Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №5

«Работа с БД в СУБД MongoDB»

по дисциплине «**Проектирование и реализация баз данных**»

Выполнил:

студент группы К3240
Бабан Виктория

Проверил:
Говорова Марина Михайловна

Санкт-Петербург

2022

Цель: овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

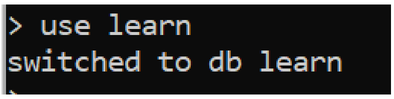
Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 5.0.8.

Практическое задание и выполнение:

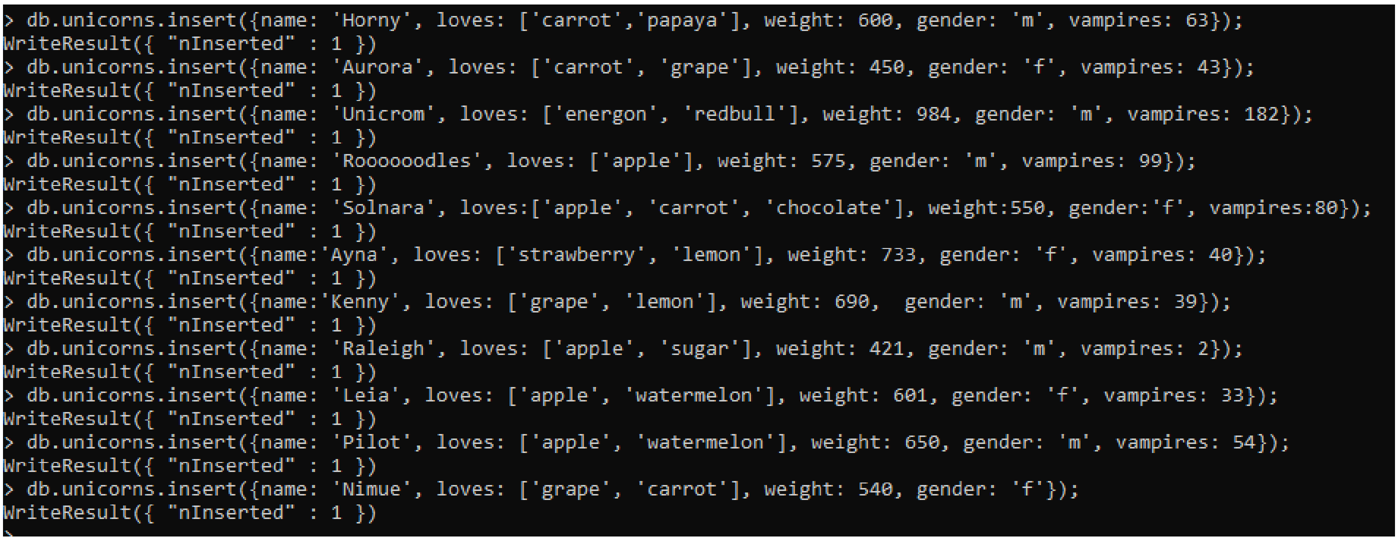
1. Вставка документов в коллекцию

1.1 Создайте базу данных learn.



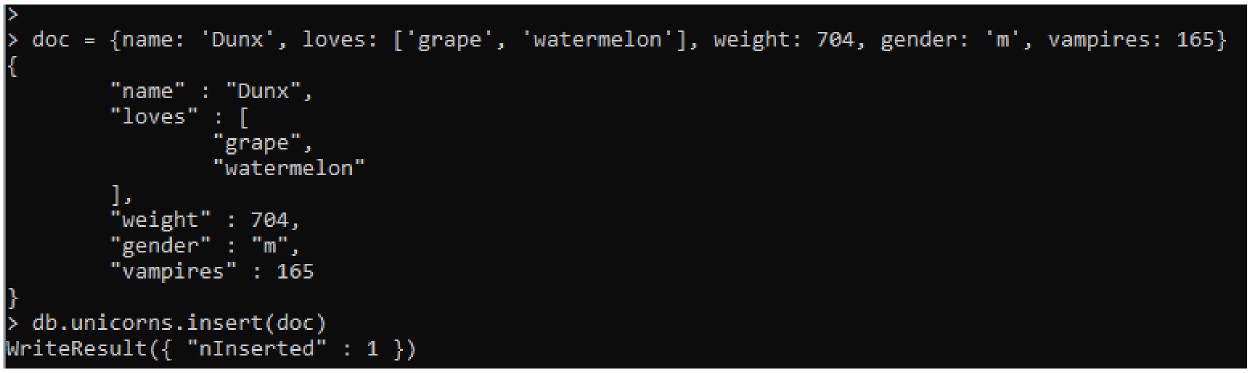
```
> use learn
switched to db learn
```

1.2 Заполните коллекцию единорогов unicorns:



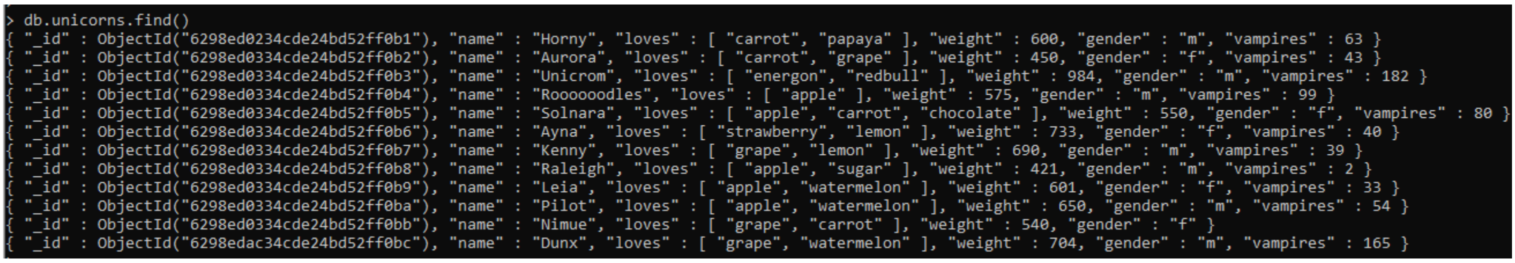
```
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
```

1.3 Используя второй способ, вставьте в коллекцию единорогов документ:



```
>
> doc = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
{
        "name" : "Dunx",
        "loves" : [
                "grape",
                "watermelon"
        ],
        "weight" : 704,
        "gender" : "m",
        "vampires" : 165
}
> db.unicorns.insert(doc)
WriteResult({ "nInserted" : 1 })
```

1.4 Проверьте содержимое коллекции с помощью метода find.



