

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №5.2

«Работа с БД в СУБД MongoDB»

по дисциплине «Проектирование и реализация баз данных»

Автор:

Петросян Анна Мнацакановна

Факультет:

Инфокоммуникационных технологий

Группа:

К3242

Преподаватель: Говорова Марина Михайловна



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2022

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

Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 5.0.8.

Оглавление:

[Вставка документов в коллекцию](#)

[Выборка данных из БД](#)

[Логические операторы](#)

[Запрос к вложенным объектам](#)

[Курсоры](#)

[Агрегированные запросы](#)

[Редактирование данных](#)

[Удаление данных из коллекции](#)

[Ссылки в бд](#)

[Настройка индексов](#)

[Управление индексами](#)

[План запроса](#)

Практическое задание и выполнение:

Вставка документов в коллекцию

1) *Создайте базу данных learn.*

```
> use learn  
switched to db learn
```

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'});

```

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

```

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

```

4) Проверьте содержимое коллекции с помощью метода *find*. (часть вывода)

```

> db.unicorns.find()
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad29"), "name" : "Aurora", "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2a"), "name" : "Unicorn", "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2b"), "name" : "Raleigh", "gender" : "m", "vampires" : 99 }

```

Выборка данных из БД

5) 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)

```

> db.unicorns.find({gender:'f'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad29"), "name" : "Aurora", "loves" : [ "carrot", "lemon" ], "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad30"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "gender" : "f", "vampires" : 33 }

```

db.unicorns.find({gender:'m'}).sort({name: 1})

```

> db.unicorns.find({gender:'m'}).sort({name: 1})
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "loves" : [ "lemon", "sugar" ], "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad31"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "gender" : "m", "vampires" : 2 }

```

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

```
db.unicorns.findOne({gender: 'f', loves: 'carrot'})
```

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad29"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
```

```
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
```

```
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad29"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
```

7) 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});
```

```
> db.unicorns.find({gender: "m"}, {loves: 0, gender: 0}).sort({name: 1});
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2e"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad31"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2f"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2b"), "name" : "Rooooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2a"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
>
```

8) Вывести список единорогов в обратном порядке добавления.

```
db.unicorns.find().sort({$natural: -1});
```

сравнение

```
> db.unicorns.find().sort({$natural: -1});
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad32"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad31"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad30"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2d"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2c"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2b"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad29"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

```
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: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
```

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

```
db.unicorns.find( { }, { loves: { $slice: 1 }, _id: 0 } );
```

```
> db.unicorns.find( { }, { loves: { $slice: 1 }, _id: 0 } );
{ "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" : "Rooooooodles", "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" }
```

Логические операторы

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

```
db.unicorns.find( { gender: "f", weight: { $gt: 500, $lt: 700 } }, { _id: 0 } );
```

```
> db.unicorns.find( { gender: "f", weight: { $gt: 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" }
>
```

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

```
db.unicorns.find( { gender: 'm', weight: { $gt: 500 }, loves: { $all: ['grape', 'lemon'] } }, { _id: 0 } );
```

```
> db.unicorns.find( { gender: 'm', weight: { $gt: 500 }, loves: { $all: ['grape', 'lemon'] } }, { _id: 0 } );
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
```

12) Найти всех единорогов, не имеющих ключ vampires.

```
db.unicorns.find( { vampires: { $exists: false } } );
```

```
> db.unicorns.find( { vampires: { $exists: false } } );
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad32"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

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

```
db.unicorns.find( { gender: 'm' }, { loves: { $slice: 1 } }).sort( { name: 1 } );
```

```
> db.unicorns.find( { vampires: { $exists: false } } );
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad32"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
> db.unicorns.find( { gender: 'm' }, { loves: { $slice: 1 } }).sort( { name: 1 } );
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2e"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad31"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2f"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2b"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2a"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

Запрос к вложенным объектам

14) Создайте коллекцию towns, включающую следующие документы:

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

```
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"}})
```

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

```
> db.towns.find()
{ "_id" : ObjectId("6286c607c82f8c0c9fa9ad33"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6286c62ac82f8c0c9fa9ad34"), "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("6286c647c82f8c0c9fa9ad35"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

15) 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"}, {"name":1, "mayor":1, "_id":0});`

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

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

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

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

Курсоры

17) Сформировать функцию для вывода списка самцов единорогов.

```
fn = function() {return this.gender=="m"}
db.unicorns.find(fn)
```

```
> fn = function() {return this.gender=="m"}
function() {return this.gender=="m"}
> db.unicorns.find(fn)
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad28"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2a"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2b"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2e"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad2f"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6286a3dcc82f8c0c9fa9ad31"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
>
```

18) Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке. 19) Вывести результат, используя `forEach`.

