NAME:AVINASH BHIWALE ROLL NO.:T1851050 PRN NO.:71901284J DIV:A

Assignment No. 13

AIM: Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques:

- find and findOne (specific values)
- Query criteria (Query conditionals, OR queries, $not, Conditional semantics)
- Type-specific queries (Null, Regular expression, Querying arrays)

THEORY:

The find() Method

Execute at least 10 queries on any suitable MongoDB database that demonstrates following:

- $ where queries
- Cursors (Limits, skips, sorts, advanced query options)
- Database commands

MongoDB's **find()** method, explained in MongoDB Query Document accepts second optional parameter that is list of fields that you want to retrieve. In MongoDB, when you execute **find()** method, then it displays all fields of a document. To limit this, you need to set a list of fields with value 1 or 0. 1 is used to show the field while 0 is used to hide the fields.

Syntax

The basic syntax of **find() method with projection is as follows** −

```
>db.COLLECTION_NAME.find({},{KEY:1})
```

Following example will display the title of the document while querying the document.

```
>db.mycol.find({},{"title":1,_id:0})
{"title":"MongoDB Overview"}
{"title":"NoSQL Overview"}
{"title":"Tutorials Point Overview"}
>
```

Apart from find() method there is findOne() method, that reruns only one document.

AND in MongoDB

Syntax

In the **find()** method, if you pass multiple keys by separating them by ',' then MongoDB treats it as **AND** condition. Following is the basic syntax of **AND** −

```
>db.mycol.find(
   {
      $and: [
         {key1: value1}, {key2:value2}
      ]
   }
).pretty()
```

Example

Following example will show all the tutorials written by 'tutorials point' and whose title is 'MongoDB Overview'.

```
>db.mycol.find({$and:[{"by":"tutorials point"},{"title":"MongoDB Overview"}]}).pretty(){

"_id":ObjectId(7df78ad8902c),

"title":"MongoDB Overview",

"description":"MongoDB is no sql database",

"by":"tutorials point",

"url":"http://www.tutorialspoint.com",

"tags":["mongodb","database","NoSQL"],

"likes":"100"

}
```

For the above given example, equivalent where clause will be ' where by = 'tutorials point' AND title = 'MongoDB Overview' '. You can pass any number of key, value pairs in find clause.

OR in MongoDB

Syntax

To query documents based on the OR condition, you need to use **$or**keyword. Following is the basic syntax of **OR –**

```
>db.mycol.find(

{

     $or:[

{key1: value1},{key2:value2}

]

}
).pretty()
```

Example

Following example will show all the tutorials written by 'tutorials point' or whose title is 'MongoDB Overview'.

```
>db.mycol.find({$or:[{"by":"tutorials point"},{"title":"MongoDB

Overview"}]}).pretty() {

"_id":ObjectId(7df78ad8902c),

"title":"MongoDB Overview",

"description":"MongoDB is no sql database",

"by":"tutorials point",

"url":"http://www.tutorialspoint.com",

"tags":["mongodb","database","NoSQL"],

"likes":"100"

}

>
```

Using AND and OR Together

Example

The following example will show the documents that have likes greater than 10 and whose title is either 'MongoDB Overview' or by is 'tutorials point'. Equivalent SQL where clause is **'where likes>10 AND (by = 'tutorials point' OR title = 'MongoDB Overview')'**

```
>db.mycol.find({"likes":{$gt:10}, $or:[{"by":"tutorials point"},

{"title":"MongoDB Overview"}]}).pretty()
```

