NAME :- DIVYANG BAGLA
PANEL :- C PC 33
C-3

FINAL YEAR B.TECH

PAGE NO:

BDA LAB ASSIGNMENT - 2

AIM :- To execute any 10 queries on suitable sample mongodb database to demonstrate various query.

OBJECTIVE :- To study and execute Mongodb under. To study & execute aggregate functions of mongodb

THEORY :- Explain different aggregation pipeline operators

① Stages Operators :-
$project :- Passes along documents with and specified fields to next stage in pipeline.
$match :- Filters documents based on material of specified conditions to next pipeline stage.
$group :- groups document by some specific expression of groups documents for distinct grouping id fields contains distinct group by key.

② Set operators :-
$ set equals :- compares b/w two & more arrays & return true if same distinct element else false.
$ set information :- $ set union $ set difference same as set operations.
$ set subset :- 2 args :- returns true when first array is subset of second or even if first arrays equals second

* Explain type of indices in Mongodb :-
① single field :- use to create index on single field of document
② compound index :- Mongodb supports user defined index on multiple fields.
③ Multikey index :- Mongodb uses the multikey index to index the values stored in array.

(4) GeoSpatial index :- supports 2D & 2D sphere index. 2-D uses planar geometry whereas other uses spherical geometry.

(5) Text index :- supports searching of string content in collection.

* INPUT :- Sample collection
* OUTPUT :- Execution of all commands.
* CONCLUSION :- Successfully performed all queries.

FAQ's =

Q.1 Which pipeline operator is similar to where and having in mongodb?

→ where ~ $where
having ~ $group <criteria>
$then $match <criteria>

Q.2 How do you know which indicies are defined over a collection?

use :- db.collection.getIndexes() method to know index defined.

Q.3 What is application of $unwind in mongodb?

used to deconstruct an array field in a document and create separate document for each items in an array.

1. Create database EMP and make Collection with name "EMPL" Insert documents into EMPL Collection with following structure/data

```
> db.createCollection("EMPL")
{ "ok" : 1 }
> db.EMPL.insertMany([{eno:1,name:"ST",salary:2000,role:"OB"}, {
eno:2,name:"MSD",salary:1500,role:"WK"},
...  { eno:3,name:"YS",salary:1000,role:"ALR"},{
eno:4,name:"RD",salary:1000,role:"MOB"},
...  { eno:5,name:"RS",salary:500,role:"OB"}, {
eno:6,name:"BK",salary:500,role:"MOB"},
...  { eno:7,name:"VK",salary:300,role:"BW"}, {
eno:8,name:"JB",salary:400,role:"BW"},
...  { eno:9,name:"HP",salary:400,role:"ALR"}, {
eno:10,name:"VS",salary:300,role:"OB"}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("6156e4f10b6180511df786cc"),
                ObjectId("6156e4f10b6180511df786cd"),
                ObjectId("6156e4f10b6180511df786ce"),
                ObjectId("6156e4f10b6180511df786cf"),
                ObjectId("6156e4f10b6180511df786d0"),
                ObjectId("6156e4f10b6180511df786d1"),
                ObjectId("6156e4f10b6180511df786d2"),
                ObjectId("6156e4f10b6180511df786d3"),
                ObjectId("6156e4f10b6180511df786d4"),
                ObjectId("6156e4f10b6180511df786d5")
        ]
}
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 2000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 1500, "role" : "WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 1000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 1000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
```

: 300, "role" : "OB" }

Queries :-

1. Display Data in proper format

```
> db.EMPL.find().pretty()
{
        "_id" : ObjectId("6156e4f10b6180511df786cc"),
        "eno" : 1,
        "name" : "ST",
        "salary" : 2000,
        "role" : "OB"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786cd"),
        "eno" : 2,
        "name" : "MSD",
        "salary" : 1500,
        "role" : "WK"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786ce"),
        "eno" : 3,
        "name" : "YS",
        "salary" : 1000,
        "role" : "ALR"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786cf"),
        "eno" : 4,
        "name" : "RD",
        "salary" : 1000,
        "role" : "MOB"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d0"),
        "eno" : 5,
        "name" : "RS",
        "salary" : 500,
        "role" : "OB"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d1"),
        "eno" : 6,
        "name" : "BK",
        "salary" : 500,
        "role" : "MOB"
```

```
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d2"),
        "eno" : 7,
        "name" : "VK",
        "salary" : 300,
        "role" : "BW"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d3"),
        "eno" : 8,
        "name" : "JB",
        "salary" : 400,
        "role" : "BW"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d4"),
        "eno" : 9,
        "name" : "HP",
        "salary" : 400,
        "role" : "ALR"
}
{
        "_id" : ObjectId("6156e4f10b6180511df786d5"),
        "eno" : 10,
        "name" : "VS",
        "salary" : 300,
        "role" : "OB"
}
```

