

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

О Т Ч Е Т

по дисциплине «Проектирование и реализация баз данных»

на тему: Работа с БД в СУБД MongoDB

Обучающийся: Олейникова Полина Леонидовна, К32402

Преподаватель: Говорова М. М.

Дата 28.05.2023

Санкт-Петербург 2023

ВВЕДЕНИЕ

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.

1 Выполнение

Практическое задание 8.1.1:

1. Создайте базу данных learn.

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

2. Заполните коллекцию единорогов unicorns:

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

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

```
> 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.

```

> db.unicorns.find()
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eaea"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eaeab"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }

```

Практическое задание 8.1.2:

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

```

> db.unicorns.find({gender:"m"}).sort({name:1})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eaeab"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
>
> db.unicorns.find({gender:"f"}).sort({name:1}).limit(3)
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
>

```

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("64738f418cc5d58a7ea1eae1"),
  "name" : "Aurora",
  "loves" : [
    "carrot",
    "grape"
  ],
  "weight" : 450,
  "gender" : "f",
  "vampires" : 43
}
>
> db.unicorns.find({gender:"f",loves:"carrot"}).limit(1)
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
>

```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({gender:"m"},{loves:0,gender:0}).sort({name:1})
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eae0"), "name" : "Dunx", "weight" : 704, "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "weight" : 600, "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "weight" : 690, "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "weight" : 650, "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "weight" : 421, "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Roooooodles", "weight" : 575, "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "weight" : 984, "vampires" : 182 }
```

Практическое задание 8.1.4:

1. Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({_id:-1})
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eae0"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

```
> db.unicorns.find().sort({$natural:-1})
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eae0"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

1. 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" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.6:

1. 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:

1. 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", "vampires" : 39 }
{ "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

Практическое задание 8.1.8:

1. Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({vampires:{$exists:false}})
{ "_id" : ObjectId("64738f418cc5d58a7eaeaea"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
```

Практическое задание 8.1.9:

1. 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("64738f8c8cc5d58a7eaeab"), "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae0"), "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae6"), "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae9"), "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae7"), "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae3"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae2"), "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
```

Практическое задание 8.2.1:

1. Создайте коллекцию towns, включающую следующие документы:

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

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

```
> db.towns.find({"mayor.party":"I"},{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" } }
```

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("64738f418cc5d58a7eaeae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae3"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7eaeae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f8c8cc5d58a7eaeab"), "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(fn);null;
null
> cursor.sort({name:1}).limit(2)
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eae0"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
```

3. Вывести результат, используя forEach.

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

4. Содержание коллекции единорогов unicorns:

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()
2
```

Практическое задание 8.2.4: вывести список предпочтений.

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

Практическое задание 8.2.5: посчитать количество особей единорогов обоих полов.

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

Практическое задание 8.2.6:

1. Выполнить команду, Проверить содержимое коллекции unicorns.


```
> db.unicorns.save({name: 'Barny', loves: ['grape'],
... weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae4"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eaaa"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eaaeb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("647396b38cc5d58a7ea1eaaef"), "name" : "Barny", "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 вапмира.
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name:"Ayna"},{name:"Ayna",weight:800,gender:"f",vampires:51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name:"Ayna"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae5"), "name" : "Ayna", "weight" : 800, "gender" : "f", "vampires" : 51 }
```

Практическое задание 8.2.8:

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

```
> db.unicorns.update({name:"Raleigh"},{$set:{loves:['redbull']}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name:"Raleigh"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
```

Практическое задание 8.2.9:

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

```
> db.unicorns.updateMany({gender:"m"},{$inc:{vampires:5}})
{ "acknowledged" : true, "matchedCount" : 8, "modifiedCount" : 8 }
> db.unicorns.find({gender:"m"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae0"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae2"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae3"), "name" : "Rooodooles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae6"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae7"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("64738f8c8cc5d58a7ea1eaaeb"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("647396b38cc5d58a7ea1eaaef"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

Практическое задание 8.2.10:

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

```
> db.towns.update({name:"Portland"},{$unset:{mayor.party:1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.towns.find()
{ "_id" : ObjectId("6473935c8cc5d58a7ea1eae3"), "name" : "Punxsutawney ", "populatiuon" : 6200, "last_sensus" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "m" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("6473936f8cc5d58a7ea1eae4"), "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("647393808cc5d58a7ea1eae5"), "name" : "Portland", "populatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams" } }
```

Практическое задание 8.2.11:

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


```
> db.unicorns.update({name:"Pilot"},{$push:{loves:"chocolate"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name:"Pilot"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae9"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
```

Практическое задание 8.2.12:

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

```
> db.unicorns.find({name:"Aurora"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae1"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
```

Практическое задание 8.2.13:

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

2. Удалите документы с беспартийными мэрами.
3. Проверьте содержание коллекции.