Conclusion: Executed MongoDB queries using AND, OR and NOT conditions

Output:

```
hp@hp-HP-Notebook:~$ mongo
MongoDB shell version: 2.6.10
connecting to: test
> show dbs;
admin (empty)
local 0.078GB
akash 0.078GB
test 0.078GB
> db.createCollection(employee)
2019-10-18T11:32:54.134+0530 ReferenceError: employee is not
defined > db.createCollection("employee")
```

```
{ "ok" : 1 } >
show collections;
Akash employee

akash student
system.indexes

> db.employee.insert({"_id":1,"lid":1,"lname":"akash","salary":90000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":2,"lid":2,"lname":kirti,"salary":80000})
2019-10-18T11:44:43.583+0530 ReferenceError: kirti is not defined
> db.employee.insert({"_id":2,"lid":2,"lname":"kirti","salary":80000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":3,"lid":3,"lname":"kirti","salary":60000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":3,"lid":3,"lname":"aditi","salary":60000})
WriteResult({
      "nInserted" : 0,
      "writeError" : { "code"
                  : 11000,
                  "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error index:
test.employee.$_id_ dup key: { : 3.0 }"
      }
})
> db.employee.insert({"_id":3,"lid":4,"lname":"aditi","salary":60000})
WriteResult({
      "nInserted" : 0,
      "writeError" : { "code"
                  : 11000,
                  "errmsg" : "insertDocument :: caused by :: 11000 E11000 duplicate key error index:
test.employee.$_id_ dup key: { : 3.0 }"
      }
})
> db.employee.insert({"_id":4,"lid":3,"lname":"aditi","salary":60000})

> db.employee.insert({"_id":5,"lid":5,"lname":"suraj","salary":40000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":6,"lid":6,"lname":"aditya","salary":30000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":7,"lid":7,"lname":"pratiksha","salary":20000})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({"_id":8,"lid":7,"lname":"tejas","salary":10000})
WriteResult({ "nInserted" : 1 })
> db.employee.find()
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find().pretty()
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.empoyee.findone()
2019-10-18T12:32:10.243+0530 TypeError: Property 'findone' of object test.empoyee is not a function
> db.empoyee.findOne()
null
> db.employee.findOne()
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" :
90000 } > db.employee.findTwo()
2019-10-18T12:32:50.549+0530 TypeError: Property 'findTwo' of object test.employee is not a
function > db.employee.find({"salary":{$gt:25000}})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
> db.empolyee.find({"salary":{"$gt:90000"}})
2019-10-18T13:36:51.642+0530 SyntaxError: Unexpected token }
> db.empolyee.find({"salary":{$gt:90000}})
> db.empolyee.find({"salary":{$lt:90000}})
> db.employee.find({"salary":{$lt:90000}})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({"salary":{$gt:90000}})
> db.employee.find({"salary":{$gt:60000}})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
> db.employee.find({"salary":{$gte:60000}})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000
} > db.employee.find({"salary":{$gte:90000}})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" :
90000 } > db.employee.find({"salary":{$lte:10000}})
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({"salary":{$lte:1000}})
> db.employee.find({"salary":{$gte:1000}})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 } >
db.employee.find({$and:[{"lid":1}{"lname":"akash"}]})
2019-10-18T13:48:44.849+0530 SyntaxError: Unexpected token
{ > db.employee.find({$and:[{"lid":1},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({$and:[{"lid":1},{"lsurname":"shinde"}]})
> db.employee.find({$and:[{"lid":1},{"lname":"kirti"}]})
> db.employee.find({$and:[{"lid":3},{"lname":"kirti"}]})
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 } >
db.employee.find({$and:[{"lid":3},{"lname":"aditi"}]}) {
"_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 } >
db.employee.find({$and:[{"lid":3},{"lname":"tejas"}]}) >
db.employee.find()
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({$and:[{"_id":6},{"lname":aditya}]})
2019-10-18T13:56:05.949+0530 ReferenceError: aditya is not defined
> db.employee.find({$and:[{"_id":6},{"lname":"aditya"}]})
{ "_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
> db.employee.find({$or:[{"_id":5},{"lname":"aditya"}]})
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 } {
"_id" : 6, "lid" : 6, "lname" : "aditya", "salary" : 30000 }
> db.employee.find({$or:[{"_id":5},{"salary":10000}]})
{ "_id" : 5, "lid" : 5, "lname" : "suraj", "salary" : 40000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({$or:[{"lname":"aditi"},{"salary":10000}]})
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({$or:[{"lname":"kirti"},{"salary":10000}]})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 8, "lid" : 7, "lname" : "tejas", "salary" : 10000 }
> db.employee.find({$or:[{"lname":"kirti"},{"salary":60000}]})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
> db.employee.find({$or:[{"lname":"kirti"},{"salary":6000}]})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
> db.employee.find({$or:[{"lname":"aditi"},{"salary":6000}]})
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
> db.employee.find({$and:[{"lname":"aditi"},{"salary":6000}]})
> db.employee.find({"salary":{$gt:60000},$or:{"lname":"akash"},{"lid":7})
...
...
> db.employee.find({"salary":{$gt:60000},$or:[{"lid":7},{"lname":akash}]})
2019-10-18T14:09:09.780+0530 ReferenceError: akash is not defined
> db.employee.find({"salary":{$gt:60000},$or:[{"lid":7},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
```

```
> db.employee.find({"salary":{$gt:10000},$or:[{"lid":7},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 7, "lid" : 7, "lname" : "pratiksha", "salary" : 20000 }
> db.employee.find({"salary":{$gt:10000},$or:[{"lid":8},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gt:80000},$or:[{"lid":8},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gt:80000},$or:[{"lid":1},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gt:80000},$or:[{"lid":2},{"lname":"akash"}]})
{ "_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gte:80000},$or:[{"lid":2},{"lname":"akash"}]}) {
"_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
> db.employee.find({"salary":{$gte:80000},$or:[{"lid":1},{"lname":"akash"}]}) {
"_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gte:80000},$or:[{"lid":3},{"lname":"akash"}]}) {
"_id" : 1, "lid" : 1, "lname" : "akash", "salary" : 90000 }
> db.employee.find({"salary":{$gte:60000},$or:[{"lid":3},{"lname":"kirti"}]})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
{ "_id" : 3, "lid" : 3, "lname" : "kirti", "salary" : 60000 }
{ "_id" : 4, "lid" : 3, "lname" : "aditi", "salary" : 60000 }
> db.employee.find({"salary":{$gt:60000},$or:[{"lid":3},{"lname":"kirti"}]})
{ "_id" : 2, "lid" : 2, "lname" : "kirti", "salary" : 80000 }
> db.employee.updsate({"_id":1},$set:{"address":"pune"})
2019-10-18T14:17:32.174+0530 SyntaxError: Unexpected token :
> db.employee.updsate({"_id":1},{$set:{"address":"pune"}}) 2019-10-18T14:18:06.162+0530
TypeError: Property 'updsate' of object test.employee is not a function >
db.employee.update({"_id":1},$set:{"address":"pune"})
2019-10-18T14:18:23.895+0530 SyntaxError: Unexpected token :
> db.employee.update({"_id":1},{$set:{"address":"pune"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.employee.findOne
function ( query , fields, options ){
    var cursor = this.find(query, fields, -1 /* limit */, 0 /* skip*/,
        0 /* batchSize */, options);

    if ( ! cursor.hasNext() )
        return null;
    var ret = cursor.next();
    if ( cursor.hasNext() ) throw "findOne has more than 1
    result!"; if ( ret.$err )
        throw "error " + tojson( ret );
    return ret;
}
> db.employee.findOne()
{
      "_id" : 1,
      "lid" : 1,
      "lname" : "akash",
      "salary" : 90000,
      "address" : "pune"
}
```

```
> db.employee.findOne().pretty()
2019-10-18T14:19:18.073+0530 TypeError: Object [object Object] has no method 'pretty'

> db.employee.update({"_id":1},{$set:{"address":"pune"}})
```