

# Experiment 4.3

## Aim

To design and implement an e-commerce catalog using MongoDB with nested documents to represent products and their variants, and to perform queries to retrieve and manipulate the data.

## Introduction

This experiment demonstrates how to create a MongoDB collection for an e-commerce catalog using nested documents. Each product document contains fields such as name, price, category, and an array of nested variant documents with details like color, size, and stock. This showcases MongoDB's flexible schema design and the ability to handle complex relationships inside a single collection.

## Procedure

1. Set up a MongoDB database for the e-commerce catalog.
2. Create a collection named `products`.
3. Define the product document structure with fields: `name`, `price`, `category`, and `variants` (an array of nested documents).
4. Insert sample products with multiple variants into the collection.
5. Perform queries to:
6. Retrieve all products.
7. Filter products by category.
8. Project specific variant details.
9. Use MongoDB shell commands or Mongoose methods to demonstrate working with nested documents.

## Code (MongoDB Shell Example)

```
use ecommerceDB;

// Insert sample products
db.products.insertMany([
  {
    name: "T-Shirt",
    price: 499,
    category: "Clothing",
    variants: [
      { color: "Red", size: "M", stock: 50 },
      { color: "Blue", size: "L", stock: 30 }
    ]
  },
  {
```

```

    name: "Sneakers",
    price: 2999,
    category: "Footwear",
    variants: [
      { color: "White", size: 9, stock: 20 },
      { color: "Black", size: 10, stock: 15 }
    ]
  }
});

// Retrieve all products
db.products.find().pretty();

// Filter products by category
db.products.find({ category: "Clothing" }).pretty();

// Project only name and variants
db.products.find({}, { name: 1, variants: 1 }).pretty();

```

## Output

- **All products:** Displays all products with their variants.
- **Filtered products by category (Clothing):** Shows only products in the Clothing category.
- **Projected fields:** Shows only the `name` and `variants` of all products.

Sample Output:

```

{
  "_id": ObjectId("..."),
  "name": "T-Shirt",
  "price": 499,
  "category": "Clothing",
  "variants": [
    { "color": "Red", "size": "M", "stock": 50 },
    { "color": "Blue", "size": "L", "stock": 30 }
  ]
}

```

## Learning Outcomes

- Understand nested document structure in MongoDB.
- Learn how to design an e-commerce catalog using flexible schemas.
- Perform CRUD operations and queries on nested documents.
- Retrieve and project specific data from nested arrays.
- Gain practical experience in modeling real-world data efficiently in MongoDB.