eCommerce Database

✔ **Users** (Consumers, Sellers, Admins)  
✔ **Addresses**  
✔ **Orders**  
✔ **Cart**  
✔ **Wishlist**  
✔ **Products**  
✔ **Sellers**  
✔ **Reviews & Ratings**  
✔ **Inventory**  
✔ **Payments**  
✔ **Shipments**  
✔ **Coupons & Discounts**

### ****Relationship Check****

|  |  |  |
| --- | --- | --- |
| **Entity 1** | **Entity 2** | **Relationship Type** |
| Users → Orders | One-to-Many | A user places multiple orders, but an order belongs to one user. |
| Users → Addresses | One-to-Many (Embedded) | A user has multiple addresses stored in their profile. |
| Users → Wishlist | One-to-One (One-to-Many with Products) | A user has a single wishlist, containing multiple products. |
| Users → Cart | One-to-One (One-to-Many with Products) | A user has one cart containing multiple products. |
| Users → Reviews & Ratings | One-to-Many with Products | A user can review multiple products. |
| Users → Payments | One-to-Many | A user can make multiple payments. |
| Users → Coupons & Discounts | Many-to-Many | A user can apply multiple coupons, and a coupon can be used by multiple users. |
| Products → Sellers | Many-to-One | A seller lists multiple products, but a product has only one seller. |
| Products → Inventory | One-to-One | Each product has a unique inventory record. |
| Products → Reviews & Ratings | One-to-Many | A product has multiple reviews from different users. |
| Orders → Products | Many-to-Many (Embedded) | An order contains multiple products, and a product can be in multiple orders. |
| Orders → Payments | One-to-One | Each order has a single payment transaction. |
| Orders → Shipments | One-to-One | Each order has one shipment tracking record. |
| Orders → Coupons & Discounts | Many-to-Many | A user can apply different coupons to different orders. |
| Shipments → Orders | One-to-One | Each order has one shipment, and a shipment belongs to one order. |

**Step 1: Users Schema (Consumers, Sellers, Admin)**

We'll structure the **Users** collection with the following considerations:

1. **Role-Based Users**: Consumers, Sellers, and Admins will be distinguished using a role field.
2. **User Details**: Basic info like name, email, phone, and address.
3. **Embedded vs. Referenced Data**:
   * Address can be embedded since it belongs only to the user.
   * Orders, Wishlist, and Cart can be referenced for flexibility.
4. **Relationships**:
   * One-to-Many: A User can place multiple orders.
   * One-to-Many: A User can have multiple addresses.

**📌 Users Schema (Embedded + Referenced)**

{

"name": "John Doe",

"email": "john.doe@example.com",

"phone": "+919876543210",

"role": "Consumer", // Can be "Consumer", "Seller", or "Admin"

"addresses": [

{

"type": "Home",

"street": "123 Street Name",

"city": "New York",

"state": "NY",

"zip": "10001",

"country": "USA",

“status”: “default”

}

],

"wishlist": ["wishlist1", "wishlist2"],

"cart": ["prod123", "prod789"],

"orders": ["order001", "order002"],

"createdAt": ISODate("2025-03-11T10:00:00Z"),

"updatedAt": ISODate("2025-03-11T10:00:00Z")

}

### ****📌 Example of a Seller User****

{

"\_id": ObjectId("user456"),

"name": "Alice Smith",

"email": "alice.smith@example.com",

"phone": "+911234567890",

"role": "Seller",

"addresses": [

{

"type": "Business",

"street": "456 Business St",

"city": "Los Angeles",

"state": "CA",

"zip": "90001",

"country": "USA"

}

],

"seller\_info": {

"store\_name": "Alice Fashion Hub",

"gst\_number": "GST456XYZ",

"products": ["prod111", "prod222"]

},

"createdAt": ISODate("2025-03-11T10:00:00Z"),

"updatedAt": ISODate("2025-03-11T10:00:00Z")

