Assignment - B1

Title :- study of open source NoSQL database MongoDB (Installation, Basic CRUD operation execution).

Problem Statement :-
Implement database with suitable example using mongoDB & implement study of open source NoSQL database: MongoDB (Installation, Basic CRUD operat^n).

Objective :- • Understand concept of NoSQL DB.
• Understand concept of mongoDB with CRUD operation.
• Understand the basic installation & administrative commands of mongoDB.

S/W & H/W :- MongoDB, Fedora os.

Outcome :- - Implement the commands.
- Implement the database in mongoDB.

Theory :-
MongoDB is an open-source document database that provides high availability, & automatic scaling.

Document database

A record in mongoDB is a document which is a data structure compared of field & value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays & arrays of documents

e.g. {
name : "Sue", ← Field : Value
age : 26,
status : "A",
groups : ["news", "sports"]
}

Create operation

create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection.

- db collection insertOne()
- db collection insertMany()

In mongoDB, insert operations target a single collection. All write operations in mongoDB are atomic on the level of single document. collection,

Collection is a group of MongoDB document. It is equivalent of an RDBMS table. A collection exists within single database. collection do not enforce a schema. documents within a collection can have different fields in a collection are of single related purpose.

Document - A document is set of key value pair document have dynamic schema. Dynamic schema means that documents in the same collection do not need to have same set of fields.

Conclusion :

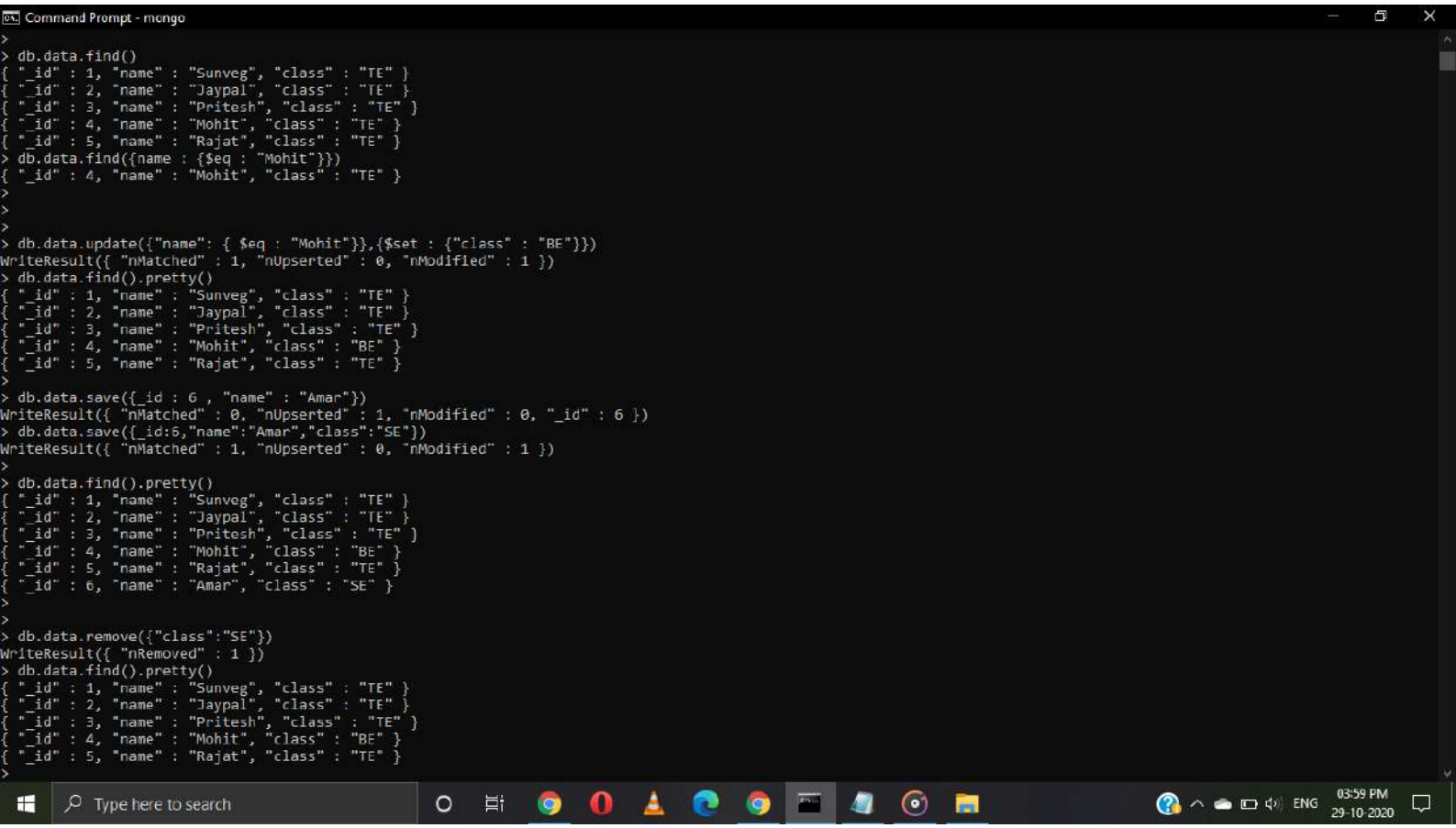
Successfully implemented mongoDB ~~to~~ CRUD operations & installation of mongoDB on system.

```
> show dbs
admin      0.000GB
conFusion  0.000GB
config     0.000GB
local      0.000GB
> use student
switched to db student
> db
student
> db.data.insert({ _id : 1 , "name" : "Sunveg" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 2 , "name" : "Jaypal" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 3 , "name" : "Pritesh" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 4 , "name" : "Mohit" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 5 , "name" : "Rajat" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
>
> db.data.find()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
> db.data.find({name : {$eq : "Mohit"}})
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
>
>
>
> db.data.update({"name": { $eq : "Mohit"}},{$set : {"class" : "BE"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
>
> db.data.save({_id : 6 , "name" : "Amar"})
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 6 })
> db.data.save({_id:6,"name":"Amar","class":"SE"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
{ "_id" : 6, "name" : "Amar", "class" : "SE" }
>
>
```

```
> db.data.remove({"class":"SE"})
WriteResult({ "nRemoved" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
>
```

