Practical – 10

Mongodb basics and Creating documents

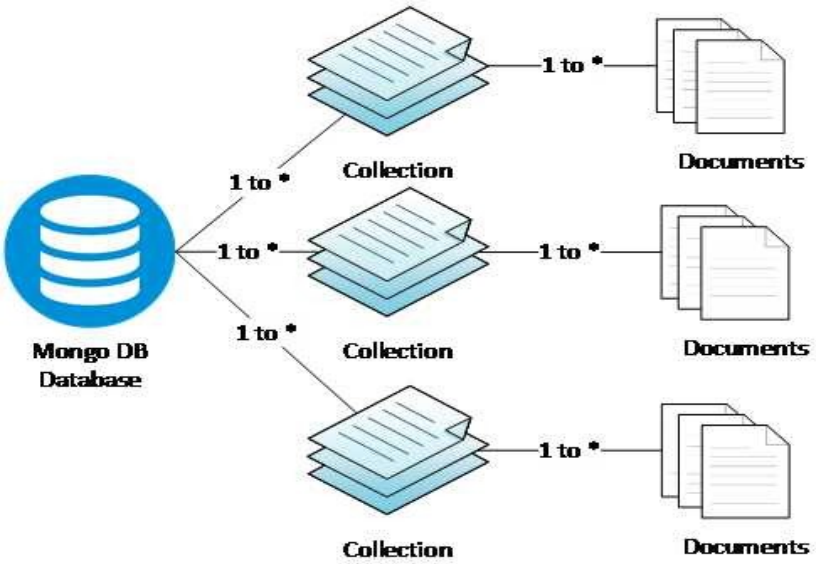
Step 1 : Install the mongodb

From https://www.mongodb.com/try/download/community site

Select version and zip option to download mongodb

Step 2 : create folder called **"data"** inside mongodb path e.g c:\programfiles\mongodb\data

e.g d:\mongodb\data



Run client and server of the mongodb

Start server of mongodb

> path of mongodb \ bin

mongodb\bin\> mongod.exe –dbpath="path of mongodb\data"

e.g d:\mongodb\bin > mongod.exe –dbpath="d:\mongodb\data"

Start mongo db client

open command prompt

go to the path where mongodb is installed from command prompt

d:\mongodb\bin>

execute following commands

d:\mongdb\bin\> mongo.exe

it will start mongodb client. And all the command will be written on mongodb client.

Commands for mongodb

Creating database

> use employee

it create employee database and switch inside the employee database

Show all databases
>show dbs

Select databases
> use databasename
e.g > use employee

Display the List of collections of database
>show collections;
ElectronicShop
emp

Help function for database
>db.help() // will display all the available function of the database

Help for collections
>db.collectionname.help()

e.g db.employee2.help() // it will list out all available functions for collections

Create the collections
The easiest way to create a collection is to insert a record (which is nothing but a document consisting of Field names and Values) into a collection. If the collection does not exist a new one will be created.

Delete the collections
>db.employee2.drop() // it whill delete the collections

>show collections

Documents:

Documents are inside the collection. Mongo stores data inside documents – so mongodb is document based database.

Maximum size of the document is 16MB. That is one document contains data only upto 16MB.

Insert one document at a time in collection

insertOne() it will create one document in collection

syntax : db.<collectionname>.insertOne({field1 : ”value1” , field2 : “value2”,..})

e.g
db.employee.insertOne({name:"subham",age : 20,dept :"sales"})

show documents from the collections

syntax. db.<collectioname>.find().pretty()

e.g
```
> db.employee.find().pretty()
{
	"_id" : ObjectId("6019136992fa350ddd0397fe"),
	"name" : "subham",
	"age" : 20,
	"dept" : "sales"
}
```

Each documents generate the uniq Objectid . Which uniquely identify the documents

creating complex nested documents

e.g db.employee.insertOne({name : "prit", age : 25, dept :"admin",
address :{ street :"new street",city :"ahmd"},
phone : { personal : "12345" , office : "4567"} })