}

### ****📌 Example of an Admin User Document****

{

"\_id": ObjectId("user789"),

"name": "Rajesh Kumar",

"email": "rajesh.kumar@ecommerce.com",

"phone": "+919876543210",

"role": "Admin",

"permissions": ["manage\_users", "manage\_orders", "manage\_products", "view\_reports"],

"createdAt": ISODate("2025-03-11T10:00:00Z"),

"updatedAt": ISODate("2025-03-11T10:00:00Z")

}

### ****📌 Products Collection Schema (Corporate E-Commerce Example)**** 🛒

{

"\_id": ObjectId("prod123"),

"name": "Samsung Galaxy S24 Ultra",

"description": "Latest Samsung flagship with a 200MP camera and AI-powered features.",

"category": "Smartphones",

"price": 129999,

"discount": {

"percentage": 10,

"finalPrice": 116999

},

"brand": "Samsung",

"sellerId": ObjectId("seller001"),

"inventory": {

"stock": 50,

"threshold": 5

},

"images": [

"https://example.com/images/s24-front.jpg",

"https://example.com/images/s24-back.jpg"

],

"ratings": {

"average": 4.8,

"totalReviews": 1500

},

"specifications": {

"RAM": "12GB",

"Storage": "512GB",

"Processor": "Snapdragon 8 Gen 3",

"Battery": "5000mAh"

},

"createdAt": ISODate("2025-03-11T10:30:00Z"),

"updatedAt": ISODate("2025-03-11T10:30:00Z")

}

### ****🔹 Explanation of Product Fields:****

✅ **Seller Reference (sellerId)** → Links to the seller in the users collection.  
✅ **Inventory Management (stock, threshold)** → Helps prevent overselling.  
✅ **Discount Calculation (percentage, finalPrice)** → Avoids recalculating discount on every fetch.  
✅ **Ratings & Reviews (average, totalReviews)** → Quick access for users.  
✅ **Flexible Specifications (specifications)** → Adapts to various product types.

### ****📌 Orders Collection Schema (Corporate E-Commerce Example)**** 🛒 📦

{

"\_id": ObjectId("order123"),

"userId": ObjectId("user001"),

"products": [

{

"productId": ObjectId("prod123"),

"quantity": 2,

"priceAtPurchase": 116999,

"totalPrice": 233998

},

{

"productId": ObjectId("prod456"),

"quantity": 1,

"priceAtPurchase": 54999,

"totalPrice": 54999

}

],

"totalAmount": 288997,

"paymentStatus": "Paid",

"orderStatus": "Shipped",

"shippingAddress": {

"fullName": "Rahul Sharma",

"addressLine1": "123, MG Road",

"city": "Bangalore",

"state": "Karnataka",

"pincode": "560001",

"country": "India",

"contact": "+91 9876543210"

},

"trackingDetails": {

"carrier": "BlueDart",

"trackingId": "BD123456789",

"estimatedDelivery": ISODate("2025-03-15T12:00:00Z")

},

"paymentMethod": "Credit Card",

"createdAt": ISODate("2025-03-11T10:30:00Z"),

"updatedAt": ISODate("2025-03-12T08:45:00Z")

}

### ****🔹 Explanation of Order Fields:****

✅ **User Reference (userId)** → Links to the users collection.  
✅ **Products Array (products)** → Stores multiple items per order with their purchase price.  
✅ **Total Calculation (totalAmount)** → Precomputed for faster order retrieval.  
✅ **Payment & Order Status (paymentStatus, orderStatus)** → Helps in tracking order processing.  
✅ **Shipping Details (shippingAddress)** → Stores full delivery information.  
✅ **Tracking (trackingDetails)** → Links order to shipment provider.