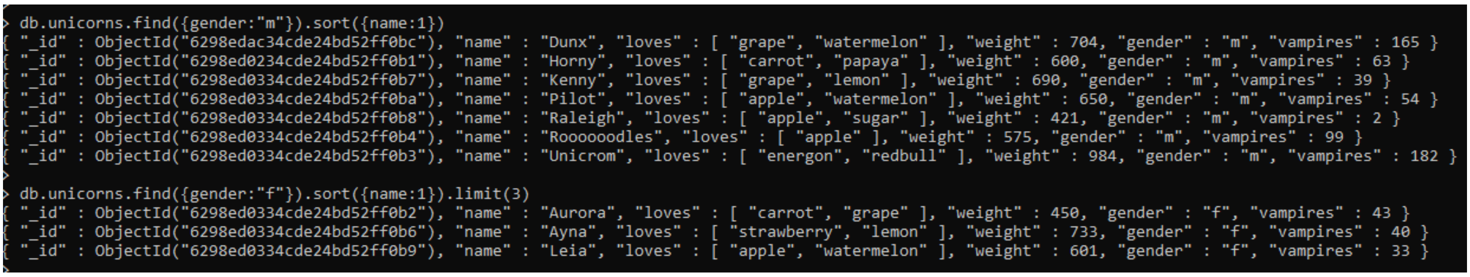
```
> db.unicorns.find()
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b2"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b5"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b6"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b9"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0bb"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

2. Выборка данных из БД

2.1 Сформируйте запросы для вывода списков самцов и самок единорогов.Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.

db.unicorns.find({gender:'m'}).sort({name: 1})

db.unicorns.find({gender:'f'}).sort({name: 1}).limit(3)

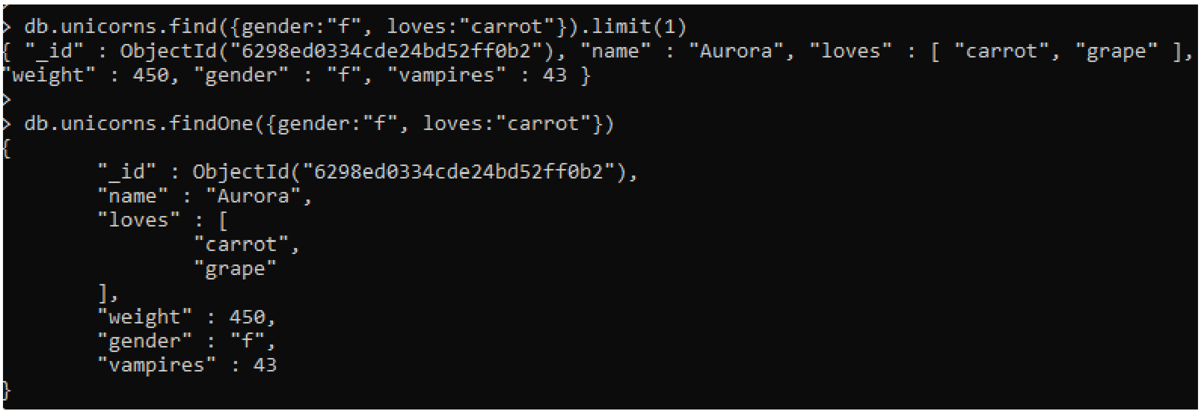


```
> db.unicorns.find({gender:"m"}).sort({name:1})
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
> db.unicorns.find({gender:"f"}).sort({name:1}).limit(3)
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b2"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b6"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b9"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
```

2.2 Найдите всех самок, которые любят carrot. Ограничьте этот список первой особьюс помощью функций findOne и limit.

db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)

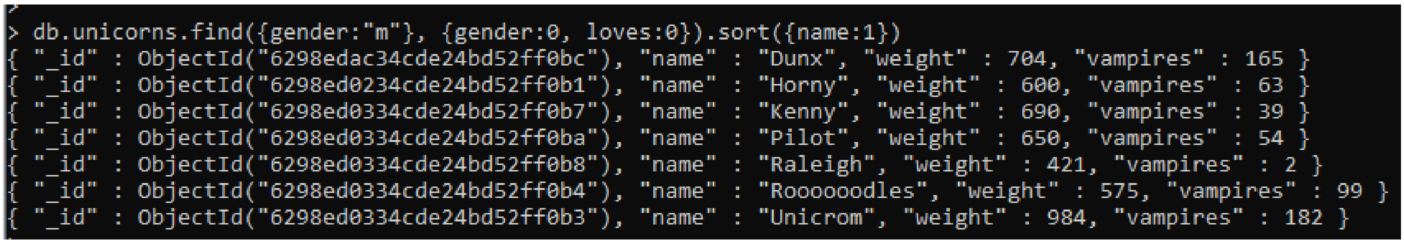
db.unicorns.findOne({gender: 'f', loves: 'carrot'})



```
> db.unicorns.find({gender:"f", loves:"carrot"}).limit(1)
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b2"), "name" : "Aurora", "loves" : [ "carrot", "grape" ],
"weight" : 450, "gender" : "f", "vampires" : 43 }
>
> db.unicorns.findOne({gender:"f", loves:"carrot"})
{
        "_id" : ObjectId("6298ed0334cde24bd52ff0b2"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
```

2.3 Модифицируйте запрос для вывода списков самцов единорогов, исключив изрезультата информацию о предпочтениях и поле.

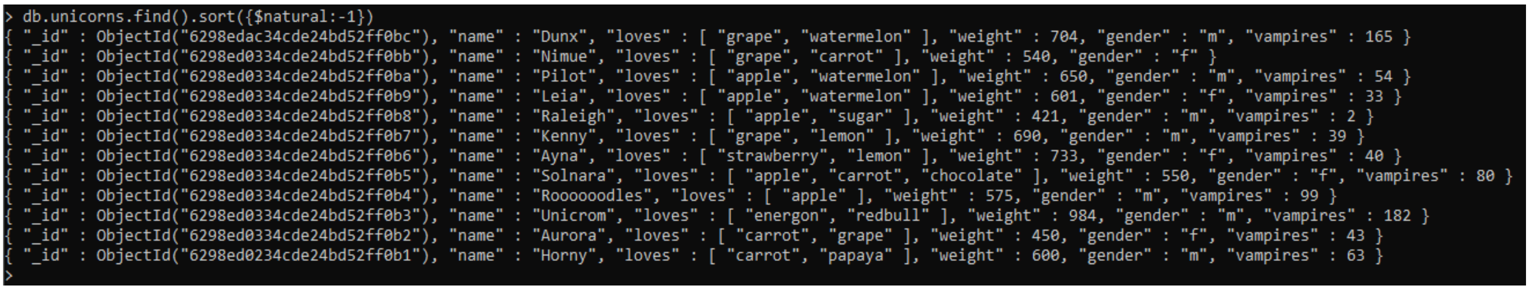
db.unicorns.find({gender: "m"}, { gender: 0, loves: 0}).sort({name: 1});



```
>
> db.unicorns.find({gender:"m"}, {gender:0, loves:0}).sort({name:1})
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Rooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

2.4 Вывести список единорогов в обратном порядке добавления.

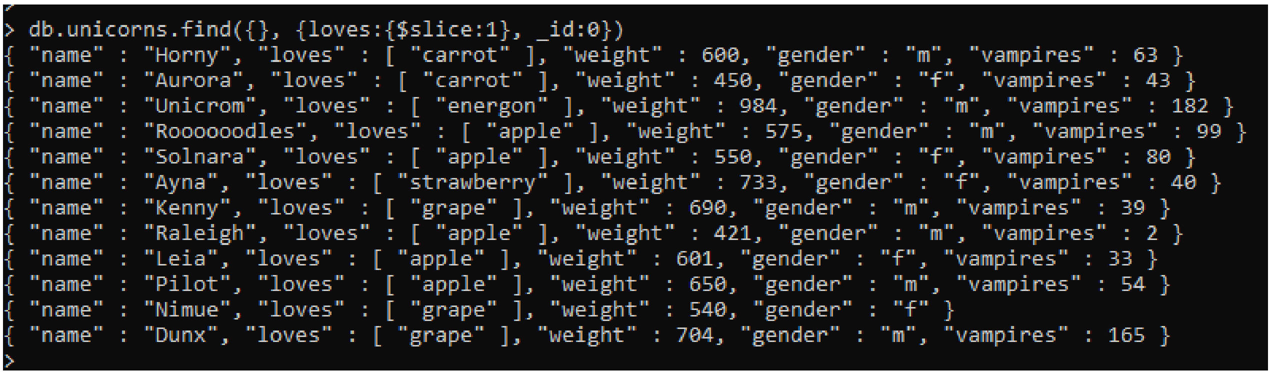
db.unicorns.find().sort({$natural: -1})



```
> db.unicorns.find().sort({$natural:-1})
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0bb"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b9"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b6"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b5"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b2"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
>
```

2.5 Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.

db.unicorns.find({ } ,{ loves: {$slice: 1}, _id: 0});

