# **MongoDB CRUD operation E-commerce Monsta**

## ****1. USERS Collection****

### Create

db.users.insertOne({

name: "Alice",

email: "alice@monsta.com",

address: { city: "Chennai", zip: "600001" },

wishlist: [],

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.users.find({ city: "Chennai" })

### Update

db.users.updateOne(

{ email: "alice@monsta.com" },

{ $set: { "address.city": "Bangalore", updatedAt: new Date() } }

)

### Delete

db.users.deleteOne({ email: "alice@monsta.com" })

## ****2. SELLERS Collection****

### Create

db.sellers.insertOne({

name: "BestElectro",

email: "seller1@store.com",

gst: "33AAAPT1234A1Z1",

address: { city: "Chennai", zip: "600010" },

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.sellers.find({ "address.city": "Chennai" })

### Update

db.sellers.updateOne(

{ name: "BestElectro" },

{ $set: { email: "contact@bestelectro.com", updatedAt: new Date() } }

)

### Delete

db.sellers.deleteOne({ name: "BestElectro" })

## ****3. PRODUCTS Collection****

### Create

db.products.insertOne({

name: "iPhone 15",

price: 99999,

categoryId: ObjectId("..."),

sellerId: ObjectId("..."),

tags: ["apple", "mobile"],

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.products.find({ price: { $gte: 50000 } })

### Update

db.products.updateOne(

{ name: "iPhone 15" },

{ $set: { price: 94999, updatedAt: new Date() } }

)

### Delete

db.products.deleteOne({ name: "iPhone 15" })

## ****4. INVENTORY Collection****

### Create

db.inventory.insertOne({

productId: ObjectId("..."),

sellerId: ObjectId("..."),

quantityAvailable: 100,

reorderLevel: 10,

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.inventory.find({ quantityAvailable: { $lt: 20 } })

### Update

db.inventory.updateOne(

{ productId: ObjectId("...") },

{ $inc: { quantityAvailable: -1 }, $set: { updatedAt: new Date() } }

)

### Delete

db.inventory.deleteOne({ productId: ObjectId("...") })

## ****5. ORDERS Collection****

### Create

db.orders.insertOne({

userId: ObjectId("..."),

items: [

{ productId: ObjectId("..."), name: "Laptop", qty: 1, priceAtPurchase: 65000 }

],

total: 65000,

status: "processing",

orderDate: new Date(),

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.orders.find({ userId: ObjectId("...") })

### Update

db.orders.updateOne(

{ \_id: ObjectId("...") },

{ $set: { status: "delivered", updatedAt: new Date() } }

)

### Delete

db.orders.deleteOne({ \_id: ObjectId("...") })

## ****6. REVIEWS Collection****

### Create

db.reviews.insertOne({

userId: ObjectId("..."),

productId: ObjectId("..."),

rating: 4,

comment: "Great product!",

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.reviews.find({ productId: ObjectId("...") })

### Update

db.reviews.updateOne(

{ \_id: ObjectId("...") },

{ $set: { rating: 5, comment: "Updated review", updatedAt: new Date() } }

)

### Delete

db.reviews.deleteOne({ \_id: ObjectId("...") })

## ****7. CART Collection****

### Create

db.cart.insertOne({

userId: ObjectId("..."),

items: [

{ productId: ObjectId("..."), name: "iPhone 15", price: 99999, quantity: 1, addedAt: new Date() }

],

createdAt: new Date(),

updatedAt: new Date()

})

### Read

db.cart.find({ userId: ObjectId("...") })

### Update

db.cart.updateOne(

{ userId: ObjectId("..."), "items.productId": ObjectId("...") },

{

$inc: { "items.$.quantity": 1 },

$set: { updatedAt: new Date() }

}

)

### Delete (empty cart)

db.cart.deleteOne({ userId: ObjectId("...") })