Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ № 5.2

по теме: Работа с БД в СУБД MongoDB.
по дисциплине: Проектирование и реализация баз данных

Специальность:
09.03.03 Мобильные и сетевые технологии

Выполнил:
студент 2 курса ИКТ группа
К3241
Сизей Омар

Проверила:
Говорова Марина Михайловна

Санкт-Петербург 2021 г.

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

1. Создайте базу данных learn

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

2. Заполните коллекцию единорогов u

```
local   0.000GB
> db.unicorns.insert({name: 'Horny', loves: ['carrot' 'grape'], weight 450, gender: 'f', vampire: 43});
uncaught exception: SyntaxError: missing ] after element list :
@(shell):1:52
> db.unicorns.insert({name: 'Horny', loves: ['carrot','grape'], weight: 450, gender: 'f', vampire: 43});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Aurora', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampire: 63});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Unicrom', loves: ['energon','redbull'], weight: 948, gender: 'm', vampire: 18});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampire: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves: ['apple','carrot','chocolate'], weight: 550, gender: 'f', vampire: 80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Ayna', loves: ['strawberry','lemon'], weight: 733, gender: 'f', vampire: 40});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Kenny', loves: ['grape','lemon'], weight: 690, gender: 'm', vampire: 39});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Raleighh', loves: ['apple','sugar'], weight: 421, gender: 'm', vampire: 2});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['apple','watermelon'], weight: 601, gender: 'f', vampire: 33});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Pilot', loves: ['apple','watermelon'], weight: 650, gender: 'm', vampire: 54});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Leia', loves: ['grape','carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Nimue', loves: ['grape','carrot'], weight: 540, gender: 'f'});
WriteResult({ "nInserted" : 1 })
>
```

nicorns

3. Используя второй способ, вставьте в коллекцию единорогов документ

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

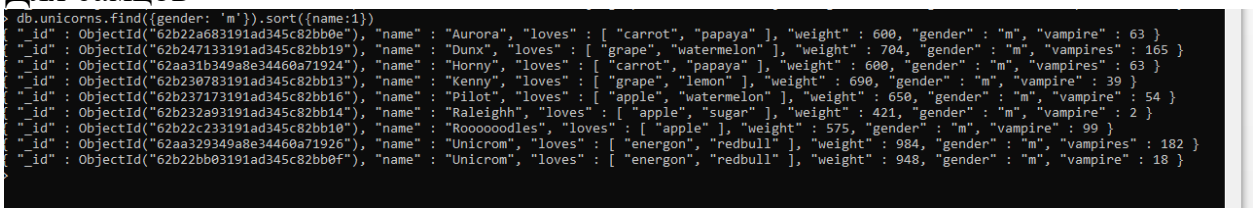
```
> db.unicorns.insert(document)
WriteResult({ "nInserted" : 1 })
>
```

4. Проверьте содержимое коллекции с помощью метода find

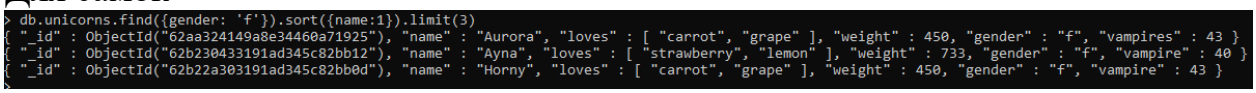
8.2.2 Выборка данных из БД

8.1.2
1. Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.

Для самцов



Для самок



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

FindOne

```
> db.unicorns.findOne({gender: 'f', loves: "carrot"})
{
        "_id" : ObjectId("62aa324149a8e34460a71925"),
        "name" : "Aurora",
        "loves" : [
                "carrot",
                "grape"
        ],
        "weight" : 450,
        "gender" : "f",
        "vampires" : 43
}
>
```

Limit

```
> db.unicorns.find({gender: 'f', loves: "carrot"}).limit(1)
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>
```

8.1.3

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

```
> db.unicorns.find({gender: 'm'}, {loves:0, gender:0}).sort({name:1})
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "weight" : 600, "vampire" : 63 }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("62b230783191ad345c82bb13"), "name" : "Kenny", "weight" : 690, "vampire" : 39 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "weight" : 650, "vampire" : 54 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "weight" : 421, "vampire" : 2 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "weight" : 575, "vampire" : 99 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "weight" : 948, "vampire" : 18 }
>
```