### ****📌 Payments Collection Schema (Corporate E-Commerce Example)**** 💳 🔄

{

"\_id": ObjectId("pay123"),

"orderId": ObjectId("order123"),

"userId": ObjectId("user001"),

"amount": 288997,

"paymentMethod": "Credit Card",

"transactionId": "TXN987654321",

"paymentStatus": "Successful",

"paymentDate": ISODate("2025-03-11T10:45:00Z"),

"cardDetails": {

"cardType": "Visa",

"last4Digits": "1234",

"expiryDate": "12/27"

},

"UPI": null,

"wallet": null,

"createdAt": ISODate("2025-03-11T10:45:00Z"),

"updatedAt": ISODate("2025-03-11T10:45:00Z")

}

### ****🔹 Explanation of Payment Fields:****

✅ **Order Reference (orderId)** → Links the payment to a specific order.  
✅ **User Reference (userId)** → Identifies the payer.  
✅ **Amount Paid (amount)** → Ensures accurate transaction records.  
✅ **Payment Method (paymentMethod)** → Can be Credit Card, Debit Card, UPI, Wallet, etc.  
✅ **Transaction ID (transactionId)** → Stores the unique payment gateway transaction ID.  
✅ **Status (paymentStatus)** → Pending, Successful, or Failed.  
✅ **Card Details (cardDetails)** → Stores masked card info for security.  
✅ **UPI & Wallet Fields** → Supports alternative payment methods.

### ****📌 Reviews & Ratings Collection Schema (Corporate E-Commerce Example)**** ⭐📢

{

"\_id": ObjectId("review001"),

"userId": ObjectId("user001"),

"productId": ObjectId("prod123"),

"rating": 4.5,

"reviewText": "The laptop performance is amazing! Worth every penny.",

"reviewDate": ISODate("2025-03-11T12:00:00Z"),

"likes": 12,

"comments": [

{

"userId": ObjectId("user002"),

"commentText": "Thanks for the review! Does it heat up during gaming?",

"commentDate": ISODate("2025-03-12T08:30:00Z")

},

{

"userId": ObjectId("user003"),

"commentText": "I'm planning to buy this. How's the battery life?",

"commentDate": ISODate("2025-03-12T09:45:00Z")

}

],

"verifiedPurchase": true,

"createdAt": ISODate("2025-03-11T12:00:00Z"),

"updatedAt": ISODate("2025-03-11T12:00:00Z")

}

### ****🔹 Explanation of Reviews & Ratings Fields:****

✅ **User Reference (userId)** → Identifies the reviewer.  
✅ **Product Reference (productId)** → Links review to a product.  
✅ **Rating (rating)** → Stores a decimal value between 1.0 and 5.0.  
✅ **Review Text (reviewText)** → Contains user feedback.  
✅ **Review Date (reviewDate)** → Records when the review was submitted.  
✅ **Likes (likes)** → Tracks how many users found this review helpful.  
✅ **Comments (comments)** → Stores responses from other users.  
✅ **Verified Purchase (verifiedPurchase)** → Ensures the reviewer actually bought the product.

### ****📌 Shipments Collection Schema (Corporate E-Commerce Example)**** 🚚 📦

{

"\_id": ObjectId("shipment001"),

"orderId": ObjectId("order001"),

"userId": ObjectId("user001"),

"shipmentStatus": "Out for Delivery",

"trackingNumber": "TRK123456789",

"carrier": "FedEx",

"estimatedDelivery": ISODate("2025-03-15T18:00:00Z"),

"shipmentDate": ISODate("2025-03-12T10:00:00Z"),

"deliveryDate": null,

"shippingAddress": {

"fullName": "John Doe",

"street": "1234 Elm Street",

"city": "New York",

"state": "NY",

"zipCode": "10001",

"country": "USA",

"contactNumber": "+1-555-1234"

},

"trackingHistory": [

{

"status": "Order Packed",

"timestamp": ISODate("2025-03-11T14:00:00Z"),

"location": "Warehouse A - NY"

},

{

"status": "Shipped",

"timestamp": ISODate("2025-03-12T08:00:00Z"),

"location": "FedEx Facility - NY"

