Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №5 «Реализация БД с использованием СУБД MongoDB. Запросы к базе данных» по дисциплине: «Проектирование и реализация баз данных»

Выполнила:
студентка II курса ИКТ
группы К3242
Скокова Алина Викторовна

Проверила:
Говорова Марина Михайловна

Санкт-Петербург
2022

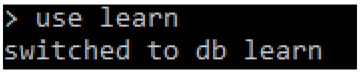
Цель лабораторной работы: овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

Выполнение.

Практическое задание 8.1.1:

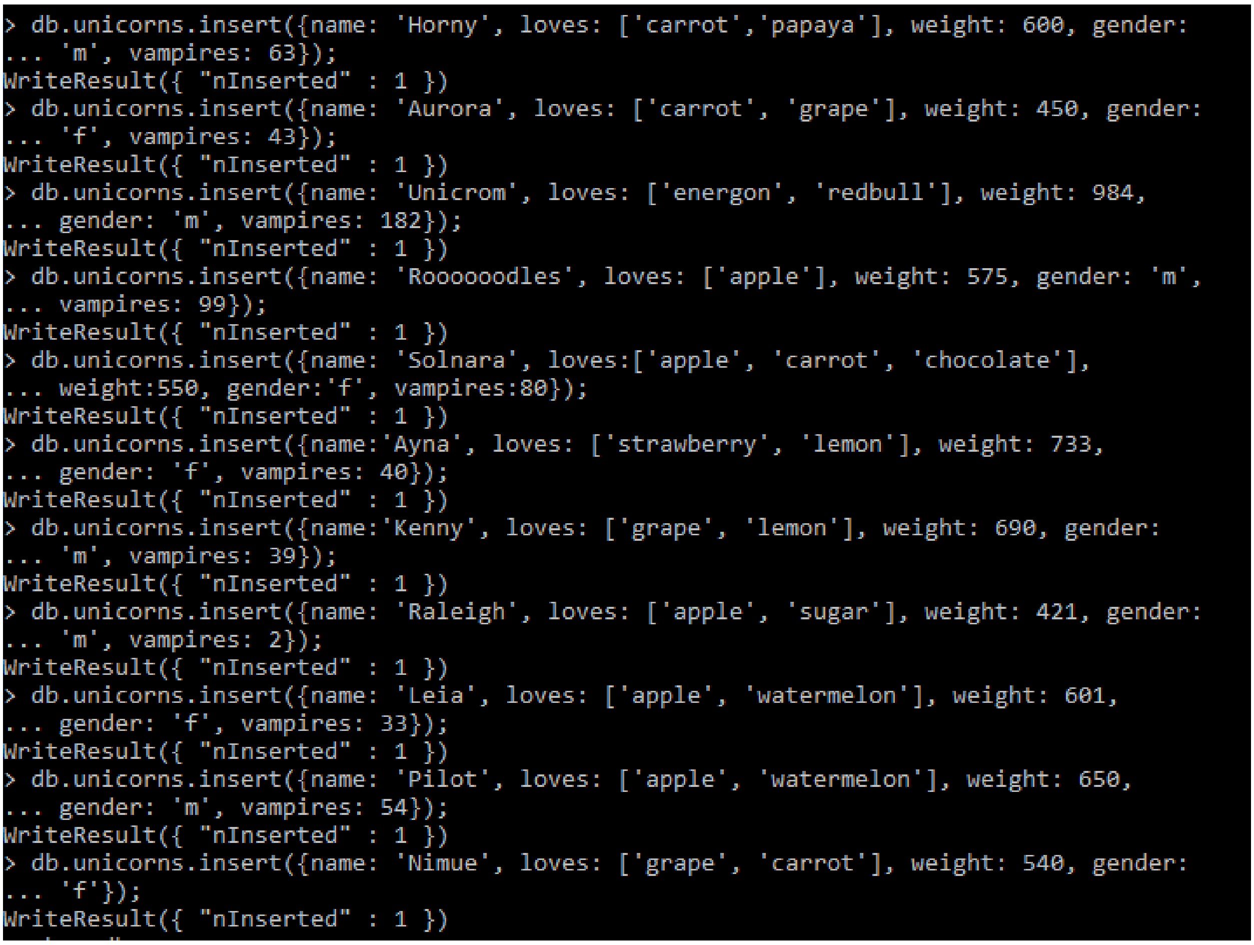
1. Создайте базу данных learn.

use learn;



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

2. Заполните коллекцию единорогов unicorns.

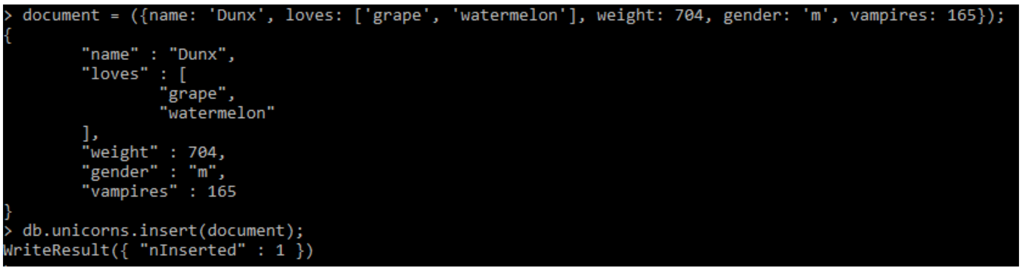


```
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender:
... 'm', vampires: 63});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender:
... 'f', vampires: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: '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 })
```

3. Используя второй способ, вставьте в коллекцию единорогов документ.

document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});

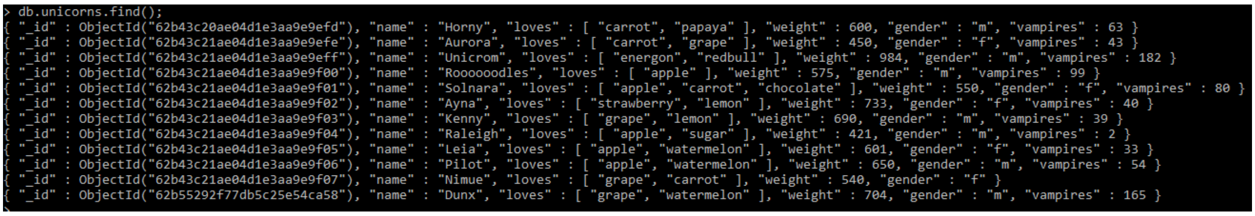
db.unicorns.insert(document);



```
> document = ({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(document);
WriteResult({ "nInserted" : 1 })
```

4. Проверьте содержимое коллекции с помощью метода find.

db.unicorns.find();



