Министерство науки и высшего образования Российской Федерации

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

Факультет инфокоммуникационных технологий

Лабораторная работа №8

по дисциплине:

«Создание таблиц базы данных POSTGRESQL. Заполнение таблиц рабочими данными»

Выполнил:
Студент 2 курса ИКТ
группы К3241
Павел Золотов

Проверил:
Говорова Марина Михайловна

Санкт-Петербург
2021

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:

Задание 8.1.1

1) Создайте базу данных learn.
2) Заполните коллекцию единорогов unicorns
3) Используя второй способ, вставьте в коллекцию единорогов документ
4) Проверьте содержимое коллекции с помощью метода find

```
> db.unicorns.insert(document)
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 42
1, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
> []
```

Задание 8.1.2

1) Сформируйте запросы для вывода списков самцов и самок единорогов.
Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.
2) Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

```
> db.unicorns.find({"gender": "f"}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
> db.unicorns.find({"gender": "m"}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
```

```
> db.unicorns.find({"gender": "f", "loves": "carrot"}).limit(1)
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
> db.unicorns.findOne({"gender": "f", "loves": "carrot"})
{
        "_id" : ObjectId("60be7c1f5baa457bbd18e985"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
```

Задание 8.1.3

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

```
> db.unicorns.find({"gender": "m"}, {loves: 0})
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "weight" : 600, "gender" : "m", "vampires" : 63
 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "weight" : 984, "gender" : "m", "vampires" :
182 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "weight" : 575, "gender" : "m", "vampires
" : 99 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "weight" : 690, "gender" : "m", "vampires" : 39
 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "weight" : 421, "gender" : "m", "vampires" :
2 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "weight" : 650, "gender" : "m", "vampires" : 54
 }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "weight" : 704, "gender" : "m", "vampires" : 165
 }
```

Задание 8.1.4

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({ $natural: -1})
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 42
1, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
```

Задание 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" ], "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" : "Roooooodles", "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 }
```

Задание 8.1.6

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

```
> db.unicorns.find({weight: {$gt : 500, $lt : 700}}, {_id: false})
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "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" }
```

Задание 8.1.7

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

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

Задание 8.1.8

Найти всех единорогов, не имеющих ключ *vampires*.

```
> db.unicorns.find({vampires: {$exists: false}})
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
```

Задание 8.1.9

Вывести список 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})
{ "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" ] }
```

Задание 8.2.1

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

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

```
> 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: ["sta
tus of liberty", "food"], mayor: { name: "Michael Bloomberg", 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 })
```

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

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

Задание 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.

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

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()
5
>
```

Задание 8.2.4

Вывести список предпочтений.

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

Задание 8.2.5

Посчитать количество особей единорогов обоих полов.

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

Задание 8.2.6

1. Выполнить команду:
`db.unicorns.save({name: “Barny”, loves: [“grape”], weight: 340, gender: «m»})`
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.save({name: "Barny", loves: ["grape"], weight: 340, gender: "m"})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 42
1, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m" }
```

Задание 8.2.7

1. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: "Ayna"}, {"name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gend
er" : "f", "vampires" : 51 }, {upsert: false})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 42
1, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m" }
```

Задание 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: ["redbull"]}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gen
der" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m" }
```

Задание 8.2.9

1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
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("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gen
der" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m", "vampires" : 5 }
```

Задание 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("60bf5de672f61e9c3258f6c8"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : I
SODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("60bf5e0372f61e9c3258f6c9"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : IS
ODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloo
mberg", "party" : "I" } }
{ "_id" : ObjectId("60bf5e1672f61e9c3258f6ca"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISOD
ate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

Задание 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("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gen
der" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate"
], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m", "vampires" : 5 }
```

Задание 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"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be7bd75baa457bbd18e984"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be7c1f5baa457bbd18e985"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon
" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be7c275baa457bbd18e986"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be7c2e5baa457bbd18e987"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be7c365baa457bbd18e988"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be7c3c5baa457bbd18e989"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60be7c4a5baa457bbd18e98a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be7c535baa457bbd18e98b"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gen
der" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be7c595baa457bbd18e98c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be7c5e5baa457bbd18e98d"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate"
], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be7c635baa457bbd18e98e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f" }
{ "_id" : ObjectId("60be80605baa457bbd18e98f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60c0d67516e5fa372774869d"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender"
 : "m", "vampires" : 5 }
```

Задание 8.2.13

1) Создайте коллекцию towns, включающую документы из задания
2) Удалите документы с беспартийными мэрами.
3) Проверьте содержание коллекции.
4) Очистите коллекцию.