```
var cursor = db.unicorns.find({gender: "m"});null;
```

```
null
```

```
> cursor.sort({name: 1}).limit(2);null;
```

```
null
```

```
> cursor.forEach(function(fn){ print (fn.name); })
```

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

Агрегированные запросы

21) Вывести количество самок единорогов весом от полутонны до 600 кг.

```
db.unicorns.find({gender: 'f', weight:{$gt: 500, $lt: 600}}).count()
```

```
> db.unicorns.find({gender: 'f', weight:{$gt: 500, $lt: 600}}).count()
2
```

22) Вывести список предпочтений.

```
db.unicorns.distinct("loves")
```

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


23) Посчитать количество особей единорогов обоих полов. с учётом Barney из следующих заданий

```
db.unicorns.aggregate({"$group": {_id:"$gender", count:{$sum:1}}})
```

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

Редактирование данных

24) Выполнить команду:

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

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

25) Проверить содержимое коллекции unicorns.

db.unicorns.find({name: "Barney"}) - Barney есть

```
> db.unicorns.find({name: "Barney"})
{ "_id" : ObjectId("6287f1c2af33bf760f5a4af8"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

26) Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вампира.

```
db.unicorns.update({name: "Ayna", gender: "f"}, {weight: 800, vampires: 51})
```

```
> db.unicorns.update({name: "Ayna"}, {name: "Ayna", loves: ['strawberry', 'lemon'], weight: 800, gender: 'f', vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

27) Проверить содержимое коллекции unicorns.

```
db.unicorns.find({name: "Ayna"})
```

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("6287fa7daf33bf760f5a4b12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

28) Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.

```
db.unicorns.update({name: "Raleigh", gender: "m"}, {$set: {loves: "redbull"}})
```

```
> db.unicorns.update({name: "Raleigh", gender: "m"}, {$set: {loves: "redbull"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

29) Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Raleigh"})
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b0c"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
```

30) Всем самцам единорогов увеличить количество убитых вампиров на 5.

```
db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}})
```

```
> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```


31) Проверить содержимое коллекции `unicorns`.

`db.unicorns.find({gender: "m"})`

```
> db.unicorns.find({gender: "m"})
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b06"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b08"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b0b"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b0c"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b0e"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b10"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6287f7d6af33bf760f5a4b11"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

32) Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

`db.towns.update({name: "Portland"}, {$unset: {"mayor.party": "D"}})`

```
> db.towns.update({name: "Portland"}, {$unset: {"mayor.party": "D"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

33) Проверить содержимое коллекции `towns`.

```
{ "_id" : ObjectId("628804f3af33bf760f5a4b15"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

34) Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.

`db.unicorns.update({name: "Pilot", gender: "m"}, {$push: {loves: "chocolate"}})`

```
> db.unicorns.update({name: "Pilot", gender: "m"}, {$push: {loves: "chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

35) Проверить содержимое коллекции `unicorns`.

`db.unicorns.find({name: "Pilot"})`

```
> db.unicorns.find({name: "Pilot"})
{ "_id" : ObjectId("6287f7b5af33bf760f5a4b0e"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
```

36) Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.

`db.unicorns.update({name: "Aurora", gender: "f"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})`

```
> db.unicorns.update({name: "Aurora", gender: "f"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

37) Проверить содержимое коллекции `unicorns`.

`db.unicorns.find({name: "Aurora"})`

```
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b07"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Удаление данных из коллекции

38) Создайте коллекцию `towns`, включающую следующие документы:

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

```
}})
```

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

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

39) Удалите документы с беспартийными мэрами.

```
db.towns.remove({"mayor.party": {$exists: false}})
```

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

40) Проверьте содержание коллекции.

```
db.towns.find({"mayor.party": {$exists: false}})
```

```
> db.towns.find({"mayor.party": {$exists: false}})
>
```

41) Очистите коллекцию.

```
db.towns.remove({})
```

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

42) Просмотрите список доступных коллекций.

```
show collections
```

```
> show collections
towns
> db.towns.find()
> _
```

так как мы очистили, а не удалили коллекцию

Ссылки в бд

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

```
> db.zones.insert({_id: "yer", short: "yer", full: "Yerevan", descr: "the best city"})
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.zones.insert({_id: "prs", short: "prs", full: "Paris", descr: "the second best city"})
```

```
WriteResult({ "nInserted" : 1 })
```

```
> db.zones.find()
{ "_id" : "yer", "short" : "yer", "full" : "Yerevan", "descr" : "the best city" }
{ "_id" : "prs", "short" : "prs", "full" : "Paris", "descr" : "the second best city" }
```