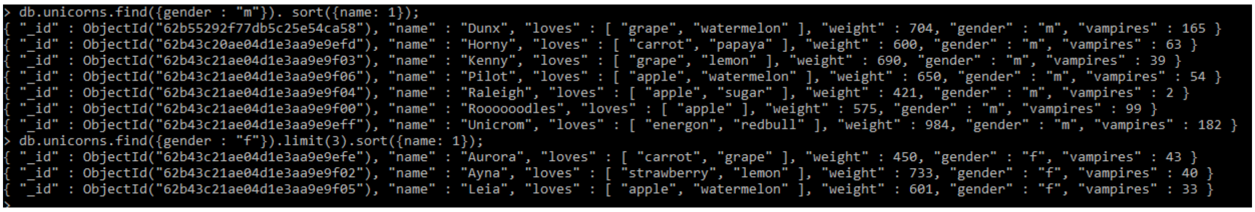
```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

Практическое задание 8.1.2:

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"}).limit(3).sort({name: 1});

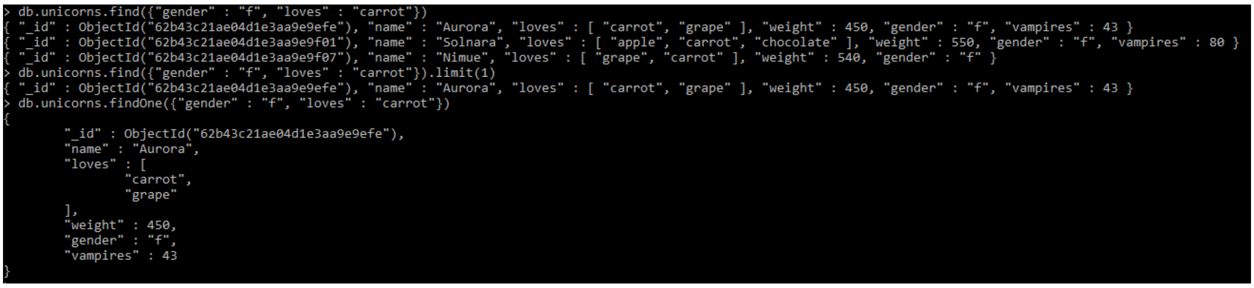


```
> db.unicorns.find({gender : "m"}). sort({name: 1});
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
> db.unicorns.find({gender : "f"}).limit(3).sort({name: 1});
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>
```

2. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

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

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

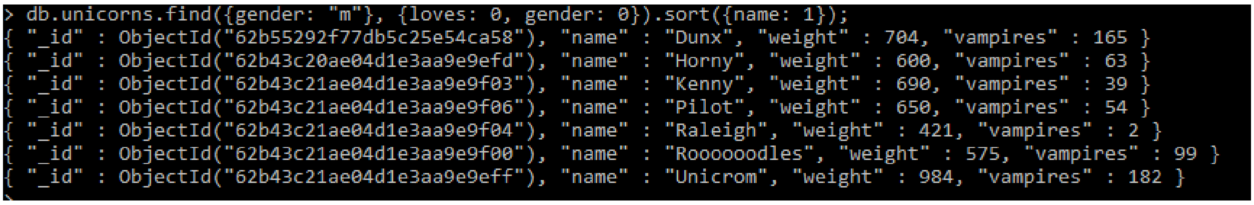


```
> db.unicorns.find({"gender" : "f", "loves" : "carrot"})
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
> db.unicorns.find({"gender" : "f", "loves" : "carrot"}).limit(1)
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.findOne({"gender" : "f", "loves" : "carrot"})
{
        "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
```

Практическое задание 8.1.3:

Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.

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

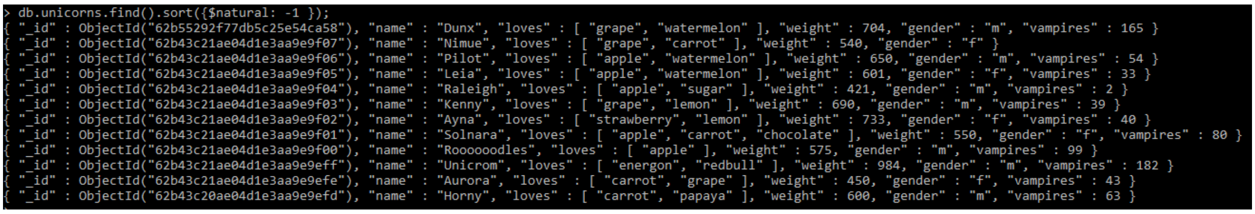


```
> db.unicorns.find({gender: "m"}, {loves: 0, gender: 0}).sort({name: 1});
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

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

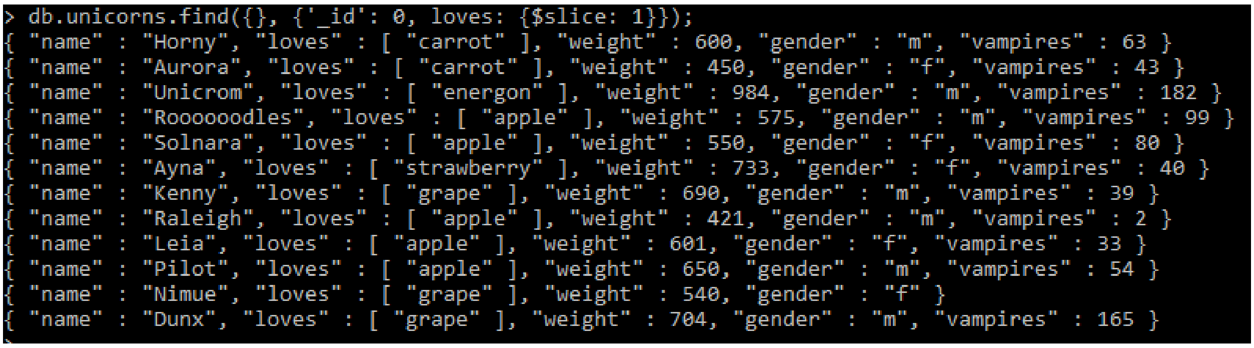


```
> db.unicorns.find().sort({$natural: -1 });
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

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

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

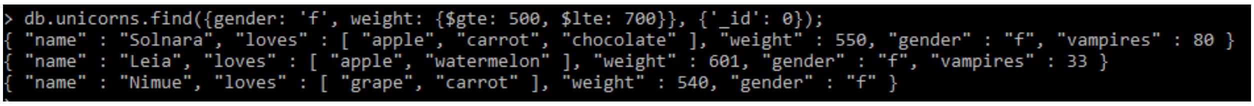


```
> db.unicorns.find({}, {'_id': 0, loves: {$slice: 1}});
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 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 }
```

Практическое задание 8.1.6:

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

db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {'_id': 0});

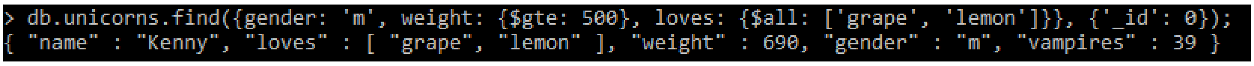


```
> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {'_id': 0});
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" :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

Практическое задание 8.1.7:

