



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 }

>
```

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
```

The dropDatabase () Method

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

Syntax:

```
db.dropDatabase()
```

The createCollection() Method

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

Syntax:

```
db.createCollection(name, options)
```

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
```



```
}  
]  
}
```

CONCLUSION:

Installing and configuring MongoDB to execute NoSQL commands is a straightforward process that is essential for those looking to work with this powerful database system. To begin, one needs to download and install MongoDB, which is available for various operating systems, ensuring that the appropriate drivers and tools are also installed. Once MongoDB is up and running, it is crucial to configure it by setting up the data directory, enabling authentication, and specifying the desired port and network settings. With these prerequisites in place, users can then leverage MongoDB to execute NoSQL commands effectively. MongoDB's intuitive JSON-like document structure, combined with its rich query and aggregation capabilities, allows for seamless data manipulation and retrieval. Furthermore, its support for various programming languages and extensive community resources make it a versatile choice for developers and organizations. In conclusion, installing and configuring MongoDB is a fundamental step that unlocks the power of NoSQL databases, enabling users to efficiently manage, query, and analyze their data.