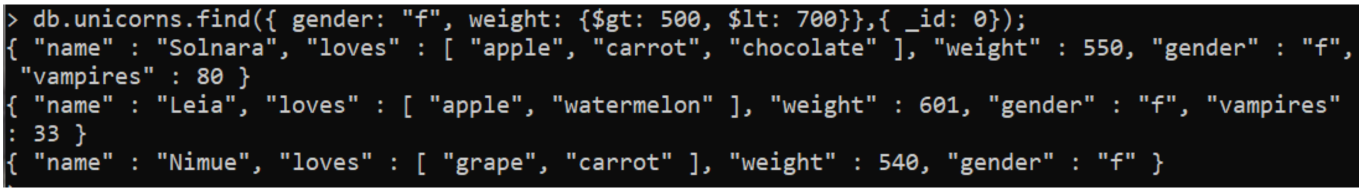


```
> db.unicorns.find({}, {loves:{$slice:1}, _id:0})
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

3. Логические операторы

3.1. Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора.

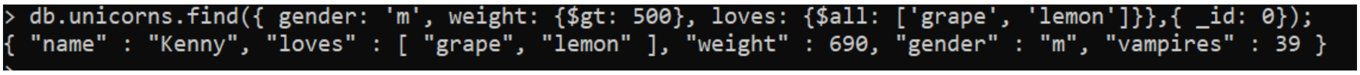
db.unicorns.find({ gender: "f", weight: {$gt: 500, $lt: 700}},{ _id: 0});



```
> db.unicorns.find({ gender: "f", weight: {$gt: 500, $lt: 700}},{ _id: 0});
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f",
 "vampires" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires"
: 33 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

3.2 Вывести список самцов единорогов весом от полутонны и предпочитающих grape ilemon, исключив вывод идентификатора.

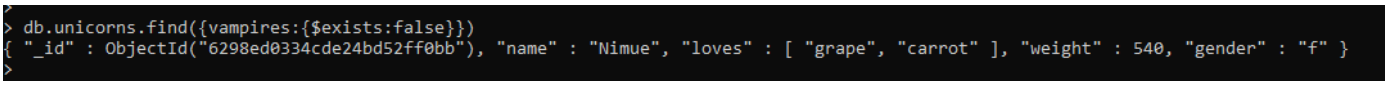
db.unicorns.find({ gender: 'm', weight: {$gt: 500}, loves: {$all: ['grape', 'lemon']}},{ _id:0})



```
> db.unicorns.find({ gender: 'm', weight: {$gt: 500}, loves: {$all: ['grape', 'lemon']}},{ _id: 0});
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

3.3 Найти всех единорогов, не имеющих ключ vampires.

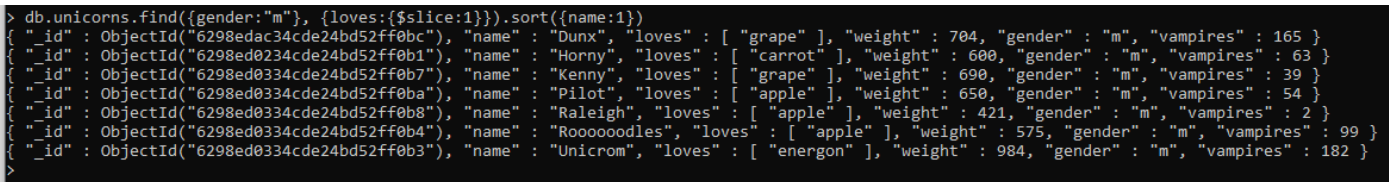
db.unicorns.find({vampires:{$exists:false}})



```
> db.unicorns.find({vampires:{$exists:false}})
{ "_id" : ObjectId("6298ed0334cde24bd52ff0bb"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
>
```

3.4 Вывести список упорядоченный список имен самцов единорогов с информацией обих первом предпочтении.

db.unicorns.find({gender: 'm'}, {loves: {$slice:1}}).sort({name:1});



```
> db.unicorns.find({gender:"m"}, {loves:{$slice:1}}).sort({name:1})
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
```

4 Запрос к вложенным объектам

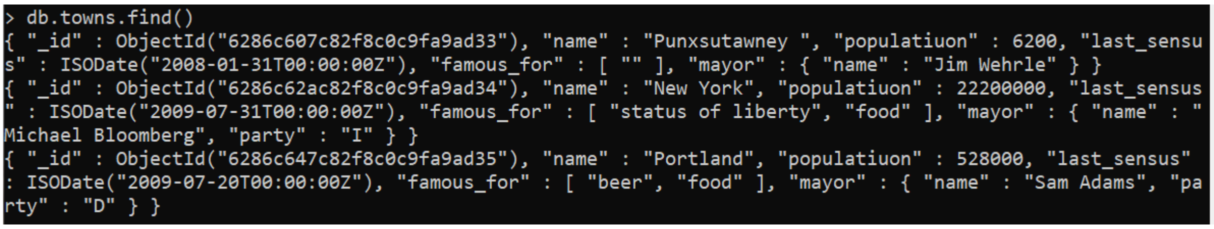
4.1 Создайте коллекцию towns, включающую следующие документы:

db.towns.insert({name: "Punxsutawney ", populatiuon: 6200,last_sensus: ISODate("2008-01-31"),famous_for: [""], mayor: { name: "Jim Wehrle"}})

db.towns.insert({name: "New York",populatiuon: 22200000, last_sensus: ISODate("2009-07-

31"), famous_for: ["status of liberty", "food"],mayor: {name: "Michael Bloomberg",party: "I"}})

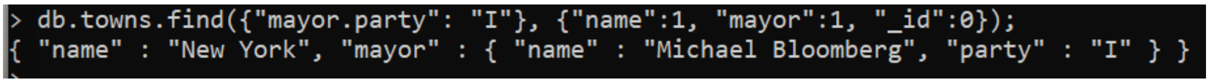
db.towns.insert({name: "Portland",populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: {name: "Sam Adams",party: "D"}})



```
> db.towns.find()
{ "_id" : ObjectId("6286c607c82f8c0c9fa9ad33"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensu
s" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6286c62ac82f8c0c9fa9ad34"), "name" : "New York", "populatiuon" : 22200000, "last_sensus
" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "
Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("6286c647c82f8c0c9fa9ad35"), "name" : "Portland", "populatiuon" : 528000, "last_sensus"
: ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "pa
rty" : "D" } }
```

4.2 Сформировать запрос, который возвращает список городов с независимымимэрами (`party="I"`). Вывести только название города и информацию о мэре.

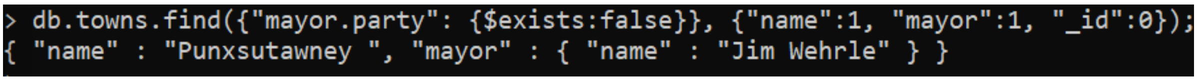
db.towns.find({"mayor.party": "I"}, {"name":1, "mayor":1, "_id":0})



```
> db.towns.find({"mayor.party": "I"}, {"name":1, "mayor":1, "_id":0});
{ "name" : "New York", "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
```

4.3 Сформировать запрос, который возвращает список беспартийных мэров (party отсутствует) Вывести только название города и информацию о мэре.

db.towns.find({"mayor.party": {$exists:false}}, {"name":1, "mayor":1, "_id":0})



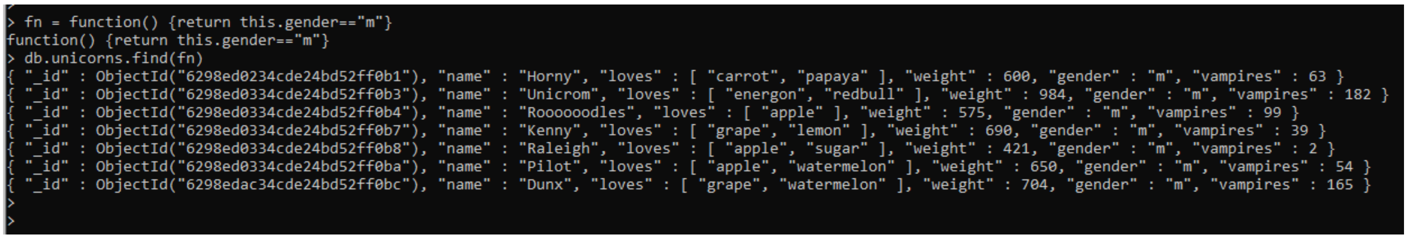
```
> db.towns.find({"mayor.party": {$exists:false}}, {"name":1, "mayor":1, "_id":0});
{ "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

