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: «__» ____ 2021 г.
Оценка ___ _____

Выполнил:
студентка группы К3243

Варгина А.В.

Санкт-Петербург 2021 г.

ЦЕЛЬ РАБОТЫ

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

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Практическое задание 8.1.1:

1) *Создайте базу данных learn.*
2) *Заполните коллекцию единорогов unicorns:*

```
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
WriteResult({ "nInserted" : 1 })
>
> db.unicorns.insert({name: 'Aurora', loves: ['carrot','grape'], weight: 450, gender: 'f', vampires: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
>
```

3) *Используя второй способ, вставьте в коллекцию единорогов документ:*

```
> doc = {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(doc)
```

4) *Проверьте содержимое коллекции с помощью метода find.*

```
> db.unicorns.find()
{ "_id" : ObjectId("60d5c2fa4ed984349829f2aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5c3614ed984349829f2ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5c36e4ed984349829f2ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5c37c4ed984349829f2ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5c38b4ed984349829f2ae"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60d5c3954ed984349829f2af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60d5c3a84ed984349829f2b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5c3b04ed984349829f2b1"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "w
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5c3ee4ed984349829f2b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60d5c4244ed984349829f2b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5c4324ed984349829f2b4"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60d5c5254ed984349829f2b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.2:

1) Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.

```
> db.unicorns.find({gender:'m'}).sort({name:1})
{ "_id" : ObjectId("60d5c5254ed984349829f2b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60d5c2fa4ed984349829f2aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5c3a84ed984349829f2b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5c4244ed984349829f2b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5c3b04ed984349829f2b1"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "w
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5c37c4ed984349829f2ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5c36e4ed984349829f2ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
```

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("60d5c3614ed984349829f2ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5c3954ed984349829f2af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60d5c3ee4ed984349829f2b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
```

```
> db.unicorns.findOne({gender:'f', loves: 'carrot'})
{
        "_id" : ObjectId("60d5c3614ed984349829f2ab"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({gender:'m'}, {loves: 0, gender: 0}).sort({name:1})
{ "_id" : ObjectId("60d5c5254ed984349829f2b5"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("60d5c2fa4ed984349829f2aa"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("60d5c3a84ed984349829f2b0"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("60d5c4244ed984349829f2b3"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("60d5c3b04ed984349829f2b1"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("60d5c37c4ed984349829f2ad"), "name" : "Roooooodles", "weight" : 575, "vampires" : 9
9 }
{ "_id" : ObjectId("60d5c36e4ed984349829f2ac"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find({gender: 'm'}).sort({$natural: -1})
{ "_id" : ObjectId("60d5c5254ed984349829f2b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60d5c4244ed984349829f2b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5c3b04ed984349829f2b1"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "w
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5c3a84ed984349829f2b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5c37c4ed984349829f2ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5c36e4ed984349829f2ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5c2fa4ed984349829f2aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

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

```
> db.unicorns.find({gender: 'm'}, {loves: {$slice: 1}, _id: 0})
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "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", "v
ampires" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 3
3 }
{ "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: true}})
{ "_id" : ObjectId("60d5c2fa4ed984349829f2aa"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5c3614ed984349829f2ab"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5c36e4ed984349829f2ac"), "name" : "Unicrom", "loves" : [ "energon", "redbull"
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5c37c4ed984349829f2ad"), "name" : "Roooooodles", "loves" : [ "apple" ], "weigh
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5c38b4ed984349829f2ae"), "name" : "Solnara", "loves" : [ "apple", "carrot", "c
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60d5c3954ed984349829f2af"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60d5c3a84ed984349829f2b0"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "we
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5c3b04ed984349829f2b1"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5c3ee4ed984349829f2b2"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60d5c4244ed984349829f2b3"), "name" : "Pilot", "loves" : [ "apple", "watermelon"
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5c5254ed984349829f2b5"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.9:

