Experiment-10

Name: Prabhakar Kr Jha **UID: 23BCS12284**
Branch: CSE **Section/Group: KRG_1-B**
Semester: 5th **Date of Performance:04/11/2025**
Subject Name: ADBMS **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_prabhakar

// 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,
```

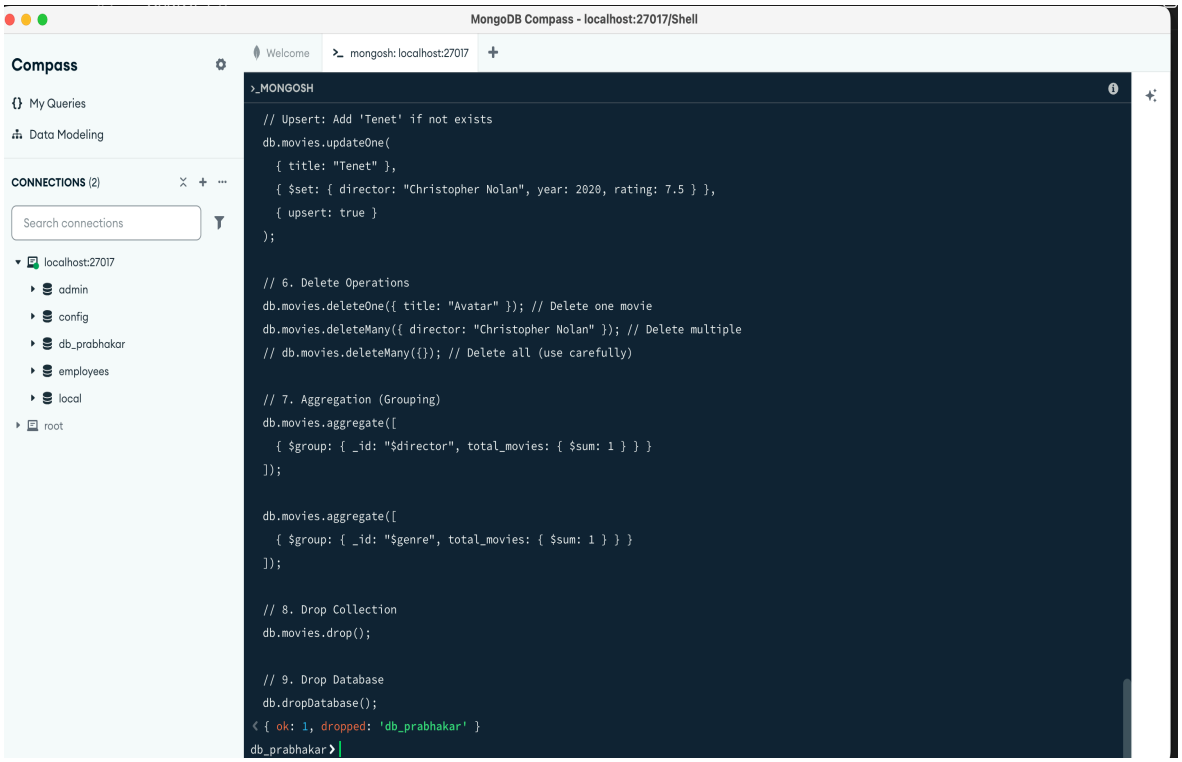
```
  genre: "Sci-Fi",
  features: ["Mind-bending", "Action"],
  added_by: "prabhakar"
})
db.movies.insertMany([
  {
    title: "Interstellar",
    director: "Christopher Nolan",
    year: 2014,
    rating: 8.6,
    genre: "Sci-Fi",
    features: ["Space", "Drama"],
    added_by: "prabhakar"
  },
  {
    title: "Avatar",
    director: "James Cameron",
    year: 2009,
    rating: 7.8,
    genre: "Adventure",
    features: ["3D", "Action"],
    added_by: "prabhakar"
  },
  {
    title: "The Dark Knight",
```

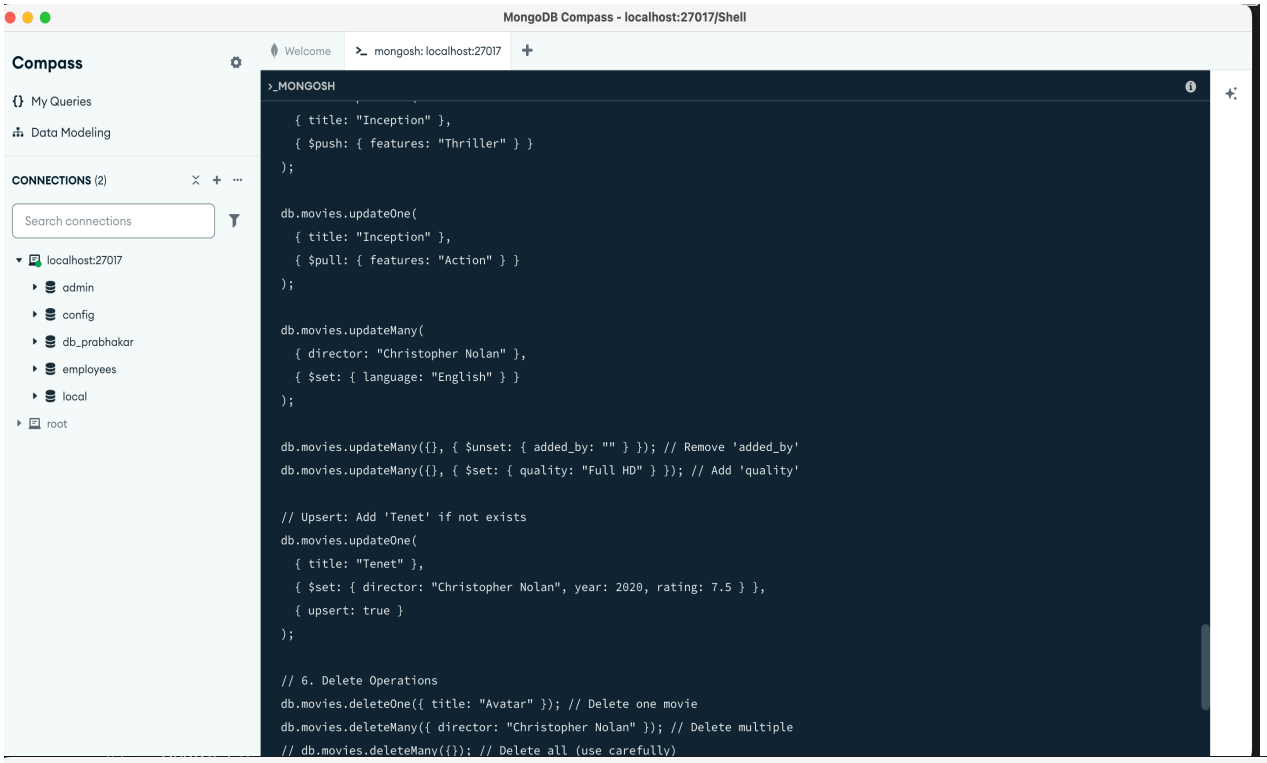
