Date: 01-04-2021

Session 2

To uninstall the package

sudo apt remove mongodb-org

To delete entire folder

sudo apt purge mongodb-org\*

Steps for creating collection Employee and inserting documents

1. show dbs;
2. use EmployeeDB;

switched to db EmployeeDB

1. db.createCollection("Employee",{size:30000,capped:true,max:12});

{ "ok" : 1 }

1. show collections;

Employee

1. Open vs code and create employee.js file and add insert 10 values
2. load("/home/divyah/Desktop/Vs\_code/employee.js");

true

1. db.Employee.find().pretty();

Keywords (Starts with $)

1. $eq = equals
2. $neq = Not equal
3. $gt = greater than
4. $lt = lesser than
5. $gte = greater than or equal
6. $lte = lesser than or equal to
7. $in = in list
8. $nin = not in list
9. $exist = exists

Query 1

List out all the records having salary >20000

db.Employee.find({

"Salary":{$gt:20000}

}).pretty();

Output

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a3"),

"Name" : {

"First\_Name" : "Anusha",

"Middle\_Name" : "R",

"Last\_Name" : "Bhat"

},

"Age" : 25,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Team Lead",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a4"),

"Name" : {

"First\_Name" : "Amala",

"Middle\_Name" : "H",

"Last\_Name" : "Hebbar"

},

"Age" : 46,

"Salary" : 60000,

"Designation" : "Product Manager",

"Role" : [

"Manager"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a6"),

"Name" : {

"First\_Name" : "Anika",

"Middle\_Name" : "Manav",

"Last\_Name" : "Kohli"

},

"Age" : 26,

"Salary" : 35000,

"Designation" : "Support",

"Role" : [

"Tester",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a7"),

"Name" : {

"First\_Name" : "Anjali",

"Middle\_Name" : "R",

"Last\_Name" : "Naik"

},

"Age" : 43,

"Salary" : 38000,

"Designation" : "QandA",

"Role" : [

"Tester"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a8"),

"Name" : {

"First\_Name" : "Alisha",

"Middle\_Name" : "R",

"Last\_Name" : "Mesta"

},

"Age" : 28,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Software Developer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a9"),

"Name" : {

"First\_Name" : "Alia",

"Middle\_Name" : "M",

"Last\_Name" : "Bhatt"

},

"Age" : 30,

"Salary" : 45000,

"Designation" : "Designer",

"Role" : [

"Team Lead",

"UI Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0aa"),

"Name" : {

"First\_Name" : "Priyanka",

"Middle\_Name" : "R",

"Last\_Name" : "Dhawan"

},

"Age" : 29,

"Salary" : 39000,

"Designation" : "Developer",

"Role" : [

"Manager",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0ab"),

"Name" : {

"First\_Name" : "Prajna",

"Middle\_Name" : "S",

"Last\_Name" : "Hegde"

},

"Age" : 31,

"Salary" : 40000,

"Designation" : "Designer",

"Role" : [

"Designer"

]

}

Query 2

List out all the employees having a middle name = “R”

db.Employee.find(

... {

... "Name.Middle\_Name":{$eq:"R"}

... }).pretty();

Output

"\_id" : ObjectId("60654cbe5d88f729b453b0a3"),

"Name" : {

"First\_Name" : "Anusha",

"Middle\_Name" : "R",

"Last\_Name" : "Bhat"

},

"Age" : 25,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Team Lead",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a7"),

"Name" : {

"First\_Name" : "Anjali",

"Middle\_Name" : "R",

"Last\_Name" : "Naik"

},

"Age" : 43,

"Salary" : 38000,

"Designation" : "QandA",

"Role" : [

"Tester"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a8"),

"Name" : {

"First\_Name" : "Alisha",

"Middle\_Name" : "R",

"Last\_Name" : "Mesta"

},

"Age" : 28,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Software Developer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0aa"),

"Name" : {

"First\_Name" : "Priyanka",

"Middle\_Name" : "R",

"Last\_Name" : "Dhawan"

},

"Age" : 29,

"Salary" : 39000,

"Designation" : "Developer",

"Role" : [

"Manager",

"Software Developer"

]

}

Query 3

Find all the records with age <42 years

db.Employee.find(

... {

... "Age":{$lt:42}

... }).pretty();

Output

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a3"),

"Name" : {

"First\_Name" : "Anusha",

"Middle\_Name" : "R",

"Last\_Name" : "Bhat"

},

"Age" : 25,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Team Lead",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a5"),

"Name" : {

"First\_Name" : "Ankita",

"Middle\_Name" : "S",

"Last\_Name" : "Sharma"

},

"Age" : 35,

"Salary" : 15000,

"Designation" : "Frontend Developer",

"Role" : [

"UI Designer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a6"),

"Name" : {

"First\_Name" : "Anika",

"Middle\_Name" : "Manav",

"Last\_Name" : "Kohli"

},

"Age" : 26,

"Salary" : 35000,

"Designation" : "Support",

"Role" : [

"Tester",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a8"),

"Name" : {

"First\_Name" : "Alisha",

"Middle\_Name" : "R",

"Last\_Name" : "Mesta"

