



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 (CBCGS)	Academic year: 2019-20
Name of Student	Kazi Jawwad A Rahim		QUIZ Score :
Roll No	28	Assignment/Experiment No.	04
<b>Title:</b> Installations of MongoDB NoSQL database and CRUD Operations			
<b>1. Course objectives applicable</b> <b>COB4.</b> Understand cloud implementation, programming and mobile cloud computing <b>COB3.</b> Understand different cloud computing services			
<b>2. Course outcomes applicable:</b> <b>CO3</b> –To use and define different cloud computing services. <b>CO6</b> –To memorize cloud computing, memorize different cloud services and cloud deployment models.			
<b>3. Learning Objectives:</b> 1. Create Connection String for MongoDB 2. Perform CRUD Operations on MongoDB			
<b>4. Practical applications of the assignment/experiment:</b> MongoDB is a leading Document Database used in Unstructured Data Processing			
<b>5. Prerequisites:</b> 1. Knowledge of NoSQL Database 2. Internet Access 3. MongoDB Server Setup			
<b>6. Hardware Requirements:</b> 1. Internet Access with Browser 2. PC to install MongoDB			
<b>7. Software Requirements:</b> Browser like Chrome, Internet Explorer Edge, MongoDB Server Setup			
<b>8. Experiment/Assignment Evaluation:</b>			
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 -	05	6
2	Neatness/presentation	01	2
3	Punctuality	01	2
Date of performance (DOP)	05/02/2020	Total marks obtained	07
Date of checking (DOC)	07-02-2020	Signature of teacher	



## Theory:

A NoSQL (originally referring to ~~NoSQL~~ "non SQL" or "non relational") database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Such databases have existed since the late 1960s, but did not obtain the "NoSQL" moniker until a surge of popularity in the early twenty-first century, triggered by the needs of <sup>web</sup> 2.0 companies such as Facebook, Google and Amazon.com. NoSQL databases are increasingly used in Big Data and real time web applications. NoSQL systems are also sometimes called as "Not Only SQL" to emphasize that they may support SQL-like query languages.

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc., and is published under a combination of the GNU Affero General Public License and Apache License.

There have been various approaches to classify NoSQL databases, each with different categories and sub-categories, some of which overlap. What follows is a basic classification by data model, with examples.



- **Column:** Accumulo, Cassandra, Druid, HBase, Vertica, SAP HANA.
- **Document:** Apache CouchDB, ArangoDB, Clusterpoint, CouchBase, cosmosDB, HyperDex, IBM Domino, MarkLogic, MongoDB, OrientDB, Qizx, RethinkDB.
- **Key-value:** Aerospike, ArangoDB, Couchbase, Dynamo, FairCom CtreeACE, FoundationDB, HyperDex, InfinityDB, MemcachedDB, Numps, Oracle NoSQL Database, OrientDB, Redis, Riak, BerkeleyDB, SDBM/Flat file, dbm.
- **Graph:** AllegroGraph, ArangoDB, InfiniteGraph, Apache Graph, MarkLogic, Neo4j, OrientDB, virtuoso.
- **Multi-Model:** ArangoDB, Couchbase, FoundationDB, InfinityDB, MarkLogic, OrientDB.

### CAP Theorem:

Eric Brewer, a professor at the University of California, Berkeley and co-founder and chief scientist at Inktomi, made the conjecture that web services cannot ensure all three of the following properties at once.

**Consistency:** The client perceives that a set of operation has occurred all at once.

**Availability:** Every operation must terminate in an intended response.

**Partition Tolerance:** Operations will complete, even if individual components are unavailable.



specifically, a web application can support, at most, only two of these properties, with any database design. Obviously any horizontal scaling strategy is based on data partitioning. Therefore, designers are forced to decide between consistency and availability.

The BASE acronym is used to describe the properties of certain databases, usually NoSQL databases. It's often referred to as the opposite of ACID. In the NoSQL world, ACID transactions are less fashionable as some databases have loosened the requirement for immediate consistency, data freshness and accuracy in order to gain other benefits like scale and resilience.

This leaves me with some questions about the definition

Basically Available, Soft state, Eventual Consistency.  
Basic availability: The database appears to work most of the time.

Soft state: Stores don't have to be write consistent, nor do different replicas have to be mutually consistent all the time.

Eventual consistency: Stores exhibit consistency at some later point.

BASE properties are much looser than ACID guarantees, but there isn't a direct one-to-one mapping between the two consistency models.

## **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("5e3a64d3e5a4956bbf43a74f"), "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:{>: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("5e3a64d368a4956bbf43a74f"), "name" : "The School of Rock" }
{ "_id" : ObjectId("5e3a650765a4956bbf43a750"), "name" : "Bill & Ted's Excellent Adventure", "year" : 1989, "cast" : [ "Keanu Reeves", "Alex Winter" ], "director" : "Stephen 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("5e3a64d368a4956bbf43a74f"), "name" : "The School of Rock" }
{ "_id" : ObjectId("5e3a650765a4956bbf43a750"), "name" : "Bill & Ted's Excellent Adventure", "year" : 1989, "cast" : [ "Keanu Reeves", "Alex Winter" ], "director" : "Stephen Herek" }

> db.JK.count()
3

> db.JK.drop()
true

> show dbs
mongodbs

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 Nonsense 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.



### Learning Outcomes Achieved:

- 1) MongoDB as NoSQL is discussed.
- 2) Students downloaded and installed MongoDB server, Students configured MongoDB Database.
- 3) SQL queries for CRUD operations were tested on the database.
- 4) CAP Theorem and BASE properties of NoSQL database are discussed in connection with MongoDB with Big Data Analytics.

### Conclusion:

- 1) Applications of the studied technique in industry
  - a. MongoDB is one of the commonly used document store.
  - b. BASE properties are of relevance in view of Big Data analytics.
- 2) Engineering Relevance
  - a. Distributed databases for storage of large volumes of data.
  - b. NoSQL for Distributed Databases.

### 3) Skills Developed

- a. Configuration of MongoDB.
- b. CRUD commands on MongoDB.