```
    director: "Christopher Nolan",
    year: 2008,
    rating: 9.0,
    genre: "Action",
    features: ["Batman", "Crime"],
    added_by: "prabhakar"
  }
])
// 6. Show all collections
show collections
// 7. Read Operations
db.movies.find().pretty()
db.movies.findOne()
db.movies.find({}, { title: 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" } }
)
```

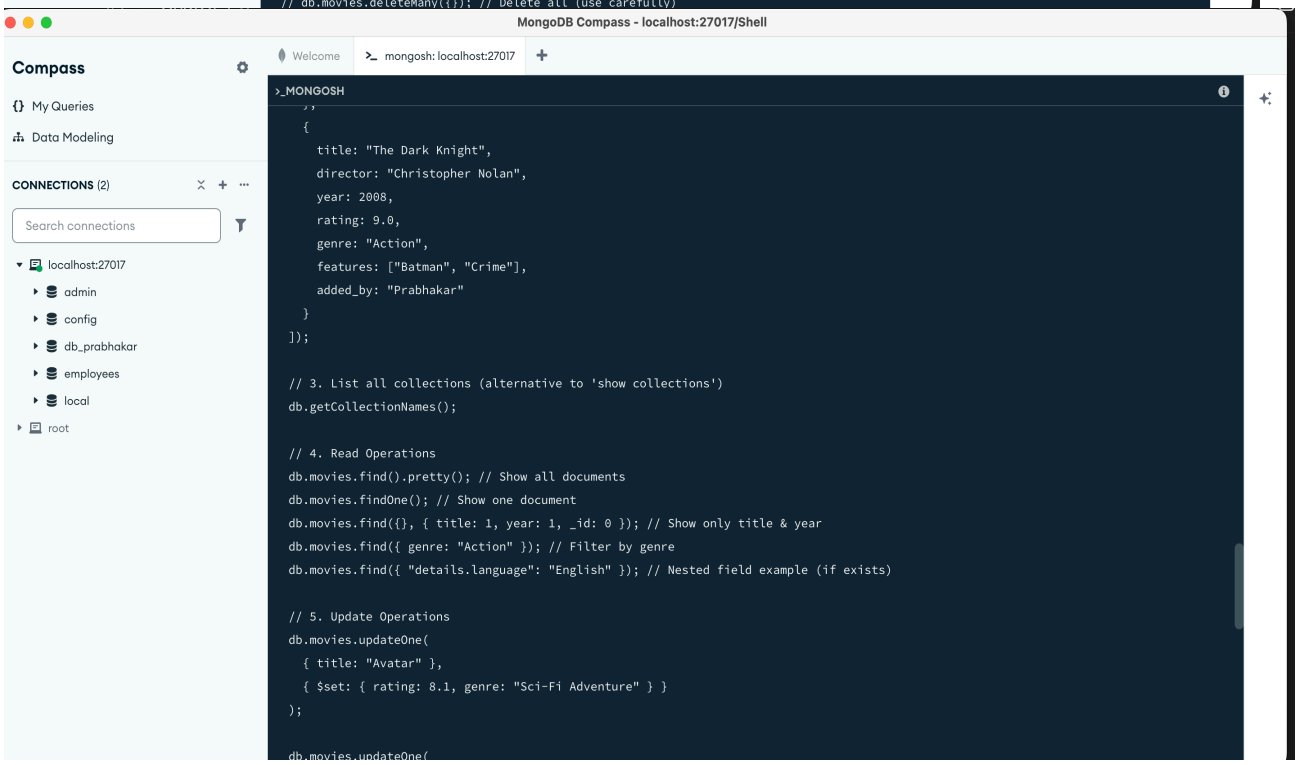
```
db.movies.updateOne(
  { title: "Inception" },
  { $pull: { features: "Action" } }
)
db.movies.updateMany(
  { director: "Christopher Nolan" },
  { $set: { language: "English" } }
)

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([
```

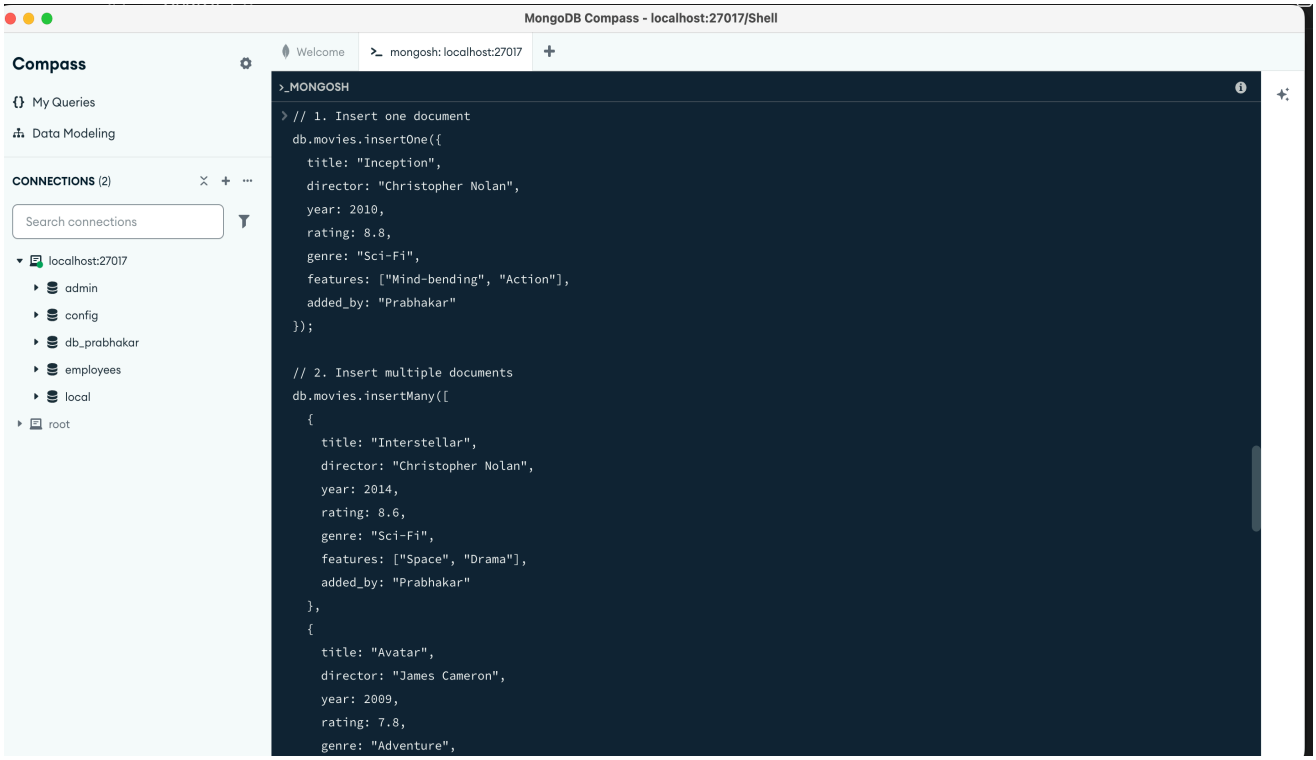
```
  { $group: { _id: "$genre", total_movies: { $sum: 1 } } }
])
// 11. Drop collection
db.movies.drop()
// 12. Drop database
db.dropDatabase()
```

4. Output:




