Create Collection “employees” with following data

[{_id: 1,name: "Eric",age: 30,position: "Full Stack Developer",salary: 60000}, {_id: 2,name: "Erica",age: 35,position: "Intern",salary: 8000},{_id: 3,name: "Erical",age: 40,position: "UX/UI Designer",salary: 56000}, {_id: 4,name: "treric7",age: 37,position: "Team Leader",salary: 85000}, {_id: 5,name: "Eliza",age: 25,position: "Software Developer",salary: 45000}, {_id: 6,name: "Trian",age: 29,position: "Data Scientist",salary: 75000}, {_id: 7,name: "Elizan",age: 25,position: "Full Stack Developer",salary: 49000}]

1) Find All Documents:

2) Find Documents by Position “Full Stack Developer”:

3) Retrieve name of employees whose age is greater than or equal to 25 and less than or equal to 40. 4) Retrieve name of the employee with the highest salary.

5) Retrieve employees with a salary greater than 50000.

6) Retrieve employees' names and positions, excluding the "_id" field.

7) Count the number of employees who have salary greater than 50000

8) Retrieve employees who are either " Software Developer" or "Full Stack Developer" and are below 30 years.

9) Increase the salary of an employee who has salary less than 50000 by 10%.

10) Delete all employees who are older than 50.

11) Give a 5% salary raise to all "Data Scientist"

12) Find documents where name like “%an”

13) Find documents where name like “Eri--” (Case Insensitive)

14) Find documents where name like “%ric%”

15) Find documents where name contains only 4 or 5 letters.

16) Find documents where name must end with digit

Answers

1) db.employees.insertMany([{_id: 1,name: "Eric",age: 30,position: "Full Stack Developer",salary: 60000}, {_id: 2,name: "Erica",age: 35,position: "Intern",salary: 8000}, {_id: 3,name: "Erical",age: 40,position: "UX/UI Designer",salary: 56000}, {_id: 4,name: "treric7",age: 37,position: "Team Leader",salary: 85000}, {_id: 5,name: "Eliza",age: 25,position: "Software Developer",salary: 45000}, {_id: 6,name: "Trian",age: 29,position: "Data Scientist",salary: 75000}, {_id: 7,name: "Elizan",age: 25,position: "Full Stack Developer",salary: 49000}])

2) db.employees.find()

3) db.employees.find({position:"Full Stack Developer"})

4) db.employees.find({ age: { $gte: 30, $lte: 40 } },{name:1,_id:0})

5) db.employees.find({},{name:1,_id:0}).sort({ salary: -1 }).limit(1)

6) db.employees.find({ salary: { $gt: 50000 } })

7) db.employees.find({salary:{$gt:50000}}).count()

8) db.employees.find({ $and: [{ $or: [{ position: "Software Developer" }, { position: "Full Stack Developer" }] }, { age: { $lt: 30 } }] })

9) db.employees.updateOne({salary:{$lt:50000}},{ $mul: { salary: 1.1 } })

10) db.employees.deleteMany({ age: { $gt: 50 } })

11) db.employees.updateMany({ position: "Data Scientist" }, { $mul: { salary: 1.05 }})

12) db.employees.find({name:{$regex:/an$/}})

13) db.employees.find({name:{$regex:/^eri[A-z]{2}$/i}})

14) db.employees.find({name:{$regex:/ric/i}})

15) db.employees.find({name:{$regex:/^[A-Za-z]{4,5}$/i}})

16) db.employees.find({name:{$regex:/[0-9]$/}})