MongoDB Shell Lab Practice – Basic Commands 1

Create database EMP and Make Collection With name "EMPL" and Follow Queries

Created Database

```
> use emp

 switched to DB emp
```

Create Collection

```
> db.createCollection("empl")
```

Insert Records Into EMPL Collection

```
> db.empl.insert([

 {no:1,name:"ST",salary:2000,role:"OB"},

 {no:2,name:"MSD",salary:1500,role:"WK"},

 {no:3,name:"YS",salary:1000,role:"ALR"},

 {no:4,name:"RD",salary:1000,role:"MOB"},

 {no:5,name:"RS",salary:500,role:"OB"},

 {no:6,name:"BK",salary:500,role:"MOB"},

 {no:7,name:"VK",salary:300,role:"BW"},

 {no:8,name:"JB",salary:400,role:"BW"},

 {no:9,name:"HP",salary:400,role:"ALR"},

 {no:10,name:"VS",salary:300,role:"OB"}])
```

Display Data in Proper Format

```
> db.empl.find().pretty()
```

Update Salary Of Employee where Name is "ST" by +8000

```
> db.empl.update({name:"ST"},{$inc:{salary:8000}})
```

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

```
> db.empl.update({},{$inc:{salary:4000}},{multi:true})
```

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

```
> db.empl.update({name:"MSD"},{$set:{role:"c and WK"}})
```

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

```
> db.emp.update({name:"RS"},{$set:{remark:"WC"}})
```

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.update({no:11},{$set:{no:11,name:"AK",salary:10000,role:"coch"}},{upsert:true})
```

Remove added New Field

```
> db.empl.update({name:"RD"},{$unset:{remark:"WC"}})
```

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

```
> db.empl.update({name:"RD"},{$mul:{salary:2}})
```

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

```
> db.empl.find({name:/^S/})
```

To Find Document From the empl collection where name begins with R

```
> db.empl.find({name:/^R/})
```

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

```
> db.empl.find({name:/K$/})
```

To Find Document From the empl collection where name ends with D

```
> db.empl.find({name:/S$/})
```

To Find Document From the empl collection where name has S in any position

```
> db.empl.find({name:/S/})
```

Regular Expression

(Note: Use Case sensitive allow For that write in Option: "i")

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

```
> db.empl.find({name:{$regex:"^S"}})
```

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

```
> db.empl.find({name:{$regex:"S",$options:"i"}})
```

Use of $in and $nin (in and notin)

(Note: There will not use {} braces in that $in and $nin)

Display Documents where in empl collection Field have OB,MOB

```
> db.empl.find({role:{$in:["OB","MOB"]}})
```

Display Documents where in empl collection Field not have OB,MOB

```
> db.empl.find({role:{$nin:["OB","MOB"]}})
```

MongoDB Shell Lab Practice – Basic Commands 2

Create a database named “mydb”

```
> use mydb
```

Show all databases

```
> show dbs
```

Show currently selected database.

```
> db
```

Create a collection named “emp”.

```
> db.createCollection(“emp”)
```

Show all collections of the selected database.

```
> show collections
```

Insert following documents in emp collection:

```
> db.emp.insert(
[
{ empno : 1,ename : "Sachin",sal : 60000,desige : "Manager",dept : "Purchase" },
{ empno : 2,ename : "Kohali",sal : 50000,desige : "Manager",dept : "Sales" }
])
```

Using the Find() function.

List all documents.

```
> db.emp.find()
```

List all documents with formatted output.

```
> db.emp.find.pretty()
```

List the document of an employee whose name is “Sachin”

```
> db.emp.find({ename:”Sachin”})
```

List the documents of employee whose salary is less than 30000

```
 > db.emp.find({sal : {$lt : 30000}})
```

List employees whose designation is manager and department is sales.

```
> db.emp.find( { desige : "Manager",dept : "Sales"})
```

List employees whose salary is less than 50000 and designation is manager or department is admin.

```
> db.emp.find(

{sal : {$lt : 50000},

$or : [

{ desige : "Manager"},{dept : "Admin"}

]})
```

Arrange the records by name in descending order.

```
> db.emp.find().sort({ename : -1})
```

List first 3 documents of emp collection.

