Practice command

- **sudo service mongodb start :**
  This commans will start services of mongodb and then after we can open mongo shell.

- **mongo (enter) :**
  After starting services of mongodb we can start mongo shell using mongo command.

- **show dbs :**
  This command will list out all databases and their size. If data does not present in dtabase then it will not listed out.

  ```
  Command :    show dbs

  O/P :   admin  0.000GB
          config 0.000GB
          local  0.000GB
  ```

- **use demoDB :**
  This command create new database named demoDB and switch to it.

  ```
  O/P:    switched to demoDB
  ```

- **show collections :**
  This command list out all collections present in current working database. Suppose we have two collection present in demoDB then it'll print.

  ```
  Command :    show collections

  O/P :   students
          teachers
  ```

- **db.students.insertOne() :**
  This command will insert one record in to students collections.

  ```
  Command :
  db.students.insertOne( {
          name: “Manish Rathod”,
          contactDetails: {
                  email: “manish.rathod@bacancy.com”,
                  phoneNumber: 7283848276
          },
          technologyAssigned: “NodeJS”
  })
  ```

```
O/P :   {
                "acknowledged" : true,
                "insertedId" : ObjectId("62264faa2a63ba5a0791c9fc")
        }
```

- **db.students.insertMany() :**

  This command insert multiple documents at a time.

  Command :

```
db.students.insertMany([
        {
                name:”Sachin Vaishnav “,
                contactDetails: {
                        email: “sachin.vaishnav@bacancy.com”,
                        phoneNumber: 9979590466
                },
                technologyAssigned: “ReactJS”
        },
        {
                name: “Krushit Dudhat”,
                contactDetails: {
                        email: ”krushit.dudhat@bacancy.com”,
                        contactNumber: 8745123690
                },
                 technologyAssigned: “NodeJS”
        }
])
```

```
O/P :   {
"acknowledged" : true,
"insertedIds" : [
                ObjectId("622652da2a63ba5a0791c9fe"),
                ObjectId("622652da2a63ba5a0791c9ff")
        ]
}
```

- **db.students.updateOne() :**

  This command will update first matched document.

  Command:

```
db.students.updateOne(
        {
                technologyAssigned: “ReactJS”
        },
        {
                $set: {technologyAssigned: “Flutter”
        }
})
```

```
O/P :   { "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

- **db.students.updateMany() :**
  This command will update all matched document.

  Command:
  ```
  db.students.updateMany(
          {
                  technologyAssigned: { $in: [“NodeJS”, “ReactJS”] }
          },
          {
                  $set: { city: “Vadodara”}
          }
  })
  ```

  O/P : { "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }

- **db.students.replaceOne() :**
  This command will replace whole document by specified document and only first matched document will be modifided.

  ```
  db.students.replaceOne(
          {  city: "Rajkot"  },
          {
                  name:"Vinayak Chavan",
                  contactDetails:         {
                          email: "vinayak.chavan@bacancy.com",
                          phoneNumber: 7894561230
                  },
          technologyAssigned: "NodeJS"
  })
  ```

  O/P : { "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

- **db.students.deleteOne() :**
  This command will delete first matched document from the collections and return object having preperties like acknowledged and deletedCount.

  Command:
  ```
  db.students.deleteOne({
          technologyAssigned: “Flutter”
  })
  ```

  O/P :{ "acknowledged" : true, "deletedCount" : 1 }

- **db.students.deleteMany() :**
  This command will delete all matched document from the collections and return object having preperties like acknowledged and deletedCount.

  Command:
  ```
  db.students.deleteOne({
          technologyAssigned: “ReactJS”
  })
  ```

```
O/P :{ "acknowledged" : true, "deletedCount" : 2 }
```

- **db.students.find() :**
  This command will print all document present in students collection without any filter.

```
Command :    db.students.find()

O/P :
{
        "_id" : ObjectId("62264faa2a63ba5a0791c9fc"),
        "name" : "Manish Rathod",
        "contactDetails" : {
                "email" : "manish.rathod@bacancy.com",
                "phoneNumber" : 7283848276
        },
         "technologyAssigned" : "NodeJS",
        "city" : "Vadodara"
}
{
        "_id" : ObjectId("622652da2a63ba5a0791c9ff"),
        "name" : "Krudhit Dudhat",
        "contactDetails" : {
                "emaill" : "krushit.dudhat@bacancy.com",
                "phoneNumber" : 9874563210
        },
        "technologyAssigned" : "NodeJS",
        "city" : "Vadodara"
}
{
        "_id" : ObjectId("6226d21f6e11e499518d1c8c"),
        "name" : "Vinayak Chavan",
        "contactDetails" : {
                "email" : "vinayak.chavan@bacancy.com",
                "phoneNumber" : 7894561230
        },
        "technologyAssigned" : "NodeJS"
        city: "Ahmedabad"
}
```

- **db.students.find().limit(n) :**
  This command will list all matched document if condition is specfied in maximum limit of number defined in limit(n) (Here is n).

```
Command :    db.students.find().limit(1)

O/P :
{
        "_id" : ObjectId("62264faa2a63ba5a0791c9fc"),
        "name" : "Manish Rathod",
        "contactDetails" : {
                "email" : "manish.rathod@bacancy.com",
                "phoneNumber" : 7283848276
```

```
            },
             "technologyAssigned" : "NodeJS",
            "city" : "Vadodara"
    }
```

- **db.students.find(filter, specifiedFieldToBeShown) :**
  This command will appplied filter on the documents present in collection and display onlly specified filed. Here we are showing only name and technologyAssigned field.

  Command:
  db.students.find({}, {_id:0, contactDetails: 0, city:0})

  O/P :

```
            {
                        "name" : "Manish Rathod",
                        "technologyAssigned" : "NodeJS"
            }
            {
                        "name" : "Krudhit Dudhat",
                        "technologyAssigned" : "NodeJS"
            }
            {
                        "name" : "Vinayak Chavan",
                        "technologyAssigned" : "NodeJS"
            }
```