8.1.4

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({ $natural: -1})
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62b23ef03191ad345c82bb18"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b23e6c3191ad345c82bb17"), "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampire" : 54 }
{ "_id" : ObjectId("62b2344a3191ad345c82bb15"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampire" : 2 }
{ "_id" : ObjectId("62b230783191ad345c82bb13"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "_id" : ObjectId("62b230433191ad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampire" : 40 }
{ "_id" : ObjectId("62b22d823191ad345c82bb11"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire" :
80 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18 }
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "_id" : ObjectId("62b22a303191ad345c82bb0d"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

8.1.5

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" : "Horny", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 948, "gender" : "m", "vampire" : 18 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampire" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampire" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "name" : "Raleighh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampire" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampire" : 54 }
{ "name" : "Leia", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

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", "vampire" : 80 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

8.1.7

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

```
> db.unicorns.find({gender:"m", weight:{$gt : 500}, loves: {$in:["grape","lemon"]}},{_id : 0})
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

8.1.8

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find ({vampires: {$exists:false}}, {_id : 0})
{ "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampire" : 40 }
{ "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "name" : "Raleighh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampire" : 2 }
{ "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampire" : 54 }
{ "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

8.1.9

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

```
> db.unicorns.find({gender:"m"}, {loves: {$slice : 1}}).sort({name : 1})
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62b230783191ad345c82bb13"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampire" : 54 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampire" : 2 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 948, "gender" : "m", "vampire" : 18 }
```

8.2.1

1. Создайте коллекцию towns, включающую следующие документы

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" : "Michale 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" } }
>
```

8.2.2

1. Сформировать функцию для вывода списка самцов единорогов.

```
> f=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(f);null;
null
> cursor.sort({name : 1}).limit(2)
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

3. Вывести результат, используя forEach.

```
> cursor.sort({name : 1}).limit(2); null;
null
> cursor.forEach(function(obj)[ print(obj.name); })
uncaught exception: SyntaxError: missing { before function body :
@(shell):1:28
> cursor.forEach(function(obj){ print(obj.name); })
Aurora
Dunx
```

8.2.3

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> db.unicorns.find({gender: "f", weight: {$gt : 500, $lt : 600}}).count()
3
>
```

8.2.4

Вывести список предпочтений.

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

8.2.5

Посчитать количество особей единорогов обоих полов.

```
> db.unicorns.aggregate({"$group":{_id:"$gender",count:{$sum:1}}})
{ "_id" : "m", "count" : 9 }
{ "_id" : "f", "count" : 7 }
>
```

8.2.6

1. Выполнить команду:

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b22a303191ad345c82bb0d"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63, "vampires" : 5 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18, "vampires" : 5 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99, "vampires" : 5 }
{ "_id" : ObjectId("62b22d823191ad345c82bb11"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire" : 80 }
{ "_id" : ObjectId("62b230433191ad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b230783191ad345c82bb13"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39, "vampires" : 5 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampire" : 2, "vampires" : 5 }
{ "_id" : ObjectId("62b2344a3191ad345c82bb15"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampire" : 54, "vampires" : 5 }
{ "_id" : ObjectId("62b23e6c3191ad345c82bb17"), "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b23ef03191ad345c82bb18"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62bb85e5a2af005c056d33e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

8.2.7

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

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Ayna"})
{ "_id" : ObjectId("62b230433191ad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon
" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
>
```

8.2.8

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

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: "Raleighh"})
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "redbull" ], "we
ight" : 421, "gender" : "m", "vampire" : 2 }
>
```

8.2.9

1. Всем самцам единорогов увеличить количество убитых вапмиров на 5.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.find()
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("62b22a303191ad345c82bb0d"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99 }
{ "_id" : ObjectId("62b22d823191ad345c82bb11"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire"
{ "_id" : ObjectId("62b23043319lad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b23078319lad345c82bb13"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampire" : 2 }
{ "_id" : ObjectId("62b2344a3191ad345c82bb15"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampire" : 54 }
{ "_id" : ObjectId("62b23e6c3191ad345c82bb17"), "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b23ef03191ad345c82bb18"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

8.2.10

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

```
> db.towns.update({name:"Portland"}, {$unset:{"mayor.party":1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("62b772c300d63213d0ded5ff"), "name" : "New york", "mayor" : { "name" : "Michale Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("62b773b800d63213d0ded600"), "name" : "Punxsutawney", "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("62bb83e0a2af005c056d33e2"), "name" : "Portland", "mayor" : { "name" : "John wick" } }
>
```

