MongoDB Assessment

Jai Goyal

Q1) You can use find to:

1. Find a document by id
2. Find a user by email
3. Find a list of all users with the same first name
4. Find all cats who are more than 12 years old
5. Find all gerbils called ‘Herbie’ who are bald, have three or more eyes, and who have exactly 3 legs
6. db.cars.find({id: ‘1’})
7. db.user.findOne({email:’jai@gmail.com’})
8. db.user.find({ fName: ‘Rick’ })
9. db.user.find(age: { $gt: 12})
10. db.pets.find({ species: "gerbil", name: "Herbie", bald: true, eyes: { $gte: 3 }, legs: 3 });

Q2 a.

Paste the following into your terminal to create a petshop with some pets in it

use petshop

db.pets.insert({name: "Mikey", species: "Gerbil"})

db.pets.insert({name: "Davey Bungooligan", species: "Piranha"})

db.pets.insert({name: "Suzy B", species: "Cat"})

db.pets.insert({name: "Mikey", species: "Hotdog"})

db.pets.insert({name: "Terrence", species: "Sausagedog"})

db.pets.insert({name: "Philomena Jones", species: "Cat"})

b. Add another piranha, and a naked mole rat called Henry.

db.pets.insert({name: "Robert Mark", species: "Piranha"})

db.pets.insert({name: "Henry", species: "Nakedmole"})

c.

db.pets.find()

db.pets.find({ $and: [ { name: 'Mikey' }, { species: 'Gerbil' } ] }, {\_id:1})

d.

db.pets.find({\_id: ObjectId('668234c62927f1fa5ccc8988')})

e.

db.pets.find({species:'Gerbil'})

f.

db.pets.find({name:’Mickey’})

g. db.pets.find({ $and: [ { name: 'Mikey' }, { species: 'Gerbil' } ] })

h. db.pets.find({"species":{$regex:"dog"}});

Q3 a.

(*function*() {

*var* names = [

'Yolanda',

'Iska',

'Malone',

'Frank',

'Foxton',

'Pirate',

'Poppelhoffen',

'Elbow',

'Fluffy',

'Paphat'

];

*var* randName = *function*() {

*var* n = names.length;

return [

names[Math.floor(Math.random() \* n)],

names[Math.floor(Math.random() \* n)]

].join(' ');

};

*var* randAge = *function*(*n*) {

return Math.floor(Math.random() \* n);

};

for (*var* i = 0; i < 1000; ++i) {

*var* person = {

name: randName(),

age: randAge(100)

};

if (Math.random() > 0.4) {

if (Math.random() < 0.5) {

person.cat = {

name: randName(),

age: randAge(18)

};

} else if (Math.random() < 0.6) {

person.piranha = {

name: randName(),

age: randAge(18)

};

} else {

person.dachshund = {

name: randName(),

age: randAge(18)

};

}

}

db.people.insert(person);

}

})();

b.

db.people.find({age:99})

c. db.people.find({age:99})

d. db.people.find( { age: { $gt: 65 } } )

e. db.people.find({age: {$gt:12, $lt:20}})

4. a.

db.people.find({

cat: {

$exists: true,

},

});

b. db.people.find({ "$and": [ {"cat": {$exists: true,}}, {"age": {$gt: 60}} ] })

c. db.people.find({ "$and": [ {"cat": {$exists: true,}}, {age: {$gt:12, $lt:20}}] })

5. a. db.stocks.find({$where:"(this.price\_2007 - this.price\_2002)/this.price\_2002 > 0.5"})

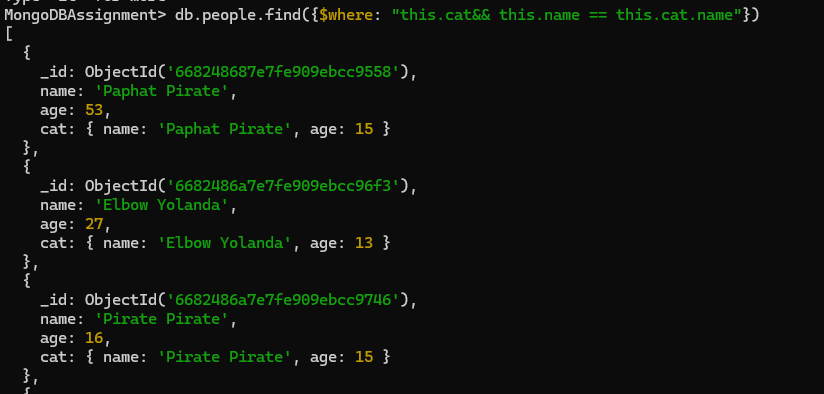
b. db.stocks.find({$where:"(this.price\_2007 - this.price\_2002)/this.price\_2002 < 0"})

