T P MADHU PRAKASH

1NT19IS176

ISE 6 C-2

Date: 2-Jun-2022

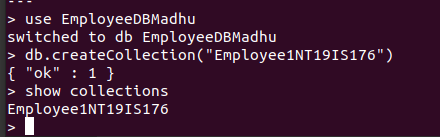
1.Problem Statement & Dataset

Create a collection named "Employee" under the "EmployeeDB" database with each document in the format shown below Table

use EmployeeDBMadhu

db.createCollection("Employee1NT19IS176")

show collections



2. Queries

1. Populate the database with atleast 15 documents

db.Employee1NT19IS176.insertMany([

{

Firstname: "T P",

Middlename: "Madhu",

Lastname: "Prakash",

Age: 25,

Salary: 75000,

Designation: "Employee",

Role: ["Software developer","Tester"],

},

{

Firstname: "Vinayak",

Middlename: "P",

Lastname: "Bhupal",

Age: 26,

Salary: 75000,

Designation: "HR",

Role: ["Manager","UI Designer"],

},

{

Firstname: "T",

Middlename: "Nikitha",

Lastname: "Reddy",

Age: 27,

Salary: 45000,

Designation: "Employee",

Role: ["Tester","Software Developer"],

},

{

Firstname: "M",

Middlename: "Kumar",

Lastname: "Babu",

Age: 27,

Salary: 70000,

Designation: "Researcher",

Role: ["Team Lead","Tester", "UI Designer"],

},

{

Firstname: "Rajesh",

Middlename: "Patil",

Lastname: "N",

Age: 36,

Salary: 50000,

Designation: "Scientist",

Role: ["UI Designer","Tester"],

},

{

Firstname: "Ullaas",

Lastname: "M",

Age: 36,

Salary: 50000,

Designation: "Scientist",

Role: ["UI Designer","Team Lead"]

},

]);

db.Employee1NT19IS176.insertMany([

{

Firstname: "Prateek",

Middlename: "P",

Lastname: "Nayak",

Age: 30,

Salary: 45000,

Designation: "Employee",

Role: ["Team Lead","Manager"]

},

{

Firstname: "Wakada",

Middlename: "Sai",

Lastname: "Sanjeev",

Age: 30,

Salary: 40000,

Designation: "Employee",

Role: ["Junior Architect","Manager"],

},

{

Firstname: "T",

Middlename: "H",

Lastname: "Lakshana",

Age: 30,

Salary: 20000,

Designation: "Employee",

Role: ["Software developer","Tester"],

},

{

Firstname: "Vishal",

Middlename: "Kumar",

Lastname: "H K",

Age: 40,

Salary: 75000,

Designation: "Scientist",

Role: ["Research and Developer","Tester"],

},

{

Firstname: "V",

Middlename: "Mohan",

Lastname: "Babu",

Age: 38,

Salary: 750000,

Designation: "President",

Role: ["Software developer","Research and Developer"]

},

]);

db.Employee1NT19IS176.insertMany([

{

Firstname: "Rajesh",

Middlename: "C",

Lastname: "R",

Age: 25,

Salary: 25000,

Designation: "Employee",

Role: ["Junior Lead","Junior Architect"],

},

{

Firstname: "Sohan",

Middlename: "M",

Lastname: "N",

Age: 33,

Salary: 60000,

Designation: "Employee",

Role: ["Junior Engineer","Software developer"],

},

{

Firstname: "Praneeth",

Middlename: "M",

Lastname: "V L S S S",

Age: 28,

Salary: 75000,

Designation: "Employee",

Role: ["Software developer","UI Designer"],

},

{

Firstname: "Laisha",

Middlename: "S",

Lastname: "A",

Age: 29,

Salary: 50000,

Designation: "Scientist",

Role: ["Research and Developer","UI Designer"],

},

{

Firstname: "Randheer",

Middlename: "Prasad",

Lastname: "B",

Age: 30,

Salary: 20000,

Designation: "System Engineer",

Role: ["Intern","Software developer"],

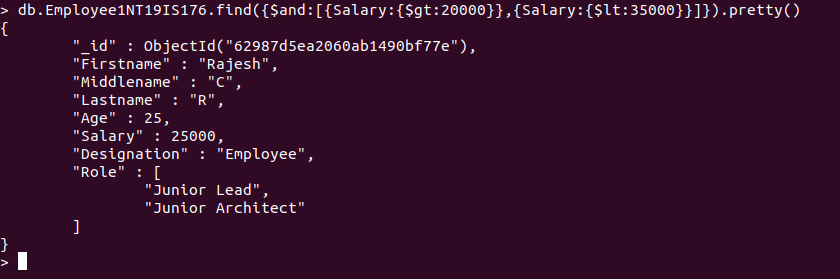
},

]);