Command Prompt - mongo

```
> show dbs
admin      0.000GB
confusion  0.000GB
config     0.000GB
local      0.000GB
> use student
switched to db student
> db
student
> db.data.insert({ _id : 1 , "name" : "Sunveg" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 2 , "name" : "Jaypal" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 3 , "name" : "Pritesh" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 3 , "name" : "Mohit" , "class" : "TE" })
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: student.data index: _id_ dup key: { _id: 3.0 }"
        }
})
> db.data.insert({ _id : 4 , "name" : "Mohit" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
> db.data.insert({ _id : 5 , "name" : "Rajat" , "class" : "TE" })
WriteResult({ "nInserted" : 1 })
>
> db.data.find()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
> db.data.find({name : {$eq : "Mohit"}})
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
>
>
>
> db.data.update({"name": { $eq : "Mohit"}},{$set : {"class" : "BE"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
```


Command Prompt - mongo
> db.data.find()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
> db.data.find({name : {$eq : "Mohit"}})
{ "_id" : 4, "name" : "Mohit", "class" : "TE" }
> db.data.update({"name": { $eq : "Mohit"}},{$set : {"class" : "BE"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
> db.data.save({_id : 6 , "name" : "Amar"})
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 6 })
> db.data.save({_id:6,"name":"Amar","class":"SE"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
{ "_id" : 6, "name" : "Amar", "class" : "SE" }
> db.data.remove({"class":"SE"})
WriteResult({ "nRemoved" : 1 })
> db.data.find().pretty()
{ "_id" : 1, "name" : "Sunveg", "class" : "TE" }
{ "_id" : 2, "name" : "Jaypal", "class" : "TE" }
{ "_id" : 3, "name" : "Pritesh", "class" : "TE" }
{ "_id" : 4, "name" : "Mohit", "class" : "BE" }
{ "_id" : 5, "name" : "Rajat", "class" : "TE" }
Type here to search
ENG
03:59 PM
29-10-2020