5 Курсоры

5.1 Сформировать функцию для вывода списка самцов единорогов.

fn = function() {return this.gender=="m"}

db.unicorns.find(fn)



```
> fn = function() {return this.gender=="m"}
function() {return this.gender=="m"}
> db.unicorns.find(fn)
{ "_id" : ObjectId("6298ed0234cde24bd52ff0b1"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b3"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b4"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b7"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0b8"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6298ed0334cde24bd52ff0ba"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6298edac34cde24bd52ff0bc"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
>
```

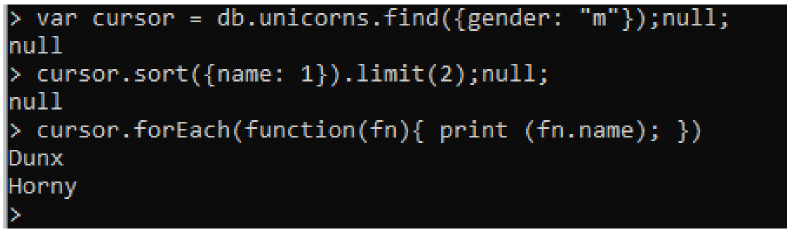
5.2 Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

5.3 Вывести результат, используя forEach.

var cursor = db.unicorns.find({gender: "m"});null;

cursor.sort({name: 1}).limit(2);null;

cursor.forEach(function(fn){ print (fn.name); })

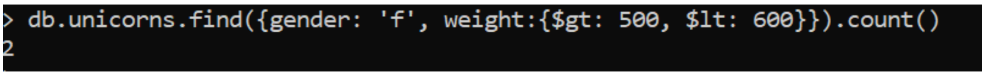


```
> var cursor = db.unicorns.find({gender: "m"});null;
null
> cursor.sort({name: 1}).limit(2);null;
null
> cursor.forEach(function(fn){ print (fn.name); })
Dunx
Horny
>
```

6 Агрегированные запросы

6.1 Вывести количество самок единорогов весом от полутонны до 600 кг.

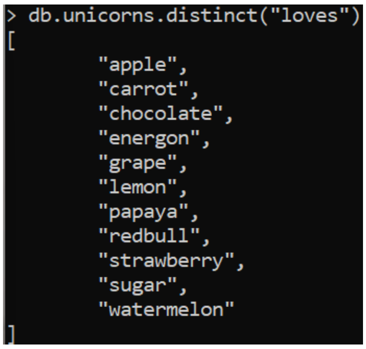
db.unicorns.find({gender: 'f', weight:{$gt: 500, $lt: 600}}).count()



```
> db.unicorns.find({gender: 'f', weight:{$gt: 500, $lt: 600}}).count()
2
```

6.2 Вывести список предпочтений.

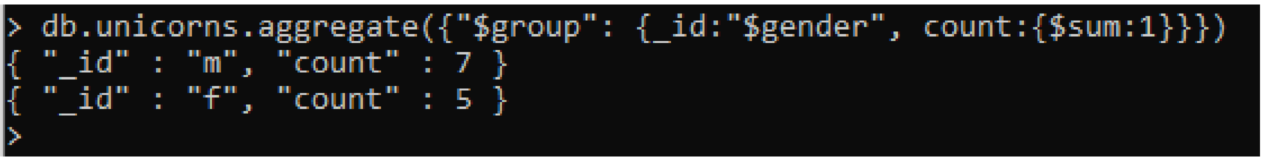
db.unicorns.distinct("loves")



```
> db.unicorns.distinct("loves")
[
        "apple",
        "carrot",
        "chocolate",
        "energon",
        "grape",
        "lemon",
        "papaya",
        "redbull",
        "strawberry",
        "sugar",
        "watermelon"
]
```

6.3 Посчитать количество особей единорогов обоих полов.

db.unicorns.aggregate({"$group": {_id:"$gender", count:{$sum:1}}})

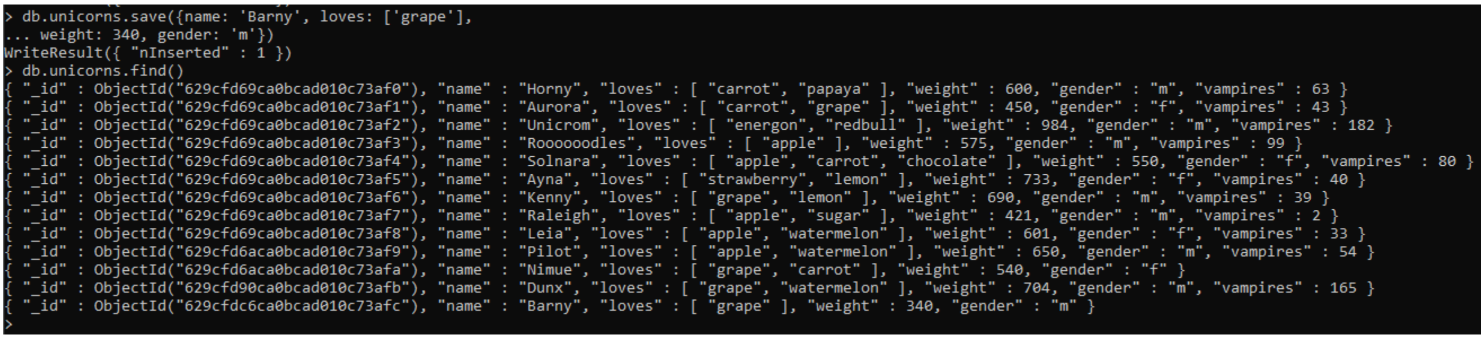


```
> db.unicorns.aggregate({"$group": {_id:"$gender", count:{$sum:1}}})
{ "_id" : "m", "count" : 7 }
{ "_id" : "f", "count" : 5 }
>
```

7 Редактирование данных

7.1 Выполнить команду:

