EXPERIMENT NO. 3

Objective(s):

Basic CRUD operations in MongoDB.

Outcome:

Students will understand NoSQL data handling and MongoDB shell usage.

Problem Statement:

Create, read, update, and delete documents in a MongoDB collection.

Background Study:

1. Introduction to NoSQL Databases
   In recent years, the need to manage large volumes of unstructured or semi-structured data has increased. Traditional relational databases, while powerful, often struggle with scalability, flexibility, and schema changes. This has led to the rise of **NoSQL databases**, which are designed to handle diverse data types and large-scale data with high availability.

2. What is MongoDB?
   MongoDB is a widely used **NoSQL database** that stores data in a flexible, JSON-like format called **BSON** (Binary JSON). Unlike traditional relational databases, MongoDB uses **collections** instead of tables and **documents** instead of rows.

   MongoDB is known for:
   - High performance
   - Schema flexibility
   - Scalability
   - Document-oriented storage

3. What are CRUD Operations?
   CRUD stands for the four basic operations that can be performed on data in any database system.

Code:

Create Database: using “use” keyword.

```
test> use BDA
switched to db BDA
```

Insert Document in Database:

```
BDA> db.Students.insertMany([ {Name: "Yaksh", Enrollment: 2203031080110, Age: 21},
{Name: "Poojan", Enrollment: 2203031080050, Age: 20}, {Name: "Bhavin", Enrollment:
2203031080042, Age: 20}])

  {
 acknowledged: true,
 insertedIds: {
'0': ObjectId('6875ee4d0491babc7f32a03c'),
'1': ObjectId('6875ee4d0491babc7f32a03d'),
'2': ObjectId('6875ee4d0491babc7f32a03e')
}
}
```

Update Document in Database:

```
BDA> db.Students.updateOne( { Name: "Yaksh" }, { $set: { Age: 22 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

BDA> db.Students.find()
[
  {
    _id: ObjectId("685a5b04878fd1bbb8a82d58"),
    Name: 'Yaksh',
    Enrollment: 2203031080110,
    Age: 22
  }
]
```

Delete Document in Database:

```
BDA> db.Students.deleteOne({ Name: "Bhavin" })
{ acknowledged: true, deletedCount: 1 }
```

```
BDA> db.Students.find()
[
  {
    _id: ObjectId("685a5b04878fd1bbb8a82d58"),
    Name: 'Yaksh',
    Enrollment: 2203031080110,
    Age: 22
  },
  {
    _id: ObjectId("685a5b04878fd1bbb8a82d59"),
    Name: 'Poojan',
    Enrollment: 2203031080050,
    Age: 20
  }
]
```

Question Bank:

1. Compare insertOne vs insertMany.
   - `insertOne()` inserts a **single document** into a MongoDB collection.
   - `insertMany()` inserts **multiple documents** at once, improving performance for bulk operations.
   - `insertMany()` returns an array of inserted IDs, while `insertOne()` returns one ID.

2. How do you create a nested document?
   - db.users.insertOne({ name: "John", address: { city: "Mumbai", pin: 400001 } });

3. How can MongoDB handle arrays or sets?
   - MongoDB supports arrays as a native data type in documents It can store multiple values in a single field using arrays. It also supports array-specific queries like `$in`, `$push`, `$addToSet`, and `$pull`.

EXPERIMENT NO. 4

Objective(s):
Store the basic information about students using List, Set, and Map in MongoDB.

Outcome:
Students will represent and store structured data using JSON-like documents.

Problem Statement:
Store students' roll number, name, DOB, and address using different MongoDB data types.

Background Study:

MongoDB is a **NoSQL database** that stores data in flexible, JSON- like **documents** rather than traditional tables. It supports **arrays** (List), **embedded documents** (Map), and simulates **Set-like** behavior through unique elements in arrays.

MongoDB's flexibility allows easy storage of complex data like lists of subjects, maps of marks per subject, and sets of hobbies.

Code:

Insert Document in MongoDB using List.

```
db.Students.insertMany([
  {
    RollNumber: 220301080110,
    Name: 'Yaksh',
    DOB: "2003-11-23",
    Address: {
      City: 'Vadodara',
      State: 'Gujarat',
      Zip: 390001
    },
    subjects: ["Math", "Physics", "Computer Science"],
    Hobby: ["Cricket", "Music", "Reading"]
  },
  {
    RollNumber: 220301080050,
    Name: 'Poojan',
    DOB: "2004-03-22",
    Address: {
      City: 'Navsari',
      State: 'Gujarat',
      Zip: 396445
    },
    Subjects: ["Math", "Physics", "Chemistery"],
    Hobby: ["Gaming", "Sketching", "Swimming"]
  },
```

```
        {
          RollNumber: 220301080007,
          Name: 'Jenish',
          DOB: "2004-11-19",
          Address: {
            City: 'Surat',
            State: 'Gujarat',
            Zip: 395007
          },
          Subjects: ["Math", "Physics", "IT"],
          Hobby: ["Photography", "Reading", "Football"]
        },
        {
          RollNumber: 220301080277,
          Name: 'Sandeep',
          DOB: "2005-05-19",
          Address: {
            City: 'Patna',
            State: 'Bihar',
            Zip: 800001
          },
          Subjects: ["Math", "Physics"],
          Hobby: ["Traveling", "Cooking", "Chess"]
        },
        {
          RollNumber: 220301080098,
          Name: 'Dhruv',
          DOB: "2004-12-05",
          Address: {
            City: 'Vadodara',
            State: 'Gujarat',
            Zip: 390002
          },
          Subjects: ["Physics", "Computer Science"],
          Hobby: ["Coding", "Gaming", "Drawing"]
        },
        {
          RollNumber: 220301080099,
          Name: 'Jigar',
          DOB: "2004-02-28",
          Address: {
            City: 'Surat',
            State: 'Gujarat',
            Zip: 395008
          },
          Subjects: ["Math", "Computer Science"],
          Hobby: ["Writing", "Gaming", "Public Speaking"]
        }
]);
```

Question Bank:

1. Write a MongoDB document to store student name, age, and list of subjects.

```
db.students.insertOne({
  name: "Yaksh",
  age: 21,
  subjects: ["Math", "Physics", "Computer Science"]
});
```

2. How can you simulate a Set in MongoDB for hobbies?

```
db.students.updateOne(
  { name: "Riya" },
  { $addToSet: { hobbies: "reading" } }
);
```

3. How do you store marks for different subjects using a Map in MongoDB?

```
db.students.insertOne({
  name: "Riya",
  marks: {
    Math: 85,
    Physics: 90,
    English: 88
  }
});
```