5) Просмотрите список доступных коллекций.

```
> db.towns.find()
{ "_id" : ObjectId("60bf5de672f61e9c3258f6c8"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : I
SODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("60bf5e0372f61e9c3258f6c9"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : IS
ODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloo
mberg", "party" : "I" } }
{ "_id" : ObjectId("60bf5e1672f61e9c3258f6ca"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISOD
ate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" }
 }
> db.towns.remove({"mayor.party": {$exists: false}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("60bf5e0372f61e9c3258f6c9"), "name" : "New York", "populatiuon" : 22200000, "last_sensus" : IS
ODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloo
mberg", "party" : "I" } }
{ "_id" : ObjectId("60bf5e1672f61e9c3258f6ca"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISOD
ate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" }
 }
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> show collections
towns
unicorns
```

Задание 8.3.1

1) Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2) Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания.
3) Проверьте содержание коллекции едиорогов.

```
> db.habitats.insert({_id: "mtn", name: "mountain", descr: "High mountain"})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: "cld", name: "cloud", descr: "Magical cloud"})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: "hvn", name: "heaven", descr: "The best place"})
WriteResult({ "nInserted" : 1 })
```

```
> db.unicorns.update({_id : ObjectId("60c108f616e5fa37277486a0")}, {$set: {habitat: {$ref:"habitats", $id: "cld"}
}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id : ObjectId("60c1095116e5fa37277486a3")}, {$set: {habitat: {$ref:"habitats", $id: "mtn"}
}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({_id : ObjectId("60c1096716e5fa37277486a8")}, {$set: {habitat: {$ref:"habitats", $id: "hvn"}
}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.unicorns.find()
{ "_id" : ObjectId("60c108ed16e5fa372774869e"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 60
0, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c108f216e5fa372774869f"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 45
0, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c108f616e5fa37277486a0"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight"
: 984, "gender" : "m", "vampires" : 182, "habitat" : DBRef("habitats", "cld") }
{ "_id" : ObjectId("60c1091a16e5fa37277486a1"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ],
 "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60c1094a16e5fa37277486a2"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "g
ender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c1095116e5fa37277486a3"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" :
733, "gender" : "f", "vampires" : 40, "habitat" : DBRef("habitats", "mtn") }
{ "_id" : ObjectId("60c1095516e5fa37277486a4"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690,
 "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c1095916e5fa37277486a5"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 42
1, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c1095e16e5fa37277486a6"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" :
601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c1096316e5fa37277486a7"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" :
 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c1096716e5fa37277486a8"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540
, "gender" : "f", "habitat" : DBRef("habitats", "hvn") }
{ "_id" : ObjectId("60c1096b16e5fa37277486a9"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" :
704, "gender" : "m", "vampires" : 165 }
```

Задание 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
}
```

Задание 8.3.3

1) Получите информацию о всех индексах коллекции unicorns .
2) Удалите все индексы, кроме индекса для идентификатора.
3) Попытайтесь удалить индекс для идентификатора.

```
> db.unicorns.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "unique" : true,
                "key" : {
                        "name" : 1
                },
                "name" : "name_1"
        }
]
> db.unicorns.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.dropIndex("_id_")
{
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
}
```

Задание 8.3.4

1) Создайте объемную коллекцию numbers, задействовав курсор:
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
2) Выберите последних четыре документа.
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 о всех индексах коллекции numbres.
6) Выполните запрос 2.
7) Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8) Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}; null;
null
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("60c1157d16e5fa3727760d49"), "value" : 99999 }
{ "_id" : ObjectId("60c1157d16e5fa3727760d48"), "value" : 99998 }
{ "_id" : ObjectId("60c1157d16e5fa3727760d47"), "value" : 99997 }
{ "_id" : ObjectId("60c1157d16e5fa3727760d46"), "value" : 99996 }
> db.numbers.explain("executionStats").find().sort({ $natural: -1 }).limit(4)
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "learn.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 2,
                "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,
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "nReturned" : 4,
                                "executionTimeMillisEstimate" : 0,
                                "works" : 5,
                                "advanced" : 4,
                                "needTime" : 1,
                                "needYield" : 0,
                                "saveState" : 0,
                                "restoreState" : 0,
                                "isEOF" : 0,
                                "direction" : "backward",
                                "docsExamined" : 4
                        }
                }
        },
        "serverInfo" : {
                "host" : "home",
                "port" : 27017,
                "version" : "4.4.6",
                "gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"
        },
        "ok" : 1
}
```

```
> 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"
        }
]
> db.numbers.explain("executionStats").find().sort({ $natural: -1 }).limit(4)
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "learn.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,
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "nReturned" : 4,
                                "executionTimeMillisEstimate" : 0,
                                "works" : 5,
                                "advanced" : 4,
                                "needTime" : 1,
                                "needYield" : 0,
                                "saveState" : 0,
                                "restoreState" : 0,
                                "isEOF" : 0,
                                "direction" : "backward",
                                "docsExamined" : 4
                        }
                }
        },
        "serverInfo" : {
                "host" : "home",
                "port" : 27017,
                "version" : "4.4.6",
                "gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"
        },
        "ok" : 1
}
```

Эффективнее работает запрос с индексом. Наблюдается ускорение с 2 мс до <1 мс.

Выводы

В результате выполнения работы были получены начальные навыки работы с MongoDB. Получен опыт создания CRUD запросов, работы с коллекциями, создание и управление индексами и применение их для сокращения времени исполнения запроса.