**DBMS Lab**

**7358 Umang Kumar**

**Assignment No. 9 (Install & study MongoDB and Implement CRUD Operations)**

Title: Study of Open Source NOSQL Database: MongoDB (Installation, Basic CRUD operations, Execution).

Objective:

* To learn and understand NOSQL Database.
* To execute CRUD Operations on MongoDB.

MongoDB shell version v4.4.17

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session {"id" : UUID("26501b3a-16b3-4663-adb2-9463c8597144") }

MongoDB server version: 4.4.17

Welcome to the MongoDB shell.

For interactive help, type "help".

For more comprehensive documentation, see

https://docs.mongodb.com/

Questions? Try the MongoDB Developer Community Forums

https://community.mongodb.com

---

The server generated these startup warnings when booting:

2022-10-27T17:42:48.501+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> db

test

> use umang

switched to db umang

> show collections

> db.create collection('Student');

uncaught exception: SyntaxError: unexpected token: identifier :

@(shell):1:10

> db.createcollection('Student');

uncaught exception: TypeError: db.createcollection is not a function :

@(shell):1:1

> db.create collection('Student')

uncaught exception: SyntaxError: unexpected token: identifier :

@(shell):1:10

> umang.create collection('Student');

uncaught exception: SyntaxError: unexpected token: identifier :

@(shell):1:14

> use umang

switched to db umang

> db.createCollection('Student');

{ "ok" : 1 }

> db.Student.insert({'Rno':'1','Name':'Ritika',Branch':'TE COMP'});

uncaught exception: SyntaxError: missing : after property id :

@(shell):1:51

> db.Student.insert({'Rno':'1','Name':'Umang,'Branch':'TE COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'2','Name':'Charcha,'Branch':'TE COMP'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'3','Name':'Denver,'Branch':'TE ENTC'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'4','Name':'Barry,'Branch':'TE Comp'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'5','Name':'Shri','Branch':'TE Comp'});

WriteResult({ "nInserted" : 1 })

> db.Student.insert({'Rno':'6','Name':'Shristy','Branch':'TE Comp'});

WriteResult({ "nInserted" : 1 })

> db.Student.find();

{ "\_id" : ObjectId("635a78895b11d62aed87b1de"), "Rno" : "1", "Name" : "Umang", "Branch" : "TE COMP" }

{ "\_id" : ObjectId("635a789f5b11d62aed87b1df"), "Rno" : "2", "Name" : "Charcha", "Branch" : "TE COMP" }

{ "\_id" : ObjectId("635a78b05b11d62aed87b1e0"), "Rno" : "3", "Name" : "Denver", "Branch" : "TE ENTC" }

{ "\_id" : ObjectId("635a78d65b11d62aed87b1e1"), "Rno" : "4", "Name" : "Barry", "Branch" : "TE Comp" }

{ "\_id" : ObjectId("635a78ea5b11d62aed87b1e2"), "Rno" : "5", "Name" : "Shri", "Branch" : "TE Comp" }

{ "\_id" : ObjectId("635a78fc5b11d62aed87b1e3"), "Rno" : "6", "Name" : "Shristy", "Branch" : "TE Comp" }

> db.Student.find().pretty();

{

"\_id" : ObjectId("635a78895b11d62aed87b1de"),

"Rno" : "1",

"Name" : "Umang",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a789f5b11d62aed87b1df"),

"Rno" : "2",

"Name" : "Charcha",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a78b05b11d62aed87b1e0"),

"Rno" : "3",

"Name" : "Denver",

"Branch" : "TE ENTC"

}

{

"\_id" : ObjectId("635a78d65b11d62aed87b1e1"),

"Rno" : "4",

"Name" : "Barry",

"Branch" : "TE Comp"

}

{

"\_id" : ObjectId("635a78ea5b11d62aed87b1e2"),

"Rno" : "5",

"Name" : "Shri",

"Branch" : "TE Comp"

}

{

"\_id" : ObjectId("635a78fc5b11d62aed87b1e3"),

"Rno" : "6",

"Name" : "Shristy",

"Branch" : "TE Comp"

}

> show dbs;

admin 0.000GB

config 0.000GB

local 0.000GB

ritika 0.000GB

> db.Student.update({'Name':'Shristy'},{$set:{'Name':'Amatya'}});

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db,Student.find().pretty();

uncaught exception: ReferenceError: Student is not defined :

@(shell):1:4

> db.Student.find().pretty();

{

"\_id" : ObjectId("635a78895b11d62aed87b1de"),

"Rno" : "1",

"Name" : "Umang",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a789f5b11d62aed87b1df"),

"Rno" : "2",

"Name" : "Charcha",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a78b05b11d62aed87b1e0"),

"Rno" : "3",

"Name" : "Denver",

"Branch" : "TE ENTC"

}

{

"\_id" : ObjectId("635a78d65b11d62aed87b1e1"),

"Rno" : "4",

"Name" : "Barry",

"Branch" : "TE Comp"

}

{

"\_id" : ObjectId("635a78ea5b11d62aed87b1e2"),

"Rno" : "5",

"Name" : "Shri",

"Branch" : "TE Comp"

}

{

"\_id" : ObjectId("635a78fc5b11d62aed87b1e3"),

"Rno" : "6",

"Name" : "Amatya",

"Branch" : "TE Comp"

}

> db.Student.remove({'Name':'Amatya'},1);

WriteResult({ "nRemoved" : 1 })

> db.Student.remove({'Name':'Amatya'},1);

WriteResult({ "nRemoved" : 0 })

> db.Student.find().pretty();

{

"\_id" : ObjectId("635a78895b11d62aed87b1de"),

"Rno" : "1",

"Name" : "Umang",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a789f5b11d62aed87b1df"),

"Rno" : "2",

"Name" : "Charcha",

"Branch" : "TE COMP"

}

{

"\_id" : ObjectId("635a78b05b11d62aed87b1e0"),

"Rno" : "3",

"Name" : "Denver",

"Branch" : "TE ENTC"

}

{

"\_id" : ObjectId("635a78d65b11d62aed87b1e1"),

"Rno" : "4",

"Name" : "Barry",

"Branch" : "TE Comp"

}

{

"\_id" : ObjectId("635a78ea5b11d62aed87b1e2"),

"Rno" : "5",

"Name" : "Shri",

"Branch" : "TE Comp"

}

> db.Student.drop();

true

> db.Student.find().pretty();