

Name: **Chandradhar Rao**
SRN: **PES1UG19CS123**
Week1 NoSQL MongoDB DBMS Lab

Create Collection

```
> use test
switched to db test
> db.createCollection("towns")
{ "ok" : 1 }
> █
```

Insert Documents

```
> db.towns.insert({
... name:"New York",
... population:22200000,
... lastCensus:ISODate("2016-07-01"),
... famousFor:["the MOMA","food","Derek Jeter"],
... mayor:{
... name:"Bi;; de Blasio",
... party:"D"
... }
... })
WriteResult({ "nInserted" : 1 })
> █
```

Display the Output

```
student@pessat196: ~
... population:22200000,
... lastCensus:ISODate
... ^C

> db.towns.insert({
... name:"New York",
... population:22200000,
... lastCensus:ISODate("2016-07-01"),
... famousFor:["the MOMA","food","Derek Jeter"],
... mayor:{
... name:"Bi;; de Blasio",
... party:"D"
... }
... })
WriteResult({ "nInserted" : 1 })
> show collections
319
inventory
marvel
pes
student
studentdetails
towns
> █
```

Find Method in DB:

```
student@pessat196: ~  
studentdetails  
towns  
> db.towns.find()  
{ "_id" : ObjectId("612d9ff276a62dd7a10470ee"), "name" : "New York", "population" : 22200000, "lastCensus" : ISODate("2016-07-01T00:00:00Z"), "famousFor" : [ "the MOMA", "food", "Derek Jeter" ], "mayor" : { "name" : "Bi;; de Blasio", "party" : "D" } }  
> db.towns.find().count()  
1
```

Type of operation:

```
> typeof db.towns.insert  
function  
> typeof db.towns  
object  
>
```

Insert City Function:

```
> function insertCity(  
... name,population,lastCensus,  
... famousFor,mayorInfo){  
... db.towns.insert({  
... name:name,  
... population:population,  
... lastCensus:lastCensus,  
... famousFor:famousFor,  
... mayor:mayorInfo  
... }));  
... }
```

Use Insert City Function and Use Find:

```
> insertCity("Punxsutawney", 6200, '2016-01-31',"Punxsutawney Phil",{  
... name:"Richard Alexander"  
... })  
> insertCity("Portland", 582000, '2016-09-20',  
... ["beer", "food", "Portlandia"], { name : "Ted Wheeler", party : "D"})  
> db.towns.find().count()  
3  
> db.towns.find()  
{ "_id" : ObjectId("612d9ff276a62dd7a10470ee"), "name" : "New York", "population" : 22200000, "lastCensus" : ISODate("2016-07-01T00:00:00Z"), "famousFor" : [ "the MOMA", "food", "Derek Jeter" ], "mayor" : { "name" : "Bi;; de Blasio", "party" : "D" } }  
{ "_id" : ObjectId("612da1d276a62dd7a10470ef"), "name" : "Punxsutawney", "population" : 6200, "lastCensus" : "2016-01-31", "famousFor" : [ "Punxsutawney Phil" ], "mayor" : { "name" : "Richard Alexander" } }  
{ "_id" : ObjectId("612da20076a62dd7a10470f0"), "name" : "Portland", "population" : 582000, "lastCensus" : "2016-09-20", "famousFor" : [ "beer", "food", "Portlandia" ], "mayor" : { "name" : "Ted Wheeler", "party" : "D" } }  
> db.towns.find({"_id" : ObjectId("612d9ff276a62dd7a10470ee")})  
{ "_id" : ObjectId("612d9ff276a62dd7a10470ee"), "name" : "New York", "population" : 22200000, "lastCensus" : ISODate("2016-07-01T00:00:00Z"), "famousFor" : [ "the MOMA", "food", "Derek Jeter" ], "mayor" : { "name" : "Bi;; de Blasio", "party" : "D" } }
```

Use \$lt:

```
student@pessat196: ~  
> db.towns.find(  
... {  
... name:/^P/,  
... population:{  
... $lt:10000  
... }  
... },  
... {  
... _id:0,  
... name:1,  
... population:1  
... })  
{ "name" : "Punxsutawney", "population" : 6200 }  
>
```

Name Starts with P:

```
student@pessat196: ~  
> db.towns.find(  
... {  
... name:/^P/,  
... population:{  
... $lt:10000  
... }  
... },  
... {  
... _id:0,  
... name:1,  
... population:1  
... })  
{ "name" : "Punxsutawney", "population" : 6200 }  
> var population_range={  
... $lt:1000000,  
... $gt:10000  
... }  
> db.towns.find({  
... name:/^P/,  
... population:population_range  
... },  
... {  
... name:1  
... }  
... )  
{ "_id" : ObjectId("612da20076a62dd7a10470f0"), "name" : "Portland" }  
>
```

