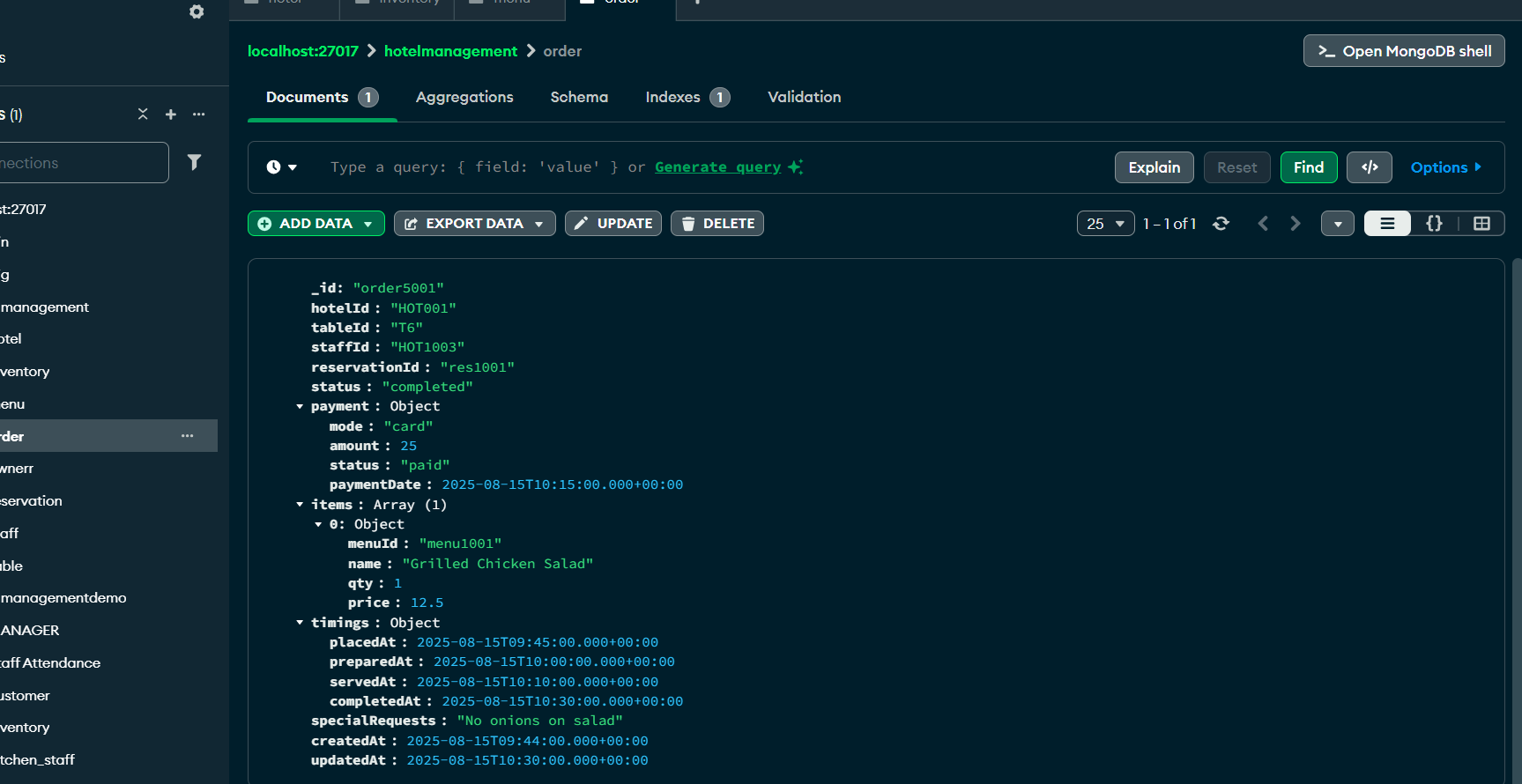
INVENTORY COLLECTION



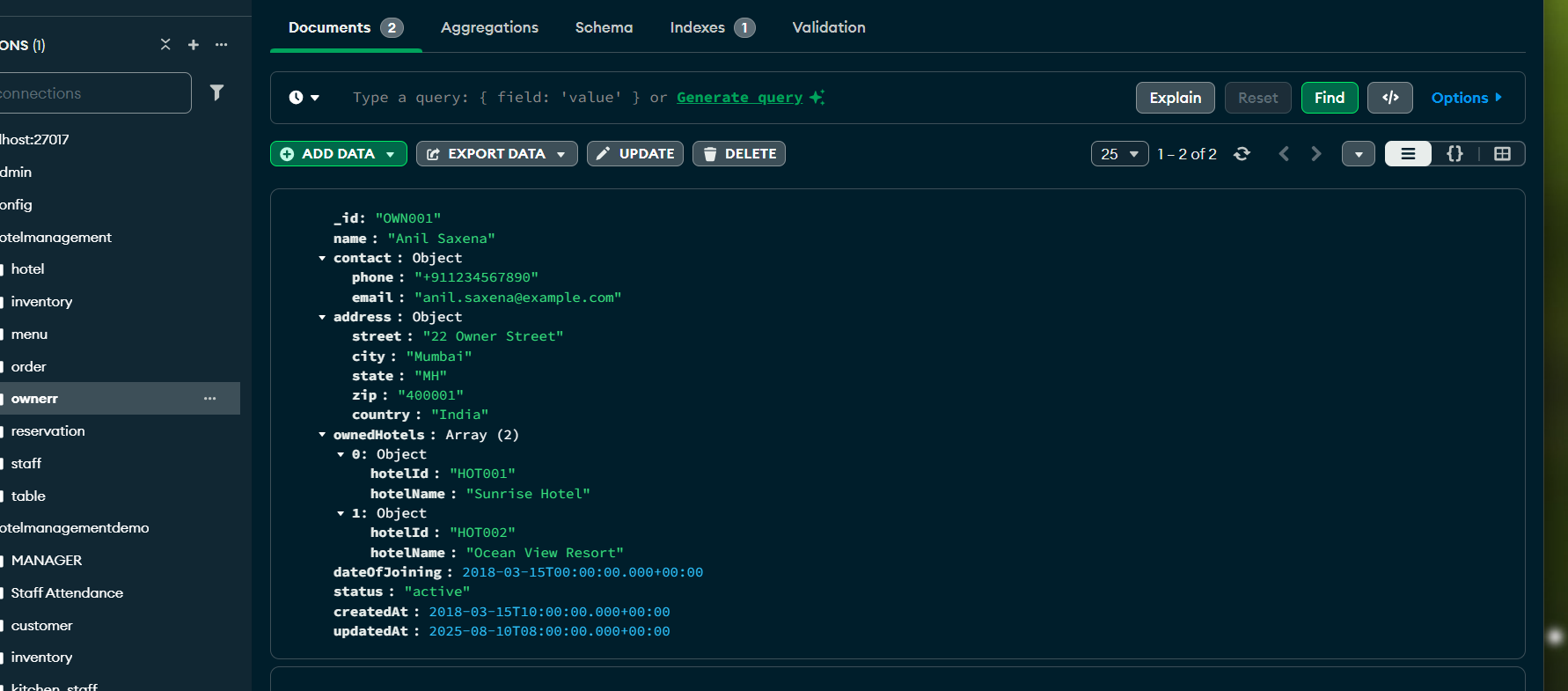
MENU COLLECTION



