

ASSIGNMENT NO: 5

AIM : Design and create a suitable collection example to perform CRUD operations in MongoDB.

INDEX TERMS: MongoDB, CRUD Operations.

THEORY

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.

Database:

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents. Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection:

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document:

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

Advantages of MongoDB:

Schema less – MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.

Structure of a single object is clear.

No complex joins.

Deep query-ability. MongoDB supports dynamic queries on documents using a document based query language that's nearly as powerful as SQL.

Ease of scale-out –MongoDB is easy to scale.

Installation:

MongoDB can be installed on the various O.S. platforms like Windows, Ubuntu and various Linux Flavors of O.S. MongoDB installation packages are available for both 32 bit and 64 bit Linux. You can download it, and install. After downloading the zip file, unzip it to the folder where you want to install. Later create a data directory and run the MongoDB commands through terminal for starting the server and the client of MongoDB

MongoDB Shell:

The mongo shell is an interactive JavaScript interface to MongoDB. You can use the mongo shell to query and update data as well as perform administrative operations. The mongo shell is a component of the MongoDB distributions. Once you have installed and have started MongoDB, connect the mongo shell to your running MongoDB instance.

CRUD Operations:

We can use the four basic operations, create, read, update, and delete (CRUD) to manipulate and view data in the shell.

Create :

The insert function adds a document to a collection. For example, suppose we want to store a blog post. First, we'll create a local variable called post that is a JavaScript object representing our document. It will have the keys "Aim", "content", and "date" (the date that it was published):

```
>post = {"Aim" : "MyBlogPost",  
... "content" : "Here'smy blogpost.",  
... "date" : new Date() }  
{  
  "Aim": "MyBlogPost",  
  "content" : "Here'smy blogpost.",  
  "date" : ISODate("2012-08-24T21:12:09.982Z")  
}
```

This object is a valid MongoDB document, so we can save it to the blog collection using the insert method:

```
>db.blog.insert(post)
```

Read :

Find() and findOne() can be used to query a collection. If we just want to see one document from a collection, we can use findOne:

```
>db.blog.findOne()  
{  
  "_id": ObjectId("5037ee4a1084eb3ffcef7228"),  
  "Aim": "MyBlogPost",  
  "content" : "Here'smy blogpost.",  
  "date" : ISODate("2012-08-24T21:12:09.982Z")  
}
```

Update :

If we would like to modify our post, we can use update. update takes (at least) two parameters: the first is the criteria to find which document to update, and the second is the new document. Suppose we decide to enable comments on the blog post we created earlier. We'll need to add an array of comments as the value for a new key in our document.

The first step is to modify the variable post and add a "comments" key:

```
>post.comments = []
```

Then we perform the update, replacing the post Aimd "My Blog Post" with our new version of the document:

```
>db.blog.update({Aim : "My BlogPost"},post)
```

Delete :

'remove' permanently deletes documents from the database. Called with no parameters, it removes all documents from a collection. It can also take a document specifying criteria for removal. For example, this would remove the post we just created:

```
>db.blog.remove({Aim : "MyBlogPost"})
```

FAQS:

Q.1. What is NoSQL?

Q2. State the advantages of NoSQL

Q3. What is a Namespace in MongoDB?

Conclusion:

Outcome of the experiment is students are able to

1. Handle open source NoSQL Database tool: MongoDB.
2. Execute CRUD operations.