

MongoDB Schema Design - Online Retail Platform

Database Name: onlineRetailDB

The database is created and selected using:

```
use onlineRetailDB;
```

1. Product Catalog Design

Collection Name: products

Schema Design:

```
{
  _id: ObjectId,
  name: String,
  category: String,
  description: String,
  price: Number,
  stock: Number,
  brand: String,
  specifications: Object,
  createdAt: ISODate,
  updatedAt: ISODate
}
```

Sample Documents:

Example 1: Electronics Product

```
{
  name: "Samsung Galaxy S26",
  category: "Mobile Phones",
  description: "6G smartphone with 512GB storage and AI features",
```

```
    price: 89999,  
    stock: 60,  
    brand: "Samsung",  
    specifications: {  
        color: "Phantom Titanium",  
        screenSize: "6.3 inches",  
        battery: "4500 mAh"  
    },  
    createdAt: ISODate("2026-02-10T10:00:00Z"),  
    updatedAt: ISODate("2026-03-05T14:00:00Z")  
}
```

Example 2: Clothing Product

```
{  
    name: "Women's Cotton Kurti",  
    category: "Clothing",  
    description: "Comfort-fit cotton kurti",  
    price: 1299,  
    stock: 150,  
    brand: "Biba",  
    specifications: {  
        size: ["S", "M", "L", "XL"],  
        color: "Teal Blue"  
    },  
    createdAt: ISODate("2026-02-15T09:30:00Z"),  
    updatedAt: ISODate("2026-02-15T09:30:00Z")  
}
```

The screenshot shows the MongoDB Compass application interface. On the left, the sidebar includes sections for 'Compass', 'My Queries', 'Data Modeling', and 'CONNECTIONS (2)'. Under 'CONNECTIONS', there are entries for 'MongoDb' and 'onlineRetailDB', which is currently selected. The main workspace shows the 'onlineRetailDB > products' collection. At the top of this view are tabs for 'Documents' (2), 'Aggregations', 'Schema', 'Indexes' (3), and 'Validation'. Below these are buttons for 'ADD DATA', 'UPDATE', 'DELETE', 'EXPORT DATA', and 'EXPORT CODE'. A search bar at the top allows querying with a field name and value or generating a query. The results section displays two documents:

```

_id: ObjectId('69838dde98805c1530f78b1f')
name: "Samsung Galaxy S26"
category: "Mobile Phones"
description: "6G smartphone with 512GB storage and AI features"
price: 8999
stock: 60
brand: "Samsung"
specifications: Object
createdAt: 2026-02-10T10:00:00.000+00:00
updatedAt: 2026-03-05T14:00:00.000+00:00

_id: ObjectId('69838dde98805c1530f78b1f')
name: "Women's Cotton Kurti"
category: "Clothing"
description: "Comfort-fit cotton kurti"
price: 1299
stock: 150
brand: "Biba"
specifications: Object
createdAt: 2026-02-15T09:30:00.000+00:00
updatedAt: 2026-02-15T09:30:00.000+00:00

```

2. Customer Orders Design

Collection Name: orders

Schema Design:

```
{
  _id: ObjectId,
  userId: ObjectId,
  orderDate: ISODate,
  products: [
    {
      productId: ObjectId,
      quantity: Number,
      priceAtPurchase: Number
    }
  ],
  totalAmount: Number,
  status: String,
}
```

```
    paymentMethod: String,  
    shippingAddress: {  
        street: String,  
        city: String,  
        state: String,  
        postalCode: String,  
        country: String  
    }  
}
```

Sample Document:

```
{  
    userId: ObjectId("USER_ID"),  
    orderDate: ISODate("2026-03-10T12:00:00Z"),  
    products: [  
        { productId: ObjectId("PRODUCT_ID_1"), quantity: 1, priceAtPurchase: 89999 },  
        { productId: ObjectId("PRODUCT_ID_2"), quantity: 2, priceAtPurchase: 1299 }  
    ],  
    totalAmount: 92597,  
    status: "Delivered",  
    paymentMethod: "UPI",  
    shippingAddress: {  
        street: "MG Road",  
        city: "Bengaluru",  
        state: "Karnataka",  
        postalCode: "560001",  
        country: "India"  
    }  
}
```

```
}
```