8.2.11

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

```
> db.unicorns.update({name:"pilot"}, {$push:{loves:"chocolate"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.unicorns.update({name:"Pilot"}, {$push:{loves:"chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b22a303191ad345c82bb0d"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63, "vampires" : 5 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18, "vampires" : 5 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99, "vampires" : 5 }
{ "_id" : ObjectId("62b22d823191ad345c82bb11"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire" : 80 }
{ "_id" : ObjectId("62b23043319lad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b23078319lad345c82bb13"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39, "vampires" : 5 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampire" : 2, "vampires" : 5 }
{ "_id" : ObjectId("62b2344a3191ad345c82bb15"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampire" : 54, "vampires" : 5
{ "_id" : ObjectId("62b23e6c3191ad345c82bb17"), "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b23ef03191ad345c82bb18"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62bb85e5a2af005c056d33e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

8.2.12

1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name:"Aurora"}, {$addToSet:{loves:{$each:["sugar","lemon"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("62aa329349a8e34460a71926"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("62b22a303191ad345c82bb0d"), "name" : "Horny", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampire" : 43 }
{ "_id" : ObjectId("62b22a683191ad345c82bb0e"), "name" : "Aurora", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampire" : 63, "vampires" : 5 }
{ "_id" : ObjectId("62b22bb03191ad345c82bb0f"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 948, "gender" : "m", "vampire" : 18, "vampires" : 5 }
{ "_id" : ObjectId("62b22c233191ad345c82bb10"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampire" : 99, "vampires" : 5 }
{ "_id" : ObjectId("62b22d823191ad345c82bb11"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampire" : 80 }
{ "_id" : ObjectId("62b230433191ad345c82bb12"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
{ "_id" : ObjectId("62b230783191ad345c82bb13"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampire" : 39, "vampires" : 5 }
{ "_id" : ObjectId("62b232a93191ad345c82bb14"), "name" : "Raleighh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampire" : 2, "vampires" : 5 }
{ "_id" : ObjectId("62b2344a3191ad345c82bb15"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampire" : 33 }
{ "_id" : ObjectId("62b237173191ad345c82bb16"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampire" : 54, "vampires"
{ "_id" : ObjectId("62b23e6c3191ad345c82bb17"), "name" : "Leia", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b23ef03191ad345c82bb18"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("62b247133191ad345c82bb19"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("62bb85e5a2af005c056d33e3"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
>
```

8.2.13

1. Создайте коллекцию towns, включающую следующие документы
2. Удалите документы с беспартийными мэрами
3. Проверьте содержание коллекции.
4. Очистите коллекцию.
5. Просмотрите список доступных коллекций.

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

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: "78", short: "ars", full: "Arsenal", descr: "hole"})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: "64", short: "vdk", full: "lokomotiv", descr: "club team"})
WriteResult({ "nInserted" : 1 })
>
```

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

```
> db.unicorns.update({name: "Aurora"}, {$set: {"city": {$ref: "zones", $id: "78"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: "Horny"}, {$set: {"city": {$ref: "zones", $id: "64"}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

3. Проверьте содержание коллекции едиорогов

```
> db.unicorns.find()
{ "_id" : ObjectId("62aa31b349a8e34460a71924"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68, "city" : DBRef("zones", "64") }
{ "_id" : ObjectId("62aa324149a8e34460a71925"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "city" : DBRef("zones", "78")
```

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}
```

8.3.2

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

```
> db.unicorns.createIndex({"name" : 1}, {"unique" : true})
{
        "ok" : 0,
        "errmsg" : "Index build failed: 2080490a-8a7f-4238-af6b-5fe164daf607: Collection learn.unicorns ( e50f1759-c418-4422-9d56-d4650b192be6 ) :: caused by :: E11000 duplicate key error collecti
n: learn.unicorns index: name_1 dup key: { name: \"Aurora\" }",
        "code" : 11000,
        "codeName" : "DuplicateKey",
        "keyPattern" : {
                "name" : 1
        },
        "keyValue" : {
                "name" : "Aurora"
        }
}
```

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

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("62bc319da2af005c056eba84"), "value" : 99999 }
{ "_id" : ObjectId("62bc319da2af005c056eba83"), "value" : 99998 }
{ "_id" : ObjectId("62bc319da2af005c056eba82"), "value" : 99997 }
{ "_id" : ObjectId("62bc319da2af005c056eba81"), "value" : 99996 }
>
```

