**Create: Adding Data**

1. Add three new products to the products collection. Each product should include the following fields:

a. product\_id: A unique identifier for the product.

b. name: The name of the product.

c. category: The category of the product (e.g., "Electronics," "Furniture").

d. price: The price of the product (e.g., 50, 200).

e. stock: The quantity of the product in stock.

**Read: Querying Data**

2. Write a query to find all products with stock less than 50. Display their name,

price, and stock fields.

3. Write a query to retrieve all products in the Furniture category. Display the full document for each product.

**Update: Modifying Data**

4. Apply a 10% discount to the price of all products in the Electronics category.

Update the documents and verify the changes by querying all Electronics

products.

5. Increase the stock of all products priced above 500 by 10 units. Update the documents and verify the changes.

**Delete: Removing Data**

6. Write a query to remove all products with a stock equal to 0. Verify that the documents have been deleted.

7. Delete a product by its product\_id. Choose one product to delete and verify

that it no longer exists in the collection.

**Additional Challenge Questions**

8. Write a query to find all products in the Electronics category with a price greater than 200 and stock greater than 20. Display their name, price, and stock.

9. Add a new field, rating, to all products in the Electronics category with a

default value of 4.5.

10. Write a query to calculate and display the total number of products in the

Furniture category.

### Create: Adding Data

1. **Add three new products**:

db.products.insertMany([

{

product\_id: "P001",

name: "Smartphone",

category: "Electronics",

price: 799,

stock: 30

},

{

product\_id: "P002",

name: "Sofa",

category: "Furniture",

price: 450,

stock: 20

},

{

product\_id: "P003",

name: "Wireless Headphones",

category: "Electronics",

price: 150,

stock: 50

}

]);

### Read: Querying Data

1. **Find all products with stock less than 50**:

db.products.find(

{ stock: { $lt: 50 } },

{ name: 1, price: 1, stock: 1, \_id: 0 }

);

1. **Retrieve all products in the Furniture category**:

db.products.find(

{ category: "Furniture" }

);

### Update: Modifying Data

1. **Apply a 10% discount to all Electronics products**:

db.products.updateMany(

{ category: "Electronics" },

{ $mul: { price: 0.9 } }

);

**Verify changes**:

db.products.find(

{ category: "Electronics" }

);

1. **Increase the stock of all products priced above 500 by 10 units**:

db.products.updateMany(

{ price: { $gt: 500 } },

{ $inc: { stock: 10 } }

);

**Verify changes**:

db.products.find(

{ price: { $gt: 500 } }

);

### Delete: Removing Data

1. **Remove all products with stock equal to 0**:

db.products.deleteMany(

{ stock: 0 }

);

**Verify deletion**:

db.products.find(

{ stock: 0 }

);

1. **Delete a product by its product\_id**:

db.products.deleteOne(

{ product\_id: "P001" }

);

**Verify deletion**:

db.products.find(

{ product\_id: "P001" }

);

### Additional Challenge Questions

1. **Find all Electronics products with price > 200 and stock > 20**:

db.products.find(

{ category: "Electronics", price: { $gt: 200 }, stock: { $gt: 20 } },

{ name: 1, price: 1, stock: 1, \_id: 0 }

);

1. **Add a new field rating to all Electronics products with default value 4.5**:

db.products.updateMany(

{ category: "Electronics" },

{ $set: { rating: 4.5 } }

);

1. **Calculate and display the total number of products in the Furniture category**:

db.products.countDocuments(

{ category: "Furniture" }

);