Assignment – B1

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

```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use I2IT;
switched to db I2IT
> db.student.insert({"id":"TI001","Name":"ABC","Credits":7.8});
WriteResult({ "nInserted" : 1 })
> show dbs;
I2IT    0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> show connections
uncaught exception: Error: don't know how to show [connections] :
shellHelper.show@src/mongo/shell/utils.js:1191:11
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> ^C
bye

C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.1
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=m
ongodb
Implicit session: session { "id" : UUID("eb610cbf-bc3a-4d47-841f-
589312575a6b") }
MongoDB server version: 4.4.1
---
The server generated these startup warnings when booting:
        2020-09-21T11:53:15.681+05:30: ***** SERVER RESTARTED *****
        2020-09-21T11:53:16.619+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;
I2IT    0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> show collections;
> ^C
bye

C:\Program Files\MongoDB\Server\4.4\bin>mongo
MongoDB shell version v4.4.1
connecting to:
mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=m
ongodb
Implicit session: session { "id" : UUID("36adbfff-99ce-4452-8d70-
359e75dd3511") }
MongoDB server version: 4.4.1
---
The server generated these startup warnings when booting:
        2020-09-21T11:53:15.681+05:30: ***** SERVER RESTARTED *****
        2020-09-21T11:53:16.619+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()
---
> use I2IT;
switched to db I2IT
> show collections;
student
>
db.student.insert({"id":"TI001","Name":"ABC","Credits":7.8,"City":"
Pune"});
WriteResult({ "nInserted" : 1 })
> stud={"id":"TI002","Name":"PQR","Credits":6.5,"City":"Pune"};
{ "id" : "TI002", "Name" : "PQR", "Credits" : 6.5, "City" : "Pune"
}
> db.student.insert(stud);
WriteResult({ "nInserted" : 1 })
> db.student.find();
{ "_id" : ObjectId("5f68483fc18652f05ff5e2ff"), "id" : "TI001",
"Name" : "ABC", "Credits" : 7.8 }
{ "_id" : ObjectId("5f6848d1d270133c4c90808b"), "id" : "TI001",
"Name" : "ABC", "Credits" : 7.8, "City" : "Pune" }
{ "_id" : ObjectId("5f68498ad270133c4c90808c"), "id" : "TI002",
"Name" : "PQR", "Credits" : 6.5, "City" : "Pune" }
> db.student.find().pretty();
{
        "_id" : ObjectId("5f68483fc18652f05ff5e2ff"),
        "id" : "TI001",
        "Name" : "ABC",
        "Credits" : 7.8
}
{
        "_id" : ObjectId("5f6848d1d270133c4c90808b"),
        "id" : "TI001",
        "Name" : "ABC",
        "Credits" : 7.8,
        "City" : "Pune"
}
{
        "_id" : ObjectId("5f68498ad270133c4c90808c"),
        "id" : "TI002",
        "Name" : "PQR",
        "Credits" : 6.5,
        "City" : "Pune"
}
> use I2IT;
switched to db I2IT
> show collections;
```

```
student
>
db.runCommand({collMod:"student",validator:{$or:[{id:{$exists:true}
},{Name:{$exists:true)}}]},validationLevel:"moderate"})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:88
>
db.runCommand({collMod:"student",validator:{$or:[{id:{$exists:true}
},{Name:{$exists:true)}}]},validationLevel:"moderate"})
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:88
>
db.runCommand({collMod:"student",validator:{$or:[{id:{$exists:true}
},{Name:{$exists:true)}}]},validationLevel:"moderate"});
uncaught exception: SyntaxError: missing } after property list :
@(shell):1:88
>
db.runCommand({collMod:"student",validator:{$or:[{id:{$exists:true}
},{Name:{$exists:true)}}]},validationLevel:"moderate"})
```