



Experiment-10

Name: Anuj Kumar Singh

Branch: CSE

Semester: 5th

Subject Name: ADBMS

UID: 23BCS10195

Section/Group: KRG_1-B

Date of Performance: 04/11/2025

Subject Code: 23CSP-333

1. Aim: Perform CRUD Operations using No-SQL Database: MongoDB

2. Objective:

- Create and manage databases and collections.
- Perform **CRUD operations** — Create, Read, Update, and Delete documents.
- Use **aggregation** for grouping and summarizing data.
- Understand how to modify, query, and manage collections efficiently in MongoDB.
- Gain practical experience with real-world data handling through command-line operations.

3. Code:

```
// 1. Show all databases
```

```
show dbs
```

```
// 2. Create or switch to your database
```

```
use db_shivanshu
```

```
// 3. Check current database
```

```
db
```

```
// 4. Create a collection (optional)
```

```
db.createCollection("movies")
```

```
// 5. Insert Operations
```

```
db.movies.insertOne({
```

```
  title: "Inception",
```

```
  director: "Christopher Nolan",
```

```
  year: 2010,
```

```
  rating: 8.8,
```



```
genre: "Sci-Fi",  
  
features: ["Mind-bending", "Action"],  
  
added_by: "Shivanshu"  
  
})
```

```
db.movies.insertMany([  
  
  {  
  
    title: "Interstellar",  
  
    director: "Christopher Nolan",  
  
    year: 2014,  
  
    rating: 8.6,  
  
    genre: "Sci-Fi",  
  
    features: ["Space", "Drama"],  
  
    added_by: "Shivanshu"
```

```
  },
```

```
  {
```

```
    title: "Avatar",  
  
    director: "James Cameron",  
  
    year: 2009,  
  
    rating: 7.8,  
  
    genre: "Adventure",  
  
    features: ["3D", "Action"],  
  
    added_by: "Shivanshu"
```

```
  },
```

```
  {
```

```
    title: "The Dark Knight",
```



```
    director: "Christopher Nolan",
    year: 2008,
    rating: 9.0,
    genre: "Action",
    features: ["Batman", "Crime"],
    added_by: "Shivanshu"
  }
])

// 6. Show all collections

show collections

// 7. Read Operations

db.movies.find().pretty()

db.movies.findOne()

db.movies.find({}, { title: 1, year: 1, _id: 0 })

db.movies.find({ genre: "Action" })

db.movies.find({ "details.language": "English" })

// 8. Update Operations

db.movies.updateOne(
  { title: "Avatar" },
  { $set: { rating: 8.1, genre: "Sci-Fi Adventure" } }
)

db.movies.updateOne(
  { title: "Inception" },
  { $push: { features: "Thriller" } }
)
```



```
db.movies.updateOne(  
  { title: "Inception" },  
  { $pull: { features: "Action" } }  
)
```

```
db.movies.updateMany(  
  { director: "Christopher Nolan" },  
  { $set: { language: "English" } }  
)
```

```
db.movies.updateMany({}, { $unset: { added_by: "" } })
```

```
db.movies.updateMany({}, { $set: { color: "Full HD" } })
```

```
db.movies.updateOne(  
  { title: "Tenet" },  
  { $set: { director: "Christopher Nolan", year: 2020, rating: 7.5 } },  
  { upsert: true }  
)
```

// 9. Delete Operations

```
db.movies.deleteOne({ title: "Avatar" })
```

```
db.movies.deleteMany({ director: "Christopher Nolan" })
```

```
db.movies.deleteMany({})
```

// 10. Grouping (Aggregation)

```
db.movies.aggregate([  
  { $group: { _id: "$director", total_movies: { $sum: 1 } } }  
)
```

```
db.movies.aggregate([
```

```
{ $group: { _id: "$genre", total_movies: { $sum: 1 } } }
```

```
)
```

```
// 11. Drop collection
```

```
db.movies.drop()
```

```
// 12. Drop database
```

```
db.dropDatabase()
```