2. List all the records having salary in the range of 20000 – 35000(Exclusive)

db.Employee1NT19IS176.find({$and:[{Salary:{$gt:20000}},{Salary:{$lt:35000}}]}).pretty()



3. List all the Employee whose Middle name is "Kumar"

db.Employee1NT19IS176.find({Middlename:"Kumar"}).pretty()



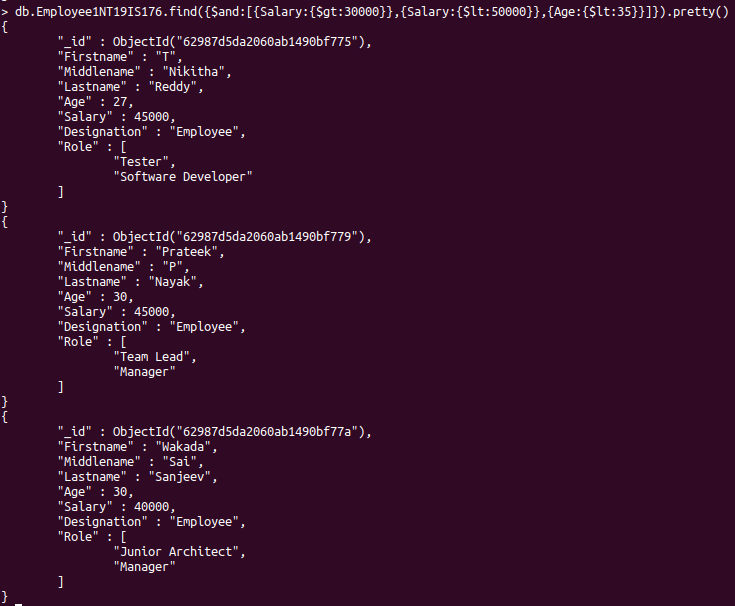
4. Count the number of Employees who has a role "Manager" in the Role field

db.Employee1NT19IS176.count({Role:"Manager"}).pretty



5. Find out all the documents who have age < 35 and salary in the range of 30000-50000

db.Employee1NT19IS176.find({$and:[{Salary:{$gt:30000}},{Salary:{$lt:50000}},{Age:{$lt:35}}]}).pretty()