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

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

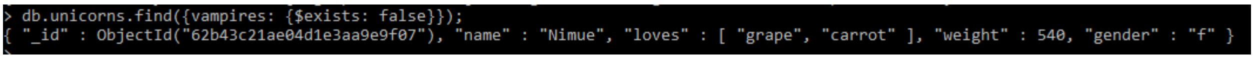


```
> db.unicorns.find({gender: 'm', weight: {$gte: 500}, loves: {$all: ['grape', 'lemon']}}, {'_id': 0});
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

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

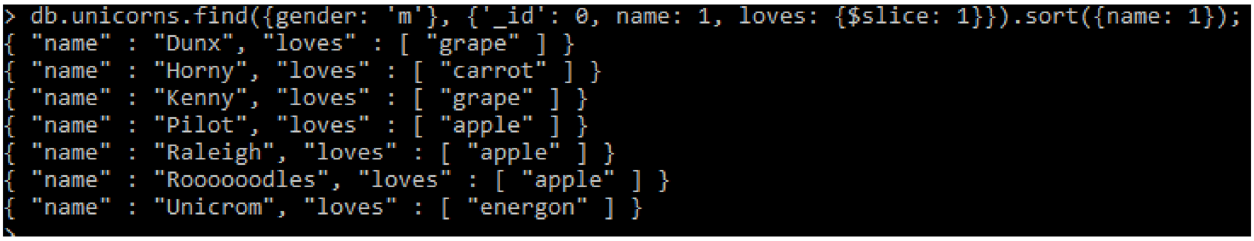


```
> db.unicorns.find({vampires: {$exists: false}});
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Практическое задание 8.1.9:

Вывести упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.

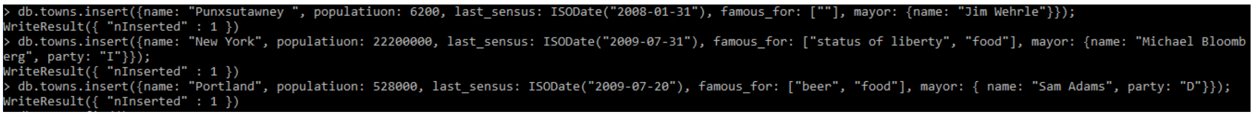
db.unicorns.find({gender: 'm'}, {'_id': 0, name: 1, loves: {$slice: 1}}).sort({name: 1});



```
> db.unicorns.find({gender: 'm'}, {'_id': 0, name: 1, loves: {$slice: 1}}).sort({name: 1});
{ "name" : "Dunx", "loves" : [ "grape" ] }
{ "name" : "Horny", "loves" : [ "carrot" ] }
{ "name" : "Kenny", "loves" : [ "grape" ] }
{ "name" : "Pilot", "loves" : [ "apple" ] }
{ "name" : "Raleigh", "loves" : [ "apple" ] }
{ "name" : "Rooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
```

Практическое задание 8.2.1:

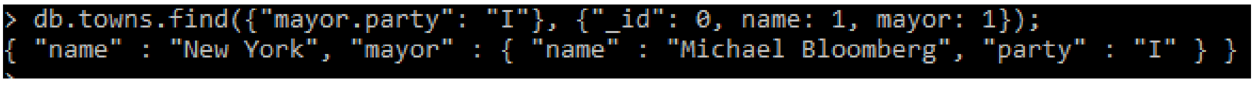
1. Создайте коллекцию towns, включающую следующие документы.



```
> db.towns.insert({name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""], mayor: {name: "Jim Wehrle"}});
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: {name: "Michael Bloomb
erg", party: "I"}});
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sam Adams", party: "D"}});
WriteResult({ "nInserted" : 1 })
```

2. Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I"). Вывести только название города и информацию о мэре.

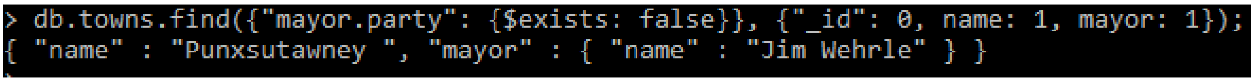
db.towns.find({"mayor.party": "I"}, {"_id": 0, name: 1, mayor: 1});



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

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}}, {"_id": 0, name: 1, mayor: 1});



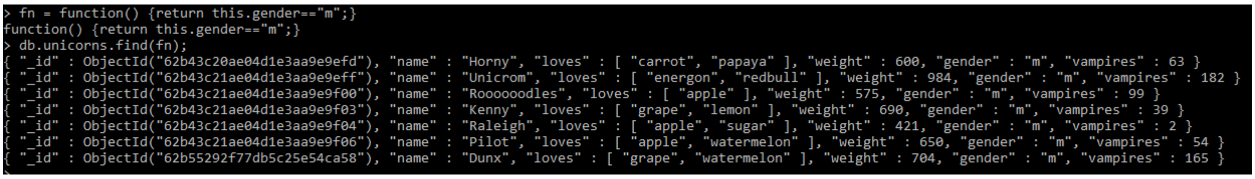
```
> db.towns.find({"mayor.party": {$exists: false}}, {"_id": 0, name: 1, mayor: 1});
{ "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

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

db.unicorns.find(fn);

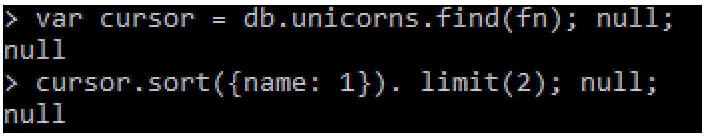


```
> fn = function() {return this.gender=="m";}
function() {return this.gender=="m";}
> db.unicorns.find(fn);
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

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

var cursor = db.unicorns.find(fn); null;

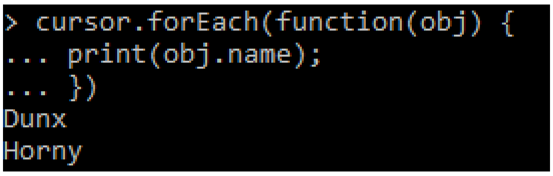
cursor.sort({name: 1}). limit(2); null;



```
> var cursor = db.unicorns.find(fn); null;
null
> cursor.sort({name: 1}). limit(2); null;
null
```

3. Вывести результат, используя forEach.

```
cursor.forEach(function(obj) {
        print(obj.name);
        })
```

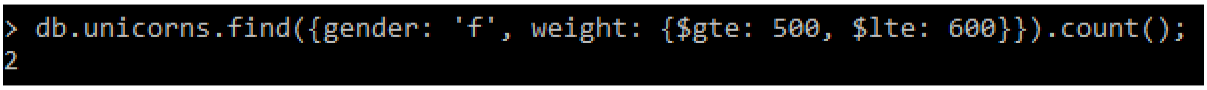


```
> cursor.forEach(function(obj) {
... print(obj.name);
... })
Dunx
Horny
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count();

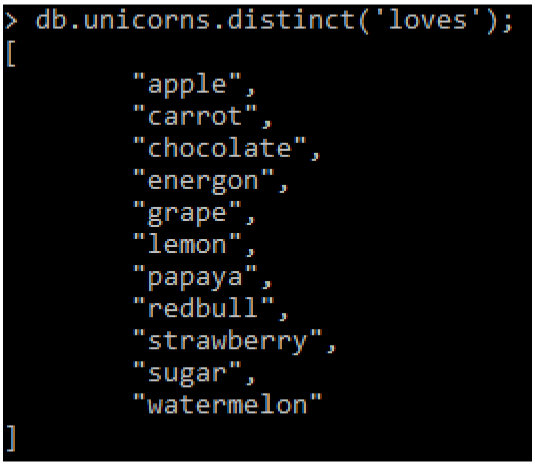


