

# MongoDB Assignment (ADARSH SUNIL K)

## 1. Retrieve all products in the "Electronics" category.

```
mongodbAssignment> db.products.find()
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  },
  {
    _id: ObjectId('223456789012345678901234'),
    product_name: 'Laptop',
    brand: 'BrandY',
    price: 999.99,
    stock_quantity: 50,
    category: [ 'Electronics', 'Computers' ],
    attributes: { color: 'Silver', storage: '512GB', RAM: '16GB' },
    reviews: [
      {
        user_id: ObjectId('287654321098765432109876'),
        rating: 5,
        comment: 'Powerful and sleek!'
      },
      {
        user_id: ObjectId('267890123456789012345678'),
        rating: 4,
        comment: 'Good value for the price.'
      }
    ]
  },
  {
    _id: ObjectId('323456789012345678901234'),
    product_name: 'Tablet',
    brand: 'BrandZ',
    price: 299.99,
    stock_quantity: 200,
```

```
    stock_quantity: 200,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Gold', storage: '64GB', RAM: '4GB' },
    reviews: [
      {
        user_id: ObjectId('387654321098765432109876'),
        rating: 3,
        comment: 'Decent performance.'
      },
      {
        user_id: ObjectId('367890123456789012345678'),
        rating: 4,
        comment: 'Great for media consumption.'
      }
    ]
  }
]
```

## 2. Find all products with a price less than \$500.

```
mongodbAssignment> db.products.find({ price: { $lt: 500 } });
[
  {
    _id: ObjectId('323456789012345678901234'),
    product_name: 'Tablet',
    brand: 'BrandZ',
    price: 299.99,
    stock_quantity: 200,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Gold', storage: '64GB', RAM: '4GB' },
    reviews: [
      {
        user_id: ObjectId('387654321098765432109876'),
        rating: 3,
        comment: 'Decent performance.'
      },
      {
        user_id: ObjectId('367890123456789012345678'),
        rating: 4,
        comment: 'Great for media consumption.'
      }
    ]
  }
]
```

## 3. Retrieve all products with a rating greater than 4.

```
mongodbAssignment> db.products.find({ "reviews.rating": { $gt: 4 } });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  },
  {
    _id: ObjectId('223456789012345678901234'),
    product_name: 'Laptop',
    brand: 'BrandY',
    price: 999.99,
    stock_quantity: 50,
    category: [ 'Electronics', 'Computers' ],
    attributes: { color: 'Silver', storage: '512GB', RAM: '16GB' },
    reviews: [
      {
        user_id: ObjectId('287654321098765432109876'),
        rating: 5,
        comment: 'Powerful and sleek!'
      },
      {
        user_id: ObjectId('267890123456789012345678'),
        rating: 4,
        comment: 'Good value for the price.'
      }
    ]
  }
]
```

## 4. Find all products with a specific color (e.g., Black).

```
mongodbAssignment> db.products.find({ "attributes.color": "Black" });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  }
]
```

## 5.Retrieve all products sorted by price in descending order.

```
mongodbAssignment> db.products.find().sort({ price: -1 });
{
  _id: ObjectId('123456789012345678901234'),
  product_name: 'Laptop',
  brand: 'BrandX',
  price: 999.99,
  stock_quantity: 50,
  category: [ 'Electronics', 'Computer' ],
  attributes: { color: 'Silver', storage: '512GB', RAM: '16GB' },
  reviews: [
    {
      user_id: ObjectId('987654321098765432109876'),
      rating: 5,
      comment: 'Powerful and sleek!'
    },
    {
      user_id: ObjectId('367890123456789012345678'),
      rating: 4,
      comment: 'Good value for the price.'
    }
  ]
},
{
  _id: ObjectId('123456789012345678901234'),
  product_name: 'Smartphone',
  brand: 'BrandX',
  price: 599.99,
  stock_quantity: 100,
  category: [ 'Electronics', 'Mobile' ],
  attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
  reviews: [
    {
      user_id: ObjectId('987654321098765432109876'),
      rating: 4,
      comment: 'Great phone!'
    },
    {
      user_id: ObjectId('367890123456789012345678'),
      rating: 5,
      comment: 'Best phone I've ever had!'
    }
  ]
},
{
  _id: ObjectId('123456789012345678901234'),
  product_name: 'Tablet',
  brand: 'BrandX',
  price: 299.99,
  stock_quantity: 200,
  category: [ 'Electronics', 'Mobile' ],
  attributes: { color: 'Gold', storage: '64GB', RAM: '4GB' },
  reviews: [
    {
      user_id: ObjectId('367890123456789012345678'),
      rating: 3,
      comment: 'Decent performance.'
    },
    {
      user_id: ObjectId('367890123456789012345678'),
      rating: 4,
      comment: 'Great for media consumption.'
    }
  ]
}
mongodbAssignment>
```

## 6.Calculate the average rating of all products.

```
mongodbAssignment> db.products.aggregate([
...   { $unwind: "$reviews" },
...   { $group: { _id: null, avgRating: { $avg: "$reviews.rating" } } }
... ]);
[ { _id: null, avgRating: 4.166666666666667 } ]
```

## 7.Find all products where the stock quantity is less than 10.

```
mongodbAssignment> db.products.find({ stock_quantity: { $lt: 10 } });
mongodbAssignment> |
```

