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 № 5

по теме: **работа с БД в СУБД MongoDB**

по дисциплине: Проектирование и реализация баз данных

Специальность: 45.03.04 Интеллектуальные системы в гуманитарной сфере

Проверила:

Говорова М.М. _____

Выполнила: студентка К3243,
Куканова У.Д._____

Санкт-Петербург 2021/2022

Практическое задание 8.1.1:

1) *Создать базу данных learn.*
2) *Заполнить коллекцию единорогов unicorns:*

```
> use learn
switched to db learn
> db.createCollection("unicorns")
{ "ok" : 1 }
> 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("62a5e225117c8b117e4f0758"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a5e237117c8b117e4f0759"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a5e23c117c8b117e4f075a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a5e243117c8b117e4f075b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a5e24c117c8b117e4f075c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a5e255117c8b117e4f075d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a5e25d117c8b117e4f075e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a5e264117c8b117e4f075f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a5e269117c8b117e4f0760"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a5e270117c8b117e4f0761"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a5e27a117c8b117e4f0762"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a5e809117c8b117e4f0763"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

Практическое задание 8.1.2:

1) Сформировать запросы для вывода списков самцов и самок единорогов. Ограничить список самок первыми тремя особями. Отсортировать списки по имени.

```
> db.unicorns.find({gender: 'm'}).sort({name: 1})
{ "_id" : ObjectId("62a5e809117c8b117e4f0763"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a5e225117c8b117e4f0758"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a5e25d117c8b117e4f075e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a5e270117c8b117e4f0761"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a5e264117c8b117e4f075f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a5e243117c8b117e4f075b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a5e23c117c8b117e4f075a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
```

```
> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("62a5e237117c8b117e4f0759"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a5e255117c8b117e4f075d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a5e269117c8b117e4f0760"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>
```

2) Найти всех самок, которые любят carrot. Ограничить этот список первой особью с помощью функций findOne и limit.

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
        "_id" : ObjectId("62a5e237117c8b117e4f0759"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("62a5e237117c8b117e4f0759"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>
```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
{ "_id" : ObjectId("62a5e225117c8b117e4f0758"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("62a5e23c117c8b117e4f075a"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("62a5e243117c8b117e4f075b"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("62a5e25d117c8b117e4f075e"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("62a5e264117c8b117e4f075f"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("62a5e270117c8b117e4f0761"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("62a5e809117c8b117e4f0763"), "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("62a5e809117c8b117e4f0763"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a5e27a117c8b117e4f0762"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a5e270117c8b117e4f0761"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a5e269117c8b117e4f0760"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a5e264117c8b117e4f075f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a5e25d117c8b117e4f075e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a5e255117c8b117e4f075d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a5e24c117c8b117e4f075c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }

{ "_id" : ObjectId("62a5e243117c8b117e4f075b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a5e23c117c8b117e4f075a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a5e237117c8b117e4f0759"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a5e225117c8b117e4f0758"), "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 }
>
```

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})
...
```

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ `vampires`.

```
> db.unicorns.find({vampires: {$exists:false}})
{ "_id" : ObjectId("62a5e27a117c8b117e4f0762"), "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({}, {loves: {$slice: 1}}).sort({name: 1})
{ "_id" : ObjectId("62a5e237117c8b117e4f0759"), "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a5e255117c8b117e4f075d"), "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a5e809117c8b117e4f0763"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a5e225117c8b117e4f0758"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a5e25d117c8b117e4f075e"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a5e269117c8b117e4f0760"), "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a5e27a117c8b117e4f0762"), "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a5e270117c8b117e4f0761"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a5e264117c8b117e4f075f"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a5e243117c8b117e4f075b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a5e24c117c8b117e4f075c"), "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a5e23c117c8b117e4f075a"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
```

Практическое задание 8.2.1:

1) *Создать коллекцию towns*

```
> db.towns.insert({name: 'Punxsutawney', population: 6200, last_sensus: ISODate('2008-01-31'), famous_for: [''], mayor: {name: 'Jim Wehrle'}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: 'New York', population: 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', population: 528000, last_sensus: ISODate('2009-07-20'), famous_for: ['beer', 'food'], mayor: {name: 'Sam Adams', party: 'D'}})
WriteResult({ "nInserted" : 1 })
>
```

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

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

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

```
> db.towns.find({'mayor.party' : null}, {_id: 0, population: 0, last_sensus: 0, famous_for: 0})
{ "name" : "Punxsutawney", "mayor" : { "name" : "Jim Wehrle" } }
>
```

Практическое задание 8.2.2:

1) *Сформировать функцию для вывода списка самцов единорогов.*

```
> fc = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
```

```
> db.unicorns.find(fc)
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

2) Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке. Вывести результат, используя forEach.

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

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

Подсчитать количество особей единорогов обоих полов.

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

Практическое задание 8.2.6:

1. *Выполнить команду:*
2. *Проверить содержимое коллекции* `unicorns`.