2. Update Salary of Employee where Name is "ST" by +8000

```
> db.EMPL.updateOne({"name" : "ST"} ,{$inc: {"salary" : 8000}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 10000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 1500, "role" : "WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 1000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 1000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
```

```
: 400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 300, "role" : "OB" }
```

3. Update Salary Of All Employee by giving an increment of +4000 each

```
> db.EMPL.updateMany({},{$inc : {"salary": 4000}})
{ "acknowledged" : true, "matchedCount" : 10, "modifiedCount" : 10 }
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 5000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
```

4. Update role of "MSD" as "C and WK"

```
> db.EMPL.updateOne({"name" : "MSD"}, {$set : {"role" : "C and WK"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 5000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
```

```
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
```

5. Add a New Field remark to document with name "RS" set Remark as WC

```
> db.EMPL.update({name:"RS"},{$set:{remark:"WC"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 5000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "salary" :
4500, "role" : "OB", "remark" : "WC" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
```

6. Add a New Field As Number 11,name AK,Salary 10000,role coch without using insert statement. But for Doing So You should have a Record Added woth number 11

```
> db.EMPL.save({eno: 11, name: "AK", "salary" : 10000, role: "coach" })
WriteResult({ "nInserted" : 1 })
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 5000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB", "remark" : "WC" }
```

```
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
{ "_id" : ObjectId("6156eac50b6180511df786d6"), "eno" : 11, "name" : "AK", "salary"
: 10000, "role" : "coach" }
```

7. remove added New Field

```
> db.EMPL.update({name: "RS"} , {$unset : {remark:
...
... "WC"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

8. Update the Field "RD" by multiplying with salary by 2

```
> db.EMPL.updateOne({name:"RD"}, {$mul : {salary : 2}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.EMPL.find()
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 10000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
{ "_id" : ObjectId("6156eac50b6180511df786d6"), "eno" : 11, "name" : "AK", "salary"
: 10000, "role" : "coach" }
```

9. To Find Document From the empl collection where name begins with S

```
> db.EMPL.find({name: /^S/})
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
```

10. To Find Document From the empl collection where name ends with K

```
> db.EMPL.find({name : /S/})
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
>
```

11. Display Documents where in empl collection field have OB,MOB

```
> db.EMPL.find({role : {$in : ["OB" , "MOB"]}})
{ "_id" : ObjectId("6156e4f10b6180511df786cc"), "eno" : 1, "name" : "ST", "salary"
: 14000, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786cf"), "eno" : 4, "name" : "RD", "salary"
: 10000, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d0"), "eno" : 5, "name" : "RS", "salary"
: 4500, "role" : "OB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d1"), "eno" : 6, "name" : "BK", "salary"
: 4500, "role" : "MOB" }
{ "_id" : ObjectId("6156e4f10b6180511df786d5"), "eno" : 10, "name" : "VS", "salary"
: 4300, "role" : "OB" }
```

12. Display Documents where in empl collection field not have OB,MOB

```
> db.EMPL.find({role : {$nin : ["OB" , "MOB"]}})
{ "_id" : ObjectId("6156e4f10b6180511df786cd"), "eno" : 2, "name" : "MSD", "salary"
: 5500, "role" : "C and WK" }
{ "_id" : ObjectId("6156e4f10b6180511df786ce"), "eno" : 3, "name" : "YS", "salary"
: 5000, "role" : "ALR" }
{ "_id" : ObjectId("6156e4f10b6180511df786d2"), "eno" : 7, "name" : "VK", "salary"
: 4300, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d3"), "eno" : 8, "name" : "JB", "salary"
: 4400, "role" : "BW" }
{ "_id" : ObjectId("6156e4f10b6180511df786d4"), "eno" : 9, "name" : "HP", "salary"
: 4400, "role" : "ALR" }
{ "_id" : ObjectId("6156eac50b6180511df786d6"), "eno" : 11, "name" : "AK", "salary"
: 10000, "role" : "coach" }
```

2.Create a database petshop with collection pets with following structure/data

{name: "Mikey", species: "Gerbil"}, {name: "Davey Bungooligan", species: "Piranha"},
{name: "Suzy B", species: "Cat"}, {name: "Mikey", species: "Hotdog"},
{name: "Terrence", species: "Sausagedog"}, {name: "Philomena Jones", species: "Cat"}

```
> db.PETS.insertMany([{name: "Mikey", species: "Gerbil"}, {name: "Davey Bungooligan", species: "Piranha"}, {name: "Suzy B", species: "Cat"}, {name: "Mikey", species: "Hotdog"}, {name: "Terrence", species: "Sausagedog"}, {name: "Philomena Jones", species: "Cat"}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("6156f49f0b6180511df786ed"),
                ObjectId("6156f49f0b6180511df786ee"),
                ObjectId("6156f49f0b6180511df786ef"),
                ObjectId("6156f49f0b6180511df786f0"),
                ObjectId("6156f49f0b6180511df786f1"),
                ObjectId("6156f49f0b6180511df786f2")
        ]
}
> db.PETS.find()
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" : "Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786ee"), "name" : "Davey Bungooligan", "species" : "Piranha" }
{ "_id" : ObjectId("6156f49f0b6180511df786ef"), "name" : "Suzy B", "species" : "Cat" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" : "Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" : "Sausagedog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f2"), "name" : "Philomena Jones", "species" : "Cat" }
```

Queries:-

1. Add another piranha, and a mole rat called Henry.