```
> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count();
2
```

Практическое задание 8.2.4:

Вывести список предпочтений.

db.unicorns.distinct('loves');

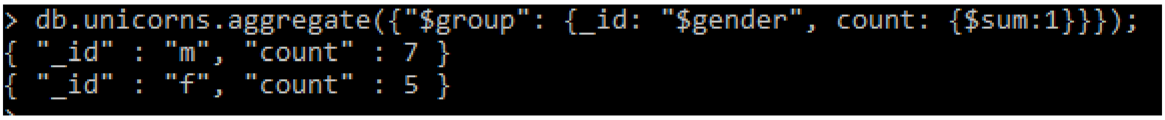


```
> db.unicorns.distinct('loves');
[
        "apple",
        "carrot",
        "chocolate",
        "energon",
        "grape",
        "lemon",
        "papaya",
        "redbull",
        "strawberry",
        "sugar",
        "watermelon"
]
```

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

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

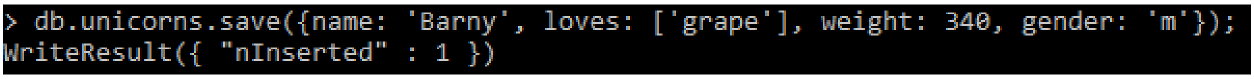


```
> db.unicorns.aggregate({"$group": {_id: "$gender", count: {$sum:1}}});
{ "_id" : "m", "count" : 7 }
{ "_id" : "f", "count" : 5 }
```

Практическое задание 8.2.6:

1. Выполнить команду:

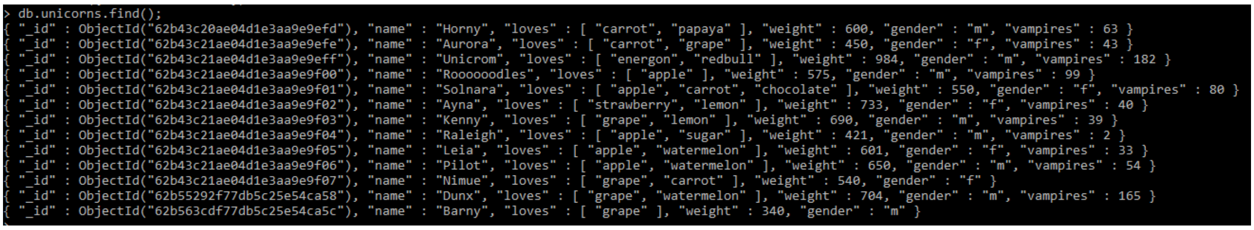
db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'});



```
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'});
WriteResult({ "nInserted" : 1 })
```

2. Проверить содержимое коллекции unicorns.

db.unicorns.find();

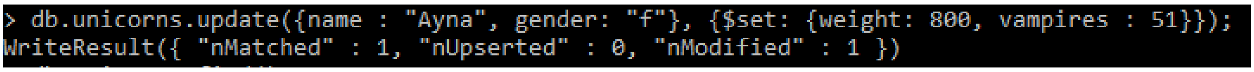


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.7:

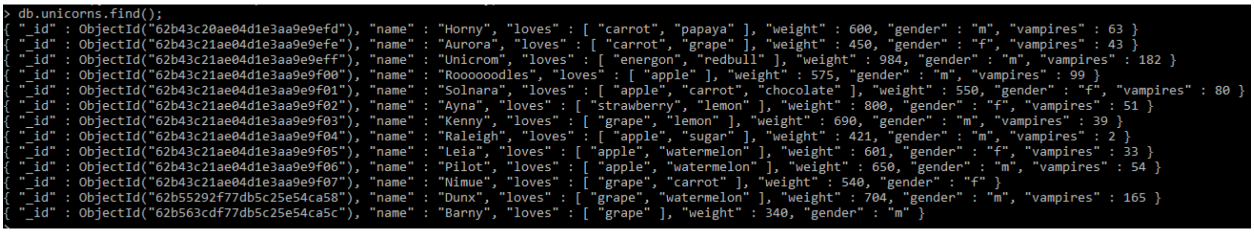
1. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вампира.

db.unicorns.update({name : "Ayna", gender: "f"}, {$set: {weight: 800, vampires : 51}});



```
> db.unicorns.update({name : "Ayna", gender: "f"}, {$set: {weight: 800, vampires : 51}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. Проверить содержимое коллекции unicorns.

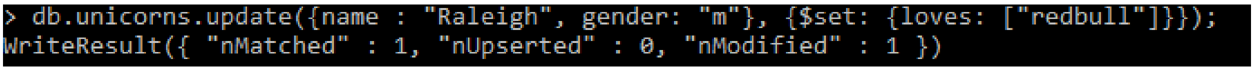


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.8:

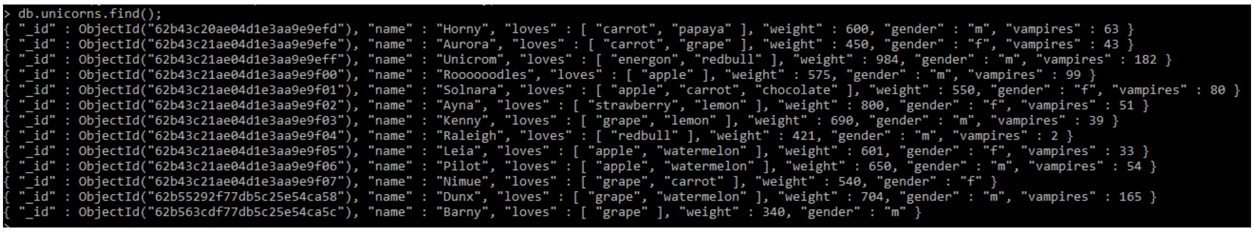
1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.

db.unicorns.update({name : "Raleigh", gender: "m"}, {$set: {loves: ["redbull"]}});



```
> db.unicorns.update({name : "Raleigh", gender: "m"}, {$set: {loves: ["redbull"]}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. Проверить содержимое коллекции unicorns.

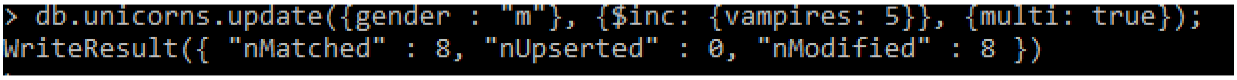


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

Практическое задание 8.2.9:

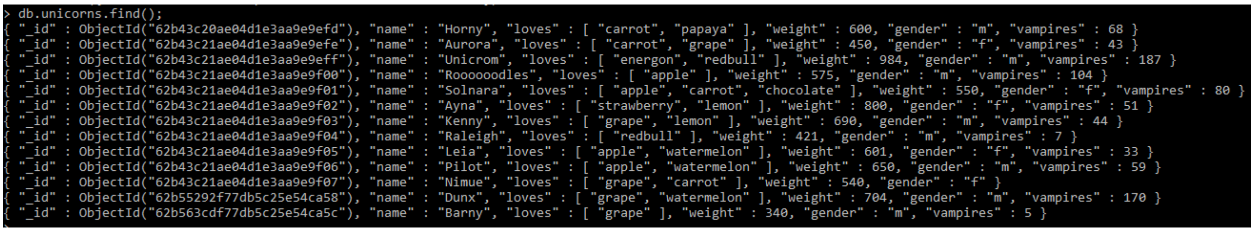
1. Всем самцам единорогов увеличить количество убитых вапмиров на 5.

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 })
```

2. Проверить содержимое коллекции unicorns.

