Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

CET4001B Big Data Technologies

School of Computer Engineering and Technology

CRUD Operations in MongoDB

LABORATORY ASSIGNMENT NO: 01

BASIC CONCEPTS OF CRUD

Basic Database Operations- Database

use *<database name>*	db	show dbs	db.dropDatabase()
switched to database provided with ciommand	To check currently selected database use the command **db**	Displays the list of databases	To Drop the database

Basic Database Operations-Collection

db.createCollection (name) **Ex:- db.createCollection(Stud)**	• To create collection
>show collections	• List out all names of collection in current database
db.*databasename*.insert **({Key : Value})** **Ex:- db.Stud.insert({{Name:"Jiya"})**	• In mongodb you don't need to create collection. MongoDB creates collection automatically, when you insert some document.
db.collection.drop() Example:- **db.Stud.drop()**	• MongoDB's **db.collection.drop()** is used to drop a collection from the database.

CRUD Operations in Mongodb

Create	Create Operations
Read	Read Operations
Update	Update Operations
Delete	Delete Operations

CRUD Operations

- Insert
- Find
- Update
- Delete

CRUD

- Create
  - db.collection.insert(<document>)
  - db.collection.save(<document>)
  - db.collection.update(<query>, <update>, { upsert: true })
- Read
  - db.collection.find(<query>, <projection>)
  - db.collection.findOne(<query>, <projection>)
- Update
  - db.collection.update(<query>, <update>, <options>)
- Delete
  - db.collection.remove(<query>, <justOne>)

CRUD example

```
> db.user.insert({
        first: "John",
        last : "Doe",
        age: 39
})
```

```
> db.user.find ()
{
        "_id" : ObjectId("51…"),
        "first" : "John",
        "last" : "Doe",
        "age" : 39
}
```

```
> db.user.update(
        {"_id" : ObjectId("51…")},
        {
                $set: {
                        age: 40,
                        salary: 7000}
        }
)
```

```
> db.user.remove({
        "first": /^J/
})
```

CRUD Operations

- Insert
- Find
- Update
- Delete

CRUD Operations - Insert

- **The insert() Method:-** To insert data into MongoDB collection, you need to use MongoDB's **insert()** or **save()**method.
- **Syntax**

```
>db.COLLECTION_NAME.insert(document)
```

- **Example**

```
>db.stud.insert({name: “Jiya”, age:15})
```

CRUD Operations - Insert

- **Insert Single Documents**

```
db.stud.insert
( {Name: “Ankit”, Rno:1, Address: “Pune”} )
```

CRUD Operations - Insert

- **Insert Multiple Documents**

```
db.stud.insert
( [
 { Name: “Ankit”, Rno:1, Address: “Pune”} ,
 { Name: “Sagar”, Rno:2},
 { Name: “Neha”, Rno:3}
 ] )
```

CRUD Operations - Insert

- **Insert Multicolumn attribute**

```
db.stud.insert(
   {
    Name: “Ritu",
    Address: { City: “Pune",
                State: “MH” },
    Rno: 6
   }
)
```

CRUD Operations

Insert

Find

Update

Delete

CRUD Operations - Find

- **The find() Method-** To display data from MongoDB collection. Displays all the documents in a non structured way.
- **Syntax**

```
>db.COLLECTION_NAME.find()
```

- **The pretty() Method-** To display the results in a formatted way, you can use **pretty()** method.
- **Syntax**

```
>db. COLLECTION_NAME.find().pretty()
```

CRUD Operations - Find

db.stud.find()

- Select All Documents in a Collection in **unstructured form**

db.stud.find().pretty()

- Select All Documents in a Collection in **structured form**

CRUD Operations - Find

Specify Equality Condition

- use the query document
  { <field>: <value> }
- Examples:
- db.stud.find(name: “Jiya" })
- db.stud.find({ _id: 5 })

CRUD Operations

Insert

Find

Update

Delete

CRUD Operations – Update

- **Syntax**

db.*CollectionName*.update(
  <query/Condition>,
  <update with $set or $unset>,
  {
    upsert: <boolean>,
    multi: <boolean>,
  }
)

CRUD Operations – Update

upsert

- If set to *True*, creates new document if no matches found.

multi

- If set to *True*, updates multiple documents that matches the query criteria

CRUD Operations – Update Examples

```
db.stud.update(
{ _id: 100 },
{ age: 25})
```

- Set age = 25 where id is 100
- First Whole document is replaced where condition is matched and only one field is remained as age:25

```
db.stud.update(
{ _id: 100 },
{ $set:{age: 25}})
```

- Set age = 25 where id is 100
- Only the age field of one document is updated where condition is matched

```
db.stud.update(
{ _id: 100 },
{ $unset:{age: 1}})
```

- To remove a age column from single document where id=100

CRUD Operations – Update Examples

```
db.stud.update(
{ _id: 100 },
{ $set: { “marks.dmsa": 50} })
```

- Set marks for dbms subject as 50 where id = 100 (only one row is updated)

```
db.stud.update(
{ class: “TE” },
{ $set: { “marks.dmsa": 50} } ,
{ multi: true } )
```

- Set marks for dbms subject as 50 where class is TE (all rows which matches the condition were updated)

```
db.stud.update(
{ class: “TE” },
{ $set: { “marks.dmsa": 50} } ,
{ upsert: true } )
```

- Set marks for dbms subject as 50 where class is TE (all rows which matches the condition were updated)
- If now row found which matches the condition it will insert new row.