6. a. db.people.find({$where: "this.cat"})

b. db.people.find({$where: "this.cat&& this.age<this.cat.age"})

c. db.people.find({$where: "this.cat&& this.name==this.cat.name"})

Yes there are such entries:



7. a. db.people.find({}, {name:true, \_id:false})

b. db.people.find({$where: "this.age>90"}, {name:true, \_id:false})

c. db.people.find({$where:"this.cat"}, {"cat.name":true, "cat.age":true, \_id:false})

8. a. db.people.find({$where:"this.cat"}, {"cat.name":true, "cat.age":true, \_id:false})

Format the stock:

1. db.stocks.updateMany( { price\_2002: { $ne: 0 } }, [ { $set: { profit: { $divide: [ { $subtract: ["$price\_2007", "$price\_2002"] }, "$price\_2002"] } } }] );

db.stocks.find({}, {"company":true, "sector":true, "profit":true})

1. db.stocks.find({$where:”profit>0.5”}, {"company":true, "sector":true, "profit":true})

9. a. db.people.count() [depreciated] =>db.people.countDocuments()

b. db.people.find({

cat: {

$exists: true,

},

}).count()

1. db.people.find({$where: "this.cat&& this.age<this.cat.age"}).count()

10. a. db.people.find().limit(5)

b. db.people.find().limit(5).skip(5);

c. db.people.find({$where : "this.piranha&&this.age>60"}, {"name":true, "age":true, \_id: false}).sort({age:-1}).limit(5)

d. db.people.find({ $where: "this.cat && this.age > 12 && this.age < 20", "cat.name": { $regex: "Yolanda" } }, { name: true, age: true, \_id: false }).sort({ age: 1 }).limit(5);

11. a. db.people.find({ cat: { $exists: true }, piranhas: { $exists: true } }).sort({ age: 1 }).limit(1);

b. db.people.find({ cat: { $exists: true }, piranhas: { $exists: true } }, { name: 1, \_id: 0 }).sort({ age: 1 }).limit(1);

c. db.people.find({ cat: { $exists: true } }, { "cat.name": 1, "cat.age": 1, \_id: 0 }).sort({ "cat.age": -1 }).limit(5);

d. db.people.find({ cat: { $exists: true } }, { "cat.name": 1, "cat.age": 1, \_id: 0 }).sort({ "cat.age": -1 }).skip(5).limit(5);

Exercise - Stocks

a. db.stocks.find().sort({ profit: -1 }).limit(10);

Q12

1. db.people.insert({ name: "Thor Thunder", age: 26, shark: { name: "Tony Shark", age: 51 } });

Q13

1. db.people.find({$where:"this.shark"})
2. db.people.findOne({$where:"this.shark"})

Q14. a. db.people.update({},{$inc: { age: 1 }}, {multi:true})

b.

db.people.updateMany({}, { $inc: { age: 1 } });

db.people.updateMany({ cat: { $exists: true } }, { $inc: { "cat.age": 1 } });

db.people.updateMany({ piranha: { $exists: true } }, { $inc: { "piranha.age": 1 } });

db.people.updateMany({ dachshund: { $exists: true } }, { $inc: { "dachshund.age": 1 } });

Q15. a.db.people.find({ name: /Pirate/ });

b. var cursor = db.people.find({ name: /Pirate/ });

cursor.forEach(function(person) {db.people.updateOne({ \_id: person.\_id },{ $set: { parrot: { name: "Polly", age: 0 } } });});

1. db.people.deleteMany({$where: "this.age>60"}).
2. db.people.deleteMany({$where: "this.shark"})

Q16. db.people.aggregate()



Q17. a. db.people.aggregate([{$match: {$and: [{ age: 10 },{ "cat.age": 10 }]}}]);

b. db.people.aggregate([{$match: {$and: [{ age: {$gt:80}},{ "cat.age":{$gt: 15} }]}}]);

18. a. db.zips.find({state:'MA'}, {zip\_code:1, \_id:0})

b. db.zips.find({population:{$lt:1000}}, {zip\_code:1})

19. a. db.stocksDummy.aggregate({$project:{ \_id:false}});

20. db.user.aggregate({ $project: { surname: "$lName", }, });

21. db.people.aggregate([ { $match: { cat: { $exists: true } } }, { $project: { \_id: 0, catName: "$cat.name" } }] );

22. db.stocks.aggregate([ { $project: { \_id: 0, IPrice: "$initial\_price", profit: 1, company\_name: 1, ticker: 1 } }] )

23. db.people.aggregate([ { $project: { \_id: 0, name: 1, capitalized\_name: { $toUpper: { $substrCP: ["$name", 0, { $strLenCP: "$name" }] } } } }] )

25 a. db.people.aggregate({ $group: { \_id: "$name" } });

b. db.people.aggregate([ { $group: { \_id: "$name" } }, { $project: { \_id: 0, name: "$\_id" } }] );