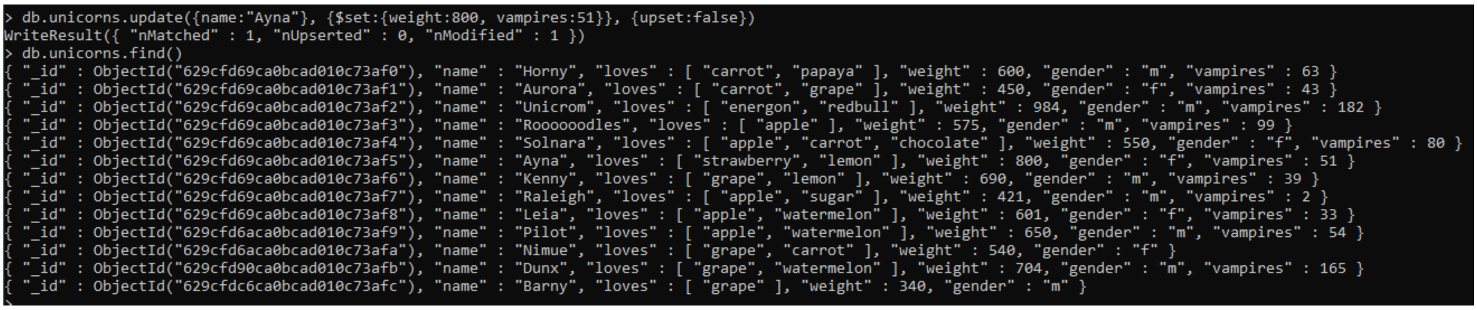
db.unicorns.save({name: 'Barny', loves: ['grape'],weight: 340, gender: 'm'})



```
> db.unicorns.save({name: 'Barny', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

7.2 Для самки единорога `Ayna` внести изменения в БД: теперь ее вес 800, она убила 51 вампира.

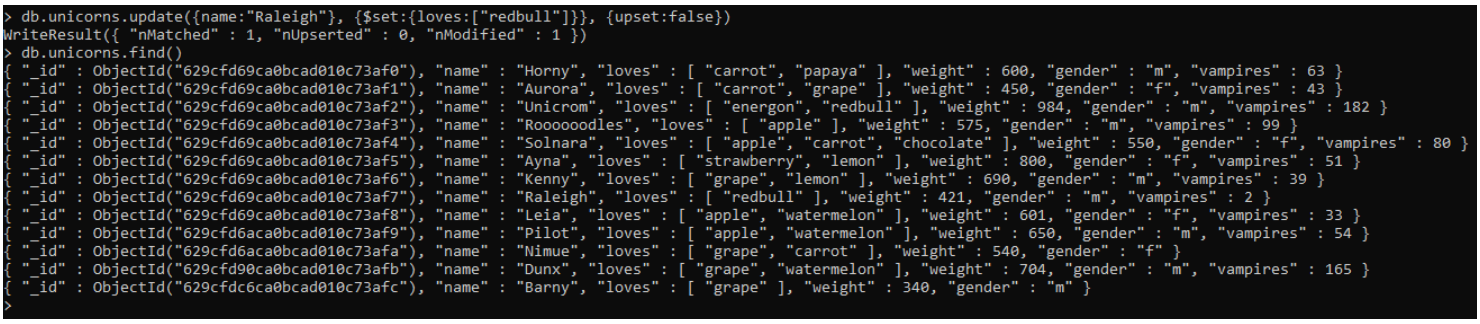
db.unicorns.update({name: “Ayna”}, {$set:{weight: 800, vampires: 51}}, {upset: false})



```
> db.unicorns.update({name:"Ayna"}, {$set:{weight:800, vampires:51}}, {upset:false})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

7.3 Для самца единорога `Raleigh` внести изменения в БД: теперь он любит рэдбул.

db.unicorns.update({name: "Raleigh"}, {$set: {loves: ["redbull"]}}, {upset: false})

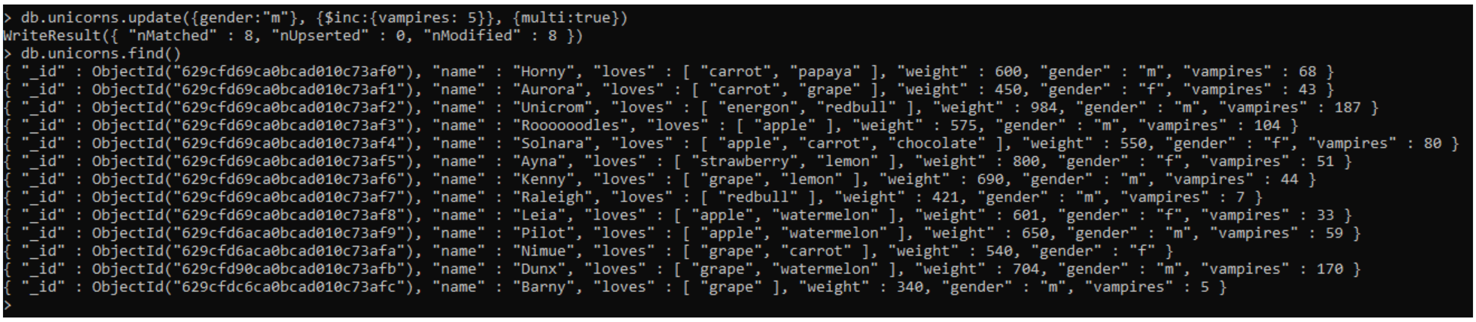


```
> db.unicorns.update({name:"Raleigh"}, {$set:{loves:["redbull"]}}, {upset:false})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

.

7.4 Всем самцам единорогов увеличить количество убитых вампиров на 5

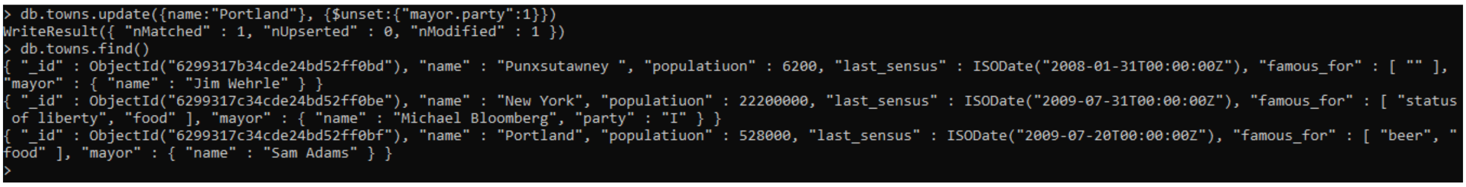
db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi:true})



```
> db.unicorns.update({gender:"m"}, {$inc:{vampires: 5}}, {multi:true})
WriteResult({ "nMatched" : 8, "nUpserted" : 0, "nModified" : 8 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>
```

7.5 Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

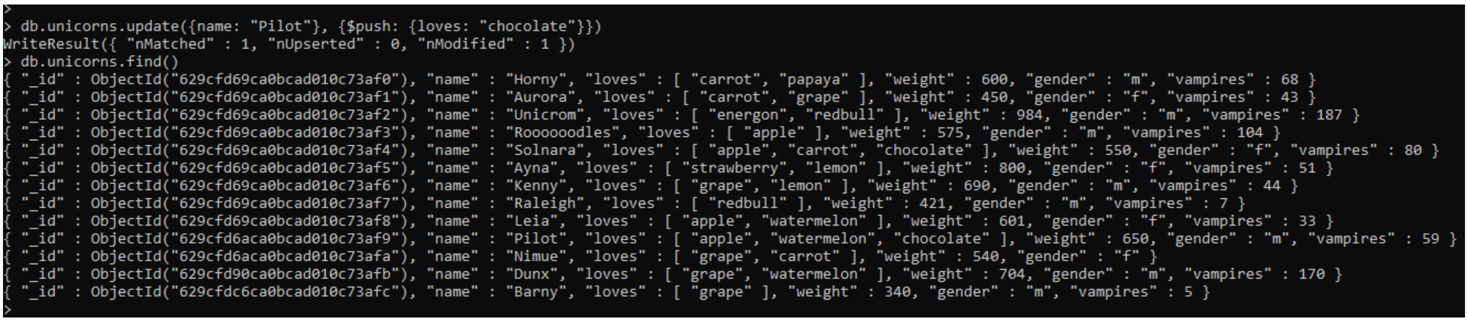
db.towns.update({name: "Portland"}, {$unset: {"mayor.party": 1}})



```
> db.towns.update({name:"Portland"}, {$unset:{"mayor.party":1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("6299317b34cde24bd52ff0bd"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ],
"mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6299317c34cde24bd52ff0be"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status
 of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("6299317c34cde24bd52ff0bf"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "
food" ], "mayor" : { "name" : "Sam Adams" } }
>
```

7.6 Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.

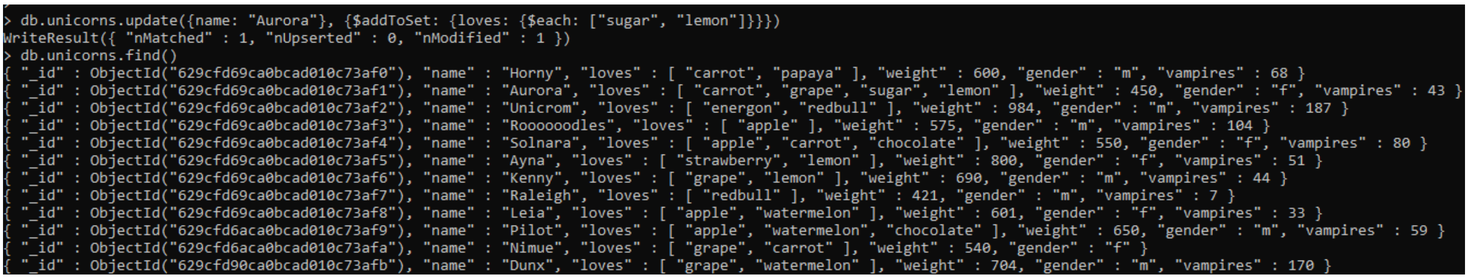
db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})



