Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе № 8 «Работа с БД в СУБД MongoDB»

по дисциплине «**Проектирование и реализация баз данных**»

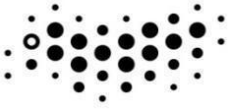
Автор: Власов М. И.

Факультет: ИКТ

Группа: K3240

Преподаватель: Говорова М. М.

Дата: 07.06.21



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2021

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

Практическое задание 8.1.1:

1. *Создайте базу данных learn.*

```
> db.createCollection("learn")
{ "ok" : 1 }
```

2. *Заполните коллекцию единорогов unicorns:*

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

```
> 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, g
```

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

```
{name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
```

```
> info = {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(info)
WriteResult({ "nInserted" : 1 })
>
```

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

```
> db.unicorns.find()
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot", "papaya"
], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon", "redbu
ll" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ], "w
eight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9a"), "name" : "Solnara", "loves" : [ "apple", "carrot"
, "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9b"), "name" : "Ayna", "loves" : [ "strawberry", "lemon
" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon" ],
 "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : [ "apple", "sugar"
], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermelon
" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "watermelo
n" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ]
, "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be16b1b10022a21d91cba1"), "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: "f"}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9b"), "name" : "Ayna", "loves" : [ "strawberry", "lemon
" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermelon
" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
> db.unicorns.find({gender: "m"}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape", "watermelon
" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot", "papaya"
], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon" ],
 "weight" : 690, "gender" : "m", "vampires" : 39 }
```

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

```
> db.unicorns.find({gender: "f", loves: "carrot"}).limit(1)
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.findOne({gender: "f", loves: "carrot"})
{
        "_id" : ObjectId("60be1684b10022a21d91cb97"),
        "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, _id: 0})
{ "name" : "Horny", "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Unicrom", "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Kenny", "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Pilot", "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Dunx", "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape", "watermelon
" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ]
, "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "watermelo
n" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermelon
" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : [ "apple", "sugar"
], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon" ],
 "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9b"), "name" : "Ayna", "loves" : [ "strawberry", "lemon
" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9a"), "name" : "Solnara", "loves" : [ "apple", "carrot"
, "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ], "w
eight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon", "redbu
ll" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape"
], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "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, $lt: 700}}, {_id: 0})
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f
", "vampires" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires
" : 33 }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

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: 0}})
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "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"}, {loves : {$slice: 1}}).sort({name: 1})
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape" ], "weight"
 : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot" ], "weigh
t" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape" ], "weight
" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple" ], "weight
" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : [ "apple" ], "weig
ht" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ], "
weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon" ], "we
ight" : 984, "gender" : "m", "vampires" : 182 }
```

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"}}
```

```
> 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: 0}}, {_id: 0, name: 1, mayor: 1})
{ "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

<u>Практическое задание 8.2.2:</u>

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

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

2. *Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.*

```
> var cursor = db.unicorns.find(male); null;
null
> cursor.sort({name: 1}).limit(2); null;
null
```

3. *Вывести результат, используя forEach.*

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

Практическое задание 8.2.6:

1. *Выполнить команду:*

```
> db.unicorns.save({name: 'Barny', loves: ['grape'],
weight: 340, gender: 'm'})
```

```
> db.unicorns.save({name: 'Barny', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
```

2. *Проверить содержимое коллекции* `unicorns`.

```
> db.unicorns.find()
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot", "papaya
" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape
" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon", "red
bull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ],
"weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9a"), "name" : "Solnara", "loves" : [ "apple", "carro
t", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9b"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon"
], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : [ "apple", "sugar
" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermel
on" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "waterme
lon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "name" : "Nimue", "loves" : [ "grape", "carrot"
 ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape", "watermel
on" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be2e68f8a5a0cfcc565ad2"), "name" : "Barny", "loves" : [ "grape" ], "weigh
t" : 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.*
2. *Проверить содержимое коллекции* `unicorns`.

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("60be2febf8a5a0cfcc565ad4"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
> db.unicorns.update({name: "Ayna"}, {name: "Ayna", loves: ["strawberry", "lemon"], weight: 800
, gender: "f", vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("60be2febf8a5a0cfcc565ad4"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

Практическое задание 8.2.8:

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