IN THIS DATA BASE THERE IS NO PRODUCTS QUANTITY LESS THAN 10

## 8. Retrieve all products with a specific brand (e.g., BrandX).

```
mongodbAssignment> db.products.find({ brand: "BrandX" });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  }
]
mongodbAssignment> |
```

## 9. Find all products where the storage is 128GB and RAM is 8GB.

```
mongodbAssignment> db.products.find({'attributes.storage': "128GB", "attributes.RAM"
... : "8GB"});
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: "Best phone I've ever had!"
      }
    ]
  }
]
mongodbAssignment> |
```

## 10. Retrieve all products with a specific user's review (e.g., user\_id: ObjectId("987654321098765432109876")).

```
mongodbAssignment> db.products.find(
... { "reviews.user_id": ObjectId("987654321098765432109876") });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: "Best phone I've ever had!"
      }
    ]
  }
]
mongodbAssignment> |
```

## 11. Calculate the total number of reviews for each product.

```
mongodbAssignment> db.products.aggregate([
...   { $project: { product_name: 1, totalReviews: { $size: "$reviews" } } }
... ]);
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    totalReviews: 2
  },
  {
    _id: ObjectId('223456789012345678901234'),
    product_name: 'Laptop',
    totalReviews: 2
  },
  {
    _id: ObjectId('323456789012345678901234'),
    product_name: 'Tablet',
    totalReviews: 2
  }
]
mongodbAssignment> |
```

## 12. Find all products with a price between \$400 and \$600.

```

mongodbAssignment> db.products.find({ price: { $gte: 400, $lte: 600 } });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: "Best phone I've ever had!"
      }
    ]
  }
]
mongodbAssignment> |

```

**13. Find all products with a specific category (e.g., "Mobile") and a stock quantity greater than 50.**

```

mongodbAssignment> db.products.find({category:"Mobile", stock_quantity:{$gt : 50
... } } );
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: "Best phone I've ever had!"
      }
    ]
  },
  {
    _id: ObjectId('323456789012345678901234'),
    product_name: 'Tablet',
    brand: 'BrandZ',
    price: 299.99,
    stock_quantity: 200,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Gold', storage: '64GB', RAM: '4GB' },
    reviews: [
      {
        user_id: ObjectId('387654321098765432109876'),
        rating: 3,
        comment: 'Decent performance.'
      },
      {
        user_id: ObjectId('367890123456789012345678'),
        rating: 4,
        comment: 'Great for media consumption.'
      }
    ]
  }
]
mongodbAssignment> |

```

**14. Find the highest rated product.**

```

mongodbAssignment> db.products.aggregate([
...   { $unwind: "$reviews" },
ng: { $max: "$reviews.rating" } } }, product_name: { $first: "$product_name" }, maxRating:
...   { $sort: { maxRating: -1 } },
...   { $limit: 1 }
... ]);
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    maxRating: 5
  }
]
mongodbAssignment> |

```

## 15. Retrieve all products sorted by brand in ascending order and then by price in descending order.

```
mongodbAssignment> db.products.find().sort({ brand: 1, price: -1 });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  },
  {
    _id: ObjectId('223456789012345678901234'),
    product_name: 'Laptop',
    brand: 'BrandY',
    price: 999.99,
    stock_quantity: 50,
    category: [ 'Electronics', 'Computers' ],
    attributes: { color: 'Silver', storage: '512GB', RAM: '16GB' },
    reviews: [
      {
        user_id: ObjectId('287654321098765432109876'),
        rating: 5,
        comment: 'Powerful and sleek!'
      },
      {
        user_id: ObjectId('267890123456789012345678'),
        rating: 4,
        comment: 'Good value for the price.'
      }
    ]
  }
]
```

```
{
  _id: ObjectId('323456789012345678901234'),
  product_name: 'Tablet',
  brand: 'BrandZ',
  price: 299.99,
  stock_quantity: 200,
  category: [ 'Electronics', 'Mobile' ],
  attributes: { color: 'Gold', storage: '64GB', RAM: '4GB' },
  reviews: [
    {
      user_id: ObjectId('387654321098765432109876'),
      rating: 3,
      comment: 'Decent performance.'
    },
    {
      user_id: ObjectId('367890123456789012345678'),
      rating: 4,
      comment: 'Great for media consumption.'
    }
  ]
}
]
mongodbAssignment> |
```

## 16. Find all products with a specific comment in their reviews (e.g., "Great phone!").

```
mongodbAssignment> db.products.find({ "reviews.comment": "Great phone!" });
[
  {
    _id: ObjectId('123456789012345678901234'),
    product_name: 'Smartphone',
    brand: 'BrandX',
    price: 599.99,
    stock_quantity: 100,
    category: [ 'Electronics', 'Mobile' ],
    attributes: { color: 'Black', storage: '128GB', RAM: '8GB' },
    reviews: [
      {
        user_id: ObjectId('987654321098765432109876'),
        rating: 4,
        comment: 'Great phone!'
      },
      {
        user_id: ObjectId('567890123456789012345678'),
        rating: 5,
        comment: 'Best phone I've ever had!'
      }
    ]
  }
]
mongodbAssignment> |
```