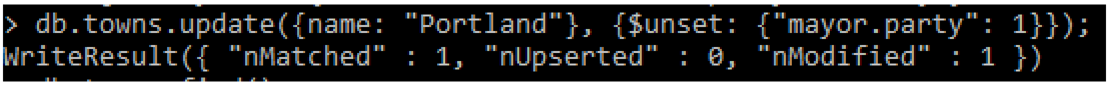


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

Практическое задание 8.2.10:

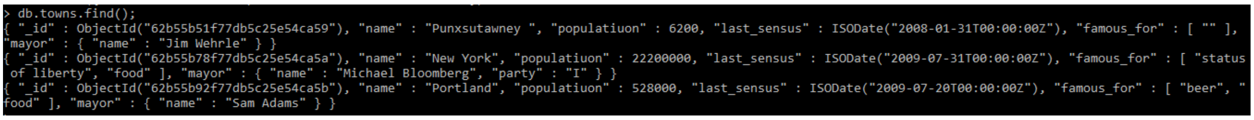
1. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

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 })
```

2. Проверить содержимое коллекции towns.

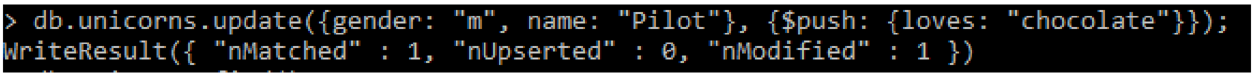


```
> db.towns.find();
{ "_id" : ObjectId("62b55b51f77db5c25e54ca59"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ],
"mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("62b55b78f77db5c25e54ca5a"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status
 of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62b55b92f77db5c25e54ca5b"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "
food" ], "mayor" : { "name" : "Sam Adams" } }
```

Практическое задание 8.2.11:

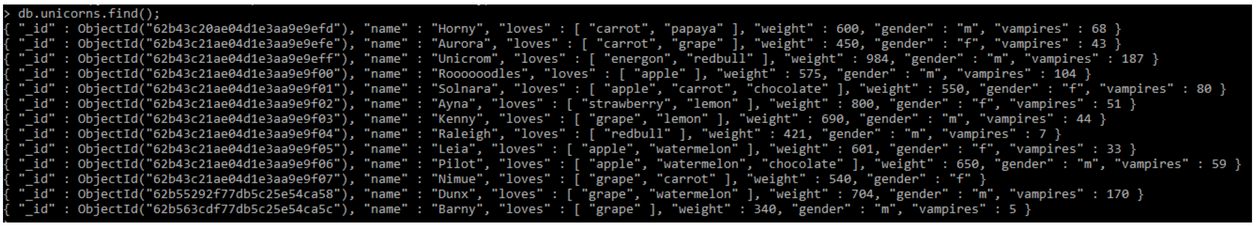
1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.

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



```
> db.unicorns.update({gender: "m", name: "Pilot"}, {$push: {loves: "chocolate"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

2. Проверить содержимое коллекции unicorns.

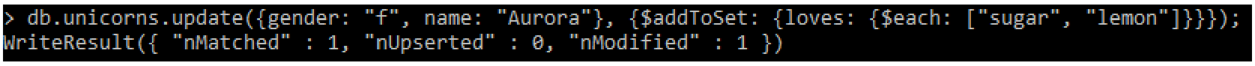


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

Практическое задание 8.2.12:

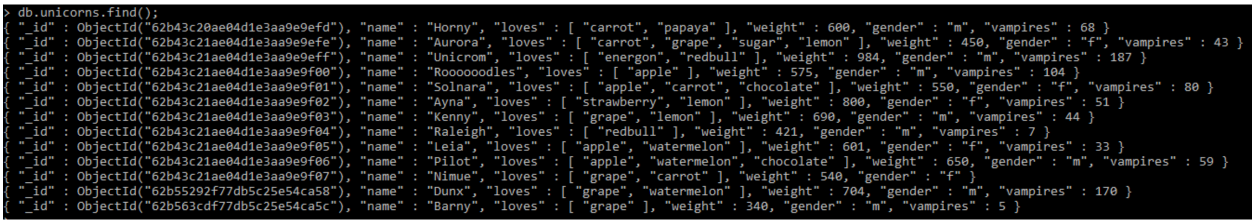
1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

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



```
> db.unicorns.update({gender: "f", name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

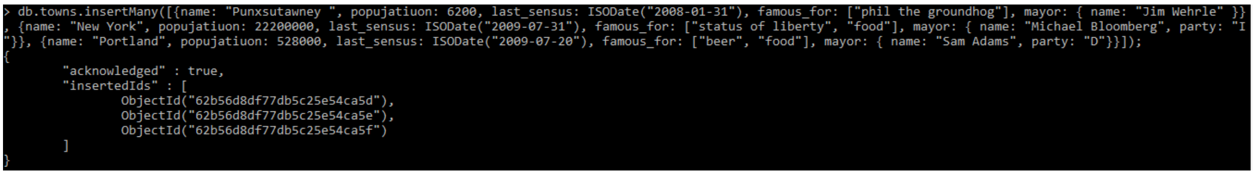
2. Проверить содержимое коллекции unicorns.



```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Rooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62b563cdf77db5c25e54ca5c"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

Практическое задание 8.2.13:

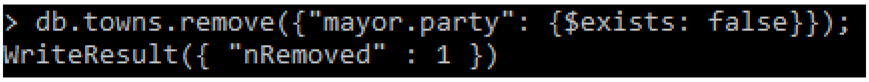
1. Создайте коллекцию towns, включающую следующие документы.



```
> db.towns.insertMany([{name: "Punxsutawney ", popujatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: ["phil the groundhog"], mayor: { name: "Jim Wehrle" }}
, {name: "New York", popujatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: { name: "Michael Bloomberg", party: "I
"}}, {name: "Portland", popujatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: { name: "Sam Adams", party: "D"}}]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62b56d8df77db5c25e54ca5d"),
                ObjectId("62b56d8df77db5c25e54ca5e"),
                ObjectId("62b56d8df77db5c25e54ca5f")
        ]
}
```

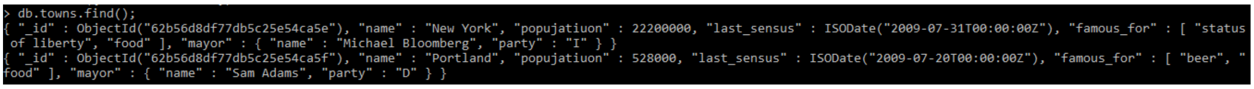
2. Удалите документы с беспартийными мэрами.

