

## BIBD MINI PROJECT

### 1. Start The MongoDB shell.

```
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sreej>mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("088410ab-9c4e-42f2-8e8c-28bedd6b3471") }
MongoDB server version: 5.0.6
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2022-04-06T14:59:33.467+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
```

### 2. Check for any existing databases.

```
> show dbs;
admin    0.000GB
config   0.000GB
local    0.000GB
```

3. So, we do not have our own existing database, hence we'll create a new one. Here, we've created a collection in the newdb database named cars and added a document of one car. So now if we check the databases on the system we can see the newdb database.

```
> use newdb
switched to db newdb
> show collections
> db.cars.insertOne({Name:"Hyundai", Model: "i10"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("624d5e496bf7275dbcb40354")
}
> show dbs;
admin    0.000GB
config   0.000GB
local    0.000GB
newdb    0.000GB
```

4. Now, to check if the document is added in the collection we run:

```
> db.cars.find().pretty()
{
  "_id" : ObjectId("624d5e496bf7275dbcb40354"),
  "Name" : "Hyundai",
  "Model" : "i10"
}
```

5. We know how to create a database. Now let's see how to delete/drop a database. Here, I've already created another sample database "delddb" with document in it.

```
> use delddb
switched to db delddb
> db.test.insertOne({Name: "abc"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("624d5fc36bf7275dbcb40355")
}
> show dbs;
admin    0.000GB
config   0.000GB
delddb   0.000GB
local    0.000GB
newdb    0.000GB
```

6. "delddb" database was deleted using db.dropDatabase().

```
> use delddb
switched to db delddb
> db.dropDatabase()
{ "ok" : 1 }
> show dbs;
admin    0.000GB
config   0.000GB
local    0.000GB
newdb    0.000GB
```

7. The basic CRUD operations include Create, Read, Update & Delete.  
 8. The Create commands are of two types "insertOne(data, options)" & "insertMany([data], options)".  
 9. The Read command are of two types "find(filter, options)" & "findOne(filter, options)".  
 10. The Update command are of three types "updateOne(filter, data, options)" ;  
 "updateMany(filter, data, options)" & "replaceOne(filter, data, options)".  
 11. The Delete command are of two types "deleteOne(filter, options)" & "deleteMany(filter, options)".  
 12. Executing the insertOne and insertMany commands:

```
> use movies
switched to db movies
> db.hollywood.insertOne({Name: "Harry Potter", Part: "3", Release_Year:"2005",Languages:["English","Hindi"],Rating:{IMDB:"8.2",Rotten_Tomatoes:"95%}})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("624d62596bf7275dbcb40357")
}

> db.hollywood.insertMany([
  {Name: "Star Wars", Part: "7", Release_Year:"2016",Languages:"English",Rating:{IMDB:"6",Rotten_Tomatoes:"25%"}},
  {Name: "The Batman", Part: "1", Release_Year:"2022",Languages:["English","Spanish"],Rating:{IMDB:"8.5",Rotten_Tomatoes:"90%}}}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("624d648a6bf7275dbcb40358"),
    ObjectId("624d648a6bf7275dbcb40359")
  ]
}
```

13. Let us now check the database.

```

> db.hollywood.find().pretty()
{
  "_id" : ObjectId("624d62596bf7275dbcb40357"),
  "Name" : "Harry Potter",
  "Part" : "3",
  "Release_Year" : "2005",
  "Languages" : [
    "English",
    "Hindi"
  ],
  "Rating" : {
    "IMDB" : "8.2",
    "Rotten_Tomatoes" : "95%"
  }
}
{
  "_id" : ObjectId("624d648a6bf7275dbcb40358"),
  "Name" : "Star Wars",
  "Part" : "7",
  "Release_Year" : "2016",
  "Languages" : "English",
  "Rating" : {
    "IMDB" : "6",
    "Rotten_Tomatoes" : "25%"
  }
}
{
  "_id" : ObjectId("624d648a6bf7275dbcb40359"),
  "Name" : "The Batman",
  "Part" : "1",
  "Release_Year" : "2022",
  "Languages" : [
    "English",
    "Spanish"
  ],
  "Rating" : {
    "IMDB" : "8.5",
    "Rotten_Tomatoes" : "90%"
  }
}

```

14. Here, we've successfully executed the insertOne and insertMany commands and also Read the data in the Document.

15. Now let's try updating the Part of Star Wars to 9 in the document using updateOne().

```

> db.hollywood.updateOne({_id:ObjectId("624d648a6bf7275dbcb40358")},{ $set:{ "Part": "9" }})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