```
>
> db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>
```

7.7 Изменить информацию о самке единорога `Aurora`: теперь она любит еще и сахар,и лимоны.

db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar","lemon"]}}})



```
> db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
```

8 Удаление данных из коллекции

8.1 Создайте коллекцию towns, включающую следующие документы:

db.towns.insert({name: "Punxsutawney ", popujatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: ["phil the groundhog"], mayor: { name: "Jim Wehrle" }})
db.towns.insert({name: "New York", popujatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: { name: "Michael Bloomberg", party: "I"}})
db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sam Adams", party: "D"}})

8.2 Удалите документы с беспартийными мэрами.

db.towns.remove({"mayor.party": {$exists: false}})

8.3 Проверьте содержание коллекции.

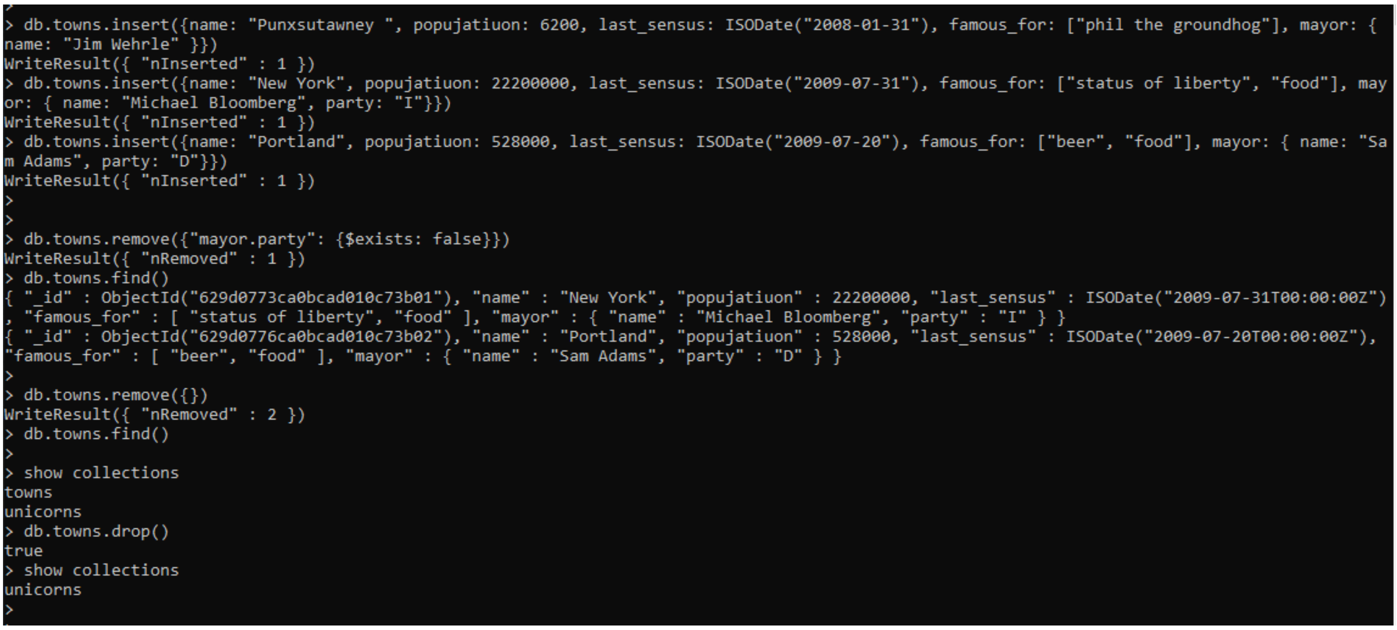
db.towns.find({"mayor.party": {$exists: false}})

8.4 Очистите коллекцию.

db.towns.remove({})

8.5 Просмотрите список доступных коллекций.

show collections

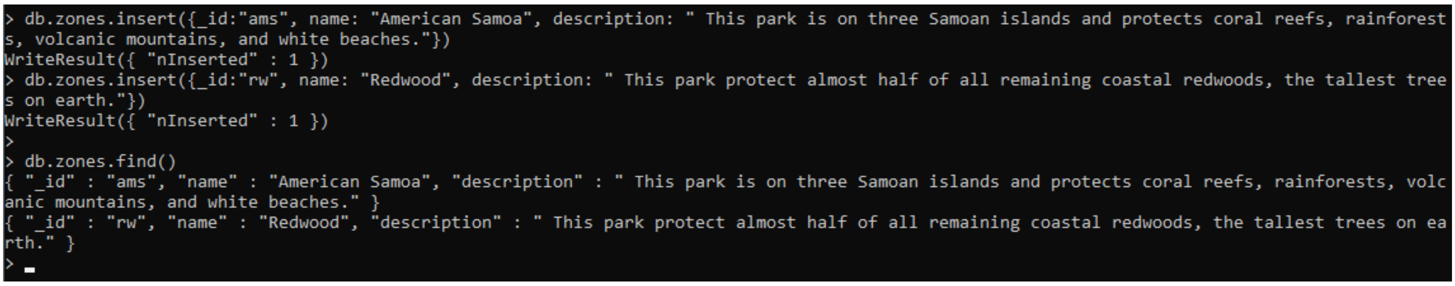


```
> db.towns.insert({name: "Punxsutawney ", popujatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: ["phil the groundhog"], mayor: {
name: "Jim Wehrle" }})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "New York", popujatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], may
or: { name: "Michael Bloomberg", party: "I"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sa
m Adams", party: "D"}})
WriteResult({ "nInserted" : 1 })
>
>
> db.towns.remove({"mayor.party": {$exists: false}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("629d0773ca0bcad010c73b01"), "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z")
, "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("629d0776ca0bcad010c73b02"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"),
"famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> db.towns.find()
>
> show collections
towns
unicorns
> db.towns.drop()
true
> show collections
unicorns
>
```

9 Ссылки в БД

9.1 Создайте коллекцию зон обитания единорогов, указав в качестве идентификаторакратко название зоны, далее включив полное название и описание.

db.zones.insert({_id:"ams", name: "American Samoa", description: " This park is on three Samoan islands and protects coral reefs, rainforests, volcanic mountains, and white beaches."})
db.zones.insert({_id:"rw", name: "Redwood", description: " This park protect almost half of all remaining coastal redwoods, the tallest trees on earth."})



