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: «__» _____ 20__г.
Оценка _____

Выполнил:
студент группы k3243
Новикова В.В.

Санкт-Петербург 2021

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

**Оборудование**: компьютерный класс.

**Программное обеспечение**: СУБД MongoDB 4.4.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

## 8.1 CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ. ВЫБОРКА ДАННЫХ

### 8.1.1 ВСТАВКА ДОКУМЕНТОВ В КОЛЛЕКЦИЮ

**Практическое 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

```
>>> use learn
switched to db learn
>>> 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 })
>>>
```

```
>>> 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 })
>>>
```

```
>>> db.unicorns.find()
{ "_id" : ObjectId("60c7be5c445ef0d6f930b07d"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c7be9e445ef0d6f930b07e"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c7bea6445ef0d6f930b07f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c7beae445ef0d6f930b080"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c7beb6445ef0d6f930b081"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60c7bebd445ef0d6f930b082"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c7bed8445ef0d6f930b083"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c7bee1445ef0d6f930b084"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c7bee9445ef0d6f930b085"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c7befb445ef0d6f930b086"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c7bf07445ef0d6f930b087"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c7bfb5445ef0d6f930b088"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>>>
```

## 8.1.2 ВЫБОРКА ДАННЫХ ИЗ БД

**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.

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

```
>>> db.unicorns.find({gender: 'm'}).sort({name: 1});
{ "_id" : ObjectId("60c7bfb5445ef0d6f930b088"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c7be5c445ef0d6f930b07d"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60c7bed8445ef0d6f930b083"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c7befb445ef0d6f930b086"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c7bee1445ef0d6f930b084"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c7beae445ef0d6f930b080"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c7bea6445ef0d6f930b07f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>>>
```

```
>>> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3);
{ "_id" : ObjectId("60c7be9e445ef0d6f930b07e"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c7bebd445ef0d6f930b082"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c7bee9445ef0d6f930b085"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>>>
```

```
>>> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);
{ "_id" : ObjectId("60c7be9e445ef0d6f930b07e"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>>>
```

```
>>> db.unicorns.findOne({gender: 'f', loves: 'carrot'});
{
    "_id" : ObjectId("60c7be9e445ef0d6f930b07e"),
    "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'}, {gender: 0, loves: 0});
{ "_id" : ObjectId("60c7be5c445ef0d6f930b07d"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("60c7bea6445ef0d6f930b07f"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("60c7beae445ef0d6f930b080"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("60c7bed8445ef0d6f930b083"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("60c7bee1445ef0d6f930b084"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("60c7befb445ef0d6f930b086"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("60c7bfb5445ef0d6f930b088"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
>>>
```

## Практическое задание 8.1.4

Вывести список единорогов в обратном порядке добавления.

```
>>> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("60c7bfb5445ef0d6f930b088"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60c7bf07445ef0d6f930b087"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60c7befb445ef0d6f930b086"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60c7bee9445ef0d6f930b085"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60c7bee1445ef0d6f930b084"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60c7bed8445ef0d6f930b083"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60c7bebd445ef0d6f930b082"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60c7beb6445ef0d6f930b081"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60c7beae445ef0d6f930b080"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60c7bea6445ef0d6f930b07f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60c7be9e445ef0d6f930b07e"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60c7be5c445ef0d6f930b07d"), "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}});
{ "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 }
>>>
```

### 8.1.3 ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

## 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});
{ "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});
{ "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}});
{ "_id" : ObjectId("60c7bf07445ef0d6f930b087"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
>>>
```

## Практическое задание 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 ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB. ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ

## 8.2.1 ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

## Практическое задание 8.2.1

1. Создайте коллекцию towns, включающую следующие документы
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: ["status 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" } }
```

## 8.2.3 КУРСОРЫ

**Практическое 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.

```
>>> males = function() {return this.gender=="m";}
function() {return this.gender=="m";}
```

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

## 8.2.4 АГРЕГИРОВАННЫЕ ЗАПРОСЫ

**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()
2
```

**Практическое задание 8.2.4**

Вывести список предпочтений.

```
>>> 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}}}])
{ "_id" : "m", "count" : 7 }
{ "_id" : "f", "count" : 5 }
```

## 8.2.5 РЕДАКТИРОВАНИЕ ДАННЫХ

## Практическое задание 8.2.6

1. Выполнить команду
2. Проверить содержимое коллекции unicorns