```

16. Check if the value is updated:

```
> db.hollywood.find().pretty()
{
  "_id" : ObjectId("624d62596bf7275dbcb40357"),
  "Name" : "Harry Potter",
  "Part" : "3",
  "Release_Year" : "2005",
  "Languages" : [
    "English",
    "Hindi"
  ],
  "Rating" : {
    "IMDB" : "8.2",
    "Rotten_Tomatoes" : "95%"
  }
},
{
  "_id" : ObjectId("624d648a6bf7275dbcb40358"),
  "Name" : "Star Wars",
  "Part" : "9",
  "Release_Year" : "2016",
  "Languages" : "English",
  "Rating" : {
    "IMDB" : "6",
    "Rotten_Tomatoes" : "25%"
  }
},
{
  "_id" : ObjectId("624d648a6bf7275dbcb40359"),
  "Name" : "The Batman",
  "Part" : "1",
  "Release_Year" : "2022",
  "Languages" : [
    "English",
    "Spanish"
  ],
  "Rating" : {
    "IMDB" : "8.5",
    "Rotten_Tomatoes" : "90%"
  }
}
```

17. Now lets try updateMany command.

```
> db.hollywood.updateMany({},{$set:{"Oscar Nominations":"unknown"}})
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }
```

18. Keeping the first parameter blank means updating all the entries.

```
> db.hollywood.find().pretty()
{
  "_id" : ObjectId("624d62596bf7275dbcb40357"),
  "Name" : "Harry Potter",
  "Part" : "3",
  "Release_Year" : "2005",
  "Languages" : [
    "English",
    "Hindi"
  ],
  "Rating" : {
    "IMDB" : "8.2",
    "Rotten_Tomatoes" : "95%"
  },
  "Oscar Nominations" : "unknown"
},
{
  "_id" : ObjectId("624d648a6bf7275dbcb40358"),
  "Name" : "Star Wars",
  "Part" : "9",
  "Release_Year" : "2016",
  "Languages" : "English",
  "Rating" : {
    "IMDB" : "6",
    "Rotten_Tomatoes" : "25%"
  },
  "Oscar Nominations" : "unknown"
},
{
  "_id" : ObjectId("624d648a6bf7275dbcb40359"),
  "Name" : "The Batman",
  "Part" : "1",
  "Release_Year" : "2022",
  "Languages" : [
    "English",
    "Spanish"
  ],
  "Rating" : {
    "IMDB" : "8.5",
    "Rotten_Tomatoes" : "90%"
  },
  "Oscar Nominations" : "unknown"
}
```

19. Now using the Find command to find an entry with a particular tag.

```
> db.hollywood.find({Languages:"Spanish"}).pretty()
{
  "_id" : ObjectId("624d648a6bf7275dbcb40359"),
  "Name" : "The Batman",
  "Part" : "1",
  "Release_Year" : "2022",
  "Languages" : [
    "English",
    "Spanish"
  ],
  "Rating" : {
    "IMDB" : "8.5",
    "Rotten_Tomatoes" : "90%"
  },
  "Oscar Nominations" : "unknown"
}
```

20. So now let's delete an entry from movies using deleteOne() where Part is 9.

```
> db.hollywood.deleteOne({Part:"9"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.hollywood.find().pretty()
{
  "_id" : ObjectId("624d62596bf7275dbcb40357"),
  "Name" : "Harry Potter",
  "Part" : "3",
  "Release_Year" : "2005",
  "Languages" : [
    "English",
    "Hindi"
  ],
  "Rating" : {
    "IMDB" : "8.2",
    "Rotten_Tomatoes" : "95%"
  },
  "Oscar Nominations" : "unknown"
}
{
  "_id" : ObjectId("624d648a6bf7275dbcb40359"),
  "Name" : "The Batman",
  "Part" : "1",
  "Release_Year" : "2022",
  "Languages" : [
    "English",
    "Spanish"
  ],
  "Rating" : {
    "IMDB" : "8.5",
    "Rotten_Tomatoes" : "90%"
  },
  "Oscar Nominations" : "unknown"
}
```

21. Now deleting movies with deleteMany() operations where Oscar Nominations is unknown.

```
> db.hollywood.deleteMany({"Oscar Nominations":"unknown"})
{ "acknowledged" : true, "deletedCount" : 2 }
> db.hollywood.find().pretty()
>
```

22. All records are deleted and hence we now have an empty collection.