

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 8

по теме: Работа с БД в СУБД MongoDB

по дисциплине: Проектирование и реализация баз данных

Специальность:

45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверил:

Говорова М.М.

Дата: «13» июня 2021 г.

Оценка _____

Выполнил:

студент группы К3243

Кармазь Д.А.

Санкт-Петербург 2020/2021

Цель 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.

Практическое задание

Выполнить различные запросы в базе данных MongoDB.

Выполнение

8.1.1

1. Создайте базу данных learn.

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

2. Заполните коллекцию единорогов unicorns.
3. Используя второй способ, вставьте в коллекцию единорогов документ.

```
> 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: 'Rooooooodles', 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 })
> 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()
{ "_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c620ea43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

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});
{ "_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c620ea43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
> db.unicorns.find({"gender": "f"}).limit(3).sort({"name": 1});
{ "_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c620e443a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c620f643a3666aa4054833"), "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":{"$exists:'carrot'}}).limit(1)
{ "_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.findOne({"gender": "f", "loves":{"$exists:'carrot'}})
{
  "_id" : ObjectId("60c620cc43a3666aa405482c"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
```

8.1.3

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

```
> db.unicorns.find({"gender": "m"}, {"loves": false, "gender": false});
{ "_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("60c620ea43a3666aa4054831"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
```

8.1.4

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({"_id": -1});
{ "_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c620ea43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c620e443a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

8.1.5

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

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

8.1.6

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

```
> db.unicorns.find({'gender' : 'f', 'weight': {$gte: 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" }
>
_
```

8.1.7

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

```
> 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 }
>
```

8.1.8

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({'vampires': {$exists: 0}})
{ "_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
>
_
```

8.1.9

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

```
> db.unicorns.find({'gender':'m'}, {"name": 1, "_id": 0, "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" : "Roooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
```

8.2.1

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

```
>
> db.towns.find()
{ "_id" : ObjectId("60c633a843a3666aa4054837"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("60c633cc43a3666aa4054838"), "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("60c633de43a3666aa4054839"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

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

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

8.2.2

1. Сформировать функцию для вывода списка самцов единорогов.
2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя forEach.

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

8.2.3

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

8.2.4

Вывести список предпочтений.

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

8.2.5

Посчитать количество особей единорогов обоих полов.

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

8.2.6

1. Выполнить команду.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c620df43a3666aa405482e"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c620ef43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c620ea43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

8.2.7

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: "Ayna", gender: 'f'}, {$set: {weight: 800, vampires: 51}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{"_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{"_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{"_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{"_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{"_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{"_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{"_id" : ObjectId("60c620e443a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{"_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{"_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{"_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{"_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{"_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{"_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

8.2.8

1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: "Raleigh", gender: 'm'}, {$set: {loves: ['redbool']}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{"_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{"_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{"_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{"_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{"_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{"_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{"_id" : ObjectId("60c620e443a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{"_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "redbool" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{"_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{"_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{"_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{"_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{"_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

8.2.9

1. Всем самцам единорогов увеличить количество убитых вампиров на 5.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({gender: 'm'}, {$inc: {vampires: 5}}, {multi: true});
WriteResult({ "nMatched" : 8, "nUpserted" : 0, "nModified" : 8 })
> db.unicorns.find()
{"_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{"_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{"_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{"_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{"_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{"_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{"_id" : ObjectId("60c620e443a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{"_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "redbool" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{"_id" : ObjectId("60c620f643a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{"_id" : ObjectId("60c620fc43a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{"_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{"_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{"_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

8.2.10

1. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.
2. Проверить содержимое коллекции towns.

```
> db.towns.update({name: 'Portland'}, {$unset: {mayor.party: 1}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{"_id" : ObjectId("60c633a843a3666aa4054837"), "name" : "Punxsutawney ", "population" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{"_id" : ObjectId("60c633cc43a3666aa4054838"), "name" : "New York", "population" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{"_id" : ObjectId("60c6436743a3666aa405483d"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

8.2.11

1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: "Pilot", gender: 'm'}, {$push: {loves: 'chocolate'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find();
{"_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{"_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{"_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{"_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{"_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{"_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{"_id" : ObjectId("60c620ef43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{"_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "redbool" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{"_id" : ObjectId("60c620f143a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{"_id" : ObjectId("60c620f243a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{"_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{"_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{"_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

8.2.12

1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: "Aurora", gender: 'f'}, {$push: {loves: {$each: ['sugar', 'lemon']}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find();
{"_id" : ObjectId("60c620c543a3666aa405482b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{"_id" : ObjectId("60c620cc43a3666aa405482c"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{"_id" : ObjectId("60c620d343a3666aa405482d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{"_id" : ObjectId("60c620d943a3666aa405482e"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{"_id" : ObjectId("60c620df43a3666aa405482f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{"_id" : ObjectId("60c620ea43a3666aa4054830"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{"_id" : ObjectId("60c620ef43a3666aa4054831"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{"_id" : ObjectId("60c620f043a3666aa4054832"), "name" : "Raleigh", "loves" : [ "redbool" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{"_id" : ObjectId("60c620f143a3666aa4054833"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{"_id" : ObjectId("60c620f243a3666aa4054834"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{"_id" : ObjectId("60c6210443a3666aa4054835"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{"_id" : ObjectId("60c6214243a3666aa4054836"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{"_id" : ObjectId("60c63de543a3666aa405483c"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

8.2.13

1. Создайте коллекцию towns.
2. Удалите документы с беспартийными мэрами.
3. Проверьте содержание коллекции.

