# MongoDB Assignment – 2

Data

**Product Collection:**

[{  \_id: "PROD001",  name: "Acoustic Guitar",  category: "String",  brand: "GuitarCo",  price: 499.99,  stock: 15,  features: ["Solid Spruce Top", "Mahogany Back & Sides"],  reviews: [    { user: "Alice", rating: 5, comment: "Amazing sound!" },    { user: "Bob", rating: 4, comment: "Great for beginners." },  ],},{  \_id: "PROD002",  name: "Electric Piano",  category: "Keyboard",  brand: "KeyMaster",  price: 799.0,  stock: 8,  features: ["88 Weighted Keys", "Multiple Voices"],  reviews: [    { user: "Charlie", rating: 5, comment: "Love the feel of the keys." },  ],},{  \_id: "PROD003",  name: "Drum Kit",  category: "Percussion",  brand: "BeatKing",  price: 1200.5,  stock: 5,  features: ["5-Piece Kit", "Cymbals Included"],  reviews: [],},{  \_id: "PROD004",  name: "Ukulele",  category: "String",  brand: "AlohaTune",  price: 89.99,  stock: 30,  features: ["Soprano Size", "Mahogany Body"],  reviews: [{ user: "Alice", rating: 4, comment: "Cute and fun!" }],},{  \_id: "PROD005",  name: "Bass Guitar",  category: "String",  brand: "BassPro",  price: 550.0,  stock: 10,  features: ["4-String", "Active Pickups"],  reviews: [],}];

**Order Collection:**

[{  \_id: "ORDER001",  customer\_id: "CUST001",  order\_date: ISODate("2023-01-10T10:00:00Z"),  items: [    { product\_id: "PROD001", quantity: 1, unit\_price: 499.99 },    { product\_id: "PROD004", quantity: 2, unit\_price: 89.99 },  ],  status: "completed",  total\_amount: 679.97,},{  \_id: "ORDER002",  customer\_id: "CUST002",  order\_date: ISODate("2023-01-15T14:30:00Z"),  items: [{ product\_id: "PROD002", quantity: 1, unit\_price: 799.0 }],  status: "pending",  total\_amount: 799.0,},{  \_id: "ORDER003",  customer\_id: "CUST001",  order\_date: ISODate("2023-02-01T09:00:00Z"),  items: [{ product\_id: "PROD001", quantity: 1, unit\_price: 499.99 }],  status: "completed",  total\_amount: 499.99,},{  \_id: "ORDER004",  customer\_id: "CUST003",  order\_date: ISODate("2023-02-05T11:45:00Z"),  items: [{ product\_id: "PROD003", quantity: 1, unit\_price: 1200.5 }],  status: "completed",  total\_amount: 1200.5,},{  \_id: "ORDER005",  customer\_id: "CUST002",  order\_date: ISODate("2023-03-01T16:00:00Z"),  items: [{ product\_id: "PROD005", quantity: 1, unit\_price: 550.0 }],  status: "pending",  total\_amount: 550.0,}]

## Q: Calculate Total Stock Value by Category.

Answer:

db.products.aggregate([  
 {  
 $group: {  
 \_id: "$category",  
 totalStockValue: { $sum: { $multiply: ["$price", "$stock"] } }  
 }  
 }  
])

## Q: Count Products per Brand.

Answer:

db.products.aggregate([  
 {  
 $group: {  
 \_id: "$brand",  
 productCount: { $count: {} }  
 }  
 }  
])

## Q: Find the Average Rating for Each Product.

Answer:

db.products.aggregate([  
 {  
 $project: {  
 name: 1,  
 avgRating: { $avg: "$reviews.rating" }  
 }  
 }  
])

## Q: Calculate Total Sales for Each Product.

Answer:

db.orders.aggregate([  
 { $unwind: "$items" },  
 {  
 $group: {  
 \_id: "$items.product\_id",  
 totalSales: { $sum: { $multiply: ["$items.quantity", "$items.unit\_price"] } }  
 }  
 }  
])

## Q: Recalculate each order's total amount using $reduce.

Answer:

db.orders.aggregate([  
 {  
 $project: {  
 order\_id: "$\_id",  
 recalculatedTotal: {  
 $reduce: {  
 input: "$items",  
 initialValue: 0,  
 in: { $add: ["$$value", { $multiply: ["$$this.quantity", "$$this.unit\_price"] }] }  
 }  
 }  
 }  
 }  
])

## Q: Find the total quantity of 'String' products sold.

Answer:

db.orders.aggregate([  
 { $unwind: "$items" },  
 {  
 $lookup: {  
 from: "products",  
 localField: "items.product\_id",  
 foreignField: "\_id",  
 as: "productInfo"  
 }  
 },  
 { $unwind: "$productInfo" },  
 { $match: { "productInfo.category": "String" } },  
 {  
 $group: {  
 \_id: "String",  
 totalQuantity: { $sum: "$items.quantity" }  
 }  
 }  
])

## Q: Find the average rating for each product that has received at least one review.

Answer:

db.products.aggregate([  
 { $match: { reviews: { $ne: [] } } },  
 {  
 $project: {  
 name: 1,  
 avgRating: { $avg: "$reviews.rating" }  
 }  
 }  
])

## Q: For each product, create a summary that includes its name and price, a simple list of the usernames who reviewed it, and a list of the order IDs in which it was sold.

Answer:

db.products.aggregate([  
 {  
 $lookup: {  
 from: "orders",  
 localField: "\_id",  
 foreignField: "items.product\_id",  
 as: "orderDocs"  
 }  
 },  
 {  
 $project: {  
 name: 1,  
 price: 1,  
 reviewers: "$reviews.user",  
 orderIds: "$orderDocs.\_id"  
 }  
 }  
])

## Q: Retrieve a list of all orders, but instead of just showing product IDs in the items array, replace them with a more detailed object containing the product's name, brand, and category.

Answer:

db.orders.aggregate([  
 { $unwind: "$items" },  
 {  
 $lookup: {  
 from: "products",  
 localField: "items.product\_id",  
 foreignField: "\_id",  
 as: "productInfo"  
 }  
 },  
 { $unwind: "$productInfo" },  
 {  
 $group: {  
 \_id: "$\_id",  
 customer\_id: { $first: "$customer\_id" },  
 order\_date: { $first: "$order\_date" },  
 status: { $first: "$status" },  
 items: {  
 $push: {  
 quantity: "$items.quantity",  
 unit\_price: "$items.unit\_price",  
 product: {  
 name: "$productInfo.name",  
 brand: "$productInfo.brand",  
 category: "$productInfo.category"  
 }  
 }  
 }  
 }  
 }  
])

## Q: Create a profile for each customer showing a list of products they have purchased. The list should not contain duplicates and should include the product name and category.

Answer:

db.orders.aggregate([  
 { $unwind: "$items" },  
 {  
 $lookup: {  
 from: "products",  
 localField: "items.product\_id",  
 foreignField: "\_id",  
 as: "productInfo"  
 }  
 },  
 { $unwind: "$productInfo" },  
 {  
 $group: {  
 \_id: "$customer\_id",  
 products: { $addToSet: { name: "$productInfo.name", category: "$productInfo.category" } }  
 }  
 }  
])