The screenshot shows the MongoDB Compass interface. The left sidebar displays 'CONNECTIONS (2)' with 'MongoDb' and 'onlineRetailDB' selected. Under 'onlineRetailDB', there are collections: 'orders' (selected), 'products', and 'users'. The main panel shows the 'onlineRetailDB > onlineRetailDB > orders' path. The 'Documents' tab is active, showing 1 document. The document details are as follows:

```
_id: ObjectId('69838ec698805c1530f78b22')
userId: ObjectId('69838df198805c1530f78b20')
orderDate: 2026-03-10T12:00:00.000+00:00
products: Array (2)
totalAmount: 92597
status: "Delivered"
paymentMethod: "UPI"
shippingAddress: Object
```

3. User Authentication Design

Collection Name: users

Schema Design:

```
{
```

```
  _id: ObjectId,
```

```
  username: String,
```

```
  email: String,
```

```
  passwordHash: String,
```

```
  role: String,
```

```
  createdAt: ISODate,
```

```
  lastLogin: ISODate
```

```
}
```

Sample Document:

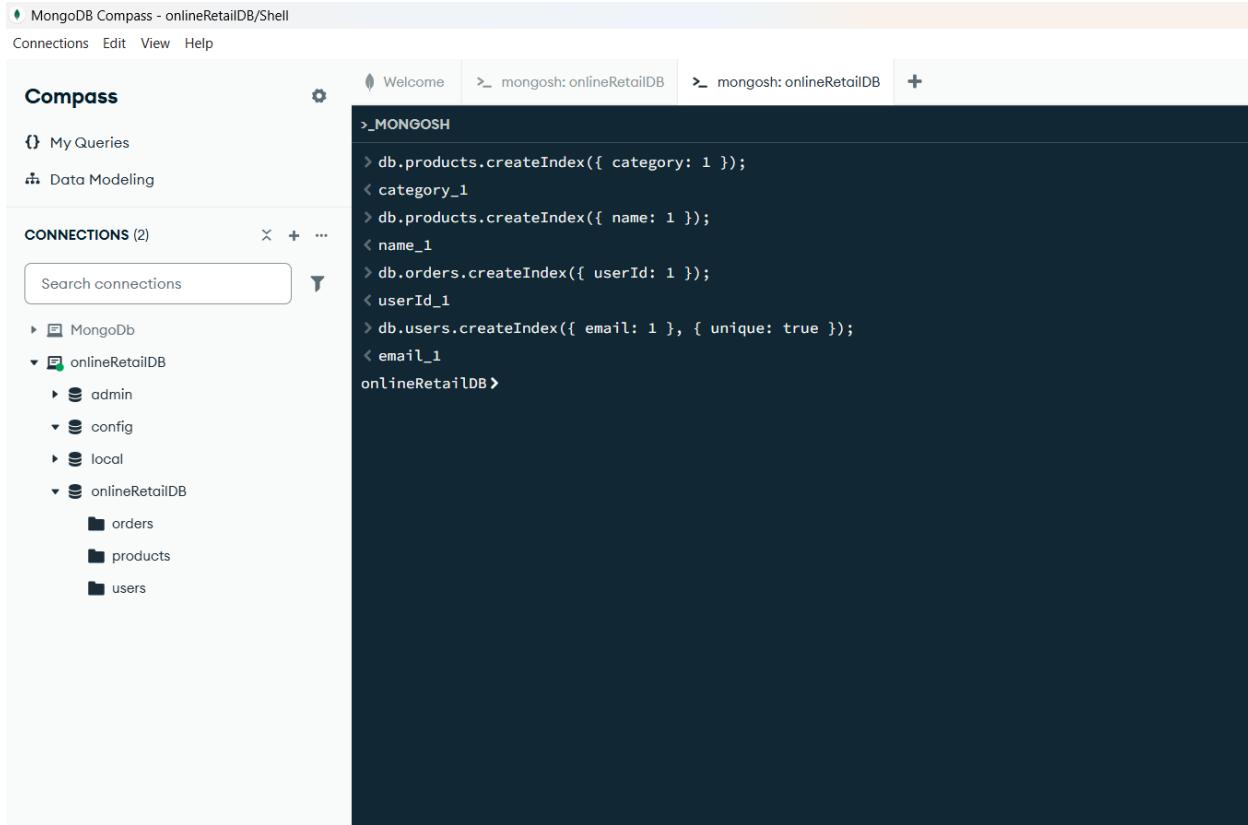
```
{  
  username: "reethu_g",  
  email: "reethu@example.com",  
  passwordHash: "$2b$10$Q29tQmFySGFzaFRlc3QxMjM0",  
  role: "customer",  
  createdAt: ISODate("2026-02-05T08:45:00Z"),  
  lastLogin: ISODate("2026-03-10T11:45:00Z")  
}
```

```
_id: ObjectId('69838df198885c1530f78b20')  
username : "reethu_g"  
email : "reethu@example.com"  
passwordHash : "$2b$10$Q29tQmFySGFzaFRlc3QxMjM0"  
role : "customer"  
createdAt : 2026-02-05T08:45:00.000+00:00  
lastLogin : 2026-03-10T11:45:00.000+00:00  
  
_id: ObjectId('69838df198885c1530f78b21')  
username : "admin_reethu"  
email : "admin@example.com"  
passwordHash : "$2b$10$Q29tQmFySGFzaFRlc3QxMjM0"  
role : "admin"  
createdAt : 2026-01-15T08:00:00.000+00:00  
lastLogin : 2026-03-08T09:00:00.000+00:00
```