3. Проанализируйте план выполнения запроса 2. Сколько потребовалось

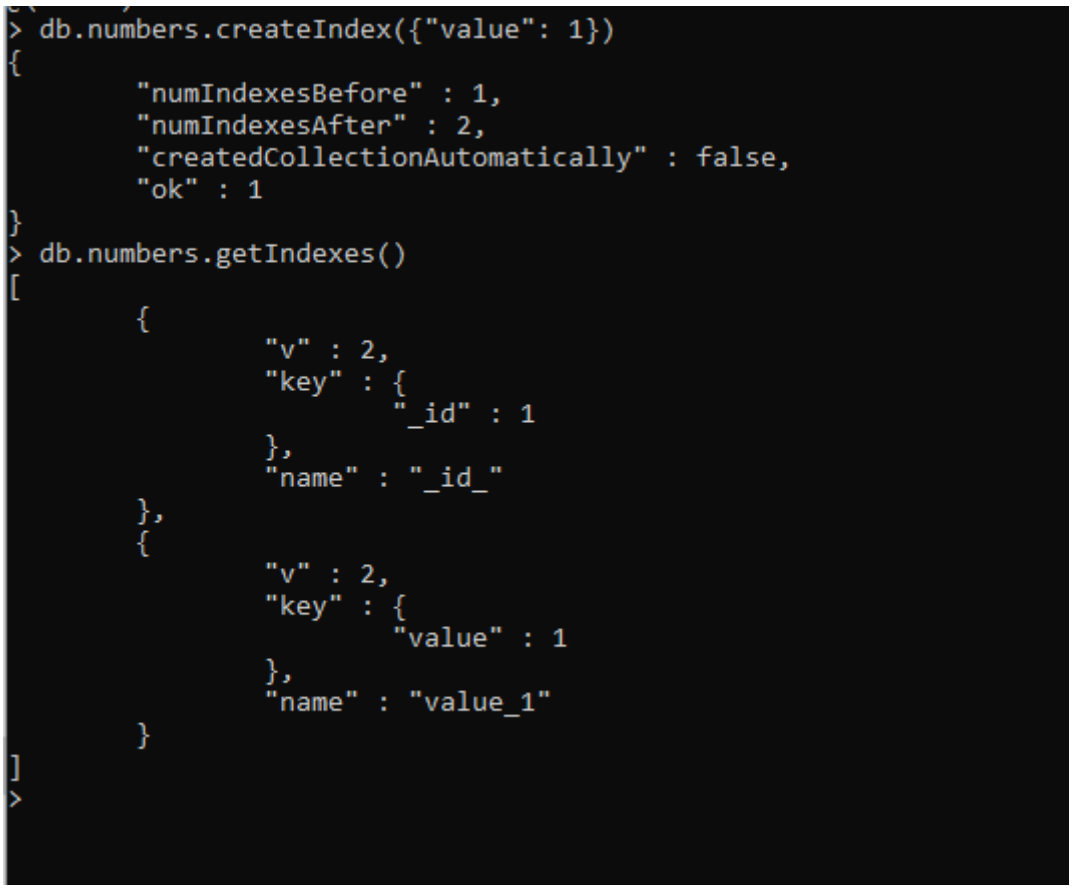
времени на выполнение запроса? (по значению параметра executionTimeMillis)

```
> db.user.explain("executionStats").find().sort({value:-1}).limit(4)
{
        "explainVersion" : "1",
        "queryPlanner" : {
                "namespace" : "learn.user",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "maxIndexedOrSolutionsReached" : false,
                "maxIndexedAndSolutionsReached" : false,
                "maxScansToExplodeReached" : false,
                "winningPlan" : {
                        "stage" : "EOF"
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 0,
                "executionTimeMillis" : 0,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 0,
                "executionStages" : {
                        "stage" : "EOF",
                        "nReturned" : 0,
                        "executionTimeMillisEstimate" : 0,
                        "works" : 1,
                        "advanced" : 0,
                        "needTime" : 0,
                        "needYield" : 0,
                        "saveState" : 0,
                        "restoreState" : 0,
                        "isEOF" : 1
                }
        },
```

4.Создайте индекс для ключа value.

5. Получите информацию о всех индексах коллекции numbres



6.Выполните запрос 2.

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

8. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?
9. Можно утверждать, что с индексами запрос будет
работать быстрее и эффективнее. Использование индекса ускоряет возврат результатов запроса оправдано.

Вывода:

В ходе работы были получены практические навыки
работа с CRUD - операции с вложенными объектами в базовой коллекции Данные MongoDB, агрегация и модификация данных со ссылками и индексами в База данных MongoDB.