**Pratheek P Nayak**

**1NT19IS117**

**BIG\_DATA\_LAB 31-05-2022**

**EXERCISE:**

**1. Populate the database with atleast 15 documents**

> db.employee.insertMany([

... {

... firstName: "Pratheek",

... lastName: "Nayak",

... age: 21,

... salary: 75000,

... designation: "Manager",

... role: ["Manager", "Team Lead", "Head"]

... },

... {

... firstName: "Sohan",

... lastName: "M",

... age: 22,

... salary: 60000,

... designation: "Employee",

... role: ["Associate", "Assistant"]

... },

... {

... firstName: "Rajesh",

... lastName: "CR",

... age: 31,

... salary: 69690,

... designation: "HOD",

... role: ["Leader", "Tester"]

... },

... {

... firstName: "Rajesh",

... lastName: "Kumar",

... age: 34,

... salary: 50000,

... designation: "Scientist",

... role: ["UI designer", "Developer"]

... },

... {

... firstName: "Praneeth",

... lastName: "MVLSSS",

... age: 40,

... salary: 35000,

... designation: "Scientist",

... role: ["Manager", "Tester"]

... },

... {

... firstName: "Shriman",

... lastName: "Mogaveer",

... age: 32,

... salary: 69000,

... designation: ["Team Lead"]

... },

... {

... firstName: "Rakshith",

... lastName: "Reddy",

... age: 31,

... salary: 45000,

... designation: ["Tester", "UI designer"]

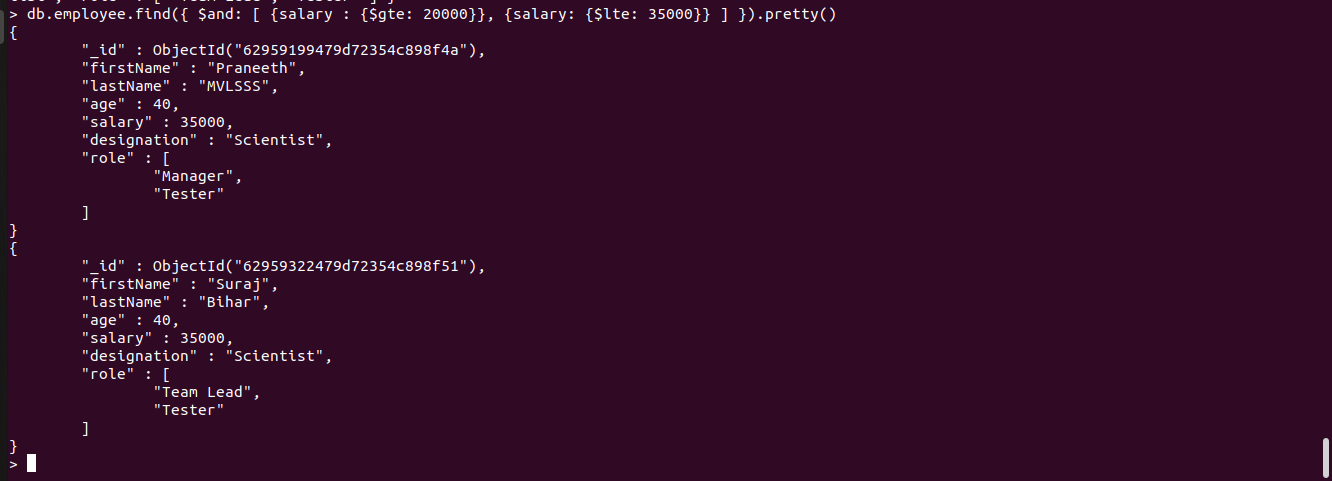
... }

... ])

> db.employee.insertMany([ { firstName: "Ullaas", lastName: "M", age: 30, salary: 65000, designation: "Manager", role: ["Manager", "Team Lead", "Head"] }, { firstName: "Vinayak", lastName: "Bhupal", age: 22, salary: 60000, designation: "Scientist", role: ["UI designer", "Assistant"] }, { firstName: "Sanjeev", lastName: "VVS", age: 31, salary: 69090, designation: "Professor", role: ["Leader", "Developer"] }, { firstName: "Vishal", lastName: "Kumar", age: 34, salary: 50000, designation: "Scientist", role: ["UI designer", "Tester"] }, { firstName: "Suraj", lastName: "Bihar", age: 40, salary: 35000, designation: "Scientist", role: ["Team Lead", "Tester"] }, { firstName: "Vansh", lastName: "Sharma", age: 32, salary: 69000, designation: ["Team Lead"] }, { firstName: "Madhu", lastName: "Prakash", age: 31, salary: 55000, designation: "HOD", role: ["UI designer"] } ])