Display documents from the collections

```
db.employee.find().pretty()
{
	"_id" : ObjectId("6019136992fa350ddd0397fe"),
	"name" : "subham",
	"age" : 20,
	"dept" : "sales"
}
{
	"_id" : ObjectId("6019157f92fa350ddd0397ff"),
	"name" : "prit",
	"age" : 25,
	"dept" : "admin",
	"address" : {
		"street" : "new street",
		"city" : "ahmd"
	},
	"phone" : {
		"personal" : "12345",
		"office" : "4567"
	}
}
```

create your own objectid for the documents

db.employee.insertOne({**_id:1**, name : "vihan", dept : "HR"})

Checking how many documents inside the collection

> db.empdata.find().count()

Read the data / find the data from document from database

reading of documents is more useful functionality of documents

find() : use to display everything inside collections
find().pretty() : display everything inside collection is proper format

find() : To search the record based on condition

syntax : db.collection_name.find({key1 " value1", key2:"value2"....})

e.g display only documents contains dept sales
db.employee.find({dept : "sales"})
{ "_id" : ObjectId("6019136992fa350ddd0397fe"), "name" : "subham", "age" : "20", "dept" : "sales" }

display the document which does have phone no
db.employee.find({phone : null}).pretty()

```
{
	"_id" : ObjectId("6019136992fa350ddd0397fe"),
	"name" : "subham",
	"age" : "20",
	"dept" : "sales"
}
{ "_id" : 1, "name" : "vihan", "dept" : "HR" }
```

Query on the subkey/nested key

e.g display the documents contains address.city =ahmd
> db.empdata.find({"address.city":"ahmd"}))

```
{
	"_id" : ObjectId("6019157f92fa350ddd0397ff"),
	"name" : "prit",
	"age" : "25",
	"dept" : "admin",
	"address" : {
		"street" : "new street",
		"city" : "ahmd"
	},
	"phone" : {
		"personal" : "12345",
		"office" : "4567"
	}
```

query document for value grater than of specific field

syntax { $gt : “value”}
>

db.employee.find({age :{ $gt: "20"}}).pretty()

```
{
        "_id" : ObjectId("6019157f92fa350ddd0397ff"),
        "name" : "prit",
        "age" : "25",
        "dept" : "admin",
        "address" : {
                "street" : "new street",
                "city" : "ahmd"
        },
        "phone" : {
                "personal" : "12345",
                "office" : "4567"
        }
}
```

Other operator used with find

Equality	{<key>:{$eg;<value>}}
Less Than	{<key>:{$lt:<value>}}
Less Than Equals	{<key>:{$lte:<value>}}
Greater Than	{<key>:{$gt:<value>}}
Greater Than Equals	{<key>:{$gte:<value>}}
Not Equals	{<key>:{$ne:<value>}

e.g **db.employee.find({dept: {$ne:"sales"}}).pretty()**

```
  "_id" : ObjectId("6019157f92fa350ddd0397ff"),
        "name" : "prit",
        "age" : "25",
        "dept" : "admin",
        "address" : {
                "street" : "new street",
                "city" : "ahmd"
        },
        "phone" : {
                "personal" : "12345",
                "office" : "4567"
        }
}
```

{ "_id" : 1, "name" : "vihan", "dept" : "HR" }

Exercise

1_ id, item, stock, info (Operation, warranty_years), tags, ratings (by, rating)

```
Sample data:
"_id":1,
"item":"Microwave oven",
“brand” : “Samsung””
stock":20,
"tags":”electronic”,
"ratings":{"by":"Jim","rate":4}
 “price”:35000
```

Perform following query
Instruction: please insert relevent data as par the query execution requirement
1. create database “itemdb”
2. create collection called “itemcollection”
3. Insert at least 7 documents in collection Itemcollection.
4. display all documents database.
5 . insert the document having own userdefine document id.
6. Create collection call “personcollection”
7. find the item whose rating rate is 4
8. find the item whose brand is samsund
9. find the item where tag is clothes
10. find the item having price > 5000
11. find the item where tag is not electronics