```
> db.zones.insert({_id:"ams", name: "American Samoa", description: " This park is on three Samoan islands and protects coral reefs, rainforest
s, volcanic mountains, and white beaches."})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id:"rw", name: "Redwood", description: " This park protect almost half of all remaining coastal redwoods, the tallest tree
s on earth."})
WriteResult({ "nInserted" : 1 })
>
> db.zones.find()
{ "_id" : "ams", "name" : "American Samoa", "description" : " This park is on three Samoan islands and protects coral reefs, rainforests, volc
anic mountains, and white beaches." }
{ "_id" : "rw", "name" : "Redwood", "description" : " This park protect almost half of all remaining coastal redwoods, the tallest trees on ea
rth." }
> _
```

9.2 Включите для нескольких единорогов в документы ссылку на зону обитания,используя второй способ автоматического связывания. Проверьте содержание коллекции единорогов.

db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})

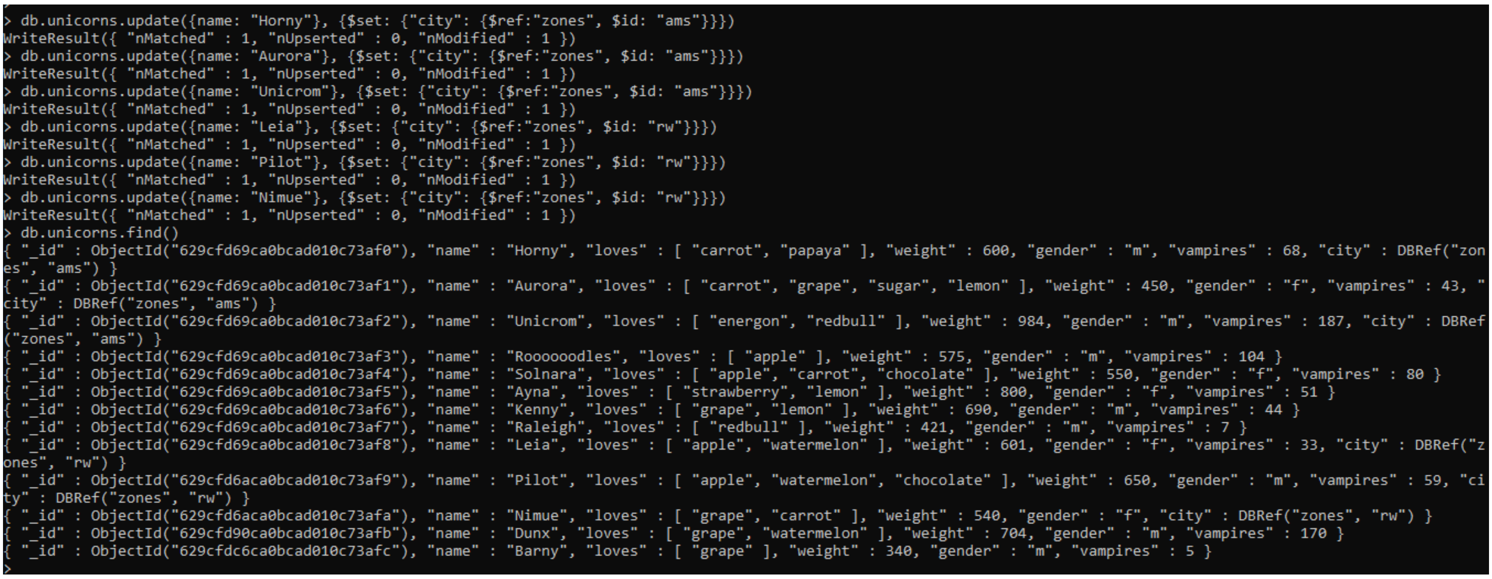
db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})

db.unicorns.update({name: "Unicrom"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})

db.unicorns.update({name: "Leia"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})

db.unicorns.update({name: "Pilot"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})

db.unicorns.update({name: "Nimue"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})

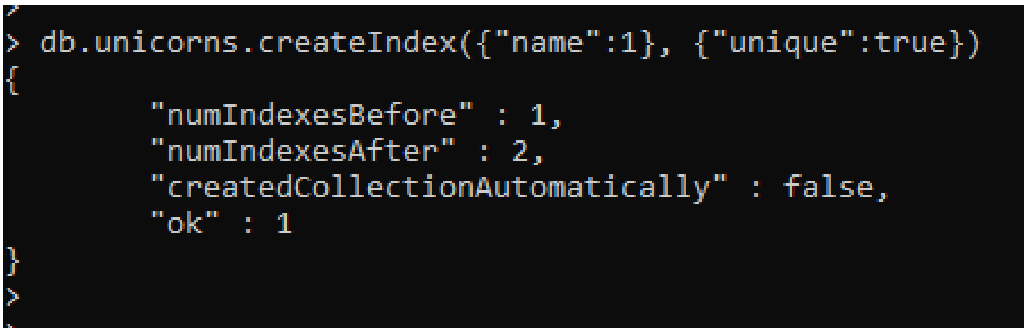


```
> db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Unicrom"}, {$set: {"city": {$ref:"zones", $id: "ams"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Leia"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Pilot"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Nimue"}, {$set: {"city": {$ref:"zones", $id: "rw"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("629cfd69ca0bcad010c73af0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "city" : DBRef("zon
es", "ams") }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af1"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "
city" : DBRef("zones", "ams") }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187, "city" : DBRef
("zones", "ams") }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("629cfd69ca0bcad010c73af8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33, "city" : DBRef("z
ones", "rw") }
{ "_id" : ObjectId("629cfd6aca0bcad010c73af9"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59, "ci
ty" : DBRef("zones", "rw") }
{ "_id" : ObjectId("629cfd6aca0bcad010c73afa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f", "city" : DBRef("zones", "rw") }
{ "_id" : ObjectId("629cfd90ca0bcad010c73afb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("629cfdc6ca0bcad010c73afc"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>
```

10 Настройка индексов

10.1 Проверьте, можно ли задать для коллекции `unicorns` индекс для ключа `name` с флагом `unique`.

db.unicorns.createIndex({"name":1}, {"unique":true})

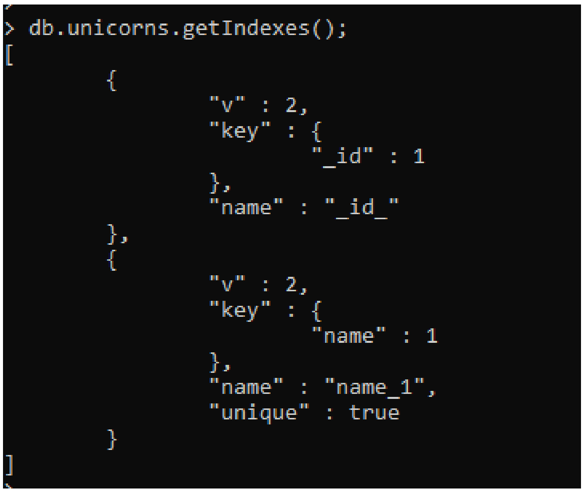


```
> db.unicorns.createIndex({"name":1}, {"unique":true})
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
>
```

11 Управление индексами

11.2 Получите информацию о всех индексах коллекции `unicorns` .

db.unicorns.getIndexes()



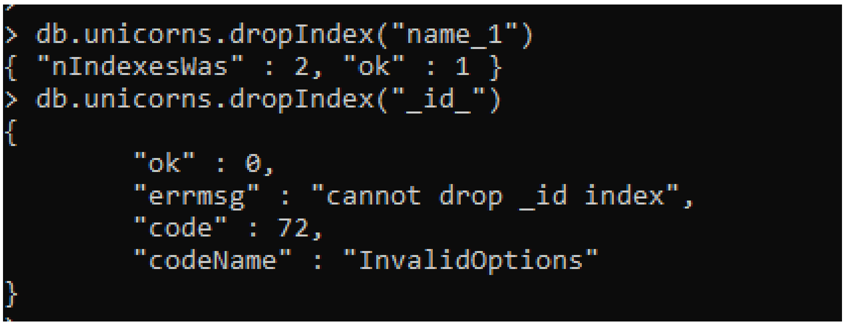
```
> db.unicorns.getIndexes();
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "name" : 1
                },
                "name" : "name_1",
                "unique" : true
        }
]
```

11.3 Удалите все индексы, кроме индекса для идентификатора

db.unicorns.dropIndex(“name_1”)

11.4 Попытайтесь удалить индекс для идентификатора.

db.unicorns.dropIndex({"_id_"})

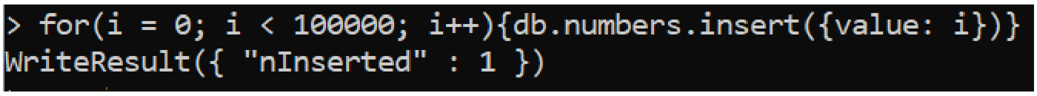


