	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri** **Information Technology Department**		
Subject name: Big Data Lab			Subject Code: ITL801
Class	BE IT	Semester – VIII (CBSGS)	Academic year: 2019-20
Name of Student	**Kazi Jawwad A Rahim**	**QUIZ Score :**	
Roll No	**28**	Assignment/Experiment No.	04

Title: Installations of MongoDB NoSQL database and CRUD Operations

1. Course objectives applicable

COB4. Understand cloud implementation, programming and mobile cloud computing

COB3.Understand different cloud computing services

2. Course outcomes applicable:

CO3 –To use and define different cloud computing services.

CO6-To memorize cloud computing, memorize different cloud services and cloud deployment models.

3. Learning Objectives:

1. Create Connection String for MongoDB
2. Perform CRUD Operations on MongoDB

4. Practical applications of the assignment/experiment:

MongoDB is a leading Document Database used in Unstructured Data Processing

5. Prerequisites:

1. Knowledge of NoSQL Database
2. Internet Access
3. MongoDB Server Setup

6. Hardware Requirements:

1. Internet Access with Browser
2. PC to install MongoDB

7. Software Requirements:

Browser like Chrome, Internet Explorer Edge, MongoDB Server Setup

8. Experiment/Assignment Evaluation:

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2

Date of performance (DOP)	**05/02/2020**	**Total marks obtained**		**10**
Date of checking (DOC)		**Signature of teacher**		

10. Implementation steps–
Executing NoSql Commands on MongoDB shell

```
ssh test@172.16.5.154

Password-test

Type mongo
```

Now run CRUD operations on shell

```
To see the list of databases in the system:
>show dbs
```

To select a database

```
> use movies
switched to db
movies

> db.comedy.insert({name:"Sairat", year:2016})
WriteResult({ "nInserted" : 1 })

> db.comedy.insert({name:'The School of Rock',year:2003})
WriteResult({ "nInserted" : 1 })

> db.comedy.find()
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016}
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }

> db.comedy.find().limit(1)
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016}

> db.comedy.findOne()
{
"_id" : ObjectId("59acf61f5726b577df63e869"),
"name" : "Sairat",
"year" : 2016
}

> db.comedy.find({year:{$gt:1994}})
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016}
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }

> db.comedy.find({year:{'$gt':1994}}, {name:true})
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat" }
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock" }

> db.comedy.insert({name:"Bill & Ted's Excellent Adventure", year:1989})
WriteResult({ "nInserted" : 1 })
```

Now Update a document

```
> db.comedy.update({name:"Bill & Ted's Excellent Adventure"},
{'$set':{director:'Stephen Herek',cast:['Keanu Reeves', 'Alex
Winter']}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }

> db.comedy.find()
{ "_id" : ObjectId("59acf61f5726b577df63e869"), "name" : "Sairat", "year" : 2016}
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
```

```
{ "_id" : ObjectId("59acf6ba5726b577df63e86b"), "name" : "Bill & Ted's Excellent
Adventure", "year" : 1989, "director" : "Stephen Herek", "cast" : ["Keanu
Reeves", "Alex Winter" ] }
```

Now remove a document

```
> db.comedy.remove({name:'Sairat'})
WriteResult({ "nRemoved" : 1 })

> db.comedy.find()
{ "_id" : ObjectId("59acf6395726b577df63e86a"), "name" : "The School of Rock",
"year" : 2003 }
{ "_id" : ObjectId("59acf6ba5726b577df63e86b"), "name" : "Bill & Ted's Excellent
Adventure", "year" : 1989, "director" : "Stephen Herek", "cast" : [ "Keanu
Reeves", "Alex Winter" ] }

> db.comedy.count()
2
```

Now drop a collection

```
> db.comedy.drop()
true

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

11. Results:

CREATE

```
> use JK
switched to db JK
```

```
> db.JK.insert({name:"sairat",year:2016})
WriteResult({ "nInserted" : 1 })
```

```
> db.JK.insert({name:'The School of Rock', year:2003})
WriteResult({ "nInserted" : 1 })
```

```
> db.JK.insert({name:"Bill & Ted's Excellent Adventure", year:1989})
WriteResult({ "nInserted" : 1 })
```

READ