CRUD Operations

Insert

Find

Update

Delete

CRUD Operations – Remove

Remove All Documents	• db.inventory.remove({})
Remove All Documents that Match a Condition	• db.inventory.remove({ type : "food" })
Remove a Single Document that Matches a Condition	• db.inventory.remove ({ type : "food" }, 1)

BATCH-01 EXERCISE

Create Company Database and Create Employee collection with following key

```
{
   "firstName": "John",
   "lastName": "Smith",
   "age": 25,
 _id: “5”
   "address":
   {
      "streetAddress": "21 2nd Street",
      "city": "New York",
      "state": "NY",
      "postalCode": "10021"
   },
   "phoneNumber":
   [
      {
        "type": "home",
        "number": "212 555-1234"
      },
      {
        "type": "fax",
        "number": "646 555-4567"
      }
   ]
“emailAddress": [ “romin.k.irani@gmail.com” , “tomhanks@gmail.com”]  }
```

CRUD Operations : Solve Queries

- Create database Company.
- Create collection Customer.
- Insert 10 documents with above mentioned structure.
- Display all Customerinformation.
- Update different records by satisfying following condition:
  - Increment value of age in one record($inc)
  - Push some more phone no ($push)
  - Pop some email address ($pop)
  - Replace existing record with new record(update)
  - set the new address for one of the record.($set)
  - Add one record if doesn't exist else update($upsert)
- Remove few documents
- Use $gte,$gt,$lt operators and fire queries.
- Find records having city 'new York'
- Find records where age is above 40
- Find all records starting with first name 'M'.

CRUD Operations : Solve Queries

- Find total count of records in your collection.
- Display all the documents where more than two phone nos.($size)
- Display the name of the employee(only) while querying the document.
- Display the documents sorted by name in descending order.
- Fire query using ‘and’ operation
- Fire query using ‘or’ operation
- Display only 2 documents while querying the document.
- Drop Collection

BATCH-02 EXERCISE

Create Restaurant Database and Create Hotel collection with following keys

```
{
        Hotel Id:
        Hotel Name:
        Type:
        Ratings
        Address:{
                Area:
                City:
                Pincode:
            }
        Rooms :[
            {
                Roomno:
                Type:
                Price:
            }
            {
                Roomno:
                Type:
                Price:
            }

            ]
        Cuisines : [“Indian”,”Italian”,”Chinese”.....]
        likes:
}
```

CRUD Operations : Solve Queries

- Create database Restaurant.
- Create collection Hotel.
- Insert 10 documents with above mentioned structure.
- Display all Hotel information.
- Update different records by satisfying following condition:
  - Increment value of likes in one record($inc)
  - Push some more cuisines ($push)
  - Pop some room ($pop)
  - Replace existing record with new record(update)
  - set the new address for one of the record.($set)
  - Add one record if doesn't exist else update($upsert)
- Remove few documents
- Use $gte,$gt,$lt operators and fire queries.
- Find records having city 'new York'
- Find records where age is above 40
- Find all records starting with name 'M'.

CRUD Operations : Solve Queries

- Find total count of records in your collection.
- Display all the documents where more than two cuisines.($size)
- Display the name of the hotel(only) while querying the document.
- Display the documents sorted by name in descending order.
- Fire query using ‘and’ operation
- Fire query using ‘or’ operation
- Display only 2 documents while querying the document.
- Drop Collection

Practice Assignment

Create Theatre Database and Create Movies collection with following keys

```
{
   "title": "",
   "year": "",
   "directors": [
      {
         "Name": "",
         "DOB": "",
         "Adress": "",
         "TelNo": ""
      }
   ],
   "writers": [
      {
         "Name": "",
         "DOB": "",
         "Adress": "",
         "TelNo": ""
      }
   ],
   "stars": [
      {
         "Name": "",
         "DOB": "",
         "Adress": "",
         "TelNo": ""      }      ]
 "tags":["comedy","action"]
 "likes":                  }
```

CRUD Operations : Solve Queries

- Create database Restaurant.
- Create collection Hotel.
- Insert 10 documents with above mentioned structure.
- Display all Hotel information.
- Update different records by satisfying following condition:
  - Increment value of likes in one record($inc)
  - Push some more tags ($push)
  - Pop some stars ($pop)
  - Replace existing record with new record(update)
  - set the new address for one of the record.($set)
  - Add one record if doesn’t exist else update($upsert)
- Remove few documents
- Use $gte,$gt,$lt operators and fire queries.
- Find records having title ‘3 idiots’
- Find records where likes more than 200
- Find all records starting with name ‘M’.

CRUD Operations : Solve Queries

- Find total count of records in your collection.
- Display all the documents where more than two writers.($size)
- Display the name of the movie(only) while querying the document.
- Display the documents sorted by name in descending order.
- Fire query using ‘and’ operation
- Fire query using ‘or’ operation
- Display only 2 documents while querying the document.
- Drop Collection