

ADVANCED DBMS LAB**Experiment No.: 8****Name: Swetha Prakash****Roll No: 46****Batch: B****Date: 03-06-2022****AIM**

Build sample collections/documents to perform query operations in MongoDB.

COMMANDS AND OUTPUT**1. Create a database.**

- use <database_name>

```
> use my_project_db
switched to db my_project_db
```

2. Check the database list.

- show dbs

```
> show dbs
AJCE      0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
rmca      0.000GB
sample    0.000GB
```

3. Create a collection.

- db.createCollection(name)

```
> db.createCollection("mydb")
{ "ok" : 1 }
> db.createCollection("movie")
{ "ok" : 1 }
```

4. Insert a value into a collection.

- db.collection_name.insert({document})

```
> db.movie.insert({"name":"Avengers: Endgame"})
WriteResult({ "nInserted" : 1 })
```

5. Insert many values into a collection.

➤ `db.collection_name.insertMany({documents})`

```
> db.developers.insertMany([{"_id": 20, devname: "John Wick", tools: "Visual Studio", born: 1948 }, {"_id": 21, devname: "Ganesh Roy", tools: "Net Beans", born: 1945 }, {"_id": 22, devname: "Deeksha Raul", tools: "Unity 3D", born: 1954 }])
```

```
> db.movie.insert([{"name": "The Invisible Man"}, {"name": "Mission Impossible"}, {"name": "Twilight"}])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

6. Display the contents in a collection.

➤ `db.collection_name.find()`

```
> db.movie.find()
{ "_id" : ObjectId("629989e5d2bbf682947f2a4c"), "name" : "Avengers: Endgame" }
{ "_id" : ObjectId("629991c5d2bbf682947f2a4f"), "name" : "The Invisible Man" }
{ "_id" : ObjectId("629991c5d2bbf682947f2a50"), "name" : "Mission Impossible" }
{ "_id" : ObjectId("629991c5d2bbf682947f2a51"), "name" : "Twilight" }
>
```

```
> db.developers.find()
{ "_id" : 20, "devname" : "John Wick", "tools" : "Visual Studio", "born" : 1948 }
{ "_id" : 21, "devname" : "Ganesh Roy", "tools" : "Net Beans", "born" : 1945 }
{ "_id" : 22, "devname" : "Deeksha Raul", "tools" : "Unity 3D", "born" : 1954 }
```

7. Display the contents in a collection in a formatted way.

➤ `db.collection_name.find().pretty()`

```
> db.developers.find().pretty()
{
  "_id" : 21,
  "devname" : "Ganesh Roy",
  "tools" : "Net Beans",
  "born" : 1945
}
{
  "_id" : 22,
  "devname" : "Deeksha Raul",
  "tools" : "Unity 3D",
  "born" : 1954
}
```

8. Update a collection.

- `Db.collection_name.update(Selection_criteria, Updated_data)`

```
> db.developers.update({"devname":"John Wick"},{$set:{"devname":"Mathew Sebastian"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.developers.find()
{ "_id" : 20, "devname" : "Mathew Sebastian", "tools" : "Visual Studio", "born" : 1948 }
{ "_id" : 21, "devname" : "Ganesh Roy", "tools" : "Net Beans", "born" : 1945 }
{ "_id" : 22, "devname" : "Deeksha Raul", "tools" : "Unity 3D", "born" : 1954 }
```

9. Delete from a collection.

- `Db.collection_name.remove(Deletion_criteria)`

```
> db.developers.remove({"devname":"Mathew Sebastian"})
WriteResult({ "nRemoved" : 1 })
> db.developers.find()
{ "_id" : 21, "devname" : "Ganesh Roy", "tools" : "Net Beans", "born" : 1945 }
{ "_id" : 22, "devname" : "Deeksha Raul", "tools" : "Unity 3D", "born" : 1954 }

> db.movie.remove({"_id" : ObjectId("629989e5d2bbf682947f2a4c")})
WriteResult({ "nRemoved" : 1 })
> db.movie.find()
{ "_id" : ObjectId("629991c5d2bbf682947f2a4f"), "name" : "The Invisible Man" }
{ "_id" : ObjectId("629991c5d2bbf682947f2a50"), "name" : "Mission Impossible" }
{ "_id" : ObjectId("629991c5d2bbf682947f2a51"), "name" : "Twilight" }
```