```
> db.JK.find()
{ "_id" : ObjectId("5e3a643865a4956bbf43a74d"), "name" : "sairat", "year" : 2016 }
{ "_id" : ObjectId("5e3a64b465a4956bbf43a74e"), "name" : "The School of Rock", "year" : 2003 }
{ "_id" : ObjectId("5e3a64d365a4956bbf43a74f"), "name" : "The School of Rock" }
{ "_id" : ObjectId("5e3a650765a4956bbf43a750"), "name" : "Bill & Ted's Excellent Adventure", "year" : 1989 }
```

```
> db.JK.find().limit(1)
{ "_id" : ObjectId("5e3a643865a4956bbf43a74d"), "name" : "sairat", "year" : 2016 }
```

```
> db.JK.findOne()
{
        "_id" : ObjectId("5e3a643865a4956bbf43a74d"),
        "name" : "sairat",
        "year" : 2016
}
```

```
> db.JK.find({year:{$gt:1994}})
{ "_id" : ObjectId("5e3a643865a4956bbf43a74d"), "name" : "sairat", "year" : 2016 }
{ "_id" : ObjectId("5e3a64b465a4956bbf43a74e"), "name" : "The School of Rock", "year" : 2003 }
```

```
> db.JK.find({year:{'$gt':1994}}, {name:true})
{ "_id" : ObjectId("5e3a643865a4956bbf43a74d"), "name" : "sairat" }
{ "_id" : ObjectId("5e3a64b465a4956bbf43a74e"), "name" : "The School of Rock" }
```

UPDATE

```
> db.JK.update({name:"Bill & Ted's Excellent Adventure"}, {'$set': {director:'Stephen Herek', cast:['Keanu Reeves', 'Alex Winter']}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.JK.find()
{ "_id" : ObjectId("5e3a643865a4956bbf43a74d"), "name" : "sairat", "year" : 2016 }
{ "_id" : ObjectId("5e3a64b465a4956bbf43a74e"), "name" : "The School of Rock", "year" : 2003 }
{ "_id" : ObjectId("5e3a64d365a4956bbf43a74f"), "name" : "The School of Rock" }
{ "_id" : ObjectId("5e3a650765a4956bbf43a750"), "name" : "Bill & Ted's Excellent Adventure", "year" : 1989, "cast" : [ "Keanu Reeves", "Alex Winter" ], "director" : "St
ephen Herek" }
```

DELETE

```
> db.JK.remove({name:'sairat'})
WriteResult({ "nRemoved" : 1 })
```

```
> db.JK.find()
{ "_id" : ObjectId("5e3a64b465a4956bbf43a74e"), "name" : "The School of Rock", "year" : 2003 }
{ "_id" : ObjectId("5e3a64d365a4956bbf43a74f"), "name" : "The School of Rock" }
{ "_id" : ObjectId("5e3a650765a4956bbf43a750"), "name" : "Bill & Ted's Excellent Adventure", "year" : 1989, "cast" : [ "Keanu Reeves", "Alex Winter" ], "director" : "St
ephen Herek" }
```

```
> db.JK.count()
3
```

```
> db.JK.drop()
true
```

```
> show dbs
```

```
mongo[test@localhost ~]$ mongo
```

References :

[1] http://nosql-database.org/ "NoSQL DEFINITION: Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable"

[2]Leavitt, Neal (2010). "Will NoSQL Databases Live Up to Their Promise?" (PDF). IEEE Computer.

[3] Mohan, C. (2013). History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla. Proc. 16th Int'l Conf. on Extending Database Technology.

[4] "NOSQL meetup Tickets, Thu, Jun 11, 2009 at 10:00 AM". Eventbrite.com. Retrieved 2017-03-06.

[5] "Amazon Goes Back to the Future With 'NoSQL' Database". WIRED. 2012-01-19. Retrieved 2017-03-06.

[6] "RDBMS dominate the database market, but NoSQL systems are catching up". DB-Engines.com. 21 Nov 2013. Retrieved 24 Nov 2013.

[7] "NoSQL (Not Only SQL)". NoSQL database, also called Not Only SQL

[8] Fowler, Martin. "NoSQL Definition". many advocates of NoSQL say that it does not mean a "no" to SQL, rather it means Not Only SQL.