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 № 8

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

Специальность:
09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «__» _____ 20__г.
Оценка _____

Выполнил:
студент группы К3241
Кривошапкина А.С.

Санкт-Петербург 2021 г

1. 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

2. Практическое задание

- Установить базу данных MongoDB
- Реализовать запросы CRUD
- Выполнить агрегацию данных

3. Выполнение

8.1 CRUD-ОПЕРАЦИИ В СУБД MONGODB. ВСТАВКА ДАННЫХ. ВЫБОРКА ДАННЫХ

8.1.1 ВСТАВКА ДОКУМЕНТОВ В КОЛЛЕКЦИЮ

Практическое задание 8.1.1:

1. Создать базу данных learn
2. Заполните коллекцию единорогов unicorns

```
> use learn
switched to db learn
> 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', vampi
res: 182})
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', v
ampires:80});
WriteResult({ "nInserted" : 1 })
> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 4
0});
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("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "w
eight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
> db.unicorns.find({"gender": "m"})
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
```

8.1.2 ВЫБОРКА ДАННЫХ ИЗ БД

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("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
```

```
> db.unicorns.find({"gender": "f"}).sort({name: 1}).limit(3)
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "w
eight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 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({"loves": "carrot"})
{
        "_id" : ObjectId("60be69328827720b39e44144"),
        "name" : "Horny",
        "loves" : [
                "carrot",
                "papaya"
        ],
        "weight" : 600,
        "gender" : "m",
        "vampires" : 63
}
> db.unicorns.find({"loves": "carrot"}).limit(1)
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 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})
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "weight" : 600, "gender" : "m", "vampi
res" : 63 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "weight" : 984, "gender" : "m", "vam
pires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "weight" : 575, "gender" : "m",
"vampires" : 99 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "weight" : 690, "gender" : "m", "vampi
res" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "weight" : 421, "gender" : "m", "vam
pires" : 2 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "weight" : 650, "gender" : "m", "vampi
res" : 54 }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "weight" : 704, "gender" : "m", "vampir
es" : 165 }
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find().sort({ $natural: -1})
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "w
eight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
```

Практическое задание 8.1.5:

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

```
> db.unicorns.find({}, {"_id": 0, "loves": {$slice: 1}})
{ "name" : "Horny", "loves" : [ "carrot" ], "weight" : 600, "gender" : "m", "vampires" : 63 }
{ "name" : "Aurora", "loves" : [ "carrot" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Unicrom", "loves" : [ "energon" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "name" : "Solnara", "loves" : [ "apple" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "name" : "Ayna", "loves" : [ "strawberry" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "name" : "Kenny", "loves" : [ "grape" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "name" : "Raleigh", "loves" : [ "apple" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "name" : "Leia", "loves" : [ "apple" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "name" : "Pilot", "loves" : [ "apple" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "name" : "Nimue", "loves" : [ "grape" ], "weight" : 540, "gender" : "f" }
{ "name" : "Dunx", "loves" : [ "grape" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
```

8.1.3 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: {$lt: 700}}, {_id: 0})
{ "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vam
pires" : 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: 0}})
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 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"}, {name: 1, "loves": {$slice: 1}, _id: 0}).sort({name: 1})
{ "name" : "Dunx", "loves" : [ "grape" ] }
{ "name" : "Horny", "loves" : [ "carrot" ] }
{ "name" : "Kenny", "loves" : [ "grape" ] }
{ "name" : "Pilot", "loves" : [ "apple" ] }
{ "name" : "Raleigh", "loves" : [ "apple" ] }
{ "name" : "Roooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
```

8.2 ЗАПРОСЫ К БАЗЕ ДАННЫХ MONGODB.

ВЫБОРКА ДАННЫХ. ВЛОЖЕННЫЕ ОБЪЕКТЫ. ИСПОЛЬЗОВАНИЕ КУРСОРОВ. АГРЕГИРОВАННЫЕ ЗАПРОСЫ. ИЗМЕНЕНИЕ ДАННЫХ

8.2.1 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.insert({name: 'Punxsutawney', population: 6200, last_sensus: ISODate('2008-01-31'), famous_fo
r: [''], mayor: {name: 'Jim Wehrle'}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: 'New York', population: 22200000, last_sensus: ISODate('2009-07-31'), famous_fo
r: ['status of liberty', 'food'], mayor: {name: 'Michael Bloomberg', party: 'I'}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: 'Portland', population: 528000, last_sensus: ISODate('2009-07-20'), famous_for:
 ['beer', 'food'], mayor: {name: 'Sam Adams', party: 'D'}})
WriteResult({ "nInserted" : 1 })
```

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

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

8.2.2 ИСПОЛЬЗОВАНИЕ JAVASCRIPT

8.2.3 КУРСОРЫ

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