```
> db.PETS.insert({name : "Nemo"}, {species: "Piranha"})
WriteResult({ "nInserted" : 1 })
> db.PETS.insert({name : "Henry"}, {species: "naked mole rat"})
WriteResult({ "nInserted" : 1 })
> db.PETS.find()
```

```
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786ee"), "name" : "Davey Bungooligan",
"species" : "Piranha" }
{ "_id" : ObjectId("6156f49f0b6180511df786ef"), "name" : "Suzy B", "species" :
"Cat" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" :
"Sausagedog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f2"), "name" : "Philomena Jones",
"species" : "Cat" }
{ "_id" : ObjectId("6156f5a00b6180511df786f3"), "name" : "Nemo" }
{ "_id" : ObjectId("6156f5bb0b6180511df786f4"), "name" : "Henry" }

2. use find to list all the pets.
> db.PETS.find()
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786ee"), "name" : "Davey Bungooligan",
"species" : "Piranha" }
{ "_id" : ObjectId("6156f49f0b6180511df786ef"), "name" : "Suzy B", "species" :
"Cat" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" :
"Sausagedog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f2"), "name" : "Philomena Jones",
"species" : "Cat" }
{ "_id" : ObjectId("6156f5a00b6180511df786f3"), "name" : "Nemo" }
{ "_id" : ObjectId("6156f5bb0b6180511df786f4"), "name" : "Henry" }

3.Find the ID of Mikey the Gerbil.

> db.PETS.findOne({name :  "Mikey"})
{
        "_id" : ObjectId("6156f49f0b6180511df786ed"),
        "name" : "Mikey",
        "species" : "Gerbil"
}

4. Use find to find Mikey by id

> db.PETS.findOne({_id: ObjectId("6156f49f0b6180511df786ed")})
{
        "_id" : ObjectId("6156f49f0b6180511df786ed"),
        "name" : "Mikey",
        "species" : "Gerbil"
}
```

5. Use find to find all the gerbils.

```
> db.PETS.find({species : "Gerbil"})
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
```

6. Find all the creatures named Mikey

```
> db.PETS.find({name: "Mikey"})
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
```

7. Find all the creatures named Mikey who are gerbils.

```
> db.PETS.find({name: "Mikey", species : "Gerbil"})
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
```

8. Find all the creatures with the string "dog" in their species.

```
> db.PETS.find({species: /dog/})
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" :
"Sausagedog" }
```

9. Add the price field in the collection for species Cat.

```
> db.PETS.update({species: "Cat"} , {$set : {price : '' }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.PETS.find()
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786ee"), "name" : "Davey Bungooligan",
"species" : "Piranha" }
{ "_id" : ObjectId("6156f49f0b6180511df786ef"), "name" : "Suzy B", "species" :
"Cat", "price" : "" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" :
"Sausagedog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f2"), "name" : "Philomena Jones",
"species" : "Cat" }
{ "_id" : ObjectId("6156f5bb0b6180511df786f4"), "name" : "Henry" }
{ "_id" : ObjectId("6156f8780b6180511df786f6"), "name" : "Nemo", "species" :
"Piranha" }
```

10. update the price field for the species Piranha .

```
> db.PETS.update({species: "Piranha"} , {$set : {price : 2000 }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.PETS.find()
{ "_id" : ObjectId("6156f49f0b6180511df786ed"), "name" : "Mikey", "species" :
"Gerbil" }
{ "_id" : ObjectId("6156f49f0b6180511df786ee"), "name" : "Davey Bungooligan",
"species" : "Piranha", "price" : 2000 }
{ "_id" : ObjectId("6156f49f0b6180511df786ef"), "name" : "Suzy B", "species" :
"Cat", "price" : "" }
{ "_id" : ObjectId("6156f49f0b6180511df786f0"), "name" : "Mikey", "species" :
"Hotdog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f1"), "name" : "Terrence", "species" :
"Sausagedog" }
{ "_id" : ObjectId("6156f49f0b6180511df786f2"), "name" : "Philomena Jones",
"species" : "Cat" }
{ "_id" : ObjectId("6156f5bb0b6180511df786f4"), "name" : "Henry" }
{ "_id" : ObjectId("6156f8780b6180511df786f6"), "name" : "Nemo", "species" :
"Piranha" }
```

11. find the first 3 species

```
>  db.PETS.aggregate(
... {$group : { _id : "$species"}}
... ,{$sort : {_id: 1}},
... {$limit : 3} );


{ "_id" : "Cat" }
{ "_id" : "Gerbil" }
{ "_id" : "Hotdog" }
```