

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет
по лабораторной работе №5.2

Тема: Работа с БД в СУБД MongoDB

Дисциплина: «Проектирование и реализация баз данных»

Выполнила:

студентка II курса ИКТ

группы К3241

Лорс Х.А.

Проверила:

Говорова М.М.

Санкт-Петербург

2022

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.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 5.0.8.

Практическое задание:

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: 'Rooooooodles', 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:*

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

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

```
> db.unicorns.find()
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f9"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fa"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fe"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294c2661597cf95c65b93ff"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294c2661597cf95c65b9400"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294c2d61597cf95c65b9401"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

ВЫБОРКА ДАННЫХ ИЗ БД

- 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("6294c2661597cf95c65b93f7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fe"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
> db.unicorns.find({gender:'m'}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("6294c2d61597cf95c65b9401"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "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.findOne({gender:'f', loves: 'carrot'})
{
  "_id" : ObjectId("6294c2661597cf95c65b93f7"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
{ "_id" : ObjectId("6294c2661597cf95c65b93f7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

- 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})
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f8"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f9"), "name" : "Rooodooles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("6294c2661597cf95c65b93ff"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("6294c2d61597cf95c65b9401"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
```

- 4) *Вывести список единорогов в обратном порядке добавления.*

```
> db.unicorns.find().sort({$natural: -1})
{ "_id" : ObjectId("6294c2d61597cf95c65b9401"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294c2661597cf95c65b9400"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("6294c2661597cf95c65b93ff"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fe"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fb"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fa"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f9"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f7"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
>
```

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

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

ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

Практическое задание 8.1.6:

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

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

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ `vampires`.

```
> db.unicorns.find({vampires:{$exists:false}})
{ "_id" : ObjectId("6294c2661597cf95c65b9400"), "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("6294c2d61597cf95c65b9401"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294c2661597cf95c65b93ff"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f9"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f8"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
```

ЗАПРОС К ВЛОЖЕННЫМ ОБЪЕКТАМ

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.find()
{ "_id" : ObjectId("6295183b1597cf95c65b9402"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("629518861597cf95c65b9403"), "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("629518a61597cf95c65b9404"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
```

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

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

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

ИСПОЛЬЗОВАНИЕ JAVASCRIPT

Практическое задание 8.2.2:

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

```
> fn=function(){return this.gender=='m'}
function(){return this.gender=='m'}
> db.unicorns.find(fn)
{ "_id" : ObjectId("6294c2661597cf95c65b93f6"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f8"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6294c2661597cf95c65b93f9"), "name" : "Rooodoodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fc"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6294c2661597cf95c65b93ff"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6294c2d61597cf95c65b9401"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
>
```

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

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

3) *Вывести результат, используя forEach.*

```
> cursor.forEach(function(fn) {print(fn.name);})
Horny
Unicrom
Rooooooodles
Kenny
Raleigh
Pilot
Dunx
>
```

4) *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: 'Rooooooodles', 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})
```

АГРЕГИРОВАННЫЕ ЗАПРОСЫ

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> db.unicorns.find({gender:'f', weight:{$gt:500, $lt:600}}).count()
4
```


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" : "f", "count" : 10 }
{ "_id" : "m", "count" : 13 }
>
```

РЕДАКТИРОВАНИЕ ДАННЫХ

Практическое задание 8.2.6:

1. *Выполнить команду:*

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

2. *Проверить содержимое коллекции unicorns.*

```
> db.unicorns.find({name: 'Barney'})
{ "_id" : ObjectId("62951f6d1597cf95c65b9410"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

Практическое задание 8.2.7:

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

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

2. *Проверить содержимое коллекции unicorns.*


```
> db.unicorns.find({name: 'Ayna'})
{ "_id" : ObjectId("6294c2661597cf95c65b93fb"), "name" : "Ayna", "loves" : [ ] }
```

Практическое задание 8.2.8:

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

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: 'Raleigh'})
{ "_id" : ObjectId("6294c2661597cf95c65b93fd"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("62951de21597cf95c65b940b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 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.

Практическое задание 8.2.10:

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

Практическое задание 8.2.11:

1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.
2. Проверить содержимое коллекции unicorns.

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.

УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ

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.

```
> db.towns.remove({"mayor.party":{"exists:false"}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find({"mayor.party":{"exists:false"}})
> db.towns.remove({})
WriteResult({ "nRemoved" : 2 })
> show collections
towns
> db.towns.find()
>
```

ССЫЛКИ В БД

Практическое задание 8.3.1:

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

```
> db.zones.insert({_id:178, short:"piter", full:"Peterburg", desc:"the second capital"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id:77, short:"mos", full:"Moscow", desc:"the firs capital"})
WriteResult({ "nInserted" : 1 })
```

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

```
> db.unicorns.update({name: "Horny"}, {$set:{"city":{"ref":"zones", $id:"77"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Aurora"}, {$set:{"city":{"ref":"zones", $id:"178"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- 3) Проверьте содержание коллекции единорогов.

```
> db.unicorns.find().limit(2)
{ "_id" : ObjectId("629628466086c2e2c9b78fc"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63, "city" : DBRef("zones", "77") }
{ "_id" : ObjectId("629628466086c2e2c9b78fd"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "178") }
```

4) *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

Практическое задание 8.3.2:

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

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

Нельзя, имена могут совпадать.

2. *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

Практическое задание 8.3.3:

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

```

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

```

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

```

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

```

3) *Попытайтесь удалить индекс для идентификатора.*

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

Не удаляется индекс для идентификатора.

ПЛАН ЗАПРОСА

Практическое задание 8.3.4:

1) 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 })
```

2) Выберите последних четыре документа.

```
> db.numbers.find().sort({$natural:-1}).limit(4)
{ "_id" : ObjectId("62962cba66086c2e2c9cffb2"), "value" : 99999 }
{ "_id" : ObjectId("62962cba66086c2e2c9cffb1"), "value" : 99998 }
{ "_id" : ObjectId("62962cba66086c2e2c9cffb0"), "value" : 99997 }
{ "_id" : ObjectId("62962cba66086c2e2c9cfffaf"), "value" : 99996 }
```

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

с курсором:

```
"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 0,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 4,
  "executionStages" : {
    "stage" : "LIMIT",
```

без курсора:

```

"executionStats" : {
  "executionSuccess" : true,
  "nReturned" : 4,
  "executionTimeMillis" : 2,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 4,
  "executionStages" : {
    "stage" : "LIMIT",
    "nReturned" : 4,
    "executionTimeMillisEstimate" : 0,
    "works" : 6,

```

4) *Создайте индекс для ключа value.*

```

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

```

5) *Получите информацию о всех индексах коллекции numbers.*

```

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

```

6) *Выполните запрос 2.*

```

> db.numbers.find().sort({$natural:-1}).limit(4)
{ "_id" : ObjectId("62962cba66086c2e2c9cffb2"), "value" : 99999 }
{ "_id" : ObjectId("62962cba66086c2e2c9cffb1"), "value" : 99998 }
{ "_id" : ObjectId("62962cba66086c2e2c9cffb0"), "value" : 99997 }
{ "_id" : ObjectId("62962cba66086c2e2c9cffaf"), "value" : 99996 }

```

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

```

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

```

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

Ответ: в данной работе время выполнения запросов с индексом и без индекса одинаково - 0 мс, поэтому нельзя оценить эффективность индексов по времени. Однако при большем количестве данных она будет заметнее.

Вывод: в ходе работы были получены практические работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.