Experiment No.3
To install and configure MongoDB to execute NoSQL commands
Date of Performance:
Date of Submission:

Aim: To install and configure MongoDB/ Cassandra/ HBase/ Hypertable and to execute NoSQL commands.

Theory:

MongoDB can be downloaded from https://www.mongodb.com/try/download/community2
Now open command prompt and run the following command:

```
C:\>move mongodb-win64-* mongodb
        1 dir(s) moved.
```

MongoDB requires a data folder to store its files. The default location for the MongoDB data directory is c:\data\db. So create the folder using the Command Prompt. Execute the following command sequence.

```
C:\>md data
C:\md data\db
```

In case mongodb is stored in some other location, navigate to that folder.

In command prompt navigate to the bin directory present into the mongodb installation folder. Suppose the installation folder is D:\set up\mongodb

```
C:\Users\XYZ>d:
D:\>cd "set up"
D:\set up>cd mongodb
D:\set up\mongodb>cd bin
D:\set up\mongodb\bin>mongod.exe --dbpath "d:\set up\mongodb\data"
```

Now to run the mongodb, open another command prompt and issue the following command:

```
D:\set up\mongodb\bin>mongo.exe
MongoDB shell version: 2.4.6
connecting to: test
>db.test.save( { a: 1 } )
>db.test.find()
{ "_id" : ObjectId(5879b0f65a56a454), "a" : 1 }
>
```

1. **The Use Command**

   MongoDB use DATABASE_NAME is used to create database. The command will create a new database, if it doesn't exist otherwise, it will return the existing database

   Syntax:

   use DATABASE_NAME

2. **The dropDatabase () Method**

   MongoDB db.dropDatabase () command is used to drop an existing database.

   Syntax:

   db.dropDatabase()

3. **The createCollection() Method**

   MongoDB db.createCollection(name, options) is used to create collection.

   Syntax:

   db.createCollection(name, options)

4. **Insert Document**

   To insert data into MongoDB collection, you need to use MongoDB's insert() or save()method

   Syntax:

   db.COLLECTION_NAME.insert(document)

Example:

```
>db.post.insert( [ {
 title: 'MongoDB Overview',
 description: 'MongoDB is no sql database',
tags: ['mongodb', 'database', 'NoSQL'],
likes: 100
}, {
 title: 'NoSQL Database',
 description: 'NoSQL database doesn't have tables',
tags: ['mongodb', 'database', 'NoSQL'], likes: 20,
comments: [ { user:'user1',
 message: 'My first comment',
dateCreated: new Date(2022,11,10,2,35),
like: 0
 } ] } ] )
```

Creating sample document:

Example:

Suppose a client needs a database design for his blog website. Website has the following requirements.

- Every post has the unique title, description and url.
- Every post can have one or more tags.
- Every post has the name of its publisher and total number of likes.
- Every Post have comments given by users along with their name, message, data-time and likes.
- On each post there can be zero or more comments.

Document:

```
{
 _id:POST_ID
title: TITLE_OF_POST,
description:   POST_DESCRIPTION,
by: POST_BY,
url: URL_OF_POST,
tags: [TAG1, TAG2, TAG3],
likes: TOTAL_LIKES,
comments: [ {
 user:'COMMENT_BY',
message:TEXT,
dateCreated:   DATE_TIME,
like: LIKES
 }, {
 user:'COMMENT_BY',
message:TEXT,
dateCreated:   DATE_TIME,
like: LIKES
 } ] }
```

Output:

Show Databases:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>mongosh
Current Mongosh Log ID: 6527d6c3deeb2dc83a6f3dbf
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.2
Using Mongosh:          2.0.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-10-12T16:33:09.634+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> show dbs
admin    40.00 KiB
config  108.00 KiB
db2       8.00 KiB
local    40.00 KiB
psm      40.00 KiB
test>
```

Create Collections:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

psm> db.createCollection("C2")
{ ok: 1 }
psm> db.createCollection("C3")
{ ok: 1 }
psm> db.createCollection("C4")
{ ok: 1 }
psm> db.createCollection("C5")
{ ok: 1 }
psm>
```