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

```
> db.unicorns.find({}, {loves: {$slice:1}, name:1, loves: 1, _id:0}).sort({name:1})
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ] }
{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ] }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ] }
{ "name" : "Horny", "loves" : [ "carrot", "papaya" ] }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ] }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ] }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ] }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ] }
{ "name" : "Raleigh", "loves" : [ "apple", "sugar" ] }
{ "name" : "Roooooodles", "loves" : [ "apple" ] }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ] }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ] }
```

Практическое задание 8.2.1:

1) Создайте коллекцию towns, включающую следующие документы:

```
{name: "Punxsutawney ",
populatiuon: 6200,
last_sensus: ISODate("2008-01-31"),
famous_for: [""],
mayor: {
   name: "Jim Wehrle"
   }}

{name: "New York",
populatiuon: 22200000,
last_sensus: ISODate("2009-07-31"),
famous_for: ["status of liberty", "food"],
mayor: {
   name: "Michael Bloomberg",
   party: "I"}}

{name: "Portland",
populatiuon: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {
   name: "Sam Adams",
   party: "D"}}
```

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

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

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

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

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

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.*

```
> f = function() {return this.gender == 'm'}
function() {return this.gender == 'm'}
> var cursor = db.unicorns.find(f); null;
null
> cursor.limit(2).sort({name:1}); 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:

Посчитать 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:*

```
> 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("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "w
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "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'], weight: 800, gende
r: 'f', vampires:51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "w
eight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "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("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "Redbull" ], "weight"
: 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "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({}, {$inc: {vampires: 5}}, {multi:true})
WriteResult({ "nMatched" : 13, "nUpserted" : 0, "nModified" : 13 })
> db.unicorns.find()
{ "_id" : ObjectId("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 56 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "Redbull" ], "weight"
: 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ],
 "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "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`.

```
> use towns
switched to db towns
> db.towns.update({name: 'Portland'}, {$set: {"mayor.party": 'I'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("60d5ca434ed984349829f2b6"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_
sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" }
}
{ "_id" : ObjectId("60d5ca644ed984349829f2b7"), "name" : "New York", "populatiuon" : 22200000, "last_s
ensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : {
"name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("60d5ca7c4ed984349829f2b8"), "name" : "Portland", "populatiuon" : 528000, "last_sen
sus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam A
dams", "party" : "I" } }
```

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("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "w
eight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 56 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "Redbull" ], "weight"
: 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "
chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "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("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sug
ar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 56 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "Redbull" ], "weight"
: 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "
chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "gender" : "m", "vampires" : 5 }
>
```

Практическое задание 8.2.13:

1) *Создайте коллекцию towns, включающую следующие документы:*

```
{name: "Punxsutawney ",
popujatiuon: 6200,
last_sensus: ISODate("2008-01-31"),
famous_for: ["phil the groundhog"],
mayor: {
   name: "Jim Wehrle"
   }}

{name: "New York",
popujatiuon: 22200000,
last_sensus: ISODate("2009-07-31"),
famous_for: ["status of liberty", "food"],
mayor: {
   name: "Michael Bloomberg",
   party: "I"}}

{name: "Portland",
popujatiuon: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {
   name: "Sam Adams",
   party: "D"}}
```

2) *Удалите документы с беспартийными мэрами.*
3) *Проверьте содержание коллекции.*
4) *Очистите коллекцию.*
5) *Просмотрите список доступных коллекций.*

```
> use towns
switched to db towns
> db.towns.remove({"mayor.party": "I"}, true)
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("60d5ca434ed984349829f2b6"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_
sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" }
}
{ "_id" : ObjectId("60d5ca7c4ed984349829f2b8"), "name" : "Portland", "populatiuon" : 528000, "last_sen
sus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam A
dams", "party" : "I" } }
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> db.rowns.find()
> show collections
towns
>
```

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, использую второй способ автоматического связывания.*
3) *Проверьте содержание коллекции едиорогов.*

