Page No.
Date

Assignment - B2

* Date of Completion: 21/10/2020
* Date of Submission: 02/11/2020

* Title: Design and develope MongoDB queries using CRUD Operations.

* Problem Statement: Design and develop MongoDB queries using CRUD Operations (Use CRUD operations, SAVE method, logical and comparison operators)

* Learning Objectives: To understand and implement the CRUD Operations in MongoDB.

* Learning Outcomes: Students will be able to
1. implement the commands on 2 ties
2. implement the database in MongoDB.

* Hardware and Software Required:
  MongoDB version 4.4.1, Windows 10 (64 bit), Intell i5 processor

* Theory:
  Collection: It is a group of MongoDB documents. It exists within a single database and doesn't enforce a schema.
  Document: It is a set of key-value pairs. They have dynamic schema, may not necessarily have same set of fields or structure.

Teacher's Sign.: ______________

CRUD operations: Create, Read, Update, Delete, documents.

Create: create or insert operations add new documents to a collection. If the collection does not exist, insert operation will create the collection.

```
db.collection.insertOne()
db.collection.insertMany()
```

In MongoDB, insert operations target a single collection. All write operations are basic atomic on the level of a single document.

eg

```
db.user.insertOne( {            ← collection
   name: "varun",               } field: value
   age: 21,                       document
   username: "varunkanwa" }
)
```

Read: Retrieve document from a collection, i.e. query a collection for documents

```
db.collection.find()
→ db.user.find(
     { age: {$ lt: 25} }             query criteria
     { name: 1, address: 1 }         projection
  ). limit(5)                        cursor modifier.
```

db.collection.findOne() → returns first document

Page No.
Date

Update Operations: Modify existing documents in a collection
db.collection.updateOne()
db.collection.updateMany()
db.collection.replaceOne()

```
db.movies.updateMany(
    {age: {$gt: 25}},                        update filter
    {$set: {movie.rating: 5.5}}              update action
)
```

Delete Option:
Remove documents from a collection:
db.collection.deleteOne()
db.collection.deleteMany()

```
⇒ db.movies.deleteMany(
    {movies.id : "comedy"}                   delete filter.
)
```

⇒ use database-name to create database
⇒ db to check your currently selected database
⇒ show dbs to check database list
⇒ db.dropDatabase() will delete selected database
⇒ db.createCollection(name, options)

$$\overset{\text{String}}{\text{name}}, \overset{\text{Document}}{\text{options}}$$

[Options: capped (bool) - fixed size collection, automatically overwrites its oldest entries when it reaches max size
autoIndexId (bool), size (number),
max (Number)]

⇒ db.collection.drop() - drop collection
It will true if dropped else false

Teacher's Sign.:

Page No.
Date

Optional query for update
1. upsert - If no document matches with one mentioned in query then new one gets upserted into the collections
⇒ {upsert: true}

2. Multi: By default in update, one document is updated. If multi is set to true then it updates all documents matching the conditions
eg: {multi: true}

Comparison Query Operators:
$eq: matches values that are equal to a specified value
$gt: matches values greater than a value
$in: matches any of the values specified in an array.
$lt: matches values lesser than a specified value.
$ne: not equal to
$nin: none of these values.

Logical operations:
$or - joins query clauses with logical or returns all documents matching either of the clauses.
$and: joins query clauses with logical and returns all documents matching all of the clauses.
$not: returns all documents not matching the given conditions.

Teacher's Sign.:

Page No.

Date

Save: It is a combination of insert and update

syntax. db.collection.save ({...})

If passed document does not exist in the collection then it is newly added in collection.

If the documentation consists of an _id which pre-exists in collection for some other document, then existing document is replaced with new document.

Conclusion: We were successfully able to learn CRUD operations, save method and implement them along with logical as well as comparison operators.

Teacher's Sign.: