4/3/25

Lab 1

Working with mongoDB

1. Create Database in mongoDB

```
use myDB;
db;
show dbs;
```

2 CRUD (Create, Read, update, delete) operations

i) db.createCollection("Student");

```
db.Student.drop();

→ db.Student.insert({_id: 1, StudName: "Rach", Grade: "6th",
  Hobbies: "Nothing"});

→ db.Student.update({_id: 1, StudName: "Rach", Grade: "6th"},
                   {$set: {Hobbies: "Playing"}}, {upsert: true});

db.Student.find({StudName: "Rach"});

db.Student.find({}, {StudName: 1, Grade: 1, _id: 0});

db.Student.find({Grade: {$eq: '6th'}}).pretty();
```

→ db.Student. ~~update~~ insert ({_id: 2, StudName: "Monisha", Grade: "6th"},
~~{$set: {~~ Hobbies: "Nothing"});

```
db.Student.find({Hobbies: {$in: ['Nothing', 'Playing']}}).
                                                    pretty();
```

Student Names begins with M

```
db.Student.find({StudName:/^M/}).pretty();
```

Student Names contains 'li' in any position

```
db.Student.find({~~fim~~ StudName:/[illegible]/ }).pretty();
```

```
db.Student.count();
```

descending order:

```
db.Student.find().sort({StudName:-1}).pretty();
```

Add new field in an existing Document

```
db.Student.update({_id:2},{$set:{Location:"Network"}})
```

Remove field

```
db.Students.update({_id:2},{$unset:{Location:
                                      "Network"}})
```

VIII finding Document based on search criteria Suppressing few fields

```
db.Student.find({_id:1},{StudName:1, Grade:1, _id:0})
```

To find those documents where the Grade is not set to 'VII'

```
db.Student.find({Grade:{$ne: 'VII'}}).pretty();
```

To find documents from the Students collection where the StudName ends with s.

```
db.Student.find({StudName:/s$/}).pretty();
```

PAGE :
DATE : / /

IX to set a particular field value to NULL

```
db.Students.update({_id:3}, {$set:{Location:null}})
```

x count the number of document in Students collections

```
db.Students.count()
```

xi count the number of documents in Student Collection with grade : VII

```
db.Students.count({Grade:"VII"})
```

retrieve first 3 documents

```
db.Students.find({Grade:"VII"}).limit(3).pretty();
```

sort the document in Ascending order

```
db.Students.find().sort({StudName:1}).pretty();
```

for desending order

```
db.Students.find().sort({StudName:-1}).pretty();
```

to skip the 1st two documents from the Students Collections

```
db.Students.find().skip(2).pretty()
```

Create a collections by name "food" and add to each document add a "fruits" array constitute of "grape", "mango" and "apple".

```
db.food.find({fruits:['grapes','mango','apple']}).pretty()
```

# To find in "fruits" array having "mango" in the first index position

```
db.food.find({"fruits.1' : 'grapes'})
```

# To find those documents from the "food" collection when the size of the array is two

```
db.food.find({"fruits" : {$size : 2}})
```

# To find the documents with a particular id and display the first two elements from the array "fruits"

```
db.food.find({_id : 1}, {"fruits" : {$slice : 2}}}
```

# To find all the documents from the food collection which have elements mango and grapes in the array "fruits"

```
db.food.find({fruits : {$all : ["mango", "grapes"]}}}
```

update on Array:
using particular id replace the element in the 1st index position of the fruits array with apple

```
db.food.update( {_id : 3}, {$set : {'fruits:1' : 'apple'}}}
```

# insert new key value pairs in the fruits array

```
db.food.update({_id:2}, {$push : {price : {grapes:80,
mango : 200, cherry : 100}}})
```