```
> db.createCollection('habitats')
{ "ok" : 1 }
> show collections
habitats
unicorns
> db.habitats.insert({_id: 'fst', name: 'forest', descriptions: 'big pinewood forest'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'ild', name: 'island', descriptions: 'small isolated island in the sea'})
WriteResult({ "nInserted" : 1 })
> db.habitats.insert({_id: 'rvr', name: 'river', descriptions: 'small river flowing through'})
WriteResult({ "nInserted" : 1 })
> db.habitats.find()
{ "_id" : "fst", "name" : "forest", "descriptions" : "big pinewood forest" }
{ "_id" : "ild", "name" : "island", "descriptions" : "small isolated island in the sea" }
{ "_id" : "rvr", "name" : "river", "descriptions" : "small river flowing through" }
> db.unicorns.update({name: 'Dunx'}, {$set: {habitat: {$ref: 'habitats', $id: 'fst'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Raleigh'}, {$set: {habitat: {$ref: 'habitats', $id: 'ild'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Pilot'}, {$set: {habitat: {$ref: 'habitats', $id: 'rvr'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60d5ce9a4ed984349829f2b9"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "w
eight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60d5ce9d4ed984349829f2ba"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sug
ar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60d5ce9f4ed984349829f2bb"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ]
, "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60d5cea04ed984349829f2bc"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight
" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60d5cea14ed984349829f2bd"), "name" : "Solnara", "loves" : [ "apple", "carrot", "ch
ocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60d5cea14ed984349829f2be"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ],
"weight" : 800, "gender" : "f", "vampires" : 56 }
{ "_id" : ObjectId("60d5cea24ed984349829f2bf"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "wei
ght" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60d5cea34ed984349829f2c0"), "name" : "Raleigh", "loves" : [ "Redbull" ], "weight"
: 421, "gender" : "m", "vampires" : 7, "habitat" : DBRef("habitats", "ild") }
{ "_id" : ObjectId("60d5cea44ed984349829f2c1"), "name" : "Leia", "loves" : [ "apple", "watermelon" ],
"weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60d5cea54ed984349829f2c2"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "
chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59, "habitat" : DBRef("habitats", "rvr") }
{ "_id" : ObjectId("60d5cea64ed984349829f2c3"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "we
ight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60d5ceaa4ed984349829f2c4"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ],
"weight" : 704, "gender" : "m", "vampires" : 170, "habitat" : DBRef("habitats", "fst") }
{ "_id" : ObjectId("60d5d0294ed984349829f2c5"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 34
0, "gender" : "m", "vampires" : 5 }
```

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})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "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,
                "unique" : true,
                "key" : {
                        "name" : 1
                },
                "name" : "name_1"
        }
]
```

2) *Удалите все индексы, кроме индекса для идентификатора.*

```
> db.unicorns.dropIndex('name_1')
{ "nIndexesWas" : 2, "ok" : 1 }
```