```
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

Практическое задание 8.2.7:

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

```
> db.unicorns.update({name: 'Ayna'}, {name: 'Ayna', weight: 800, vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "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'}, {$set: {loves: 'RedBull'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : "RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "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}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : "RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

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: 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("62a614d798ee6fd6c9fca215"), "name" : "Punxsutawney", "population" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z")
, "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("62a6156f98ee6fd6c9fca216"), "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("62a615d298ee6fd6c9fca217"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"),
"famous_for" : [ "beer", "food" ] }
>
```

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("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : "RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

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("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : "RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

Практическое задание 8.2.13:

1) *Удалить документы с беспартийными мэрами.*
2) *Проверить содержание коллекции.*
3) *Очистить коллекцию.*
4) *Просмотреть список доступных коллекций.*

```
> db.towns.remove({'mayor.party': null}, true)
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("62a6156f98ee6fd6c9fca216"), "name" : "New York", "population" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of libert
y", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62a615d298ee6fd6c9fca217"), "name" : "Portland", "population" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ] }
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> db.stats()
{
        "db" : "test",
        "collections" : 2,
        "views" : 0,
        "objects" : 13,
        "avgObjSize" : 121,
        "dataSize" : 1573,
        "storageSize" : 73728,
        "indexes" : 2,
        "indexSize" : 73728,
        "totalSize" : 147456,
        "scaleFactor" : 1,
        "fsUsedSize" : 319276867584,
        "fsTotalSize" : 511443988480,
        "ok" : 1
}
>
```

Практическое задание 8.3.1:

1) *Создать коллекцию зон обитания единорогов, указать в качестве идентификатора кратко название зоны, далее включив полное название и описание.*

```
> db.home.insert({_id: 'fo', name: 'forest'})
WriteResult({ "nInserted" : 1 })
> db.home.insert({_id: 'fi', name: 'field'})
WriteResult({ "nInserted" : 1 })
>
```

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

```
> db.unicorns.update({name: 'Aurora'}, {$set: {habitat:{$ref:'home', $id: 'fo'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Horny'}, {$set: {habitat:{$ref:'home', $id: 'fi'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Ayna'}, {$set: {habitat:{$ref:'home', $id: 'fo'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Kenny'}, {$set: {habitat:{$ref:'home', $id: 'fi'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

3) Проверьте содержание коллекции единорогов.

```
> db.unicorns.find()
{ "_id" : ObjectId("62a618e798ee6fd6c9fca218"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "habitat" : DBRef("home", "fi") }
{ "_id" : ObjectId("62a618ef98ee6fd6c9fca219"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "habitat" : DBRef("home", "fo") }
{ "_id" : ObjectId("62a618f598ee6fd6c9fca21a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62a618fd98ee6fd6c9fca21b"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("62a6190698ee6fd6c9fca21c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("62a6190c98ee6fd6c9fca21d"), "name" : "Ayna", "weight" : 800, "vampires" : 51, "habitat" : DBRef("home", "fo") }
{ "_id" : ObjectId("62a6191298ee6fd6c9fca21e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39, "habitat" : DBRef("home", "fi") }
{ "_id" : ObjectId("62a6191998ee6fd6c9fca21f"), "name" : "Raleigh", "loves" : "RedBull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62a6191f98ee6fd6c9fca220"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("62a6192598ee6fd6c9fca221"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("62a6192b98ee6fd6c9fca222"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62a6199698ee6fd6c9fca223"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62a6229998ee6fd6c9fca224"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

Практическое задание 8.3.2:

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

```
> db.unicorns.ensureIndex({'name': 1}, {'unique': true});
uncaught exception: TypeError: db.unicorns.ensureIndex is not a function :
@(shell):1:1
>
```