```
> 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"],
... mayor: {
...   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: "Sam Adams",
...   party: "D"}}
... );
WriteResult({ "nInserted" : 1 })
> db.towns.remove({'mayor.party': {'$exists: false'}});
WriteResult({ "nRemoved" : 1 })
> db.towns.find();
{"_id" : ObjectId("60c6459743a3666aa405483f"), "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("60c645a443a3666aa4054840"), "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.
5. Просмотрите список доступных коллекций.

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

8.3.1

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

```
> db.createCollection("places");
{ "ok" : 1 }
> db.places.insert({"_id": 'frs', "name": 'forest', 'description': "beautiful green forest"});
WriteResult({ "nInserted" : 1 })
> db.places.insert({"_id": 's', 'name': 'sea', 'description': 'warm blue sea'});
WriteResult({ "nInserted" : 1 })
> db.places.insert({"_id": 'mnt', 'name': 'mountain', 'description': 'high rocky mountain'});
uncaught exception: SyntaxError: missing : after property id :
@(shell):1:41
> db.places.insert({"_id": 'mnt', 'name': 'mountain', 'description': 'high rocky mountain'});
WriteResult({ "nInserted" : 1 })
```

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

3. Проверьте содержание коллекции единорогов.

```
> db.unicorns.update({'name': 'Aurora'}, {$set: {place: {$ref: "places", $id: "frs"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({'name': 'Pilot'}, {$set: {place: {$ref: "places", $id: "s"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({'name': 'Dunx'}, {$set: {place: {$ref: "places", $id: "mnt"}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find();
{ "_id" : ObjectId("60c649e943a3666aa4854845"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c649f543a3666aa4854846"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43, "place" : DBRef("places", "frs") }
{ "_id" : ObjectId("60c649fc43a3666aa4854847"), "name" : "Unicorn", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c649fc43a3666aa4854848"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60c649fc43a3666aa4854849"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c649fc43a3666aa485484a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c649fc43a3666aa485484b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c64a0343a3666aa485484c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c64a0343a3666aa485484d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54, "place" : DBRef("places", "s") }
{ "_id" : ObjectId("60c64a0343a3666aa485484e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c64a1643a3666aa485484f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165, "place" : DBRef("places", "mnt") }
{ "_id" : ObjectId("60c64a6043a3666aa4854850"), "name" : "Roocoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
```

8.3.2

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

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

8.3.3

1. Получите информацию о всех индексах коллекции unicorns .

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

2. Удалите все индексы, кроме индекса для идентификатора.
3. Попробуйте удалить индекс для идентификатора.

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

8.3.4

1. Создайте объемную коллекцию numbers, задействовав курсор.
2. Выберите последних четыре документа.

```
> db.createCollection("numbers");
{ "ok" : 1 }
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i});}
WriteResult({ "nInserted" : 1 })
> db.numbers.find().sort({$natural: -1}).limit(4);
{ "_id" : ObjectId("60c65352d023990a9ecd80cf"), "value" : 99999 }
{ "_id" : ObjectId("60c65352d023990a9ecd80e"), "value" : 99998 }
{ "_id" : ObjectId("60c65352d023990a9ecd80d"), "value" : 99997 }
{ "_id" : ObjectId("60c65352d023990a9ecd80c"), "value" : 99996 }
> db.numbers.explain("executionStats").find().sort({$natural: -1}).limit(4);
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "test.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {
    },
    "winningPlan" : {
      "stage" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 0,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 4,
    "executionStages" : {
      "stage" : "LIMIT",
      "nReturned" : 4,
      "executionTimeMillisEstimate" : 0,
      "works" : 6,
      "advanced" : 4,
      "needTime" : 1,

```

3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра `executionTimeMillis`)
4. Создайте индекс для ключа `value`.
5. Получите информацию обо всех индексах коллекции `numbers`.

```
> db.numbers.ensureIndex({"value": 1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.numbers.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "value" : 1
    },
    "name" : "value_1"
  }
]
```

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

```
db.numbers.find().sort({$natural: -1}).limit(4);
{"_id" : ObjectId("60c65352d023990a9ecd880f"), "value" : 99999 }
{"_id" : ObjectId("60c65352d023990a9ecd880e"), "value" : 99998 }
{"_id" : ObjectId("60c65352d023990a9ecd880d"), "value" : 99997 }
{"_id" : ObjectId("60c65352d023990a9ecd880c"), "value" : 99996 }
db.numbers.explain("executionStats").find().sort({$natural: -1}).limit(4);

{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "test.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {

    },
    "winningPlan" : {
      "stage" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 0,
    "totalKeysExamined" : 0,
    "totalDocsExamined" : 4,
    "executionStages" : {
      "stage" : "LIMIT",
      "nReturned" : 4,
      "executionTimeMillisEstimate" : 0,
      "works" : 6,
      "advanced" : 4,
      "needTime" : 1,
      "needYield" : 0,
      "saveState" : 0,
      "restoreState" : 0,
      "isEOF" : 1,
    }
  }
}
```

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

Выводы

В ходе лабораторной работы были созданы и выполнены различные запросы в БД MongoDB.