List Collections:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

psm> db.createCollection("C2")
{ ok: 1 }
psm> db.createCollection("C3")
{ ok: 1 }
psm> db.createCollection("C4")
{ ok: 1 }
psm> db.createCollection("C5")
{ ok: 1 }
psm> show collections
C1
C2
C3
C4
C5
test
psm>
```

Insert Document in Collection:

```
psm> db.C1.insert({Id:1, First_Name:'Prathmesh', Last_Name:'Malvi', Email:'Prathmeshmalvi10@gmail.com', Address:'Jambhulpada'})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6527dd7fdeeb2dc83a6f3dc0") }
}
psm> db.C1.insert({Id:2, First_Name:'Rutuja', Last_Name:'Malvi', Email:'Rutujamalvi10@gmail.com', Address:'Jambhulpada'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6527ddb1deeb2dc83a6f3dc1") }
}
psm> db.C1.insert({Id:3, First_Name:'Suresh', Last_Name:'Malvi', Email:'Sureshmalvi10@gmail.com', Address:'Jambhulpada'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6527ddd2deeb2dc83a6f3dc2") }
}
psm>
```

Get Collections:

```
psm> db.C1.find().pretty()
[
  {
    _id: ObjectId("6527dd7fdeeb2dc83a6f3dc0"),
    Id: 1,
    First_Name: 'Prathmesh',
    Last_Name: 'Malvi',
    Email: 'Prathmeshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddb1deeb2dc83a6f3dc1"),
    Id: 2,
    First_Name: 'Rutuja',
    Last_Name: 'Malvi',
    Email: 'Rutujamalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddd2deeb2dc83a6f3dc2"),
    Id: 3,
    First_Name: 'Suresh',
    Last_Name: 'Malvi',
    Email: 'Sureshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  }
]
psm>
```

Update Collection:

```
psm> db.C1.update({First_Name:'Suresh'},{$set:{First_Name:'Geeta'}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
psm> db.C1.find().pretty()
[
  {
    _id: ObjectId("6527dd7fdeeb2dc83a6f3dc0"),
    Id: 1,
    First_Name: 'Prathmesh',
    Last_Name: 'Malvi',
    Email: 'Prathmeshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddb1deeb2dc83a6f3dc1"),
    Id: 2,
    First_Name: 'Rutuja',
    Last_Name: 'Malvi',
    Email: 'Rutujamalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddd2deeb2dc83a6f3dc2"),
    Id: 3,
    First_Name: 'Geeta',
    Last_Name: 'Malvi',
    Email: 'Sureshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  { _id: ObjectId("6527e02fdeeb2dc83a6f3dc3"), First_Name: 'Suresh' }
]
psm>
```

Drop Collection:

```
psm> db.C1.find().pretty()
[
  {
    _id: ObjectId("6527dd7fdeeb2dc83a6f3dc0"),
    Id: 1,
    First_Name: 'Prathmesh',
    Last_Name: 'Malvi',
    Email: 'Prathmeshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddb1deeb2dc83a6f3dc1"),
    Id: 2,
    First_Name: 'Rutuja',
    Last_Name: 'Malvi',
    Email: 'Rutujamalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  {
    _id: ObjectId("6527ddd2deeb2dc83a6f3dc2"),
    Id: 3,
    First_Name: 'Geeta',
    Last_Name: 'Malvi',
    Email: 'Sureshmalvi10@gmail.com',
    Address: 'Jambhulpada'
  },
  { _id: ObjectId("6527e02fdeeb2dc83a6f3dc3"), First_Name: 'Suresh' }
]
psm> db.C1.drop()
true
```

Drop Collection:

```
psm> db.dropDatabase()
{ ok: 1, dropped: 'psm' }
psm> show dbs
admin     40.00 KiB
config   108.00 KiB
db2        8.00 KiB
local     40.00 KiB
psm>
```

Conclusion:

MongoDB is a robust and versatile NoSQL database system that is well-suited for applications that require flexibility, scalability, and high performance. It is document-based data model and support for horizontal scaling make it a compelling choice for a wide range of use cases, from content management to real-time analytics and everything in between. As a document database, MongoDB makes it easy for developers to store structured or unstructured data.

In this experiment, we successfully implemented the Installation and Configuration of MongoDB along with performing basic NoSQL operations.