3) *Попытайтесь удалить индекс для идентификатора*

```
> 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:*

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

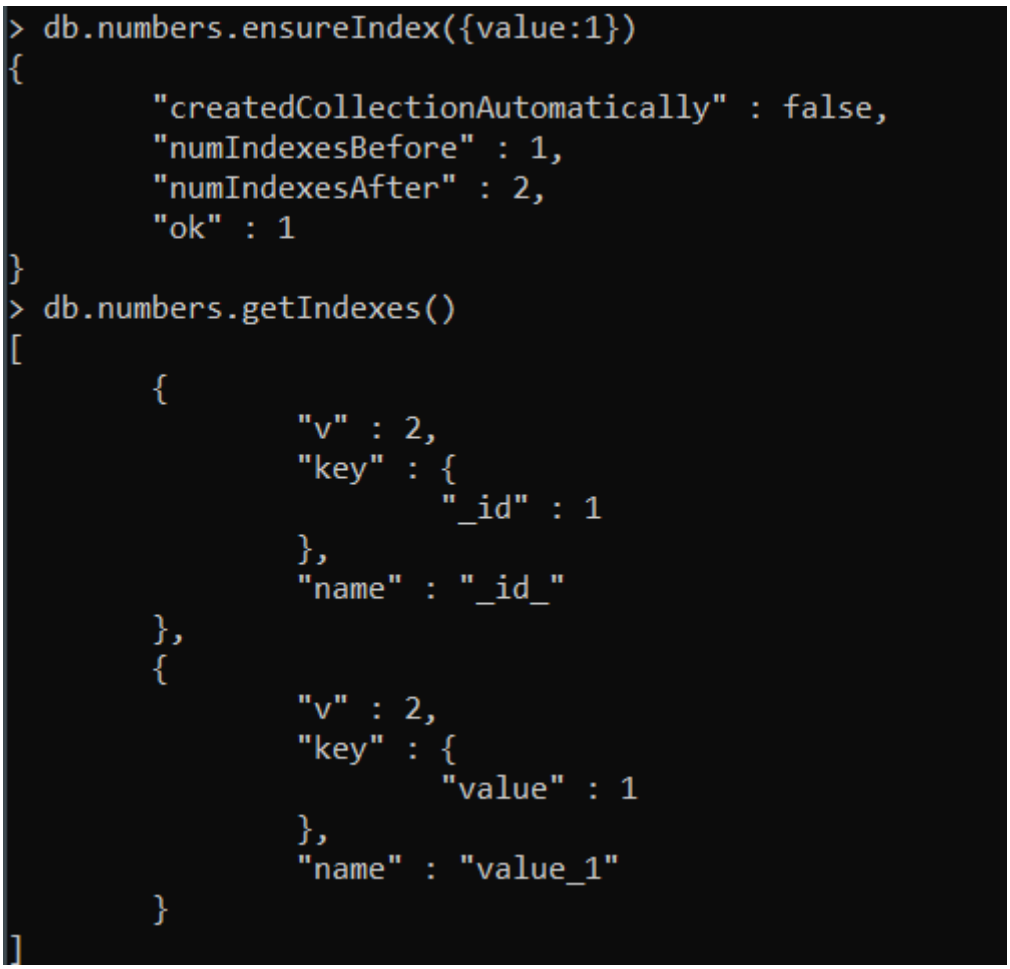
```
> use numbers
switched to db numbers
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
> db.numbers.find().limit(10)
{ "_id" : ObjectId("60d5d7794ed984349829f2c6"), "value" : 0 }
{ "_id" : ObjectId("60d5d7794ed984349829f2c7"), "value" : 1 }
{ "_id" : ObjectId("60d5d7794ed984349829f2c8"), "value" : 2 }
{ "_id" : ObjectId("60d5d7794ed984349829f2c9"), "value" : 3 }
{ "_id" : ObjectId("60d5d7794ed984349829f2ca"), "value" : 4 }
{ "_id" : ObjectId("60d5d7794ed984349829f2cb"), "value" : 5 }
{ "_id" : ObjectId("60d5d7794ed984349829f2cc"), "value" : 6 }
{ "_id" : ObjectId("60d5d7794ed984349829f2cd"), "value" : 7 }
{ "_id" : ObjectId("60d5d7794ed984349829f2ce"), "value" : 8 }
{ "_id" : ObjectId("60d5d7794ed984349829f2cf"), "value" : 9 }
```

2) *Выберите последних четыре документа.*

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

                },
                "winningPlan" : {
                        "stage" : "LIMIT",
                        "limitAmount" : 4,
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "direction" : "backward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 4,
                "executionTimeMillis" : 6,
                "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,
```

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?*


> 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" : "numbers.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" : {
```

Изначально время выполнения команды занимало 6 миллисекунд. Индексирование сократило время выполнения с 6 мс до 0 мс, из этого следует, что запрос с индексом более эффективен.

ВЫВОДЫ

MongoDB предоставляет мощный CLI интерфейс для выполнения CRUD операций, отличительной особенностью является интеграция полноценного языка программирования: Javascript.