Use \$gte:

```
student@pessat196: ~  
... {  
...   name:/^P/,  
...   population:{  
...     $lt:10000  
...   }  
... },  
... {  
...   _id:0,  
...   name:1,  
...   population:1  
... })  
{ "name" : "Punxsutawney", "population" : 6200 }  
> var population_range={  
...   $lt:1000000,  
...   $gt:10000  
... }  
> db.towns.find(  
...   name:/^P/,  
...   population:population_range  
... },  
... {  
...   name:1  
... }  
... )  
{ "_id" : ObjectId("612da20076a62dd7a10470f0"), "name" : "Portland" }  
> db.towns.find(  
... {lastCensus:{  
...   $gte:ISODate('2016-10-01')}},  
... { _id:0,name:1  
... }  
... )  
> db.towns.find( {lastCensus:{ $gte:ISODate('2016-06-01')}} , { _id:0,name:  
{ "name" : "New York" }  
>
```

Use \$all:

```
> db.towns.find(  
... {famousFor:'food'},  
... { _id:0,name:1,famousFor:1})  
{ "name" : "New York", "famousFor" : [ "the MOMA", "food", "Derek Jeter" ] }  
{ "name" : "Portland", "famousFor" : [ "beer", "food", "Portlandia" ] }  
> db.towns.find(  
... { famousFor : /MOMA/ },  
... { _id : 0, name : 1, famousFor : 1 } )  
{ "name" : "New York", "famousFor" : [ "the MOMA", "food", "Derek Jeter" ] }  
> db.towns.find(  
... {famousFor:{$all:['food','beer']}},  
... { _id : 0, name:1, famousFor:1 }  
... )  
{ "name" : "Portland", "famousFor" : [ "beer", "food", "Portlandia" ] }  
> { _id : 0, name:1, famousFor:1 }
```


Use \$Exists:


```
student@pessat196: ~  
> db.towns.find(  
... { famousFor : { $nin : ['food', 'beer'] } },  
... { _id : 0, name:1, famousFor:1 }  
... )  
{ "name" : "Punxsutawney", "famousFor" : [ "Punxsutawney Phil" ] }  
> db.towns.find(  
... {'mayorparty':{'$exists:false}},  
... { _id:0,name:1,mayor:1}  
... )  
{ "name" : "New York", "mayor" : { "name" : "Bi;; de Blasio", "party" : "D"  
} }  
{ "name" : "Punxsutawney", "mayor" : { "name" : "Richard Alexander" } }  
{ "name" : "Portland", "mayor" : { "name" : "Ted Wheeler", "party" : "D" } }  
>
```

Use \$nin:

```
> db.towns.find(  
... { famousFor : { $nin : ['food', 'beer'] } },  
... { _id : 0, name:1, famousFor:1 }  
... )  
{ "name" : "Punxsutawney", "famousFor" : [ "Punxsutawney Phil" ] }  
> db.towns.find(  
... {'mayorparty':{'$exists:false}},  
... { _id:0,name:1,mayor:1}  
... )  
{ "name" : "New York", "mayor" : { "name" : "Bi;; de Blasio", "party" : "D"  
} }  
{ "name" : "Punxsutawney", "mayor" : { "name" : "Richard Alexander" } }  
{ "name" : "Portland", "mayor" : { "name" : "Ted Wheeler", "party" : "D" } }  
> db.countries.insert(  
... _id:"us",  
... name:"United States",  
... exports:{  
... foods : [  
... { name : "bacon", tasty : true },  
... { name : "burgers" }  
... ]  
... }  
... } )  
WriteResult({ "nInserted" : 1 })
```

Insert into new collection:

```
student@pessat196: ~  
> db.countries.insert({  
.. _id : "mx",  
.. name : "Mexico",  
.. exports : {  
.. foods : [{  
.. name : "salsa",  
.. tasty : true,  
.. condiment : true  
.. }]  
.. }  
.. })  
WriteResult({ "nInserted" : 1 })  
> db.countries.count()  
2  
> db.countries.find(  
.. {'exports.foods.name':'bacon','exports.food.tasty':true},  
.. {_id:0,name:1}  
.. )  
2021-08-31T09:24:36.410+0530 E QUERY [js] uncaught exception: SyntaxEr  
r: missing name after . operator :  
js(shell):2:30  
> db.countries.find( {'exports.foods.name':'bacon','exports.food.tasty':tr  
{, {_id:0,name:1} )  
> db.countries.find(  
.. { 'exports.foods.name' : 'bacon', 'exports.foods.tasty' : true},  
.. { _id : 0, name : 1 }  
.. )  
[ "name" : "United States" ]  
> 
```

```
> db.countries.find(  
... {  
... 'exports.foods':{$elemMatch:{name:'bacon',tasty:true}}},{ _id : 0, nam  
: 1 } )  
[ "name" : "United States" ]  
> 
```