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



```
> db.towns.remove({"mayor.party": {$exists: false}});
WriteResult({ "nRemoved" : 1 })
```

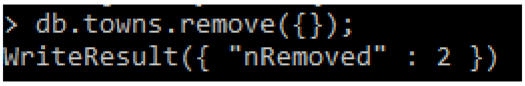
3. Проверьте содержание коллекции.



```
> db.towns.find();
{ "_id" : ObjectId("62b56d8df77db5c25e54ca5e"), "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("62b56d8df77db5c25e54ca5f"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "
food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

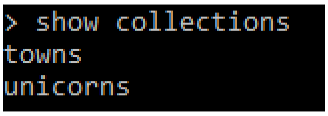
4. Очистите коллекцию.

db.towns.remove({});



```
> db.towns.remove({});
WriteResult({ "nRemoved" : 2 })
```

5. Просмотрите список доступных коллекций.



```
> show collections
towns
unicorns
```

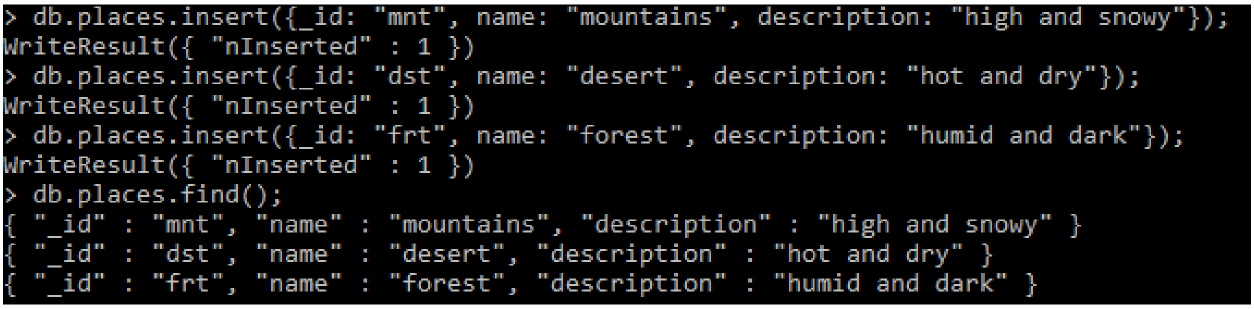
Практическое задание 8.3.1:

1. Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.

db.places.insert({_id: "mnt", name: "mountains", description: "high and snowy"});

db.places.insert({_id: "dst", name: "desert", description: "hot and dry"});

db.places.insert({_id: "frt", name: "forest", description: "humid and dark"});



```
> db.places.insert({_id: "mnt", name: "mountains", description: "high and snowy"});
WriteResult({ "nInserted" : 1 })
> db.places.insert({_id: "dst", name: "desert", description: "hot and dry"});
WriteResult({ "nInserted" : 1 })
> db.places.insert({_id: "frt", name: "forest", description: "humid and dark"});
WriteResult({ "nInserted" : 1 })
> db.places.find();
{ "_id" : "mnt", "name" : "mountains", "description" : "high and snowy" }
{ "_id" : "dst", "name" : "desert", "description" : "hot and dry" }
{ "_id" : "frt", "name" : "forest", "description" : "humid and dark" }
```

2. Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.

db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9efe")},{$set: {place: {$ref: "places", $id: "mnt"}}});

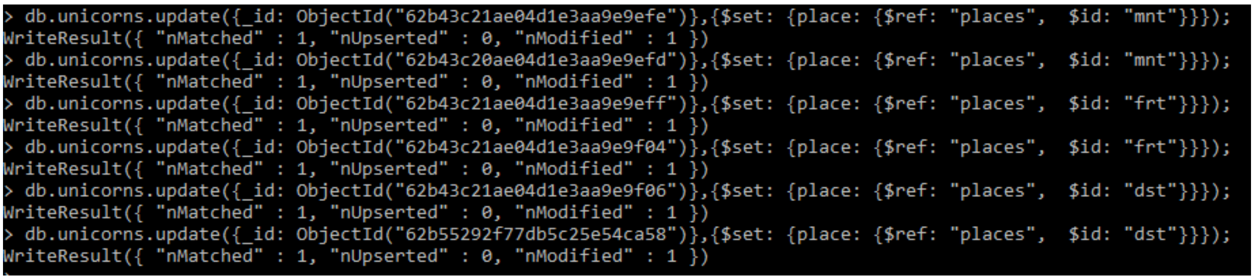
db.unicorns.update({_id: ObjectId("62b43c20ae04d1e3aa9e9efd")},{$set: {place: {$ref: "places", $id: "mnt"}}});

db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9eff")},{$set: {place: {$ref: "places", $id: "frt"}}});

db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9f04")},{$set: {place: {$ref: "places", $id: "frt"}}});

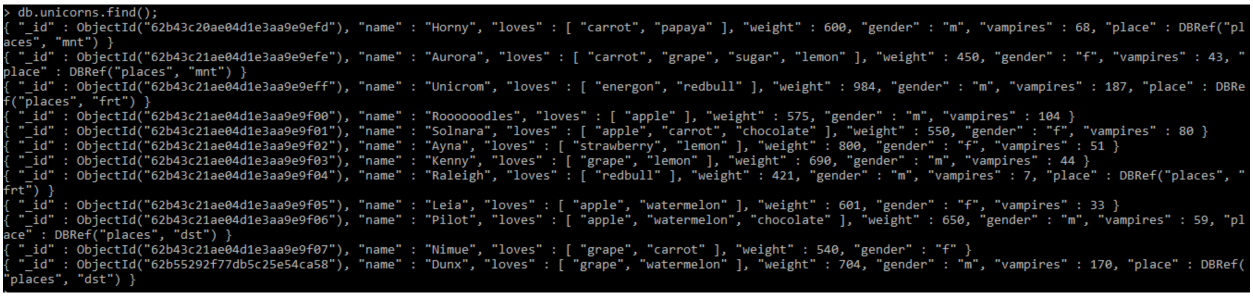
db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9f06")},{$set: {place: {$ref: "places", $id: "dst"}}});

db.unicorns.update({_id: ObjectId("62b55292f77db5c25e54ca58")},{$set: {place: {$ref: "places", $id: "dst"}}});



```
> db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9efe")},{$set: {place: {$ref: "places",  $id: "mnt"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id: ObjectId("62b43c20ae04d1e3aa9e9efd")},{$set: {place: {$ref: "places",  $id: "mnt"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9eff")},{$set: {place: {$ref: "places",  $id: "frt"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9f04")},{$set: {place: {$ref: "places",  $id: "frt"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id: ObjectId("62b43c21ae04d1e3aa9e9f06")},{$set: {place: {$ref: "places",  $id: "dst"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id: ObjectId("62b55292f77db5c25e54ca58")},{$set: {place: {$ref: "places",  $id: "dst"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3. Проверьте содержание коллекции единорогов.