4. Querying and Indexing

Indexes Used:

```
db.products.createIndex({ category: 1 });  
db.products.createIndex({ name: 1 });  
db.orders.createIndex({ userId: 1 });  
db.users.createIndex({ email: 1 }, { unique: true });
```



5. Sample Queries

1. Retrieve all products from a category

```
db.products.find({ category: "Mobile Phones" }).toArray();
```

```
> db.products.find({ category: "Mobile Phones" }).toArray()
< [
  {
    _id: ObjectId('69838dde98805c1530f78b1e'),
    name: 'Samsung Galaxy S26',
    category: 'Mobile Phones',
    description: '6G smartphone with 512GB storage and AI features',
    price: 89999,
    stock: 60,
    brand: 'Samsung',
    specifications: {
      color: 'Phantom Titanium',
      screenSize: '6.3 inches',
      battery: '4500 mAh'
    },
    createdAt: 2026-02-10T10:00:00.000Z,
    updatedAt: 2026-03-05T14:00:00.000Z
  }
]
```

2. Find a specific product by name

```
db.products.find({ name: /Samsung/i }).toArray();
```

```
> db.products.find({ name: /Samsung/i }).toArray()
< [
  {
    _id: ObjectId('69838dde98805c1530f78b1e'),
    name: 'Samsung Galaxy S26',
    category: 'Mobile Phones',
    description: '6G smartphone with 512GB storage and AI features',
    price: 89999,
    stock: 60,
    brand: 'Samsung',
    specifications: {
      color: 'Phantom Titanium',
      screenSize: '6.3 inches',
      battery: '4500 mAh'
    },
    createdAt: 2026-02-10T10:00:00.000Z,
    updatedAt: 2026-03-05T14:00:00.000Z
  }
]
```