4. Output:

```
shivanshuranjansingh@Shivanshu-MacBook ~ % mongosh
Current Mongosh Log ID: 690399ad55d128fad2120e1b
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:      7.0.25
Using Mongosh:      2.5.8

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

-----
The server generated these startup warnings when booting
2025-10-30T22:06:06.549+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
|
|_
|_  admin            48.00 KiB
|_  config           108.00 KiB
|_  db_shivanshu      68.00 KiB
|_  local             72.00 KiB
test> use db_shivanshu
switched to db db_shivanshu
db_shivanshu> db.createCollection("movies")
|
|_
|_  { ok: 1 }
db_shivanshu>
```

```
db_shivanshu> db.movies.insertOne({
|_  title: "Inception",
|_  director: "Christopher Nolan",
|_  year: 2010,
|_  rating: 8.8,
|_  genre: "Sci-Fi",
|_  features: ["Mind-bending", "Action"],
|_  added_by: "Shivanshu"
|_ })
|_ {
|_   acknowledged: true,
|_   insertedId: ObjectId('690399f655d128fad2120e1c')
|_ }
db_shivanshu> db.movies.insertMany([
|_  {
|_    title: "Interstellar",
|_    director: "Christopher Nolan",
|_    year: 2014,
|_    rating: 8.6,
|_    genre: "Sci-Fi",
|_    features: ["Space", "Drama"],
|_    added_by: "Shivanshu"
|_  },
|_  {
|_    title: "Avatar",
|_    director: "James Cameron",
|_    year: 2009,
|_    rating: 7.8,
|_    genre: "Adventure",
|_    features: ["3D", "Action"],
|_    added_by: "Shivanshu"
|_  },
|_  {
|_    title: "The Dark Knight",
|_    director: "Christopher Nolan",
|_    year: 2008,
|_    rating: 9.0,
|_    genre: "Action",
|_    features: ["Batman", "Crime"],
|_    added_by: "Shivanshu"
|_  }
|_ ])
|_ {
|_   acknowledged: true,
|_   insertedIds: {
|_     '0': ObjectId('690399fe55d128fad2120e1d'),
|_     '1': ObjectId('690399fe55d128fad2120e1e'),
|_     '2': ObjectId('690399fe55d128fad2120e1f')
|_   }
|_ }
db_shivanshu>
```

```
db_shivanshu> db.movies.find()
|_ [
|_   {
|_     _id: ObjectId('690399f655d128fad2120e1c'),
|_     title: 'Inception',
|_     director: 'Christopher Nolan',
|_     year: 2010,
|_     rating: 8.8,
|_     genre: 'Sci-Fi',
|_     features: [ 'Mind-bending', 'Action' ],
|_     added_by: 'Shivanshu'
|_   },
|_   {
|_     _id: ObjectId('690399fe55d128fad2120e1d'),
|_     title: 'Interstellar',
|_     director: 'Christopher Nolan',
|_     year: 2014,
|_     rating: 8.6,
|_     genre: 'Sci-Fi',
|_     features: [ 'Space', 'Drama' ],
|_     added_by: 'Shivanshu'
|_   },
|_   {
|_     _id: ObjectId('690399fe55d128fad2120e1e'),
|_     title: 'Avatar',
|_     director: 'James Cameron',
|_     year: 2009,
|_     rating: 7.8,
|_     genre: 'Adventure',
|_     features: [ '3D', 'Action' ],
|_     added_by: 'Shivanshu'
|_   },
|_   {
|_     _id: ObjectId('690399fe55d128fad2120e1f'),
|_     title: 'The Dark Knight',
|_     director: 'Christopher Nolan',
|_     year: 2008,
|_     rating: 9,
|_     genre: 'Action',
|_     features: [ 'Batman', 'Crime' ],
|_     added_by: 'Shivanshu'
|_   }
|_ ]
db_shivanshu> db.movies.findOne()
|_ {
|_   _id: ObjectId('690399f655d128fad2120e1c'),
|_   title: 'Inception',
|_   director: 'Christopher Nolan',
|_   year: 2010,
|_   rating: 8.8,
|_   genre: 'Sci-Fi',
|_   features: [ 'Mind-bending', 'Action' ],
|_   added_by: 'Shivanshu'
|_ }
```

```
db_shivanshu> db.movies.find({}, { title: 1, year: 1, _id: 0 })
[[
  {
    title: 'Inception', year: 2010 },
    { title: 'Interstellar', year: 2014 },
    { title: 'Avatar', year: 2009 },
    { title: 'The Dark Knight', year: 2008 }
  ]
]
db_shivanshu> db.movies.find({ genre: "Action" })
[[
  {
    _id: ObjectId('690399fe55d128fad2120e1f'),
    title: 'The Dark Knight',
    director: 'Christopher Nolan',
    year: 2008,
    rating: 9,
    genre: 'Action',
    features: [ 'Batman', 'Crime' ],
    added_by: 'Shivanshu'
  }
]
]
db_shivanshu> db.movies.find({ "details.language": "English" })
[[
]
]
db_shivanshu> █
```

```
db_shivanshu> db.movies.updateOne(
  { title: "Avatar" },
  { $set: { rating: 8.1, genre: "Sci-Fi Adventure" } }
)
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
]
]
db_shivanshu> db.movies.updateOne(
  { title: "Inception" },
  { $push: { features: "Thriller" } }
)
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
]
]
db_shivanshu> db.movies.updateOne(
  { title: "Inception" },
  { $pull: { features: "Action" } }
)
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
]
]
db_shivanshu> db.movies.updateMany(
  { director: "Christopher Nolan" },
  { $set: { language: "English" } }
)
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 3,
    modifiedCount: 3,
    upsertedCount: 0
  }
]
]
db_shivanshu> █
```

```
db_shivanshu> db.movies.updateMany({}, { $unset: { added_by: "" } })
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 4,
    modifiedCount: 4,
    upsertedCount: 0
  }
]
]
db_shivanshu> db.movies.updateMany({}, { $set: { color: "Full HD" } })
[[
  {
    acknowledged: true,
    insertedId: null,
    matchedCount: 4,
    modifiedCount: 4,
    upsertedCount: 0
  }
]
]
db_shivanshu> db.movies.updateOne(
  { title: "Tenet" },
  { $set: { director: "Christopher Nolan", year: 2020, rating: 7.5 } },
  { upsert: true }
)
[[
  {
    acknowledged: true,
    insertedId: ObjectId('69039ac0e681e6eb85a7e326'),
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 1
  }
]
]
db_shivanshu> █
```

```
db_shivanshu> db.movies.drop()
[ ]
true
db_shivanshu> db.dropDatabase()
[ ]
{ ok: 1, dropped: 'db_shivanshu' }
db_shivanshu>
```

- Understand the structure and working of MongoDB as a NoSQL database.
- Create and manage databases and collections using MongoDB commands.
- Perform CRUD (Create, Read, Update, Delete) operations on documents.
- Apply aggregation operations for data grouping and analysis.
- Gain practical skills in handling and modifying data using MongoDB shell.