Insert one and Insert Many Commands:

```
student@pessat196: ~  
> db.inventory.insertOne(  
... { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }  
... )  
{  
  "acknowledged" : true,  
  "insertedId" : ObjectId("612dabae8b1d7cc3c02f9b92")  
}  
> 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" : [  
    ObjectId("612dabcc8b1d7cc3c02f9b93"),  
    ObjectId("612dabcc8b1d7cc3c02f9b94"),  
    ObjectId("612dabcc8b1d7cc3c02f9b95")  
  ]  
}  
> db.collection.find( { qty: { $gt: 4 } } )  
> db.inventory.find( { qty: { $gt: 4 } } )  
{ "_id" : ObjectId("6127166be23a1e43e4eebacb"), "item" : "canvas", "qty" : 100, "tags" : [ "cotton" ], "size" : { "h" : 28, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebacc"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "size" : { "h" : 14, "w" : 21, "uom" : "cm" } }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebacd"), "item" : "mat", "qty" : 85, "tags" : [ "grey" ], "size" : { "h" : 27, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebace"), "item" : "mousepad", "qty" : 25, "tags" : [ "gel", "blue" ], "size" : { "h" : 19, "w" : 22.85, "uom" : "cm" } }
```

Delete Many Command:

```
> db.inventory.deleteMany({status:"A"})  
{ "acknowledged" : true, "deletedCount" : 0 }
```

Find with \$gt:

```
student@pessat196: ~  
: "cm" } } },  
... { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w:  
22.85, uom: "cm" } } } )  
{  
  "acknowledged" : true,  
  "insertedIds" : [  
    ObjectId("612dabcc8b1d7cc3c02f9b93"),  
    ObjectId("612dabcc8b1d7cc3c02f9b94"),  
    ObjectId("612dabcc8b1d7cc3c02f9b95")  
  ]  
}  
> db.collection.find( { qty: { $gt: 4 } } )  
> db.inventory.find( { qty: { $gt: 4 } } )  
{ "_id" : ObjectId("6127166be23a1e43e4eebacb"), "item" : "canvas", "qty" : 1  
00, "tags" : [ "cotton" ], "size" : { "h" : 28, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebacc"), "item" : "journal", "qty" :  
25, "tags" : [ "blank", "red" ], "size" : { "h" : 14, "w" : 21, "uom" : "cm"  
} }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebacd"), "item" : "mat", "qty" : 85,  
"tags" : [ "grey" ], "size" : { "h" : 27, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612718f5e23a1e43e4eebace"), "item" : "mousepad", "qty" :  
25, "tags" : [ "gel", "blue" ], "size" : { "h" : 19, "w" : 22.85, "uom" : "  
cm" } }  
{ "_id" : ObjectId("612dabae8b1d7cc3c02f9b92"), "item" : "canvas", "qty" : 1  
00, "tags" : [ "cotton" ], "size" : { "h" : 28, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612dabcc8b1d7cc3c02f9b93"), "item" : "journal", "qty" :  
25, "tags" : [ "blank", "red" ], "size" : { "h" : 14, "w" : 21, "uom" : "cm"  
} }  
{ "_id" : ObjectId("612dabcc8b1d7cc3c02f9b94"), "item" : "mat", "qty" : 85,  
"tags" : [ "gray" ], "size" : { "h" : 27.9, "w" : 35.5, "uom" : "cm" } }  
{ "_id" : ObjectId("612dabcc8b1d7cc3c02f9b95"), "item" : "mousepad", "qty" :  
25, "tags" : [ "gel", "blue" ], "size" : { "h" : 19, "w" : 22.85, "uom" : "  
cm" } }  
>
```

Update Operation:

```
> db.people.update(  
... {name:"Andy"},  
... {  
... name:"Andy",  
... rating:1,  
... score:1  
... },  
... {upsert:true}  
... )  
WriteResult({  
  "nMatched" : 0,  
  "nUpserted" : 1,  
  "nModified" : 0,  
  "_id" : ObjectId("612daca210c1ffdc08e752ae")  
})
```

All CRUD Operations : **DONE**

Delete Operation: **DONE**