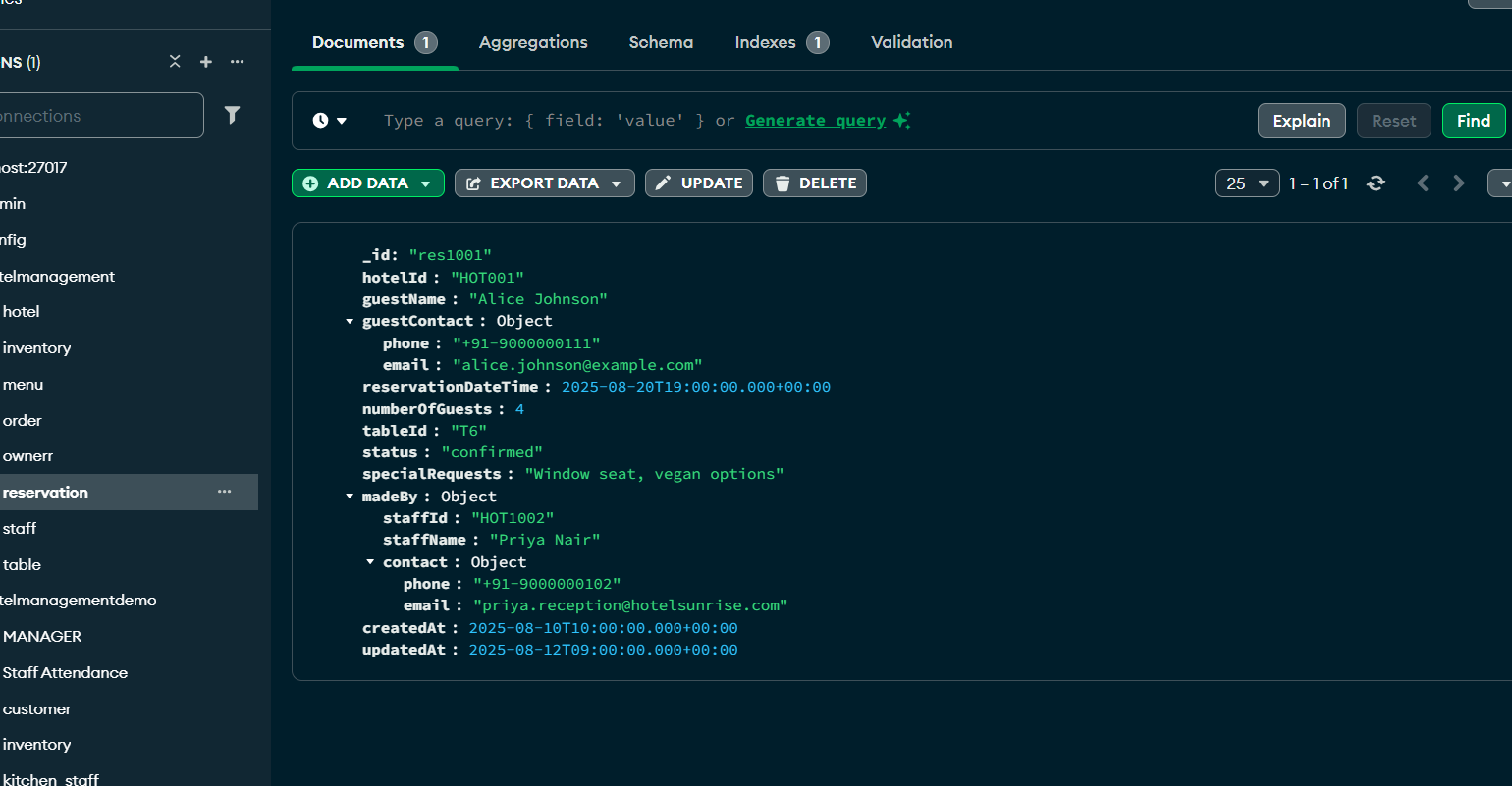
ORDER COLLECTION



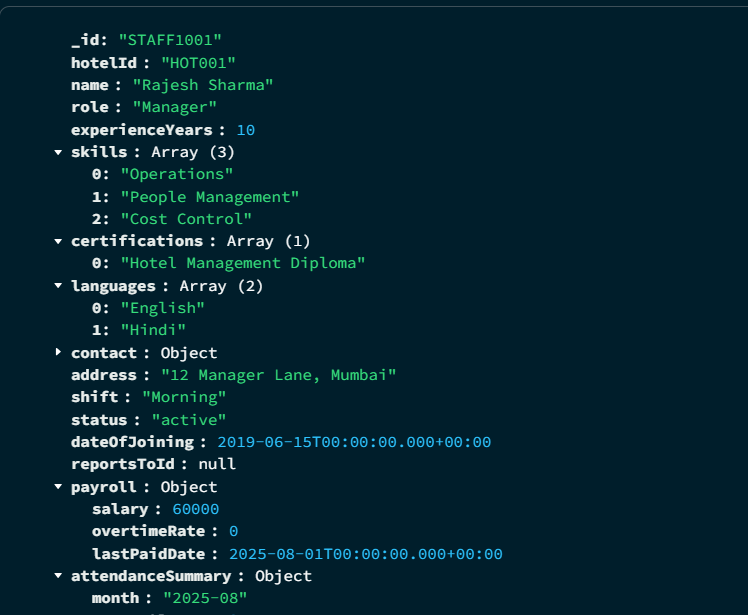
OWNER COLLECTION



RESERRVATION COLLECTION



STAFF COLLECTION



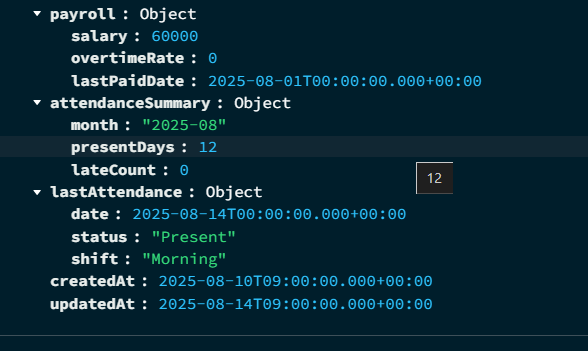


TABLE COLLECTION