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

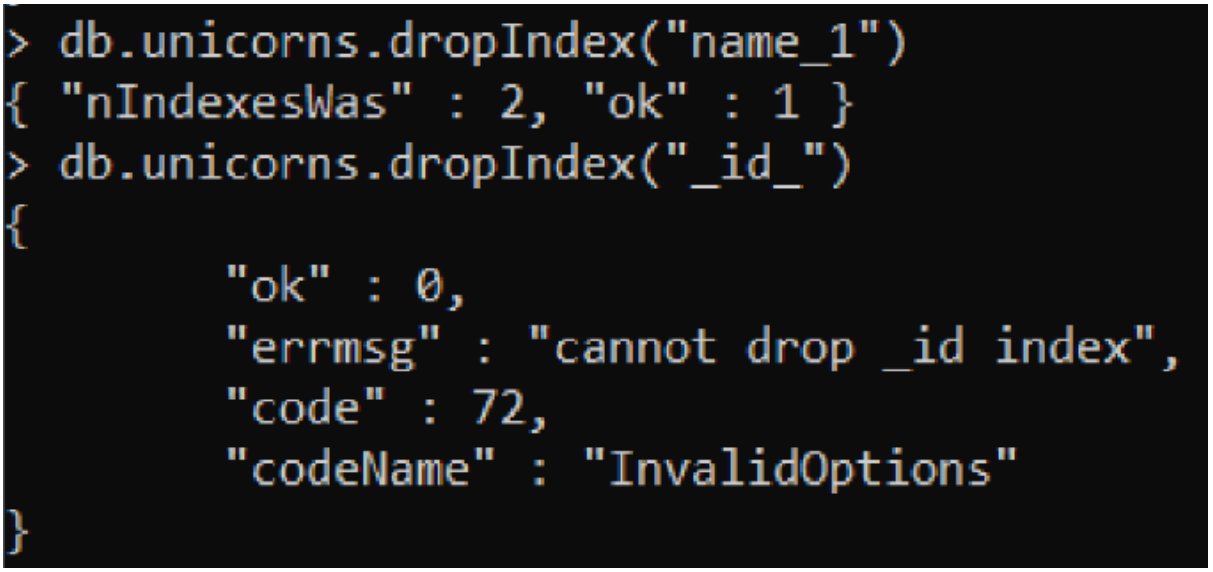
Практическое задание 8.3.3:

1) *Получите информацию о всех индексах коллекции* `unicorns` .

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

2) Удалите все индексы, кроме индекса для идентификатора.

3) Попытайтесь удалить индекс для идентификатора.



Практическое 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.*

```
> var cursor = db.numbers.find(); null;
null
```

```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
> var cursor = db.numbers.find(); null;
null
> db.numbers.find()
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb0"), "value" : 0 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb1"), "value" : 1 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb2"), "value" : 2 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb3"), "value" : 3 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb4"), "value" : 4 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb5"), "value" : 5 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb6"), "value" : 6 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb7"), "value" : 7 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb8"), "value" : 8 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcb9"), "value" : 9 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcba"), "value" : 10 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcbb"), "value" : 11 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcbc"), "value" : 12 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcbd"), "value" : 13 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcbe"), "value" : 14 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcbf"), "value" : 15 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcc0"), "value" : 16 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcc1"), "value" : 17 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcc2"), "value" : 18 }
{ "_id" : ObjectId("62a6f8b92c8ab1bf2059bcc3"), "value" : 19 }
Type "it" for more
>
```

```
> db.numbers.find().sort({value: -1}).limit(4)
{ "_id" : ObjectId("62a6f91d2c8ab1bf205b434f"), "value" : 99999 }
{ "_id" : ObjectId("62a6f91d2c8ab1bf205b434e"), "value" : 99998 }
{ "_id" : ObjectId("62a6f91d2c8ab1bf205b434d"), "value" : 99997 }
{ "_id" : ObjectId("62a6f91d2c8ab1bf205b434c"), "value" : 99996 }
>
```

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

```
> db.numbers.explain("executionStats").find().sort({value: -1}).limit(4)
{
```

```
    "executionTimeMillis" : 165,
```

5) Создайте индекс для ключа `value`*.*

Получите информацию о всех индексах коллекции `numbres`.

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

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

```
> db.numbers.explain("executionStats").find().sort({value: -1}).limit(4)
```

```
  "executionTimeMillis" : 2,
```

Вывод: с индексом запрос выполнился более эффективно - 163 миллисекунды быстрее.