```
> find_male = function() {return this.gender == 'm'; }
function() {return this.gender == 'm'; }
> db.unicorns.find(find_male)
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
```

2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя forEach.

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

8.2.4 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()
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', counter: {$sum: 1}}}])
{ "_id" : "m", "counter" : 7 }
{ "_id" : "f", "counter" : 5 }
```

8.2.5 РЕДАКТИРОВАНИЕ ДАННЫХ

Практическое задание 8.2.6:

1. Выполнить команду:

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

Проверить содержимое коллекции unicorns.

```
> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
WriteResult({ "nInserted" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "w
eight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340,
 "gender" : "m" }
```

Практическое задание 8.2.7:

1. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вампира
2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.update({name: 'Ayna'}, {name: 'Ayna', weight: 800, vampires: 51})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "wei
ght" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340,
 "gender" : "m" }
```

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()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340,
 "gender" : "m" }
```

Практическое задание 8.2.9:

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

```
> db.unicorns.update({'gender': 'm'}, {$inc: {vampires: 5}}, {multi: true})
WriteResult({ "nMatched" : 8, "nUpserted" : 0, "nModified" : 8 })
> db.unicorns.find()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "
weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "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("60be80cc8827720b39e44153"), "name" : "Punxsutawney", "population" : 6200, "last_sens
us" : ISODate("2008-01-31T00:00:00Z"), "famous_for" : [ "" ], "mayor" : { "name" : "Jim Wehrle" } }
{ "_id" : ObjectId("60be80db8827720b39e44154"), "name" : "New York", "population" : 22200000, "last_sens
us" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "name
" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("60be80ea8827720b39e44155"), "name" : "Portland", "population" : 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()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "wei
ght" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "ch
ocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340,
 "gender" : "m", "vampires" : 5 }
```

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.update({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.find()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar
", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51 }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "ch
ocolate" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("60be954a8827720b39e44158"), "name" : "Barny", "loves" : [ "grape" ], "weight" : 340,
 "gender" : "m", "vampires" : 5 }
```

8.2.6 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"}}
```

```
> 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_f
or: ["status of liberty", "food"],
... mayor: {name: "Michael Bloomberg", party: "I"}})
WriteResult({ "nInserted" : 1 })
> db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for
: ["beer", "food"], mayor: {name: "Sam Adams", party: "D"}}
... )
WriteResult({ "nInserted" : 1 })
```

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("60be9c9e8827720b39e4415a"), "name" : "New York", "popujatiuon" : 22200000, "last_sen
sus" : ISODate("2009-07-31T00:00:00Z"), "famous_for" : [ "status of liberty", "food" ], "mayor" : { "nam
e" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("60be9cea8827720b39e4415b"), "name" : "Portland", "popujatiuon" : 528000, "last_sensu
s" : 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.remove({})
WriteResult({ "nRemoved" : 2 })
> db.towns.find()
>
```

8.3 ССЫЛКИ И РАБОТА С ИНДЕКСАМИ В БАЗЕ ДАННЫХ MONGODB

8.3.1 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: 'mp', name: 'magic place', description: 'cool'})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: 'al', name: 'alfea', description: 'nice'})
WriteResult({ "nInserted" : 1 })
> db.zones.insert({_id: 'mr', name: 'mordor', description: 'scary'})
WriteResult({ "nInserted" : 1 })
```

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

```
> db.unicorns.update({name: 'Horny'}, {$set: {zone: {$ref: 'zones', $id: 'mp'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Ayna'}, {$set: {zone: {$ref: 'zones', $id: 'al'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Aurora'}, {$set: {zone: {$ref: 'zones', $id: 'mo'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.unicorns.update({name: 'Dunx'}, {$set: {zone: {$ref: 'zones', $id: 'mo'}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

3. Проверьте содержание коллекции единорогов