```
>>> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
>>> db.unicorns.find()
{ "_id" : ObjectId("60cbba366c2f0378e833c074"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60cbba3d6c2f0378e833c075"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60cbba446c2f0378e833c076"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60cbba4b6c2f0378e833c077"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60cbba536c2f0378e833c078"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbba5b6c2f0378e833c079"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60cbba6b6c2f0378e833c07a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60cbba716c2f0378e833c07b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60cbba786c2f0378e833c07c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbba7f6c2f0378e833c07d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60cbba886c2f0378e833c07e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbbaa56c2f0378e833c07f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60cbbccc6c2f0378e833c080"), "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 вапмира.
2. Проверить содержимое коллекции unicorns.

```
>>> db.unicorns.update({name: 'Ayna'}, {name: 'Ayna', loves: ['strawberry','lemon'],gender: 'f',weight:800, vampires:51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>>> db.unicorns.find()
{ "_id" : ObjectId("60cbbff6afa0632d6a33b5aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60cbc015afa0632d6a33b5ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60cbc01bafa0632d6a33b5ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60cbc021afa0632d6a33b5ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60cbc028afa0632d6a33b5ae"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbc02eafa0632d6a33b5af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60cbc044afa0632d6a33b5b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60cbc04aafa0632d6a33b5b1"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60cbc050afa0632d6a33b5b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbc057afa0632d6a33b5b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60cbc05eafa0632d6a33b5b4"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbc06aafa0632d6a33b5b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60cbc089afa0632d6a33b5b6"), "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.
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("60cbbff6afa0632d6a33b5aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60cbc015afa0632d6a33b5ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60cbc01bafa0632d6a33b5ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60cbc021afa0632d6a33b5ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60cbc028afa0632d6a33b5ae"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbc02eafa0632d6a33b5af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60cbc044afa0632d6a33b5b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60cbc04aafa0632d6a33b5b1"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60cbc050afa0632d6a33b5b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbc057afa0632d6a33b5b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60cbc05eafa0632d6a33b5b4"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbc06aafa0632d6a33b5b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60cbc089afa0632d6a33b5b6"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>>>
```

## Практическое задание 8.2.9

1. Всем самцам единорогов увеличить количество убитых вапмиров 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("60cbbff6afa0632d6a33b5aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60cbc015afa0632d6a33b5ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60cbc01bafa0632d6a33b5ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60cbc021afa0632d6a33b5ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60cbc028afa0632d6a33b5ae"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbc02eafa0632d6a33b5af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60cbc044afa0632d6a33b5b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60cbc04aafa0632d6a33b5b1"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60cbc050afa0632d6a33b5b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbc057afa0632d6a33b5b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60cbc05eafa0632d6a33b5b4"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbc06aafa0632d6a33b5b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60cbc089afa0632d6a33b5b6"), "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.

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("60cbc3bcafa0632d6a33b5b7"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" :
[ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("60cbc3c4afa0632d6a33b5b8"), "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("60cbc3caafa0632d6a33b5b9"), "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.

2. Проверить содержимое коллекции unicorns.

```
>>> db.unicorns.update({name:'Pilot'},{$push:{loves:'chocolate'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>>> db.unicorns.find()
{ "_id" : ObjectId("60cbc5ec208c641d1e515aaf"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60cbc5f1208c641d1e515ab0"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60cbc5f7208c641d1e515ab1"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60cbc5fc208c641d1e515ab2"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60cbc601208c641d1e515ab3"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbc606208c641d1e515ab4"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60cbc614208c641d1e515ab5"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60cbc61a208c641d1e515ab6"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60cbc620208c641d1e515ab7"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbc625208c641d1e515ab8"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires"
: 59 }
{ "_id" : ObjectId("60cbc62b208c641d1e515ab9"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbc636208c641d1e515aba"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60cbc661208c641d1e515abe"), "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.

2. Проверить содержимое коллекции unicorns.

```
>>> db.unicorns.update({name:'Aurora'},{$addToSet:{loves:{$each:['sugar','lemon']}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>>> db.unicorns.find()
{ "_id" : ObjectId("60cbc5ec208c641d1e515aaf"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60cbc5f1208c641d1e515ab0"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f",
"vampires" : 43 }
{ "_id" : ObjectId("60cbc5f7208c641d1e515ab1"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60cbc5fc208c641d1e515ab2"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60cbc601208c641d1e515ab3"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbc606208c641d1e515ab4"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60cbc614208c641d1e515ab5"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60cbc61a208c641d1e515ab6"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60cbc620208c641d1e515ab7"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbc625208c641d1e515ab8"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires"
: 59 }
{ "_id" : ObjectId("60cbc62b208c641d1e515ab9"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbc636208c641d1e515aba"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60cbc661208c641d1e515abe"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
>>>
```

### 8.2.6 УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ

## Практическое задание 8.2.13

1. Создайте коллекцию towns

2. Удалите документы с беспартийными мэрами.

3. Проверьте содержание коллекции.

4. Очистите коллекцию.

5. Просмотрите список доступных коллекций.

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

```
>>> db.towns.find()
{ "_id" : ObjectId("60cbcb714d2807511652c62f"), "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("60cbcb774d2807511652c630"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [
"beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