```
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : [ "apple", "sugar
" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
> db.unicorns.update({name: "Raleigh"}, {$set: {loves: "redbull"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : "redbull", "weigh
t" : 421, "gender" : "m", "vampires" : 2 }
```

Практическое задание 8.2.9:

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

```
{ "_id" : ObjectId("60be2febf8a5a0cfcc565ad4"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
> db.unicorns.update({}, {$inc: {vampires: 5}}, {multi: 1})
WriteResult({ "nMatched" : 13, "nUpserted" : 0, "nModified" : 13 })
> db.unicorns.find()
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot", "papaya
" ], "weight" : 600, "gender" : "m", "vampires" : 88 }
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape
" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon", "red
bull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ],
"weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9a"), "name" : "Solnara", "loves" : [ "apple", "carro
t", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon"
], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : "redbull", "weigh
t" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermel
on" ], "weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "waterme
lon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "name" : "Nimue", "loves" : [ "grape", "carrot"
 ], "weight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape", "watermel
on" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60be2e68f8a5a0cfcc565ad2"), "name" : "Barny", "loves" : [ "grape" ], "weigh
t" : 340, "gender" : "m", "vampires" : 5 }
{ "_id" : ObjectId("60be2febf8a5a0cfcc565ad4"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 800, "gender" : "f", "vampires" : 56 }
```

Практическое задание 8.2.10:

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

```
> db.towns.find()
{ "_id" : ObjectId("60be2297f8a5a0cfcc565acf"), "name" : "Punxsutawney ", "populatiuon" : 6200,
 "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "
Jim Wehrle" } }
{ "_id" : ObjectId("60be22a4f8a5a0cfcc565ad0"), "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("60be22abf8a5a0cfcc565ad1"), "name" : "Portland", "populatiuon" : 528000, "l
ast_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "
name" : "Sam Adams", "party" : "D" } }
> db.towns.update({name: "Portland"}, {$unset: {"mayor.party": 1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("60be2297f8a5a0cfcc565acf"), "name" : "Punxsutawney ", "populatiuon" : 6200,
 "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "
Jim Wehrle" } }
{ "_id" : ObjectId("60be22a4f8a5a0cfcc565ad0"), "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("60be22abf8a5a0cfcc565ad1"), "name" : "Portland", "populatiuon" : 528000, "l
ast_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.find({name: "Pilot"})
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "waterme
lon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
> db.unicorns.update({name: "Pilot"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "waterme
lon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
```

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.find({name: "Aurora"})
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape
" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
> db.unicorns.update({name: "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape
", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
```

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.*

```
> db.towns.find()
{ "_id" : ObjectId("60be2297f8a5a0cfcc565acf"), "name" : "Punxsutawney ", "populatiuon" : 6200,
 "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "
Jim Wehrle" } }
{ "_id" : ObjectId("60be22a4f8a5a0cfcc565ad0"), "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("60be22abf8a5a0cfcc565ad1"), "name" : "Portland", "populatiuon" : 528000, "l
ast_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "
name" : "Sam Adams" } }
> db.towns.remove({"mayor.party": {$exists: 0}})
WriteResult({ "nRemoved" : 2 })
> db.towns.find()
{ "_id" : ObjectId("60be22a4f8a5a0cfcc565ad0"), "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" } }
>
```

4. *Очистите коллекцию.*
5. *Просмотрите список доступных коллекций.*

```
> db.towns.remove({})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
> show collections
learn
towns
unicorns
```

<u>Практическое задание 8.3.1:</u>

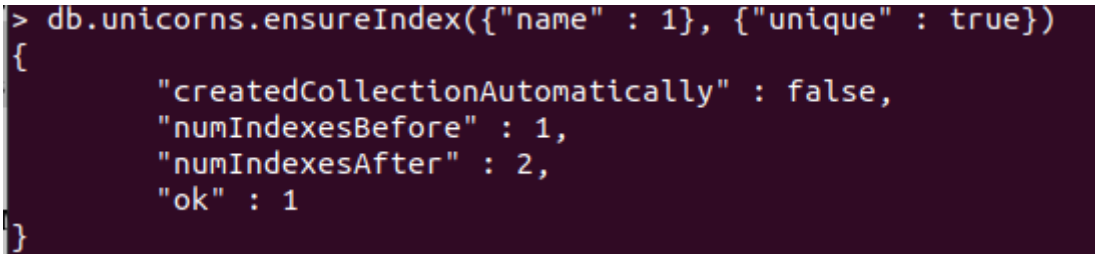
1. *Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.*

```
> db.zones.insert({_id: "zone1", name: "first zone", description: "we are the first"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: "zone2", name: "zone2", description: "we are better than the first"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: "zone3", name: "zo3ne", description: "you will be safe here"})
WriteResult({ "nInserted" : 1 })
```