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

добавила для первых трёх

```
db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref: "zones", $id: "yer"}}})
```

```
> db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref: "zones", $id: "yer"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.unicorns.find({name: "Horny"})
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b06"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "city" : DBRef("zones", "yer") }
```

```
db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref: "zones", $id: "prs"}}})
```

```
> db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref: "zones", $id: "prs"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.unicorns.find({name: "Aurora"})
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b07"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "prs") }
```

```
db.unicorns.update({name: "Unicrom"}, {$set: {"city": {$ref: "zones", $id: "yer"}}})
```

```
> db.unicorns.update({name: "Unicrom"}, {$set: {"city": {$ref: "zones", $id: "yer"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.unicorns.find({name: "Unicrom"})
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b08"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182, "city" : DBRef("zones", "yer") }
```

45) Проверьте содержание коллекции единорогов.

```
> db.unicorns.find().limit(3)
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b06"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "city" : DBRef("zones", "yer") }
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b07"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "prs") }
{ "_id" : ObjectId("6287f7b4af33bf760f5a4b08"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182, "city" : DBRef("zones", "yer") }
```

46) Содержание коллекции 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', 44), 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: 'Dunx', loves: ['grape', 'watermelon'],
weight: 704, gender: 'm', vampires: 165});

```

Настройка индексов

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

```
db.unicorns.ensureIndex({"name":1}, {"unique":true})
```

```

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

```

Нельзя создать уникальный индекс, что логично исходя из назначения `"name"`, потому могут быть единороги-тёзки

48) Содержание коллекции единорогов `unicorns`:

```

db.unicorns.insert({name: 'Horny', dob: new Date(1992,2,13,7,47),
loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0),
loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10),
loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires:
182});

db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18,
44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1),
loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f',
vampires:80});

db.unicorns.insert({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30),
loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires:
40});

```

```

db.unicorns.insert({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42),
loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57),
loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53),
loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires:
33});

db.unicorns.insert({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3),
loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires:
54});

db.unicorns.insert ({name: 'Nimue', dob: new Date(1999, 11, 20, 16,
15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert ({name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18),
loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires:
165})

```

Управление индексами

49)Получите информацию о всех индексах коллекции unicorns.

```
db.unicorns.getIndexes()
```

```

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

```

50)Удалите все индексы, кроме индекса для идентификатора.

```
db.unicorns.dropIndexes()
```

```

> db.unicorns.dropIndexes()
{
  "nIndexesWas" : 1,
  "msg" : "non-_id indexes dropped for collection",
  "ok" : 1
}

```

51)Попытайтесь удалить индекс для идентификатора.

```
db.unicorns.dropIndex({"_id":1})
```

не удалось

```

> db.unicorns.dropIndex({"_id":1})
{
  "ok" : 0,
  "errmsg" : "cannot drop _id index",
  "code" : 72,
  "codeName" : "InvalidOptions"
}

```

План запроса

52) Создайте объемную коллекцию `numbers`, задействовав курсор:

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

```
> db.createCollection("numbers")
{ "ok" : 1 }
> for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
WriteResult({ "nInserted" : 1 })
```

53) Выберите последних четыре документа.

```
db.numbers.find().sort({ $natural: -1 }).limit(4)
```

```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("6288d390ed20642d49f0f60c"), "value" : 99999 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60b"), "value" : 99998 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60a"), "value" : 99997 }
{ "_id" : ObjectId("6288d390ed20642d49f0f609"), "value" : 99996 }
```

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

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

без курсора `var cursor=db.numbers.find().sort({ $natural: -1 }).limit(4);` с курсором

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 7,
```

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,
```

[объясняется малым кол-вом данных](#)

[\(ссылка на форум\)](#)

55) Создайте индекс для ключа `value`.

```
db.numbers.createIndex({"value": 1})
```

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

56) Получите информацию о всех индексах коллекции `numbers`.

```
db.numbers.getIndexes()
```

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

57) Выполните запрос 2.

`db.numbers.find().sort({ $natural: -1 }).limit(4)`

```
> db.numbers.find().sort({ $natural: -1 }).limit(4)
{ "_id" : ObjectId("6288d390ed20642d49f0f60c"), "value" : 99999 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60b"), "value" : 99998 }
{ "_id" : ObjectId("6288d390ed20642d49f0f60a"), "value" : 99997 }
{ "_id" : ObjectId("6288d390ed20642d49f0f609"), "value" : 99996 }
```

58) Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,
```

Без индекса и курсора время выполнения = 7 ms. Без индекса с курсором = 0. С индексом = 0

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

С курсором: время работы запроса с индексом равно времени работы запроса без него, следовательно, по критерию быстродействия невозможно оценить эффективность запросов по отношению друг к другу. На самом деле, скорее всего, разница есть, но из-за малого кол-ва данных она тоже мала и будет видна на больших данных.

Выводы:

Созданы коллекции и проведены манипуляции в базе данных MongoDB.