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

```
>>> show collections
towns
unicorns
>>>
```

## 8.3 ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

### 8.3.1 ССЫЛКИ В БД

**Практическое 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.zones.insert({_id:'eq',name:'equestria',description:'here live adult unicorns'})
WriteResult({ "nInserted" : 1 })
>>> db.zones.insert({_id:'po',name:'ponyville',description:'here live young unicorns'})
WriteResult({ "nInserted" : 1 })
```

```
>>> db.unicorns.update({name:'Unicrom'},{$set:{zone:{$ref:'zones',$id:'eq'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>>> db.unicorns.update({name:'Raleigh'},{$set:{zone:{$ref:'zones',$id:'po'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>>> db.unicorns.update({name:'Aurora'},{$set:{zone:{$ref:'zones',$id:'po'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
>>> db.unicorns.find()
{ "_id" : ObjectId("60cbda92596fd81a1925e8ce"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60cbda99596fd81a1925e8cf"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" :
DBRef("zones", "po") }
{ "_id" : ObjectId("60cbdaa0596fd81a1925e8d0"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182,
"zone" : DBRef("zones", "eq") }
{ "_id" : ObjectId("60cbdaa5596fd81a1925e8d1"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60cbdaaa596fd81a1925e8d2"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" :
80 }
{ "_id" : ObjectId("60cbdaaf596fd81a1925e8d3"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60cbdabe596fd81a1925e8d4"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60cbdac4596fd81a1925e8d5"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2, "zone" :
DBRef("zones", "po") }
{ "_id" : ObjectId("60cbdac9596fd81a1925e8d6"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60cbdad3596fd81a1925e8d7"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60cbdad9596fd81a1925e8d8"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60cbdae7596fd81a1925e8d9"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>>>
```

### 8.3.2 НАСТРОЙКА ИНДЕКСОВ

**Практическое 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
}
```

## 8.3.3 УПРАВЛЕНИЕ ИНДЕКСАМИ

**Практическое 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"
}
```

**Практическое задание 8.3.4**

1. Создайте объемную коллекцию numbers, задействовав курсор
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})}
WriteResult({ "nInserted" : 1 })
```

```
>>> db.numbers.find({value:{$gte:99996}})
{ "_id" : ObjectId("60cbe6247d64ca24b3d281b0"), "value" : 99996 }
{ "_id" : ObjectId("60cbe6247d64ca24b3d281b1"), "value" : 99997 }
{ "_id" : ObjectId("60cbe6247d64ca24b3d281b2"), "value" : 99998 }
{ "_id" : ObjectId("60cbe6247d64ca24b3d281b3"), "value" : 99999 }
```

```
>>> db.users.explain("executionStats").find({executionTimeMillis:1})
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "learn.users",
        "indexFilterSet" : false,
        "parsedQuery" : {
            "executionTimeMillis" : {
                "$eq" : 1
            }
        },
        "winningPlan" : {
            "stage" : "EOF"
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 0,
        "executionTimeMillis" : 0,
        "totalKeysExamined" : 0,
        "totalDocsExamined" : 0,
        "executionStages" : {
            "stage" : "EOF",
            "nReturned" : 0,
            "executionTimeMillisEstimate" : 0,
            "works" : 1,
            "advanced" : 0,
            "needTime" : 0,
            "needYield" : 0,
            "saveState" : 0,
            "restoreState" : 0,
            "isEOF" : 1
        }
    },
    "serverInfo" : {
        "host" : "62a20948ff73",
        "port" : 27017,
        "version" : "4.4.6",
        "gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"
    },
    "ok" : 1
}
```

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

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

```
>>> db.numbers.explain("executionStats").find({})
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "learn.numbers",
        "indexFilterSet" : false,
        "parsedQuery" : {

        },
        "winningPlan" : {
            "stage" : "COLLSCAN",
            "direction" : "forward"
        },
        "rejectedPlans" : [ ]
    },
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 100000,
        "executionTimeMillis" : 38,
        "totalKeysExamined" : 0,
        "totalDocsExamined" : 100000,
        "executionStages" : {
            "stage" : "COLLSCAN",
            "nReturned" : 100000,
            "executionTimeMillisEstimate" : 4,
            "works" : 100002,
            "advanced" : 100000,
            "needTime" : 1,
            "needYield" : 0,
            "saveState" : 100,
            "restoreState" : 100,
            "isEOF" : 1,
            "direction" : "forward",
            "docsExamined" : 100000
        }
    },
    "serverInfo" : {
        "host" : "62a20948ff73",
        "port" : 27017,
        "version" : "4.4.6",
        "gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"
    },
    "ok" : 1
}
```

**Выводы:** в ходе работы были проведены операции добавления, редактирования данных, а также поиск и фильтр данных.