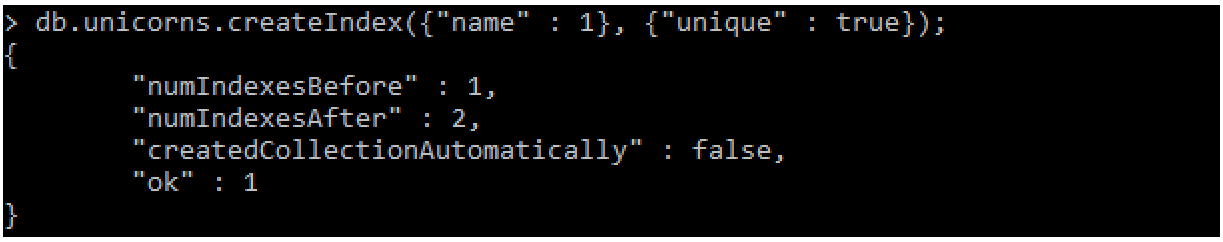


```
> db.unicorns.find();
{ "_id" : ObjectId("62b43c20ae04d1e3aa9e9efd"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "place" : DBRef("pl
aces", "mnt") }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9efe"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "
place" : DBRef("places", "mnt") }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9eff"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187, "place" : DBRe
f("places", "frt") }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f00"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f01"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f02"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f03"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f04"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7, "place" : DBRef("places", "
frt") }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f05"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f06"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59, "pl
ace" : DBRef("places", "dst") }
{ "_id" : ObjectId("62b43c21ae04d1e3aa9e9f07"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b55292f77db5c25e54ca58"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170, "place" : DBRef(
"places", "dst") }
```

Практическое задание 8.3.2:

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

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



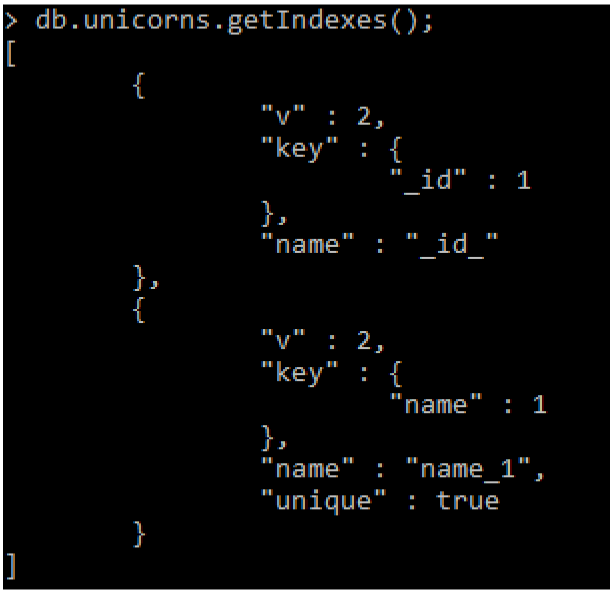
```
> db.unicorns.createIndex({"name" : 1}, {"unique" : true});
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
```

Можно, имена не повторяются.

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns.

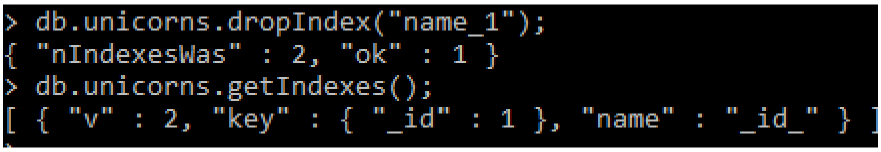
db.unicorns.getIndexes();



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

2. Удалите все индексы, кроме индекса для идентификатора.

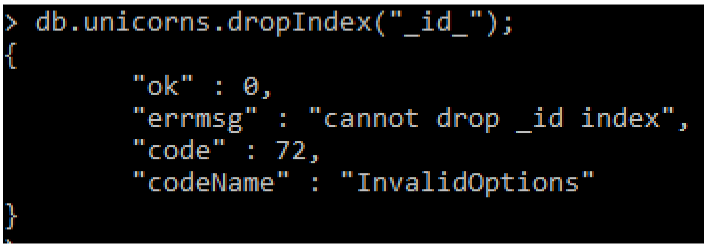
db.unicorns.dropIndex("name_1");



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

3. Попытайтесь удалить индекс для идентификатора.

db.unicorns.dropIndex("_id_");



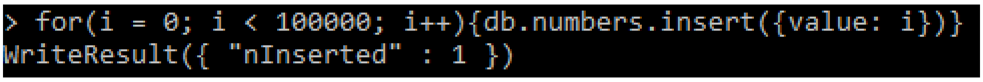
```
> db.unicorns.dropIndex("_id_");
{
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
}
```

Индекс для идентификатора удалить нельзя.

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers, задействовав курсор.

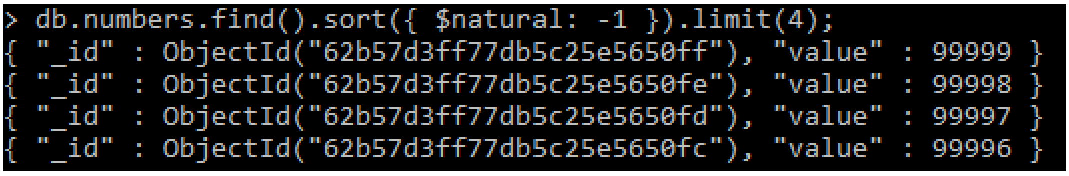
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}



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

2. Выберите последних четыре документа.

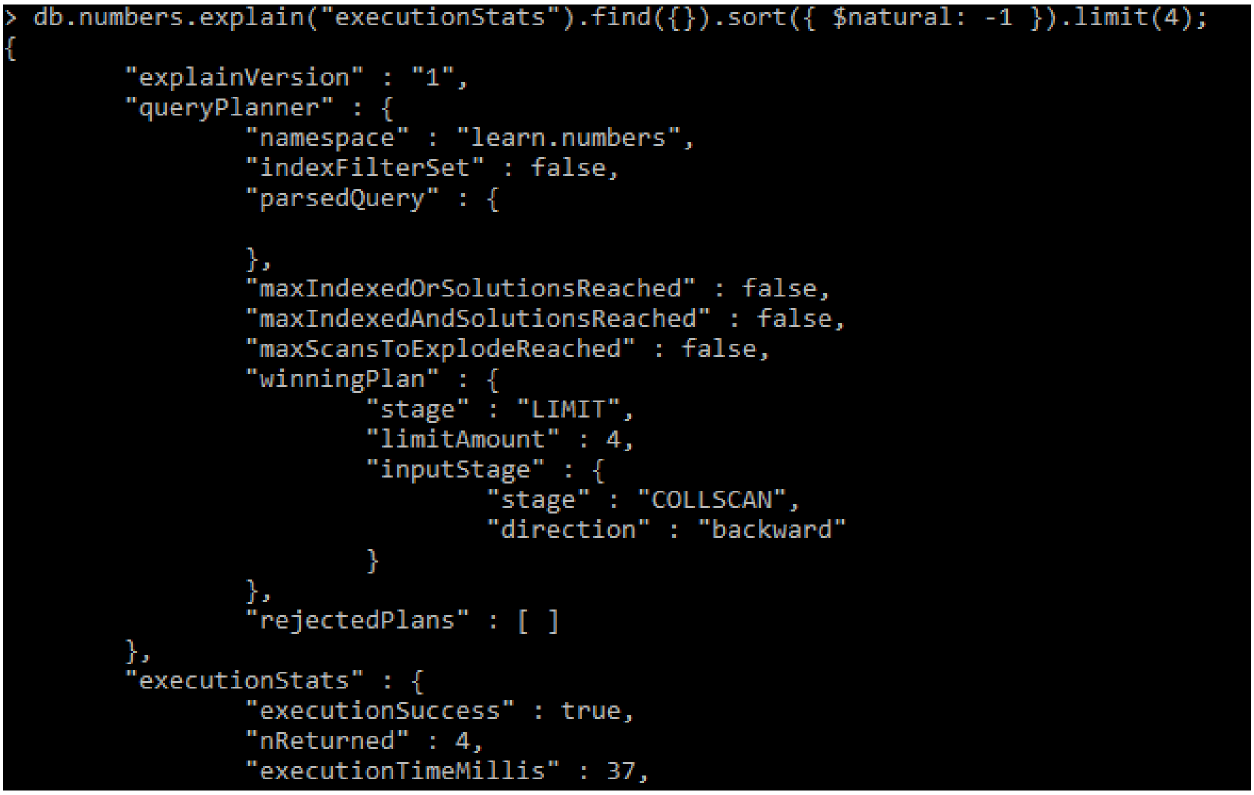
db.numbers.find().sort({ $natural: -1 }).limit(4);



```
> db.numbers.find().sort({ $natural: -1 }).limit(4);
{ "_id" : ObjectId("62b57d3ff77db5c25e5650ff"), "value" : 99999 }
{ "_id" : ObjectId("62b57d3ff77db5c25e5650fe"), "value" : 99998 }
{ "_id" : ObjectId("62b57d3ff77db5c25e5650fd"), "value" : 99997 }
{ "_id" : ObjectId("62b57d3ff77db5c25e5650fc"), "value" : 99996 }
```