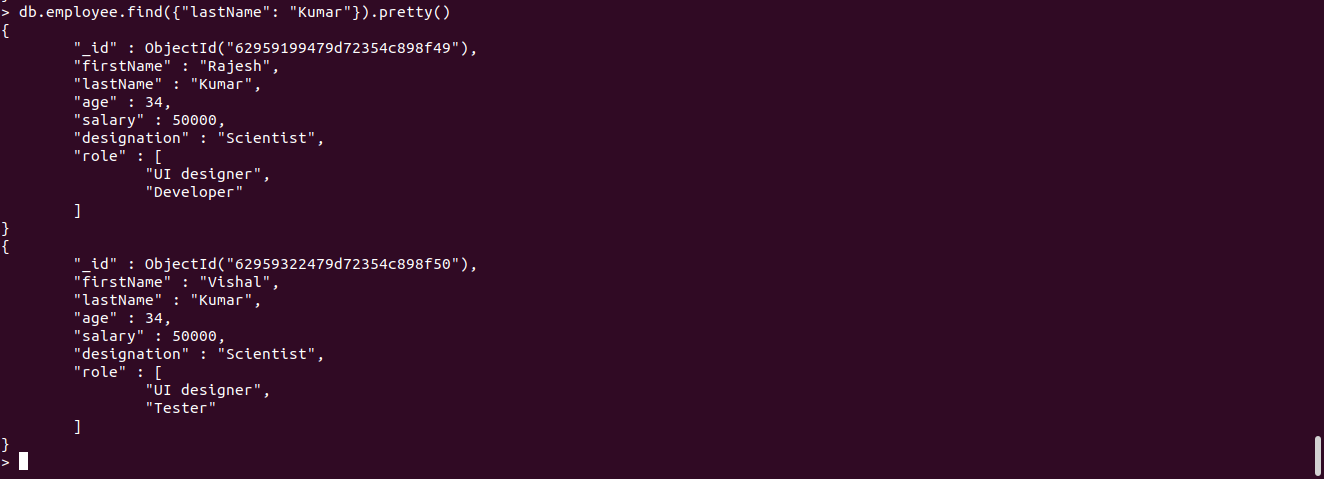
**2. List all the records having salary in the range of 20000 – 35000(Exclusive)**

> db.employee.find({ $and: [ {salary : {$gte: 20000}}, {salary: {$lte: 35000}} ] }).pretty()

****

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

> db.employee.find({"lastName": "Kumar"}).pretty()

****

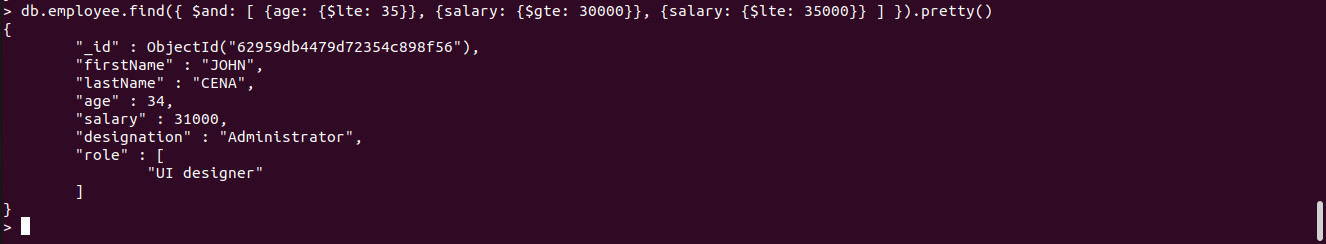
**4. Count the number of Employees who has a role "Manager" in the Role field**

> db.employee.count({role: "Manager"})



**5. Find out all the documents who have age < 35 and salary in the range of 30000-35000**

> db.employee.find({ $and: [ {age: {$lte: 35}}, {salary: {$gte: 30000}}, {salary: {$lte: 35000}} ] }).pretty()



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

> db.employee.remove({

... $and:

... [

... {firstName: "Rajesh"},

... {designation: "Scientist"}

... ]

... })



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

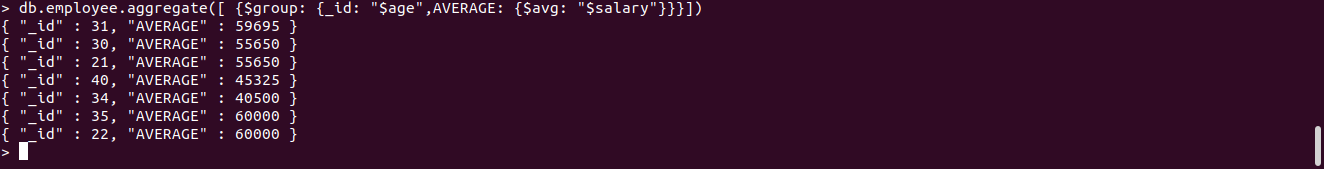
> db.employee.updateMany({role: "Team Lead"}, {$set: {salary: 55650}})



{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }

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

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

{ "result" : "result", "ok" : 1 }

