**MongoDB Data Modeling: Smart Knowledge Hub**

**Prepared by: Arnav B, Krishna Chaithanya K, Shruthe Raja**

**Collections Overview**

The following table lists the 10 collections used in our Smart knowledge Hub Platform, each with a specific purpose to support various functionalities such as user management, product cataloging, and order processing.

|  |  |
| --- | --- |
| Collection | Purpose |
| users | Customers using the platform |
| admins | Store administrators managing the platform |
| products | Items available for sale |
| categories | Product groupings (e.g., Electronics, Sports) |
| orders | Records of customer purchases |
| payments | Transaction details for orders |
| reviews | User feedback on products |
| inventory | Stock levels and availability per product |
| suppliers | Vendor information supplying products |
| shipping | Logistics and delivery details for orders |

**Relationships and Modeling Approaches**

This table describes the key relationships between collections, the type of relationship, and the modeling approach (embedding or referencing) chosen based on access patterns and scalability needs. 🔗Reference ✅Embed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Relationship | Type | Description | Field Reference | Justification |
| 1  🔗 | User ↔ Orders | One-to-Many | A user can place multiple orders; each order belongs to one user. | orders.user\_id → users.\_id | Orders can grow large; separating ensures scalability. |
| 2  🔗 | User ↔ Reviews | One-to-Many | A user can write many reviews; each review belongs to one user. | reviews.user\_id → users.\_id | Reviews are queried across products; keep separate for flexibility. |
| 3  🔗 | Product ↔ Categories | Many-to-One | Each product belongs to one category; a category has many products. | products.category → categories.\_id | Categories are static; referencing avoids duplication. |
| 4  🔗 | Product ↔ Inventory | One-to-One | Each product has a corresponding stock entry in inventory. | inventory.product\_id → products.\_id | Inventory updates frequently; separate for efficient updates. |
|  | Product ↔ Reviews | One-to-Many | A product can have multiple reviews from users. | reviews.product\_id → products.\_id | Allows independent review management and aggregation. |
| 6  🔗 | Order ↔ Payments | One-to-One | Each order has one associated payment record. | payments.order\_id → orders.\_id | Payment details are queried separately; referencing is suitable. |
| 7  🔗 | Order ↔ Shipping | One-to-One | Each order has one associated shipping record. | shipping.order\_id → orders.\_id | Shipping updates independently; keep separate for clarity. |
| 8  ✅ | Order ↔ Items (Product Snapshot) | One-to-Many (Embedded) | Each order includes multiple purchased products with details at purchase. | orders.items[] | Snapshot ensures historical accuracy of product price/quantity. |
| 9 🔗 | Product ↔ Supplier | Many-to-One | Each product is supplied by one supplier; a supplier provides many. | inventory.supplier\_id → suppliers.\_id | Suppliers are queried less often; referencing avoids redundancy. |
| 10  ✅ | Admin ↔ Permissions | One-to-Many (Embedded) | Each admin has a set of permissions defining their access level. | admins.permissions[] | Permissions are small and static; embedding is efficient. |

**Use Case Modeling Recommendations**

The following table outlines recommended modeling strategies for common use cases in our e-commerce platform to optimize query performance.

|  |  |
| --- | --- |
| Use Case | Recommended Modeling |
| Order items with product snapshot | Embed in orders.items[] |
| Product belongs to a category | Reference category |
| Product inventory status | Reference inventory |
| Review made by a user | Reference user\_id |
| Payment for an order | Reference order\_id |

**Sample Schemas**

**a. users**

{  
 "\_id": ObjectId("..."),  
 "name": "John Doe",  
 "email": "john@example.com",  
 "password\_hash": "hashedpassword123",  
 "address": {  
 "street": "123 Main St",  
 "city": "New York",  
 "zip": "10001",  
 "country": "USA"  
 },  
 "registration\_date": ISODate("2025-07-01T10:00:00Z"),  
 "preferences": ["electronics", "books"]  
}

**b. admins**

{  
 "\_id": ObjectId("..."),  
 "name": "Jane Admin",  
 "email": "jane@store.com",  
 "role": "manager",  
 "permissions": ["manage\_users", "update\_inventory"],  
 "last\_login": ISODate("2025-07-23T14:30:00Z")  
}

**c. products**

{  
 "\_id": ObjectId("..."),  
 "name": "Wireless Headphones",  
 "description": "Noise-cancelling over-ear headphones with 20-hour battery life.",  
 "price": 149.99,  
 "stock\_quantity": 50,  
 "category": "electronics",  
 "images": ["https://example.com/headphones1.jpg"],  
 "ratings": 4.5  
}

**d. orders**

{  
 "\_id": ObjectId("..."),  
 "user\_id": ObjectId("..."),  
 "order\_date": ISODate("2025-07-20T09:15:00Z"),  
 "status": "shipped",  
 "total\_amount": 199.98,  
 "items": [  
 {  
 "product\_id": ObjectId("..."),  
 "quantity": 1,  
 "price": 149.99  
 },  
 {  
 "product\_id": ObjectId("..."),  
 "quantity": 1,  
 "price": 89.99  
 }  
 ]  
}

**e. reviews**

{  
 "\_id": ObjectId("..."),  
 "product\_id": ObjectId("..."),  
 "user\_id": ObjectId("..."),  
 "rating": 5,  
 "comment": "Excellent sound quality and comfortable fit!",  
 "review\_date": ISODate("2025-07-22T11:45:00Z")  
}

**Query Scenarios**

|  |  |
| --- | --- |
| Scenario | Fields to Use |
| Show all products in a category | Filter by category in products |
| Show product inventory | Join products and inventory on product\_id |
| Show all reviews for a product | Filter reviews by product\_id |
| Get user’s order history | Filter orders by user\_id |
| Track shipping status for an order | Filter shipping by order\_id |