

BDT Experiment 9

DATE:

Kareena Shah

60004210243

C32

Aim: Perform CRUD operations using MongoDB

Theory:

MongoDB is a popular, open-source, NoSQL database management system that uses a document-oriented data model.

Here are some features & key concepts:

- (1) Document-Oriented
- (2) Schema-less
- (3) High Performance
- (4) Horizontal Scalability
- (5) Replication
- (6) Sharding
- (7) Aggregation Framework
- (8) Indexing
- (9) Geospatial Indexing
- (10) Grid FS

CRUD operations

(1) Create

```
db.collection.insertOne({field1: value1});
```

(2) Read

```
db.collection.find({field: value});
```

(3)	Update
	<code>db.collection.updateOne({field: value});</code>
(4)	Delete
	<code>db.collection.deleteOne({field: value});</code>

Using mongo compass:

MongoDB Compass is a good GUI tool for managing mongoDB collections and doing data queries. It is free to use and can be run on macOS, Windows, and Linux.

A sample data set was created and used in order to test the functionality of the MongoDB database.

1.To show databases present in MongoDB Atlas.

`show dbs;`

```
BDA 80.00 KiB
BDA1 8.00 KiB
admin 120.00 KiB
config 60.00 KiB
local 160.00 KiB
myDB 8.00 KiB
poulations 56.00 KiB
test 80.00 KiB
```

2. To select the database.

`use admin;`

'switched to db admin'

3. To show all collections present in database.

`show collections;`

employees Student system.version

4.To create a new collection. `db.createCollection('Movie');`

{ ok: 1 }

5.To insert a single row in collection.

`db.Movie.insertOne({ID:1,Title:"Terminator",Director:"Sanjay",Year:1985, Budget:25000000});`

{ acknowledged: true,
insertedId: ObjectId("64302a42c27158d639b83646") }

`db.Movie.insertOne({ID:2,Title:"DDLJ",Director:"Karan",Year:1989,Budget: 200`

```
00000});  
{ acknowledged: true,  
insertedId: ObjectId("64302a86c27158d639b83647") }
```

6.To display values in collection. db.Movie.find({ID:2});
{ _id: ObjectId("64302a86c27158d639b83647"),

ID: 2,
Title: 'DDLJ', Director: 'Karan', Year: 1989,
Budget: 20000000 }

```
db.Movie.find();  
{ _id: ObjectId("64302a42c27158d639b83646"), ID: 1,  
Title: 'Terminator', Director: 'Sanjay', Year: 1985,  
Budget: 25000000 }  
{ _id: ObjectId("64302a86c27158d639b83647"), ID: 2,  
Title: 'DDLJ',
```

7.To insert multiple rows in collection.

```
db.Movie.insertMany([{ID:3,Title:"Dabang",Director:"salman",Year:2012,Bu dge  
t:30000000},{ID:4,Title:"Bagban",Director:"Dharmendra",Year:2010,Budget:  
150 00000}]);  
{ acknowledged: true, insertedIds:  
{ '0': ObjectId("64302b79c27158d639b83648"), '1': ObjectId("64302b79c27158d639b83649")  
}}
```

8.To display values in collection. db.Movie.find();
{ _id: ObjectId("64302a42c27158d639b83646"),
ID: 1,
Title: 'Terminator', Director: 'Sanjay', Year: 1985,
Budget: 25000000 }
{ _id: ObjectId("64302a86c27158d639b83647"), ID: 2,
Title: 'DDLJ', Director:
'Karan',
Year: 1989,
Budget: 20000000 }
{ _id: ObjectId("64302b79c27158d639b83648"), ID: 3,
Title: 'Dabang', Director: 'salman', Year: 2012,
Budget: 30000000 }
{ _id: ObjectId("64302b79c27158d639b83649"), ID: 4,
Title: 'Bagban',
Director: 'Dharmendra', Year: 2010,
Budget: 15000000 }

9.To update value in the collection.

```
db.Movie.updateOne({'Title':'Terminator'},{$set: {'Title':'Robot'  
}}); { acknowledged: true,  
insertedId: null, matchedCount: 1, modifiedCount: 1,
```

```

upsertedCount: 0 }
db.Movie.find();
{ _id: ObjectId("64302a42c27158d639b83646"), ID:
1,
Title: 'Robot', Director:
'Sanjay', Year: 1985,
Budget: 25000000 }
{ _id: ObjectId("64302a86c27158d639b83647"), ID: 2,
Title: 'DDLJ', Director: 'Karan', Year: 1989,
Budget: 20000000 }
{ _id: ObjectId("64302b79c27158d639b83648"), ID: 3,
Title: 'Dabang', Director: 'salman', Year: 2012,
Budget: 30000000 }
{ _id: ObjectId("64302b79c27158d639b83649"), ID: 4,
Title: 'Bagban', Director: 'Dharmendra', Year: 2010,
Budget: 15000000 }
db.Movie.find({$and:[{Year:{$lt:"2015-01-01"}},{Budget: {$gt:500000}}]})

```

9.To find maximum budget movies group by year.

```

db.Movie.aggregate([{$group : { _id : "$Year", num_tutorial : {$max :
"$Budget"}}}])
{ _id: 1985, num_tutorial: 25000000 }
{ _id: 1989, num_tutorial: 20000000 }
{ _id: 2010, num_tutorial: 15000000 }
{ _id: 2012, num_tutorial: 30000000 }

```

Conclusion : Thus, we understood & implemented
CRUD operations in MongoDB