```
> db.unicorns.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.dropIndex("_id_")
{
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
}
```

12 План запроса

12.2 Создайте объемную коллекцию `numbers`, задействовав курсор:

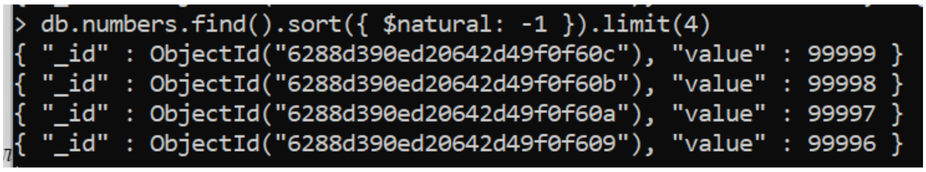
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}



```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
```

12.3 Выберите последних четыре документа.

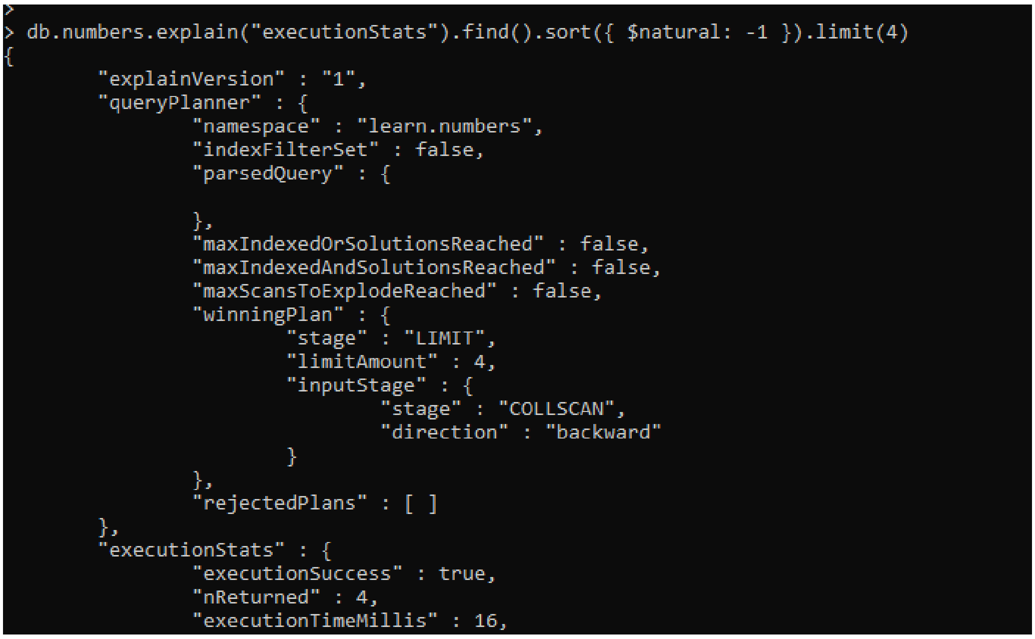
db.numbers.find().sort({ $natural: -1 }).limit(4)



```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("6288d390ed20642d49f0f60c"), "value" : 99999 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60b"), "value" : 99998 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60a"), "value" : 99997 }
{ "_id" : ObjectId("6288d390ed20642d49f0f609"), "value" : 99996 }
```

12.4 Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)

db.numbers.explain("executionStats").find().sort({ $natural: -1 }).limit(4)

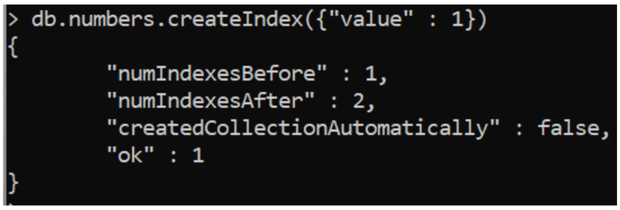


```
>
> db.numbers.explain("executionStats").find().sort({ $natural: -1 }).limit(4)
{
        "explainVersion" : "1",
        "queryPlanner" : {
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "maxIndexedOrSolutionsReached" : false,
                "maxIndexedAndSolutionsReached" : false,
                "maxScansToExplodeReached" : false,
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 16,
```

12.5 Создайте индекс для ключа value.

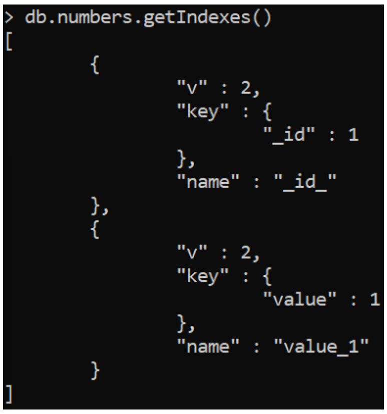
db.numbers.createIndex({"value" : 1})



```
> db.numbers.createIndex({"value" : 1})
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
```

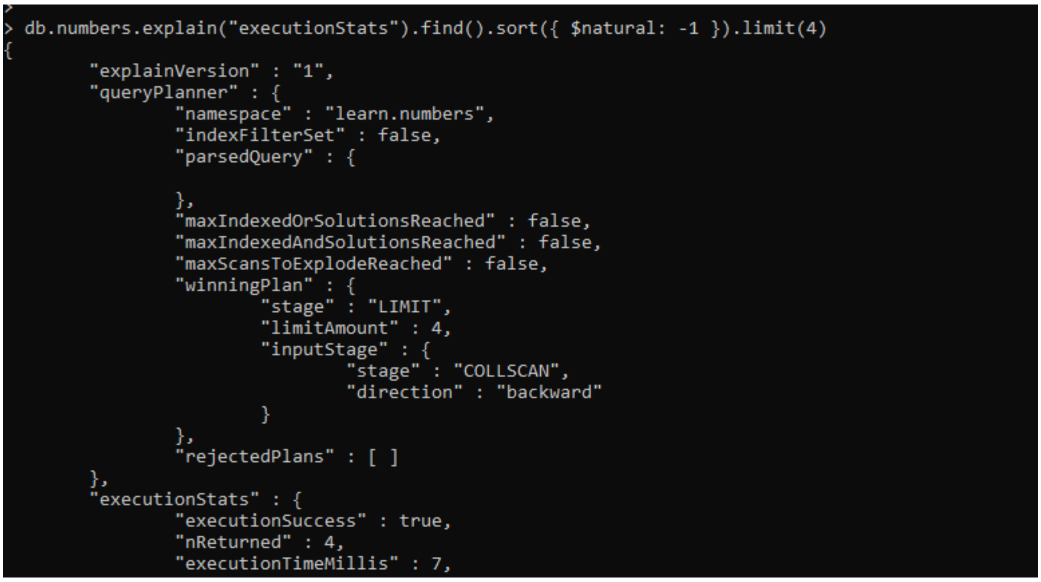
12.6 Получите информацию о всех индексах коллекции numbres.

db.numbers.getIndexes()



```
> db.numbers.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "value" : 1
                },
                "name" : "value_1"
        }
]
```

12.7 Выполните запрос 2. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?



```
>
> db.numbers.explain("executionStats").find().sort({ $natural: -1 }).limit(4)
{
        "explainVersion" : "1",
        "queryPlanner" : {
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "maxIndexedOrSolutionsReached" : false,
                "maxIndexedAndSolutionsReached" : false,
                "maxScansToExplodeReached" : false,
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 7,
```

12.8 Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

Без индекса время выполнения = 16 ms. С индексом = 7 ms. Следовательно запросы с индексами работают быстрее и эффективнее.

Выводы:

В ходе выполнения данной работы я овладела практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.