},

{

"status": "Out for Delivery",

"timestamp": ISODate("2025-03-12T12:30:00Z"),

"location": "FedEx Hub - NY"

}

],

"createdAt": ISODate("2025-03-11T12:00:00Z"),

"updatedAt": ISODate("2025-03-12T12:30:00Z")

}

### ****🔹 Explanation of Shipments Fields:****

✅ **Order Reference (orderId)** → Links shipment to an order.  
✅ **User Reference (userId)** → Identifies the customer.  
✅ **Shipment Status (shipmentStatus)** → Current delivery status (e.g., "Pending," "Shipped," "Out for Delivery," "Delivered").  
✅ **Tracking Number (trackingNumber)** → Unique identifier for tracking the shipment.  
✅ **Carrier (carrier)** → Name of the delivery service (FedEx, UPS, DHL).  
✅ **Estimated Delivery (estimatedDelivery)** → Expected arrival date.  
✅ **Shipment Date (shipmentDate)** → When the package was shipped.  
✅ **Delivery Date (deliveryDate)** → When the package was delivered (null if not yet delivered).  
✅ **Shipping Address (shippingAddress)** → Contains recipient’s full address and contact details.  
✅ **Tracking History (trackingHistory)** → Records shipment progress with timestamps and locations.

### ****📌 Coupons & Discounts Collection Schema (Corporate E-Commerce Example)**** 🎟️ 💰

{

"\_id": ObjectId("coupon001"),

"code": "WELCOME10",

"description": "Get 10% off on your first order!",

"discountType": "percentage",

"discountValue": 10,

"maxDiscountAmount": 500,

"minOrderAmount": 1000,

"validFrom": ISODate("2025-03-01T00:00:00Z"),

"validTill": ISODate("2025-03-31T23:59:59Z"),

"applicableCategories": ["Electronics", "Fashion"],

"applicableProducts": [ObjectId("product001"), ObjectId("product005")],

"applicableUsers": [ObjectId("user001"), ObjectId("user002")],

"redemptionLimit": 1000,

"redeemedCount": 150,

"status": "active",

"createdAt": ISODate("2025-02-25T10:00:00Z"),

"updatedAt": ISODate("2025-03-01T12:00:00Z")

}

### ****🔹 Explanation of Coupons & Discounts Fields:****

✅ **Coupon Code (code)** → Unique discount code (e.g., "WELCOME10").  
✅ **Description (description)** → Short info about the discount.  
✅ **Discount Type (discountType)** → "percentage" (e.g., 10% off) or "fixed" (e.g., ₹200 off).  
✅ **Discount Value (discountValue)** → The discount amount (percentage or fixed).  
✅ **Max Discount Amount (maxDiscountAmount)** → Maximum discount allowed.  
✅ **Min Order Amount (minOrderAmount)** → Minimum purchase required to use the coupon.  
✅ **Valid From & Till (validFrom, validTill)** → Date range when the coupon is applicable.  
✅ **Applicable Categories (applicableCategories)** → List of product categories eligible for the discount.  
✅ **Applicable Products (applicableProducts)** → Specific products eligible for the coupon.  
✅ **Applicable Users (applicableUsers)** → Users who can use the coupon (if restricted).  
✅ **Redemption Limit (redemptionLimit)** → Maximum number of times the coupon can be used.  
✅ **Redeemed Count (redeemedCount)** → How many times the coupon has been used.  
✅ **Status (status)** → Can be "active," "expired," or "disabled."