```
> db.emp.find().limit(3)
```

Skip first 3 documents of emp collection.

```
> db.emp.find().skip(3)
```

List 3rd and 4th documents of emp collection.

```
> db.emp.find().limit(2).skip(2)
```

Count no. of employees.

```
> db.emp.find().count()
```

List distinct designation.

```
> db.emp.distinct(“desige”)
```

Using the Aggregate() function.

Calculate annual salary of employees.

```
>
db.emp.aggregate( [ {    $project:{_id:0,ename:1,sal:1,desige:1,dept:1,Annual_Salary:{$multiply:["$sal",12]}}}]).pretty()
```

Display no. of employees designation wise.

```
> db.emp.aggregate( [ {  $group :  { _id:"$dept",Employees:{$sum:1}}}] )
```

Display total salary of employees department wise.

```
>  db.emp.aggregate( [ {    $group:    { _id:"$dept",Tot_Sal:{$sum:"$sal"} }} ] )
```

Display employee whose salary is highest.

```
> db.emp.aggregate( [ {  $sort:{sal: -1}}, {  $limit: 1} ] )
```

List highest salary of the employee department wise.

```
> db.emp.aggregate( [ {  $group:{ _id : "$dept",Highest_Sal : {$max:"$sal"} } } ] )
```

Calculate total salary of employees department wise and list whose total is greater than 70000.

```
> db.emp.aggregate( [ { $group: {_id:"$dept",total_sal:{$sum:"$sal"}}},{ $match:
{total_sal:{$gt:70000}}}])
```

Using the Index() function.

Create index on empno field.

```
> db.emp.ensureIndex( { empno:1} )
```

Check the statistics of the emp collection.

```
> db.emp.stats()
```

Get the list of all indexes created on the emp collection.

```
> db.emp.getIndexes()
```

Use hint method.

```
> db.emp.find( { empno:3}).hint({empno:1})
```

Use explain method.

```
> db.emp.find( { empno:3} ).hint({ empno:1}).explain()
```

Using the Update() function.

Increment salary by 2000 whose name is Sachin.

```
> db.emp.update({ ename:"Sachin"},{$inc:{sal:2000}})
```

Increment salary by 2000 of all the employees.

```
> db.emp.update({}, { $inc:{sal:2000}},{multi:true})
```

Update designation of Sachin with CEO.

```
>  db.emp.update({ename:"Sachin"},{$set:{desig:"CEO"}})
```

Update designation of Sachin with ‘MD’ and insert as new document.

```
> db.emp.update({ename:"Sachin"},{$set:{desig:"MD"}},{upsert:true})
```

Add a new field “remark” to document with name “dhoni” and set remark “head”

```
> db.emp.update({ename:"Dhoni"},{$set:{remaek:"head"}})
```

Remove the added new field.

```
> db.emp.update({ename:"Dhoni"},{$unset:{remaek:"head"}})
```

Using the Save() function.

Replace the whole document whose _id is 1.

```
> db.emp.save( {_id: ObjectId("59860ee940b855fbf3566a20") ,
empno:1,ename:"Tendulkar",sal:"40000",desig:"Clerk",dept:"Admin"})
```

Using Regular Expression.

To find documents from the emp collection where the ename begins with “S”. (with and without regex)

```
> db.emp.find({ename:/^S/})

> db.emp.find({ename:{$regex:"^S"}})
```

To find documents from the emp collection where the ename ends with “n”. (with and without regex)

```
> db.emp.find({ename:/n$/})

> db.emp.find({ename:{$regex:"n$"}})
```

Modify above query with case insensitivity.

```
> db.emp.find({ename:/N$/i})

> db.emp.find({ename:{$regex:"n$",$options:"i"}})
```

To find documents from the emp collection where ename has an “a” in any position.

```
> db.emp.find({ename:/a/})

> db.emp.find({ename:{$regex:".*a.*"}})
```

Dealing with null values :

Update the document with null value in designation field where name is “Kohli”.

```
> db.emp.update( {ename:"Kohali"}, {$set:{desig:null}} )
```

To search for null values in designation field.

```
> db.emp.find( {desig:null})
```

To remove designation field having null values in emp collection where name is “Kohli”.

```
> db.emp.update( { ename:"Kohali"},{$unset:{desig:null}})
```