3. Retrieve all orders for a specific user

```
db.orders.find({ userId: userId }).toArray();
```

```
> db.orders.find({ userId: userId }).toArray()
< [
  {
    _id: ObjectId('69838ec698805c1530f78b22'),
    userId: ObjectId('69838df198805c1530f78b20'),
    orderDate: 2026-03-10T12:00:00.000Z,
    products: [
      {
        productId: ObjectId('69838dde98805c1530f78b1e'),
        quantity: 1,
        priceAtPurchase: 89999
      },
      {
        productId: ObjectId('69838dde98805c1530f78b1f'),
        quantity: 2,
        priceAtPurchase: 1299
      }
    ],
    totalAmount: 92597,
    status: 'Delivered',
    paymentMethod: 'UPI',
    shippingAddress: {
      street: 'MG Road',
      city: 'Bengaluru',
      state: 'Karnataka',
      postalCode: '560001',
      country: 'India'
    }
  }
]
```

4. Find all users registered after a date

```
db.users.find({createdAt: { $gte: ISODate("2026-02-01") }}).toArray();
```

```
> db.users.find({ createdAt: { $gte: new Date("2026-02-01") } }).toArray()
< [
  {
    _id: ObjectId('69838df198805c1530f78b20'),
    username: 'reethu_g',
    email: 'reethu@example.com',
    passwordHash: '$2b$10$Q29tQmFySGFzaFRlc3QxMjM0',
    role: 'customer',
    createdAt: 2026-02-05T08:45:00.000Z,
    lastLogin: 2026-03-10T11:45:00.000Z
  }
]
```

5. Retrieve orders with product details (Aggregation)

```
db.orders.aggregate([
  { $match: { userId: userId } },
  {
    $lookup: {
      from: "products",
      localField: "products.productId",
      foreignField: "_id",
      as: "productDetails"
    }
  }
]).toArray();
```

```
> MONGOSH
< [
  {
    _id: ObjectId('69838ec698805c1530f78b22'),
    userId: ObjectId('69838df198805c1530f78b20'),
    orderDate: 2026-03-10T12:00:00.000Z,
    products: [
      {
        productId: ObjectId('69838dde98805c1530f78b1e'),
        quantity: 1,
        priceAtPurchase: 89999
      },
      {
        productId: ObjectId('69838dde98805c1530f78b1f'),
        quantity: 2,
        priceAtPurchase: 1299
      }
    ],
    totalAmount: 92597,
    status: 'Delivered',
    paymentMethod: 'UPI',
    shippingAddress: {
      street: 'MG Road',
      city: 'Bengaluru',
      state: 'Karnataka',
      postalCode: '560001',
      country: 'India'
    },
    productDetails: [
      {
        _id: ObjectId('69838dde98805c1530f78b1f'),
        name: "Women's Cotton Kurti",
        category: 'Clothing',
        description: 'Comfort-fit cotton kurti',
        price: 1299,
        stock: 150,
        brand: 'Biba',
        specifications: { size: [ 'S', 'M', 'L', 'XL' ], color: 'Teal Blue' },
        createdAt: 2026-02-15T09:30:00.000Z,
        updatedAt: 2026-02-15T09:30:00.000Z
      },
      {
        _id: ObjectId('69838dde98805c1530f78b1e'),
        name: 'Samsung Galaxy S26',
        category: 'Mobile Phones',
        description: '6G smartphone with 512GB storage and AI features',
        price: 89999,
        stock: 60,
        brand: 'Samsung',
        specifications: {
          color: 'Phantom Titanium',
          screenSize: '6.3 inches',
          battery: '4500 mAh'
        },
        createdAt: 2026-02-18T10:00:00.000Z,
        updatedAt: 2026-03-05T14:00:00.000Z
      }
    ]
  }
]
```

6. Benefits of MongoDB for This Use Case

- **Scalability:** Supports horizontal scaling for millions of products.
- **Flexibility:** Allows dynamic fields for different product categories.
- **Performance:** Indexing improves query efficiency.
- **Aggregation:** Enables complex queries like order summaries.
- **Integration:** Easily integrates with backend technologies like Node.js.