},

"Age" : 28,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Software Developer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a9"),

"Name" : {

"First\_Name" : "Alia",

"Middle\_Name" : "M",

"Last\_Name" : "Bhatt"

},

"Age" : 30,

"Salary" : 45000,

"Designation" : "Designer",

"Role" : [

"Team Lead",

"UI Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0aa"),

"Name" : {

"First\_Name" : "Priyanka",

"Middle\_Name" : "R",

"Last\_Name" : "Dhawan"

},

"Age" : 29,

"Salary" : 39000,

"Designation" : "Developer",

"Role" : [

"Manager",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0ab"),

"Name" : {

"First\_Name" : "Prajna",

"Middle\_Name" : "S",

"Last\_Name" : "Hegde"

},

"Age" : 31,

"Salary" : 40000,

"Designation" : "Designer",

"Role" : [

"Designer"

]

}

Query 4

Count the number of records having the team role as Manager

db.Employee.find( { "Role":{$eq:"Manager"}}).count();

Output

2

Query 5

List out all the employees in the age group 30 to 42

db.Employee.find( { "Age":{$gt:30,$lt:42} } ).pretty();

Output

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a5"),

"Name" : {

"First\_Name" : "Ankita",

"Middle\_Name" : "S",

"Last\_Name" : "Sharma"

},

"Age" : 35,

"Salary" : 15000,

"Designation" : "Frontend Developer",

"Role" : [

"UI Designer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0ab"),

"Name" : {

"First\_Name" : "Prajna",

"Middle\_Name" : "S",

"Last\_Name" : "Hegde"

},

"Age" : 31,

"Salary" : 40000,

"Designation" : "Designer",

"Role" : [

"Designer"

]

}

Count of the same

db.Employee.find( { "Age":{$gt:30,$lt:42} }).count();

2

Query 6

List out all the employees who is having salary range from 20k to 40k

db.Employee.find( { "Salary":{ $lt:40000,$gt:20000 } } ).pretty();

Output

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a3"),

"Name" : {

"First\_Name" : "Anusha",

"Middle\_Name" : "R",

"Last\_Name" : "Bhat"

},

"Age" : 25,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Team Lead",

"Software Developer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a6"),

"Name" : {

"First\_Name" : "Anika",

"Middle\_Name" : "Manav",

"Last\_Name" : "Kohli"

},

"Age" : 26,

"Salary" : 35000,

"Designation" : "Support",

"Role" : [

"Tester",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a7"),

"Name" : {

"First\_Name" : "Anjali",

"Middle\_Name" : "R",

"Last\_Name" : "Naik"

},

"Age" : 43,

"Salary" : 38000,

"Designation" : "QandA",

"Role" : [

"Tester"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0a8"),

"Name" : {

"First\_Name" : "Alisha",

"Middle\_Name" : "R",

"Last\_Name" : "Mesta"

},

"Age" : 28,

"Salary" : 30000,

"Designation" : "Developer",

"Role" : [

"Software Developer",

"Designer"

]

}

{

"\_id" : ObjectId("60654cbe5d88f729b453b0aa"),

"Name" : {

"First\_Name" : "Priyanka",

"Middle\_Name" : "R",

"Last\_Name" : "Dhawan"

},

"Age" : 29,

"Salary" : 39000,

"Designation" : "Developer",

"Role" : [

"Manager",

"Software Developer"

]

}

Count of same

db.Employee.find({ "Salary":{$lt:40000,$gt:20000}}).count();

5

Different ways of inserting documents

1. **load(.js file)**
2. **Using insert()**

db.Employee.insert({ Name:{First\_Name:"Apoorva",Middle\_Name:"U",Last\_Name:"Shetty"},Age:27,Salary:40000,Designation:"Tester",Role:["Tester","Team Lead"]});

WriteResult({ "nInserted" : 1 })

Following document is added to the existing document

{

"\_id" : ObjectId("6065560b5d88f729b453b0ad"),

"Name" : {

"First\_Name" : "Apoorva",

"Middle\_Name" : "U",

"Last\_Name" : "Shetty"

},

"Age" : 27,

"Salary" : 40000,

"Designation" : "Tester",

"Role" : [

"Tester",

"Team Lead"

]

}

1. **Using save()**

var emp={};

emp.Name="Amulya";

Amulya

emp.Age=30;

30

emp.Salary=16000;

16000

emp.Designation="Software Developer";

Software Developer

db.Employee.save(emp);

WriteResult({ "nInserted" : 1 })

db.Employee.find().pretty();

Following document is added to the existing document

{

"\_id" : ObjectId("6065575f5d88f729b453b0ae"),

"Name" : "Amulya",

"Age" : 30,

"Salary" : 16000,

"Designation" : "Software Developer"

}

Deleting collection and database

**db.Employee.drop();**

true

**db.dropDatabase();**

{ "dropped" : "EmployeeDB", "ok" : 1 }