2. *Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.*
3. *Проверьте содержание коллекции едиорогов.*

<u>Практическое задание 8.3.2:</u>

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



2. *Содержание коллекции единорогов* `unicorns`*:*

```
> db.unicorns.find()
{ "_id" : ObjectId("60be1684b10022a21d91cb96"), "name" : "Horny", "loves" : [ "carrot", "papaya
" ], "weight" : 600, "gender" : "m", "vampires" : 88 }
{ "_id" : ObjectId("60be1684b10022a21d91cb97"), "name" : "Aurora", "loves" : [ "carrot", "grape
", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 48 }
{ "_id" : ObjectId("60be1684b10022a21d91cb98"), "name" : "Unicrom", "loves" : [ "energon", "red
bull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be1684b10022a21d91cb99"), "name" : "Roooooodles", "loves" : [ "apple" ],
"weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9a"), "name" : "Solnara", "loves" : [ "apple", "carro
t", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 85 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9c"), "name" : "Kenny", "loves" : [ "grape", "lemon"
], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9d"), "name" : "Raleigh", "loves" : "redbull", "weigh
t" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9e"), "name" : "Leia", "loves" : [ "apple", "watermel
on" ], "weight" : 601, "gender" : "f", "vampires" : 38 }
{ "_id" : ObjectId("60be1684b10022a21d91cb9f"), "name" : "Pilot", "loves" : [ "apple", "waterme
lon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be1684b10022a21d91cba0"), "name" : "Nimue", "loves" : [ "grape", "carrot"
 ], "weight" : 540, "gender" : "f", "vampires" : 5 }
{ "_id" : ObjectId("60be1773b10022a21d91cba3"), "name" : "Dunx", "loves" : [ "grape", "watermel
on" ], "weight" : 704, "gender" : "m", "vampires" : 170, "zone" : DBRef("zones", "zone3") }
{ "_id" : ObjectId("60be2e68f8a5a0cfcc565ad2"), "name" : "Barny", "loves" : [ "grape" ], "weigh
t" : 340, "gender" : "m", "vampires" : 5, "zone" : DBRef("zones", "zone2") }
{ "_id" : ObjectId("60be2febf8a5a0cfcc565ad4"), "name" : "Ayna", "loves" : [ "strawberry", "lem
on" ], "weight" : 800, "gender" : "f", "vampires" : 56, "zone" : DBRef("zones", "zone1") }
>
```

<u>Практическое задание 8.3.3:</u>

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.dropIndexes("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
```

3. *Попытайтесь удалить индекс для идентификатора.*

```
> db.unicorns.dropIndexes("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.dropIndexes("_id_")
uncaught exception: Error: error dropping indexes : {
        "ok" : 0,
        "errmsg" : "cannot drop _id index",
        "code" : 72,
        "codeName" : "InvalidOptions"
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DBCollection.prototype.dropIndexes@src/mongo/shell/collection.js:704:11
@(shell):1:1
```

Практическое задание 8.3.4:

1. *Создайте объемную коллекцию* `numbers`*, задействовав курсор:*

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

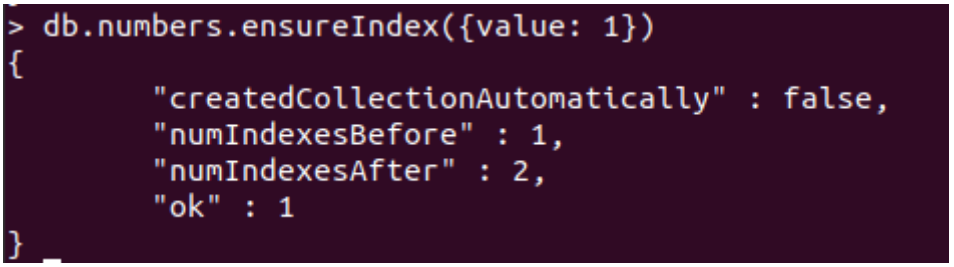
```
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