MongoDB Compass - localhost:27017/Shell
Compass
My Queries
Data Modeling
CONNECTIONS (2)
Search connections
localhost:27017
admin
config
db_prabhakar
employees
local
root
Welcome
mongosh: localhost:27017
>_MONGOSH
{ 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: "" } }); // Remove 'added_by'
db.movies.updateMany({}, { $set: { quality: "Full HD" } }); // Add 'quality'
// Upsert: Add 'Tenet' if not exists
db.movies.updateOne(
{ title: "Tenet" },
{ $set: { director: "Christopher Nolan", year: 2020, rating: 7.5 } },
{ upsert: true }
);
// 6. Delete Operations
db.movies.deleteOne({ title: "Avatar" }); // Delete one movie
db.movies.deleteMany({ director: "Christopher Nolan" }); // Delete multiple
// db.movies.deleteMany({}); // Delete all (use carefully)


MongoDB Compass - localhost:27017/Shell
Compass
My Queries
Data Modeling
CONNECTIONS (2)
Search connections
localhost:27017
admin
config
db_prabhakar
employees
local
root
Welcome
mongosh: localhost:27017
>_MONGOSH
{
title: "The Dark Knight",
director: "Christopher Nolan",
year: 2008,
rating: 9.0,
genre: "Action",
features: ["Batman", "Crime"],
added_by: "Prabhakar"
}
]);
// 3. List all collections (alternative to 'show collections')
db.getCollectionNames();
// 4. Read Operations
db.movies.find().pretty(); // Show all documents
db.movies.findOne(); // Show one document
db.movies.find({}, { title: 1, year: 1, _id: 0 }); // Show only title & year
db.movies.find({ genre: "Action" }); // Filter by genre
db.movies.find({ "details.language": "English" }); // Nested field example (if exists)
// 5. Update Operations
db.movies.updateOne(
{ title: "Avatar" },
{ $set: { rating: 8.1, genre: "Sci-Fi Adventure" } }
);



Learning Outcomes:

- 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