```
> db.unicorns.find()
{ "_id" : ObjectId("60be69328827720b39e44144"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "wei
ght" : 600, "gender" : "m", "vampires" : 63, "zone" : DBRef("zones", "mp") }
{ "_id" : ObjectId("60be694f8827720b39e44145"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar
", "lemon" ], "weight" : 450, "gender" : "f", "vampires" : 43, "zone" : DBRef("zones", "mo") }
{ "_id" : ObjectId("60be69978827720b39e44146"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ],
"weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("60be699e8827720b39e44147"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight"
: 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("60be69b48827720b39e44148"), "name" : "Solnara", "loves" : [ "apple", "carrot", "choc
olate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("60be69c98827720b39e44149"), "name" : "Ayna", "weight" : 800, "vampires" : 51, "zone"
 : DBRef("zones", "al") }
{ "_id" : ObjectId("60be69db8827720b39e4414a"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weigh
t" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("60be69ec8827720b39e4414b"), "name" : "Raleigh", "loves" : "redbull", "weight" : 421,
 "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("60be6a008827720b39e4414c"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "w
eight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("60be6a138827720b39e4414d"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "ch
ocolate" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("60be6a1b8827720b39e4414e"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weig
ht" : 540, "gender" : "f" }
{ "_id" : ObjectId("60be6a7b8827720b39e4414f"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "w
eight" : 704, "gender" : "m", "vampires" : 165, "zone" : DBRef("zones", "mo") }
```

8.3.2 НАСТРОЙКА ИНДЕКСОВ

Практическое задание 8.3.2:

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

```
> db.unicorns.ensureIndex({name: 1}, {unique: true})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.unicorns.insert({name: 'Dunx'});
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 11000,
                "errmsg" : "E11000 duplicate key error collection: learn.unicorns index: name_1 dup key:
 { name: \"Dunx\" }"
        }
})
```

8.3.3 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_"
        },
        {
                "v" : 2,
                "unique" : true,
                "key" : {
                        "name" : 1
                },
                "name" : "name_1"
        }
]
```

2. Удалите все индексы, кроме индекса для идентификатора

```
> db.unicorns.dropIndexes('name_1')
{ "nIndexesWas" : 2, "ok" : 1 }
```

3. Попытайтесь удалить индекс для идентификатора.

```
> db.unicorns.dropIndexes('_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
```

8.3.3 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})}
```

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

2. Выберите последних четыре документа.

```
> db.numbers.find().skip(99996)
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895ea9"), "value" : 99996 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eaa"), "value" : 99997 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eab"), "value" : 99998 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eac"), "value" : 99999 }
```

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

```
> db.numbers.explain('executionStats').find({executionTimeMills: 1})
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "test.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "executionTimeMills" : {
                                "$eq" : 1
                        }
                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "executionTimeMills" : {
                                        "$eq" : 1
                                }
                        },
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 0,
                "executionTimeMillis" : 244,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 100000,
                "executionStages" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "executionTimeMills" : {
                                        "$eq" : 1
                                }
                        },
                        "nReturned" : 0,
                        "executionTimeMillisEstimate" : 30,
                        "works" : 100002,
                        "advanced" : 0,
                        "needTime" : 100001,
                        "needYield" : 0,
                        "saveState" : 100,
                        "restoreState" : 100,
                        "isEOF" : 1,
                        "direction" : "forward",
                        "docsExamined" : 100000
                }
        },
        "serverInfo" : {
                "host" : "Ayta",
                "port" : 27017,
                "version" : "4.4.6",
                "gitVersion" : "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7"
        },
        "ok" : 1
}
```

4. Создайте индекс для ключа value

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

5. Получите информацию о всех индексах коллекции numbers.

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

6. Выполните запрос 2

```
> db.numbers.find().skip(99996)
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895ea9"), "value" : 99996 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eaa"), "value" : 99997 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eab"), "value" : 99998 }
{ "_id" : ObjectId("60bf6f8dbf1e7b7eb9895eac"), "value" : 99999 }
```

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

```
> db.numbers.explain('executionStats').find({executionTimeMills: 1})
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "test.numbers",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "executionTimeMills" : {
                                "$eq" : 1
                        }
                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "executionTimeMills" : {
                                        "$eq" : 1
                                }
                        },
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "executionStats" : {
                "executionSuccess" : true,
                "nReturned" : 0,
                "executionTimeMillis" : 110,
                "totalKeysExamined" : 0,
                "totalDocsExamined" : 100000,
                "executionStages" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "executionTimeMills" : {
                                        "$eq" : 1
                                }
                        },
```

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

Второй запрос был в два раза быстрее

4. Выводы

В результате выполненной работы:

- Были выполнены CRUD-операции
- Были выполнены запросы на выборку данных, вложенные объекты, агрегацию и изменение данных
- Были выполнены запросы со ссылками и индексами в базе данных