WriteResult({ "nInserted" : 1 })
>
> db.numbers.find()
{ "_id" : ObjectId("60be3d2673c33e667304056a"), "value" : 0 }
{ "_id" : ObjectId("60be3d2673c33e667304056b"), "value" : 1 }
{ "_id" : ObjectId("60be3d2673c33e667304056c"), "value" : 2 }
{ "_id" : ObjectId("60be3d2673c33e667304056d"), "value" : 3 }
{ "_id" : ObjectId("60be3d2673c33e667304056e"), "value" : 4 }
{ "_id" : ObjectId("60be3d2673c33e667304056f"), "value" : 5 }
{ "_id" : ObjectId("60be3d2673c33e6673040570"), "value" : 6 }
{ "_id" : ObjectId("60be3d2673c33e6673040571"), "value" : 7 }
{ "_id" : ObjectId("60be3d2673c33e6673040572"), "value" : 8 }
{ "_id" : ObjectId("60be3d2673c33e6673040573"), "value" : 9 }
{ "_id" : ObjectId("60be3d2673c33e6673040574"), "value" : 10 }
{ "_id" : ObjectId("60be3d2673c33e6673040575"), "value" : 11 }
{ "_id" : ObjectId("60be3d2673c33e6673040576"), "value" : 12 }
{ "_id" : ObjectId("60be3d2673c33e6673040577"), "value" : 13 }
{ "_id" : ObjectId("60be3d2673c33e6673040578"), "value" : 14 }
{ "_id" : ObjectId("60be3d2673c33e6673040579"), "value" : 15 }
{ "_id" : ObjectId("60be3d2673c33e667304057a"), "value" : 16 }
{ "_id" : ObjectId("60be3d2673c33e667304057b"), "value" : 17 }
{ "_id" : ObjectId("60be3d2673c33e667304057c"), "value" : 18 }
{ "_id" : ObjectId("60be3d2673c33e667304057d"), "value" : 19 }
```

2. *Выберите последних четыре документа.*

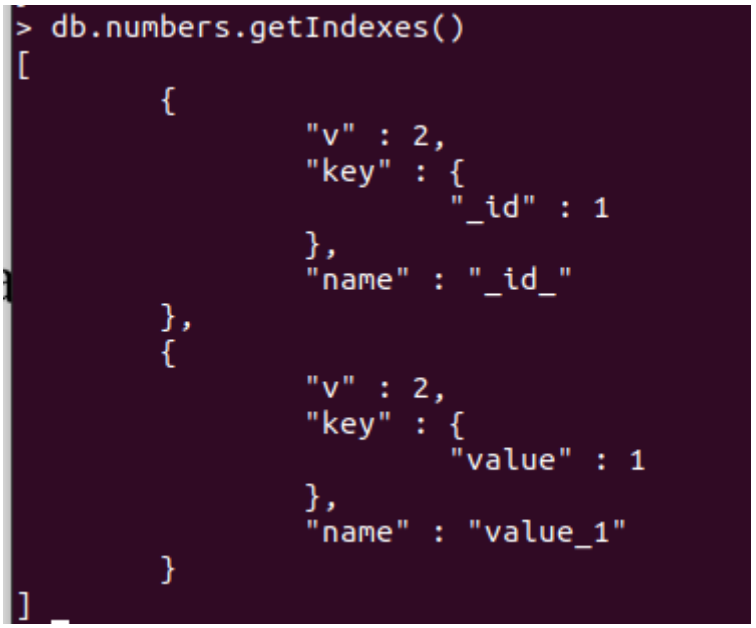
```
> last_four = db.numbers.find().sort({value: -1}).limit(4)
{ "_id" : ObjectId("60be3d5173c33e6673058c09"), "value" : 99999 }
{ "_id" : ObjectId("60be3d5173c33e6673058c08"), "value" : 99998 }
{ "_id" : ObjectId("60be3d5173c33e6673058c07"), "value" : 99997 }
{ "_id" : ObjectId("60be3d5173c33e6673058c06"), "value" : 99996 }
> last_four.explain("executionStats")
```

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* `numbers`.



6. *Выполните запрос 2.*

7. *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*

Около 0 мс

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

Запрос с индексом эффективнее

Вывод: в ходе лабораторной работы мы овладели 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.