Exercise: CRUD Operations in MongoDB

Objective: Perform basic CRUD (Create, Read, Update, Delete) operations on a MongoDB collection.

- Connect to your MongoDB server (usually on localhost and port 27017).

3. Create a New Database and Collection:

- Switch to a new database (e.g., `testDB`).

```
test> use testdb
switched to db testdb
testdb>
```

- Create a new collection named `employees` and insert sample data.

```
testdb> db.createCollection("employee");
{ ok: 1 }
```

```
testdb> db.employees.insertOne({ firstname: "Yash", lastname: "Diwate", posi
tion: "Software Developer", salary: 75000 });
{
  acknowledged: true,
  insertedId: ObjectId('66bda5185fdcd6397e228fb5')
}
```

1. Create (Insert Documents)

Task: Insert 3 employee records into the `employees` collection.

- Action: Add documents that include information such as names, positions, and salaries for three employees.

```
testdb> db.employees.insertMany([{ firstname: "Vaibhav", lastname: "Nikumbh"
, position: "Data Analyst", salary: 72000 },{ firstname: "Anush", lastname:
"Gajbhiye", position: "hardware Engineer", salary: 68000 },{ firstname: "Pri
nce", lastname: "Dwivedi", position: "Database Administrator", salary: 78000
 },{ firstname: "Sakshi", lastname: "Kalbhor", position: "Sales", salary: 70
000 }]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66bda5bc5fdcd6397e228fb6'),
    '1': ObjectId('66bda5bc5fdcd6397e228fb7'),
    '2': ObjectId('66bda5bc5fdcd6397e228fb8'),
    '3': ObjectId('66bda5bc5fdcd6397e228fb9')
  }
```

2. Read (Query Documents)

Task: Perform the following queries on the `employees` collection:

1. Find All Documents:

- Action: Retrieve and view all documents in the `employees` collection.

```
testdb> db.employees.find();
[
  {
    _id: ObjectId('66bda5185fdcd6397e228fb5'),
    firstname: 'Yash',
    lastname: 'Diwate',
    position: 'Software Developer',
    salary: 75000
  },
  {
    _id: ObjectId('66bda5bc5fdcd6397e228fb6'),
    firstname: 'Vaibhav',
    lastname: 'Nikumbh',
    position: 'Data Analyst',
    salary: 72000
  },
  {
    _id: ObjectId('66bda5bc5fdcd6397e228fb7'),
```

2. Find a Single Document with a Specific Query:

- Action: Retrieve the document for a specific employee, such as one named "Alice".

```
testdb> db.employees.findOne({ firstname: "Alice" });
{
  _id: ObjectId('66bda7105fdcd6397e228fba'),
  firstname: 'Alice',
  lastname: 'pathak',
  position: 'Data Analyst',
  salary: 65000
}
```

3. Find Employees with a Salary Greater than $65,000:

- Action: Retrieve all documents where the salary is greater than $65,000.

```
testdb> db.employees.find({ salary: { $gt: 65000 } })
[
  {
    _id: ObjectId('66bda5185fdcd6397e228fb5'),
    firstname: 'Yash',
    lastname: 'Diwate',
    position: 'Software Developer',
    salary: 75000
  },
  {
    _id: ObjectId('66bda5bc5fdcd6397e228fb6'),
    firstname: 'Vaibhav',
    lastname: 'Nikumbh',
    position: 'Data Analyst',
    salary: 72000
```

3. Update (Modify Documents)

Task: Perform the following updates on the `employees` collection:

1. Update a Single Document:

- Action: Modify the salary of the employee named "Alice" to a new amount.

```
testdb> db.employees.updateOne({ firstname: "Alice" }, { $set: { salary: 800
00 }});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
testdb> db.employees.findOne({ firstname: "Alice" });
{
  _id: ObjectId('66bda7105fdcd6397e228fba'),
  firstname: 'Alice',
  lastname: 'pathak',
  position: 'Data Analyst',
  salary: 80000
}
```

2. Update Multiple Documents:

- Action: Increase the salary for all employees with the position "Developer" by a specified amount.

4. Delete (Remove Documents)

Task: Perform the following deletions on the `employees` collection:

1. Delete a Single Document:

- Action: Remove the document for the employee named "Bob".

```
testdb> db.employees.deleteOne({ firstname: "Bob" });
{ acknowledged: true, deletedCount: 0 }
testdb>
```

2. Delete Multiple Documents:

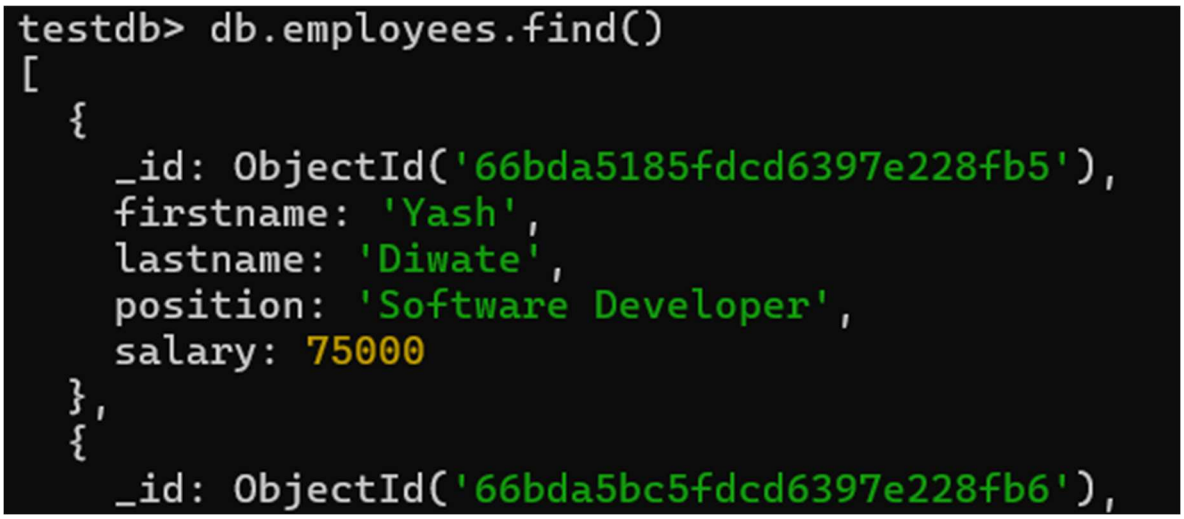
- Action: Remove all employees whose salary is less than a certain amount.

5. Verify the Operations

Task: Confirm that your CRUD operations have been applied correctly:

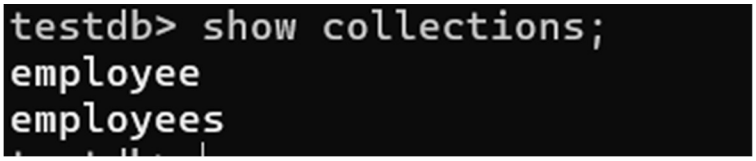
1. Find All Remaining Documents:

- Action: Check and ensure that the remaining documents in the `employees` collection reflect the changes made.



2. List Collections to Ensure `employees` Collection Exists:

- Action: Verify that the `employees` collection exists and has been updated.



3. Drop the Collection (Optional):

- Action: Optionally, you can remove the entire `employees` collection to clean up.

Completion Check:

- Confirm that each CRUD operation (Create, Read, Update, Delete) was executed correctly.

- Verify the content of the `employees` collection after each operation to ensure it matches expectations.