### ****📌 Inventory Collection Schema (Corporate E-Commerce Example)**** 📦🔥

{

"\_id": ObjectId("inventory001"),

"productId": ObjectId("product001"),

"warehouseId": ObjectId("warehouse001"),

"currentStock": 150,

"reservedStock": 20,

"availableStock": 130,

"reorderLevel": 10,

"reorderQuantity": 50,

"lastStockUpdate": ISODate("2025-03-10T14:00:00Z"),

"status": "in-stock",

"supplierId": ObjectId("supplier001"),

"createdAt": ISODate("2025-02-01T10:00:00Z"),

"updatedAt": ISODate("2025-03-10T14:00:00Z")

}

### ****🔹 Explanation of Inventory Fields:****

✅ **Product ID (productId)** → Links to the **Products** collection.  
✅ **Warehouse ID (warehouseId)** → Identifies where the stock is stored.  
✅ **Current Stock (currentStock)** → Total units in stock.  
✅ **Reserved Stock (reservedStock)** → Stock allocated for pending orders.  
✅ **Available Stock (availableStock)** → Stock available for new orders (Current Stock - Reserved Stock).  
✅ **Reorder Level (reorderLevel)** → Minimum stock level before triggering a restock.  
✅ **Reorder Quantity (reorderQuantity)** → Quantity to reorder when stock is low.  
✅ **Last Stock Update (lastStockUpdate)** → Last time stock was updated.  
✅ **Status (status)** → Can be "in-stock," "out-of-stock," or "low-stock."  
✅ **Supplier ID (supplierId)** → Links to the **Suppliers** collection.

## 📌 Cart Collection Schema (Corporate E-Commerce Example) 🛒

The **Cart** collection stores items a user intends to purchase but hasn't ordered yet. Each user has a single cart containing multiple products.

{

"\_id": ObjectId("cart001"),

"userId": ObjectId("user123"),

"items": [

{

"productId": ObjectId("prod123"),

"quantity": 2,

"priceAtAddition": 116999

},

{

"productId": ObjectId("prod789"),

"quantity": 1,

"priceAtAddition": 4999

}

],

"totalItems": 3,

"totalPrice": 238997,

"updatedAt": ISODate("2025-03-11T12:45:00Z")

}

### 🔹 Explanation of Cart Fields:

✅ **User Reference (userId)** → Links the cart to a specific user.  
✅ **Items Array (items)** → Contains multiple products with their quantity and price at the time they were added.  
✅ **Price at Addition (priceAtAddition)** → Stores the price when the product was added to prevent price fluctuations affecting the cart.  
✅ **Total Items (totalItems)** → Precomputed count of all items in the cart.  
✅ **Total Price (totalPrice)** → Sum of all item prices for quick retrieval.  
✅ **UpdatedAt (updatedAt)** → Timestamp for tracking cart modifications.

## 📌 Wishlist Collection Schema (Corporate E-Commerce Example) ❤️

The **Wishlist** collection stores user-favorite products they plan to purchase in the future.

{

"\_id": ObjectId("wishlist001"),

"userId": ObjectId("user123"),

"items": [

{

"productId": ObjectId("prod456"),

"addedAt": ISODate("2025-03-10T15:30:00Z")

},

{

"productId": ObjectId("prod789"),

"addedAt": ISODate("2025-03-11T08:45:00Z")

}

],

"totalItems": 2,

"updatedAt": ISODate("2025-03-11T12:45:00Z")

}

### 🔹 Explanation of Wishlist Fields:

✅ **User Reference (userId)** → Identifies the wishlist owner.  
✅ **Items Array (items)** → Contains multiple products a user has saved.  
✅ **Added At (addedAt)** → Timestamp indicating when the item was added to the wishlist.  
✅ **Total Items (totalItems)** → Precomputed count of all wishlist items.  
✅ **UpdatedAt (updatedAt)** → Timestamp for tracking wishlist modifications.

### 🔥 ****Key Differences Between Cart & Wishlist****

|  |  |  |
| --- | --- | --- |
| Feature | Cart 🛒 | Wishlist ❤️ |
| Purpose | Stores items user plans to buy soon | Stores items user is interested in but not buying yet |
| Contains | Product ID, quantity, price | Product ID, date added |
| Total Items | Required (for calculations) | Optional |
| User Checkout | Possible (moves to Orders) | Not directly linked to checkout |
| Auto Removal | Items removed when purchased | Items stay until removed manually |