**MONGODB HANDSON**

**Question 1**:

* Model a simple e-commerce system with two collections: Products and Orders.
* Insert 5 product documents with productName, price, stockCount, and category.
* Insert 3 order documents with orderId, customerName, items (array of product references), and totalAmount.
* Create a query that retrieves all orders containing a specific product.

**Question** 2:

* Design two models: one for Book and another for Author. Create a query where each book document contains an embedded array of author names.
* Create a second design where the Book document references Author by their unique ID in a separate Authors collection. Write a query that retrieves all books and includes their author's details.

**Question** 3:

* Create a schema for a Student collection that requires name, email, and age fields with specific (e.g., age should be a number between 10 and 100, email should follow a valid email format).
  + insert a valid a student details

**Projection**

* **Question**: Write queries that:
  1. Retrieve all students but only include their name and email.
  2. Retrieve all students and exclude the address field.
  3. Retrieve all students who have an address but only show their name, age, and grade.

**Update Operators**

* **Question**: Write queries that:
  1. Update the age of a student using the $set operator.
  2. Use the $inc operator to increase a student's age by 1.
  3. Use the $push operator to add a new hobby to a student's hobbies array.
  4. Use the $pull operator to remove a specific hobby from a student's hobbies array.

ANSWERS:

QUESTION 1:

**Collection 1: Products**

* **Fields**:
  + productName: String
  + price: Number
  + stockCount: Number
  + category: String

**Collection 2: Orders**

* **Fields**:
  + orderId: String
  + customerName: String
  + items: Array (references to Products)
  + totalAmount: Number

{

"\_id": ObjectId("product1\_id"),

"productName": "Laptop",

"price": 1000,

"stockCount": 50,

"category": "Electronics"

}

{

"\_id": ObjectId("product2\_id"),

"productName": "Smartphone",

"price": 500,

"stockCount": 100,

"category": "Electronics"

}

{

"\_id": ObjectId("product3\_id"),

"productName": "Headphones",

"price": 150,

"stockCount": 200,

"category": "Accessories"

}

Orders

{

"\_id": ObjectId("order1\_id"),

"orderId": "ORD001",

"customerName": "Alice",

"items": [

ObjectId("product1\_id"),

ObjectId("product3\_id")

],

"totalAmount": 1150

}

{

"\_id": ObjectId("order2\_id"),

"orderId": "ORD002",

"customerName": "Bob",

"items": [

ObjectId("product2\_id"),

ObjectId("product4\_id")

],

"totalAmount": 550

}

{

"\_id": ObjectId("order3\_id"),

"orderId": "ORD003",

"customerName": "Charlie",

"items": [

ObjectId("product1\_id"),

ObjectId("product5\_id")

],

"totalAmount": 1300

}

1. Use the **Find** query to search for orders containing a specific product. The query will look like this:

{

"items": ObjectId("product1\_id")

}

1. D
2. Answers

{

"name": "John Doe",

"email": "johndoe@example.com",

"age": 20

}

* **Retrieve All Students but Only Include Their name and email**

**Steps in Mongo Compass:**

1. Go to the **Students** collection in Mongo Compass.
2. Click on **Find**.
3. In the query section, paste the above query into the **Filter** field.
4. Click **Find** to see the result.

2. To retrieve all students and exclude the address field, use the following query:

**Steps in Mongo Compass:**

1. Go to the **Students** collection in Mongo Compass.
2. Click on **Find**.
3. In the query section, paste the above query into the **Filter** field.
4. Click **Find** to see the result.

**Retrieve All Students Who Have an address but Only Show Their name, age, and grade**

**Steps in Mongo Compass:**

1. Go to the **Students** collection in Mongo Compass.
2. Click on **Find**.
3. In the **Filter** field, paste the query's **filter** part:

{ "address": { "$exists": true } }

1. In the **Projection** field, paste the query's **projection** part:

{ "name": 1, "age": 1, "grade": 1, "\_id": 0 }

1. Click **Find** to see the result.

**1. Update the age of a student using the $set operator**

1. **Click on the Students Collection**.
2. **Use the Update Tab**:
   * On the **Students** collection page, click on **Update** from the top bar.
3. **Insert the Query**:
   * In the **Filter** field, specify the student you want to update (e.g., by name, \_id, or any other field). Example: { "name": "John Doe" }.
   * In the **Update** field, use the $set operator to change the age:

{ "$set": { "age": 25 } }

1. **Click Update**:
   * Click **Update** to apply the change. The student's age will be updated.