```
> db.towns.remove({'mayor.party':{'exists':false}})
WriteResult({ "nRemoved" : 1 })
> db.towns.find()
{ "_id" : ObjectId("647399e68cc5d58a7ea1eaf1"), "name" : "New York", "popujatiuon" : 22200000, "last_sensus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("647399f28cc5d58a7ea1eaf2"), "name" : "Portland", "popujatiuon" : 528000, "last_sensus" : ISODate("2009-07-20T00:00:00Z"), "famous_for" : [ "beer", "food" ], "mayor" : { "name" : "Sam Adams", "party" : "D" } }
```

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

```
> db.towns.drop()
true
> show collections
unicorns
```

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.insertMany([
  {id: "forest", name: "forest", description: "A forest with lots of trees"},
  {id: "b_forest", name: "birch forest", description: "A forest with lots of birch trees"},
  {id: "j_forest", name: "jungle", description: "A forest with lots of jungle trees"},
  {id: "t_forest", name: "taiga", description: "A forest with lots of spruce trees"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("64739b588cc5d58a7ea1eaf3"),
    ObjectId("64739b588cc5d58a7ea1eaf4"),
    ObjectId("64739b588cc5d58a7ea1eaf5"),
    ObjectId("64739b588cc5d58a7ea1eaf6")
  ]
}
> db.zones.find()
{ "_id" : ObjectId("64739b588cc5d58a7ea1eaf3"), "id" : "forest", "name" : "forest", "description" : "A forest with lots of trees" }
{ "_id" : ObjectId("64739b588cc5d58a7ea1eaf4"), "id" : "b_forest", "name" : "birch forest", "description" : "A forest with lots of birch trees" }
{ "_id" : ObjectId("64739b588cc5d58a7ea1eaf5"), "id" : "j_forest", "name" : "jungle", "description" : "A forest with lots of jungle trees" }
{ "_id" : ObjectId("64739b588cc5d58a7ea1eaf6"), "id" : "t_forest", "name" : "taiga", "description" : "A forest with lots of spruce trees" }

```

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

```

> db.unicorns.update({name: "Leia"}, {$set: {zone: { $ref: "zones", $id: "j_forest" }}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find({name: "Leia"})
{ "_id" : ObjectId("64738f418cc5d58a7ea1eae8"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "F", "vampires" : 33, "zone" : DBRef("zones", "j_forest") }

```

Практическое задание 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})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
> db.unicorns.insert({name: "Pilot"})
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: learn.unicorns index: name_1 dup key: { name: \"Pilot\" }"
  }
})

```

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns .

```

> db.unicorns.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "name" : 1
    },
    "name" : "name_1",
    "unique" : true
  }
]

```

2. Удалите все индексы, кроме индекса для идентификатора.

```
> db.unicorns.dropIndexes("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
> db.unicorns.getIndexes()
[ { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_" } ]
>
```

3. Попробуйте удалить индекс для идентификатора.

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

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers, задействовав курсор:
2. Выберите последних четыре документа.

```
> db.numbers.find().sort({$natural:-1}).limit(4)
{ "_id" : ObjectId("64739d838cc5d58a7ea371a3"), "value" : 99999 }
{ "_id" : ObjectId("64739d838cc5d58a7ea371a2"), "value" : 99998 }
{ "_id" : ObjectId("64739d838cc5d58a7ea371a1"), "value" : 99997 }
{ "_id" : ObjectId("64739d838cc5d58a7ea371a0"), "value" : 99996 }
>
```

3. 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)
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "learn.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {

    },
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 5,

```

4. Создайте индекс для ключа value.
5. Получите информацию о всех индексах коллекции numbers.

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

6. Выполните запрос 2.

```

> db.numbers.explain("executionStats").find().sort({$natural:-1}).limit(4)
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "learn.numbers",
    "indexFilterSet" : false,
    "parsedQuery" : {
      },
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "LIMIT",
      "limitAmount" : 4,
      "inputStage" : {
        "stage" : "COLLSCAN",
        "direction" : "backward"
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 4,
    "executionTimeMillis" : 0,
  }
}

```

7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса? 0
8. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?
Время уменьшилось на 5 миллисекунд. Запрос с индексами эффективнее.

ЗАКЛЮЧЕНИЕ

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

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