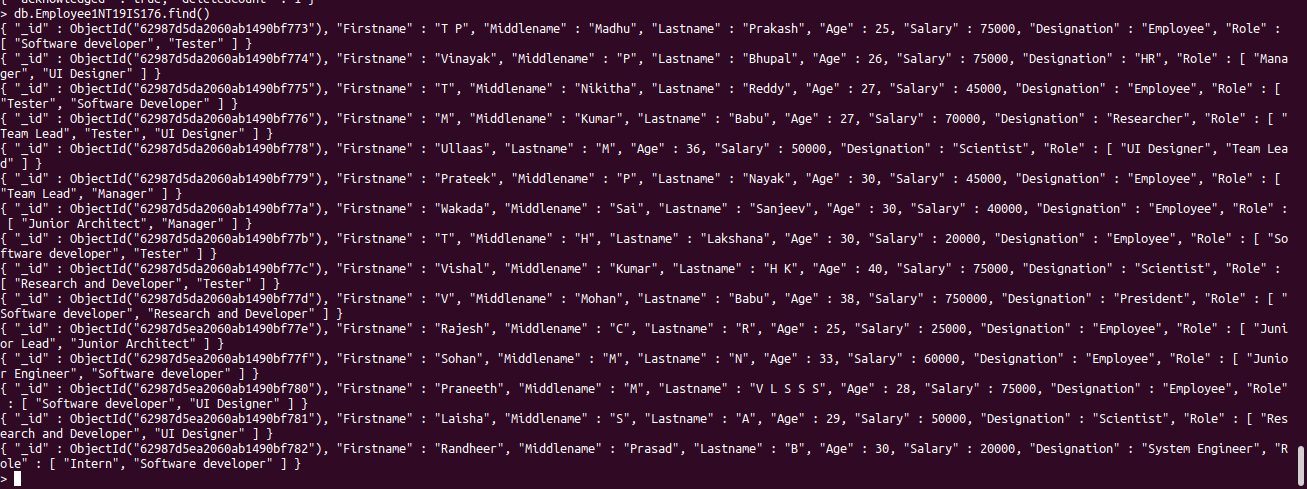


6. Delete an Employee whose "Firstname" is "Rajesh" and having the designation as "Scientist"

db.Employee1NT19IS176.deleteOne({$and:[{Firstname:"Rajesh"},{Designation:"Scientist"}]})



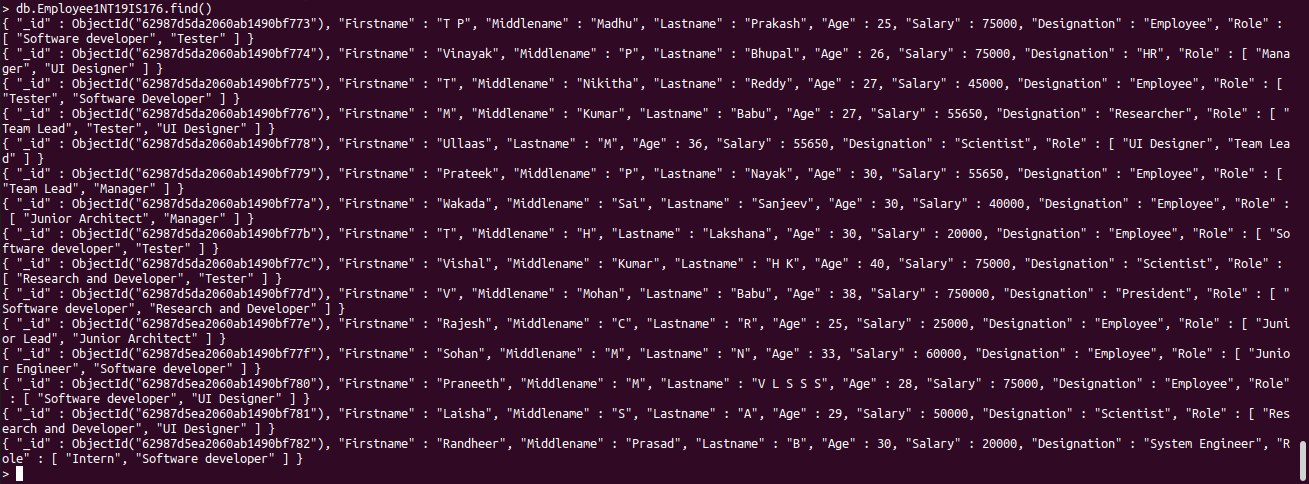
db.Employee1NT19IS176.find()



7. Update all the Employees whose role is "Team Lead" with a salary of 55650 INR

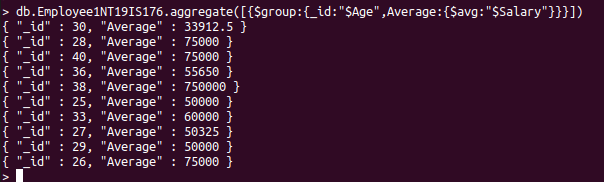
db.Employee1NT19IS176.updateMany({Role:"Team Lead"},{$set:{Salary:55650}})

 db.Employee1NT19IS176.find()



8. Group all the Employees by their age(common age should be there) and calculate the average salary obtained in the each group

db.Employee1NT19IS176.aggregate([{$group:{\_id:"$Age",Average:{$avg:"$Salary"}}}])

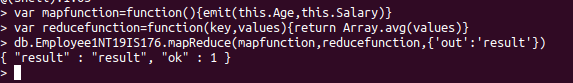


9. Apply the map-reduce to perform the above operation and obtain the results

var mapfunction=function(){emit(this.Age,this.Salary)}

var reducefunction=function(key,values){return Array.avg(values)}

db.Employee1NT19IS176.mapReduce(mapfunction,reducefunction,{'out':'result'})



db.result.find()

