**AIM:**

To Write a program in MongoDB-CRUD operations, indexing, Sharding, Deployment.

**PROCEDURE:**

Step 1: Start->open command prompt
Step 2: cd enter the path of the mongodb in command prompt
cd c:\Program filles\mongodb\server\3.2\bin
Step 3:Enter “mongodb” and hit enter ,mongo
Step 4: If error occurs,then we have to create a folder in c: call data of subfolder DB In data.
Step 5: Insert a command on the cmd C:\program files \ Mongo DB\server\3.2\bin>mongod.....storage engine=mmapui...d path
C:\data\DB then DB starts.
Step 6: Open another new command prompt and then repeat step2
Step 7:Enter “mongo” and hit enter c:\Program filles\mongodb\server\3.2\bin>Mongo
Step 8: Then follow the follow in Query statement that to be executed in Mongo DB

**CRUD Operation:**

**1. Create Operation**
**2. Read Operation**
**3. Update Operation**
**4. Delete Operation**

**Create Operation:**

It is used to create a new collection.

**Syntax:**

>>>use collection_name

**Sample Query:**

**INPUT:-**

**>>>use inventory**

**OUTPUT:-**


**Insert Operation:-**

It is used to add one or more documents to the collection. It has two types,

insertOne-used to add only one document to the collection.
insertMany-used to add more than one document to the collection.

**Syntax:-insertOne():-**

**db.collection.insertOne(<document>,{ writeConcern: <document> } )**

**SAMPLE QUERY:-**

**INPUT:-**

>>>db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } })

**OUTPUT:-**

```
inventory> db.inventory.insertOne({ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } })
{
  acknowledged: true,
  insertedId: ObjectId('66607b200047ef8479cdcdf6')
}
```

**Syntax:-**

**insertMany():-**

db.collection.insertMany(
[ <document 1> , <document 2>, ... ],
{
writeConcern: <document>,
ordered: <boolean>
}
)

**SAMPLE QUERY:-**

**INPUT:-**

```
>>>db.inventory.insertMany([
{ item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
{ item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
{ item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" }
}
])
```

**OUTPUT:-**

```
inventory> db.inventory.insertMany([
... { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
... { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
... { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66607bc60047ef8479cdcdf7'),
    '1': ObjectId('66607bc60047ef8479cdcdf8'),
    '2': ObjectId('66607bc60047ef8479cdcdf9')
  }
}
```

**READ OPERATION:-**

It is used to retrieve documents from the collection based on some constraints.

**Syntax:-**

db.collection.find(query, { <field1>: <value>, <field2>: <value> ... })

**SAMPLE QUERY:-**

INPUT:-

```
>>>db.inventory.find( {} )
```

**OUTPUT:-**

```
mongosh mongodb://127.0.0.
inventory> db.inventory.find( {} )
[
  {
    _id: ObjectId('66607a21941d6cd405cdcdf6'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('66607b200047ef8479cdcdf6'),
    item: 'canvas',
    qty: 100,
    tags: [ 'cotton' ],
    size: { h: 28, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('66607bc60047ef8479cdcdf7'),
    item: 'journal',
    qty: 25,
    tags: [ 'blank', 'red' ],
    size: { h: 14, w: 21, uom: 'cm' }
  },
  {
    _id: ObjectId('66607bc60047ef8479cdcdf8'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  },
  {
    _id: ObjectId('66607bc60047ef8479cdcdf9'),
    item: 'mousepad',
    qty: 25,
    tags: [ 'gel', 'blue' ],
    size: { h: 19, w: 22.85, uom: 'cm' }
  }
]
```

**SAMPLE QUERY:-**
**INPUT:-**
>>>db.inventory.find( {qty:85} )
**OUTPUT:-**

```
inventory> db.inventory.find( {qty:85} )
[
  {
    _id: ObjectId('66607bc60047ef8479cdcdf8'),
    item: 'mat',
    qty: 85,
    tags: [ 'gray' ],
    size: { h: 27.9, w: 35.5, uom: 'cm' }
  }
]
```

**UPDATE OPERATION:-**

It is used to modify (add/replace)one or more documents in the collection. It consists of 3 types:

updateOne

update many

**Syntax:-**

db.collection.updateOne(<filter>, <update>, <options>)
db.collection.updateMany(<filter>, <update>, <options>)
db.collection.replaceOne(<filter>, <update>, <options>)

**SAMPLE QUERY:-updateOne()**

**INPUT:-**

>>>db.inventory.updateOne( { item: "paper" }, { $set: { "size.uom": "cm"}, $currentDate: { lastModified: true } })

**OUTPUT:-**

```
inventory> db.inventory.updateOne( { item: "paper" }, { $set: { "size.uom": "cm"},
... $currentDate: { lastModified: true } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

**SAMPLE QUERY:-updateMany()**

**INPUT:-**

>>>db.inventory.updateMany( { "qty": { $lt: 50 } }, { $set: { "size.uom": "in", status: "P" }, $currentDate: { lastModified: true } })

**OUTPUT:-**

```
inventory> db.inventory.updateMany( { "qty": { $lt: 50 } }, { $set: { "size.uom": "in", status:
... "P" }, $currentDate: { lastModified: true } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

**DELETE OPERATION:-**

It is used to delete one or more documents/column from the collection based on the constraints.

**Syntax:-**

db.collection.deleteMany()
db.collection.deleteOne()

**SAMPLE QUERY:-deleteOne()**

**INPUT:-**

>>>db.inventory.deleteOne( { qty:85 } )

**OUTPUT:-**

```
inventory> db.inventory.deleteOne( { qty:85 } )
{ acknowledged: true, deletedCount: 1 }
```

**SAMPLE QUERY:-deleteMany()**

**INPUT:-**

>>>db.inventory.deleteMany( { qty:25 } )

**OUTPUT:-**

```
inventory> db.inventory.deleteMany( { qty:25 } )
{ acknowledged: true, deletedCount: 2 }
```

**RESULT:**