Name: Divya Jha

Roll no: L021

Sap id: 40778230025

Practical No: MongoDB Crud Operations

Create Operations

Creating a New Database
use userdb;

Output:
switched to db userdb

Note:- In MongoDB, a database is not actually created until it has atleast one collection !

Creating Collection
Method 1: Creating Empty Collection

db.createCollection("posts")
show collections

Method 2 : creating a collection during the insert process

db.posts.insertOne({"title": "Post 1"})

Output:
userdb> show collections
posts

Creating Documents

There are two ways to create new documents to a collection

1. **insertOne():**

db.users.insertOne({ name: "Angela", age: 27 });

Output :

{
  acknowledged: true,
  insertedId: ObjectId('65e22ff7dcee93f570585704')
}

2. **insertMany():**

db.users.insertMany([{name: "User1", age: 22}, {name: "User2", age: 25}])

Output:

{

```
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65e230e0dcee93f570585708'),
    '1': ObjectId('65e230e0dcee93f570585709')
  }
}
```

Read Operations

1. **Read All the Documents**

```
db.users.find()
```

output:

```
[
  {
    _id: ObjectId('65e23306dcee93f57058570f'),
    name: 'User1',
    age: 22,
    city: 'Mumbai'
  },
  {
    _id: ObjectId('65e23306dcee93f570585710'),
    name: 'User1',
    age: 25,
    city: 'Pune'
  },
  {
    _id: ObjectId('65e23306dcee93f570585711'),
    name: 'User2',
    age: 30,
    city: 'Kalyan'
  }
]
```

2. **Read Document with specific query**

Getting the document where name is "User1"

```
db.users.find({name: "User1"})
```

output:

```
[
  {
    _id: ObjectId('65e23306dcee93f57058570f'),
    name: 'User1',
    age: 22,
    city: 'Mumbai'
  },
  {
    _id: ObjectId('65e23306dcee93f570585710'),
    name: 'User1',
    age: 25,
```

```
    city: 'Pune'
  }
]
```

3. **Read Document with query & projection**

Hiding files id and age using projection

```
db.users.find({name: "User1"}, {_id:0, age:0})
```

output:

```
[
  { name: 'User1', city: 'Mumbai' },
  { name: 'User1', city: 'Pune' }
]
```

4. **findOne() : Returns a single document object**

```
db.users.findOne({name: "User1"}, {_id:0, age:0})
```

This returns the first document in the user's collection where the name field is "User1".

output:

```
[
  { name: 'User1', city: 'Mumbai' },
]
```

Update Operations

1. **updateOne()**

```
 db.users.updateOne({ age: { $lt: 23 } }, { $set: { status: "active" } })
```

Output

```
userdb> db.users.find({ age: { $lt: 23 } })
[
  {
    _id: ObjectId('65e23306dcee93f57058570f'),
    name: 'User1',
    age: 22,
    city: 'Mumbai',
    status: 'active'
  }
]
```

2. **updateMany()**

```
db.users.updateMany({ age: { $gt: 23 } }, { $set: { status: "inactive" } })
```

Output

```
userdb> db.users.find({ age: { $gt: 23 } })
[
  {
    _id: ObjectId('65e23306dcee93f570585710'),
    name: 'User1',
    age: 25,
    city: 'Pune',
```

```
    status: 'inactive'
  },
  {
    _id: ObjectId('65e23306dcee93f570585711'),
    name: 'User2',
    age: 30,
    city: 'Kalyan',
    status: 'inactive'
  }
]
```

Delete Operations

Delete Documents

i. deleteOne()

```
db.users.deleteOne({ name: "User2" })
```

Output

```
userdb> db.users.find({},{_id:0})
[
  { name: 'User1', age: 22, city: 'Mumbai', status: 'active' },
  { name: 'User1', age: 25, city: 'Pune', status: 'inactive' }
]
```

ii. deleteMany()

```
db.users.deleteMany({name:"User1"})
```

Output

```
db.users.find()
Empty
```

3. Delete Collection

```
db.users.drop()
```

Output

```
userdb> show collections
posts
```

4. Delete Database

```
db.dropDatabase()
```

Output

```
{ ok: 1, dropped: 'userdb' }
```