3. 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);



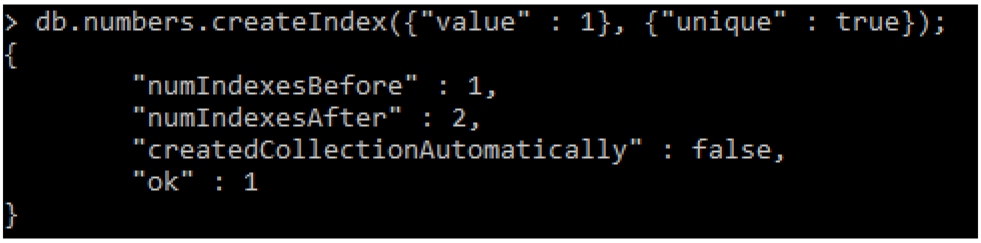
```
> db.numbers.explain("executionStats").find({}).sort({ $natural: -1 }).limit(4);
{
        "explainVersion" : "1",
        "queryPlanner" : {
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "maxIndexedOrSolutionsReached" : false,
                "maxIndexedAndSolutionsReached" : false,
                "maxScansToExplodeReached" : false,
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 37,
```

Запрос выполнялся в течение 37 миллисекунд.

4. Создайте индекс для ключа value.

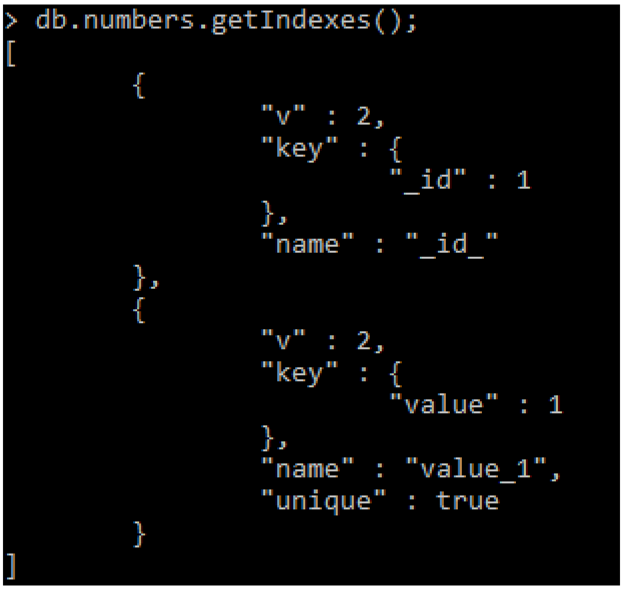
db.numbers.createIndex({"value" : 1}, {"unique" : true});



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

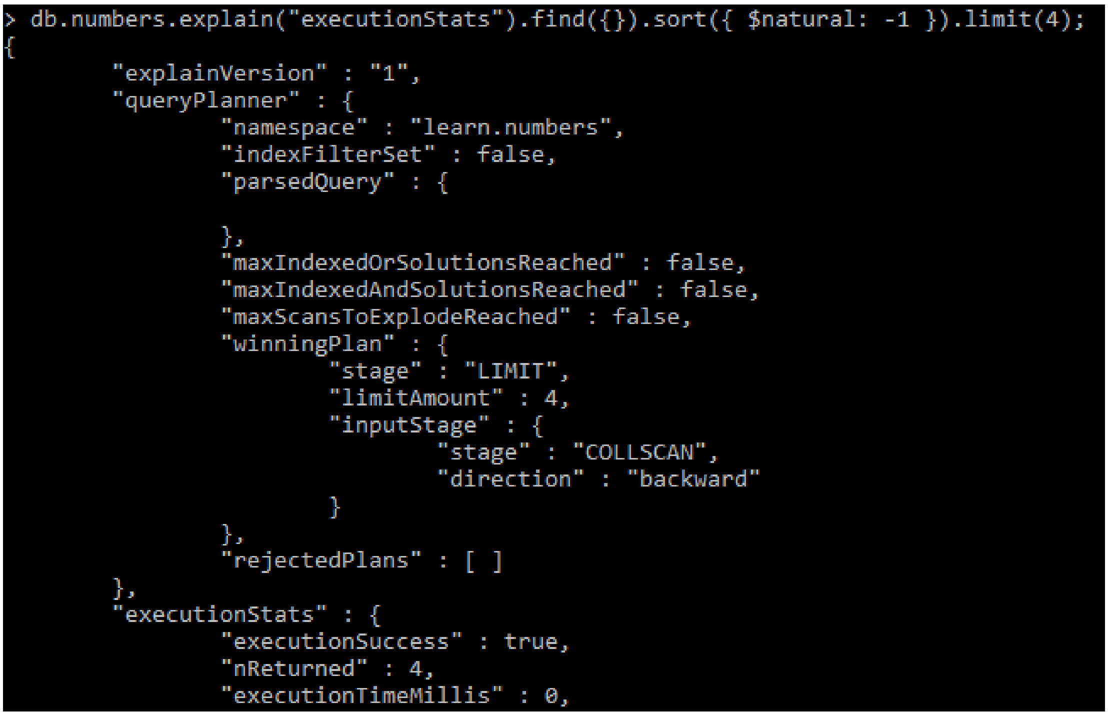
5. Получите информацию о всех индексах коллекции numbers.

db.numbers.getIndexes();



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

6. Выполните запрос 2.
7. 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" : 0,
```

На выполнение запроса потребовалось чуть больше 0 миллисекунд.

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

Более эффективен запрос с индексом, поскольку после создания индекса существенно снизилось время выполнения запроса.

Выводы.

Я овладела практическими навыками работы в базе данных MongoDB, а именно навыками работы с CRUD-операциями и вложенными объектами в коллекции, агрегации и изменения данных, работы со ссылками и индексами.