

BIRUNTHA KRISHNAMOORTHY

MONGODB EXERCISE : CRUD

Week 12: CRUD OPERATON

Create Operation:

Creating a new database named “library” and switching to it.

```
mongosh mongodb://127.0.0.1:27017
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bindhu>mongosh
Current Mongosh Log ID: 675dad100e60e5cc04893bf7
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB:  8.0.4
Using Mongosh:  2.3.4
mongosh 2.3.6 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-12-14T14:42:58.363+05:30: Access control is not enabled for the database. Read and write access to data and configuration is
unrestricted
-----

test> use library
switched to db library
library>
```

Create a collection named “books” with the fields title, author, and published_year. Insert into the “books” collection with the details of my favourite book.

```
test> use library
switched to db library
library> db.createCollection("books")
{ ok: 1 }
library>

test> use library
switched to db library
library> db.createCollection("books")
{ ok: 1 }
library> db.books.insertOne({title:"More than words",author:"Jill Santopolo",published_year:2020})
{ acknowledged: true, insertedId: ObjectId('675dafef481b30b050893bf8') }
library>
```

Read Operation

Retrieve all documents from the “books” collection

```
library> db.books.find()
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020
  },
  {
    _id: ObjectId('675db171481b30b050893bf9'),
    title: 'Golden Girl',
    author: 'Elin Hilderbrand',
    published_year: 2021
  },
  {
    _id: ObjectId('675db171481b30b050893bfa'),
    title: 'Someone Else's Shoe',
    author: 'Jojo Moyes',
    published_year: 2023
  },
  {
    _id: ObjectId('675db171481b30b050893bfb'),
    title: 'What you wish for',
    author: 'Katherine Center',
    published_year: 2020
  }
]
library>
```

Find and display a certain document by using some condition

```
library> db.books.find({author:"Jojo Moyes"})
[
  {
    _id: ObjectId('675db171481b30b050893bfa'),
    title: 'Someone Else's Shoe',
    author: 'Jojo Moyes',
    published_year: 2023
  }
]
library>
```

Fetch and display the document with the earliest published year.

```
mongosh mongodb://127.0.0.1:27020
library> db.books.find().sort({published_year:1}).limit(1)
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020
  }
]
library> db.books.find().sort({published_year:1})
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020
  },
  {
    _id: ObjectId('675db171481b30b050893bfb'),
    title: 'What you wish for',
    author: 'Katherine Center',
    published_year: 2020
  },
  {
    _id: ObjectId('675db171481b30b050893bf9'),
    title: 'Golden Girl',
    author: 'Elin Hilderbrand',
    published_year: 2021
  },
  {
    _id: ObjectId('675db171481b30b050893bfa'),
    title: 'Someone Else's Shoe',
    author: 'Jojo Moyes',
    published_year: 2023
  }
]
library>
```

Update Operation

Update the published year of the book with title “Golden Girl.”

```
]
library> db.books.update({title:"Golden Girl"},{$set:{published_year:2024}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
library>
```

Add a new field “genre” with the value “Fiction” to all documents

```
mongosh mongod://127.0.0.1 x + v
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
library> db.books.update({},{$set:{genre:"Fiction"}},{multi:true})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
library> db.books.find()
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bf9'),
    title: 'Golden Girl',
    author: 'Elin Hilderbrand',
    published_year: 2024,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bfa'),
    title: 'Someone Else's Shoe',
    author: 'Jojo Moyes',
    published_year: 2023,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bfb'),

```

Delete Operation

Remove the document with the title “Golden Girl.”

```
library> db.books.remove({title:"Golden Girl"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
library> db.books.find()
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bfa'),
    title: 'Someone Else's Shoe',
    author: 'Jojo Moyes',
    published_year: 2023,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bfb'),
    title: 'What you wish for',
    author: 'Katherine Center',
    published_year: 2020,
    genre: 'Fiction'
  }
]
library>
```

Delete all documents where the published year is after 2020

```
library> db.books.remove({published_year:{$gt:2020}})
{ acknowledged: true, deletedCount: 1 }
library> db.books.find()
[
  {
    _id: ObjectId('675dafef481b30b050893bf8'),
    title: 'More than words',
    author: 'Jill Santopolo',
    published_year: 2020,
    genre: 'Fiction'
  },
  {
    _id: ObjectId('675db171481b30b050893bf8'),
    title: 'What you wish for',
    author: 'Katherine Center',
    published_year: 2020,
    genre: 'Fiction'
  }
]
library>
```