

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

“Национальный исследовательский университет ИТМО”

Факультет инфокоммуникационных технологий

**ЛАБАРАТОРНАЯ РАБОТА №5**

**“Работа с БД в СУБД MongoDB”**

**По дисциплине:**

**“Базы данных”**

**Выполнил студент:**

Жаров Александр Павлович

Группа К32392

**Преподаватель:**

Говорова Марина Михайлова

Санкт-Петербург

2022

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

**Программное обеспечение:** Компьютерный класс

## Выполнение заданий

### Часть 1. Crud-операции. Вставка данных. Выборка данных

#### Задание 8.1.1

1. Создайте базу данных learn.

Use learn;

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

2. Заполните коллекцию единорогов unicorns:

db.unicorns.insertMany([

{name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63},

{name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},

{name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182},

{name: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99},

{name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80},

{name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},

{name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},

{name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},

{name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},

{name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},

{name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'})

)

```
> db.unicorns.insertMany([
...   {name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63},
...   {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},
...   {name: 'Unicrom', loves: ['enegon', 'redbull'], weight: 984, gender: 'm', vampires: 182},
...   {name: 'Rooodoodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99},
...   {name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80},
...   {name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},
...   {name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},
...   {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},
...   {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},
...   {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},
...   {name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'}
... ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("64778de4c5a920ab0069502b"),
    ObjectId("64778de4c5a920ab0069502c"),
    ObjectId("64778de4c5a920ab0069502d"),
    ObjectId("64778de4c5a920ab0069502e"),
    ObjectId("64778de4c5a920ab0069502f"),
    ObjectId("64778de4c5a920ab00695030"),
    ObjectId("64778de4c5a920ab00695031"),
    ObjectId("64778de4c5a920ab00695032"),
    ObjectId("64778de4c5a920ab00695033"),
    ObjectId("64778de4c5a920ab00695034"),
    ObjectId("64778de4c5a920ab00695035")
  ]
}
```

3. Используя второй способ, вставьте в коллекцию единорогов документ:  
doc = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires:  
165};

db.unicorns.insertOne(doc);

```
> doc = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165};
{
  "name" : "Dunx",
  "loves" : [
    "grape",
    "watermelon"
  ],
  "weight" : 704,
  "gender" : "m",
  "vampires" : 165
}
> db.unicorns.insertOne(doc);
{
  "acknowledged" : true,
  "insertedId" : ObjectId("64778f76c5a920ab00695036")
}
```

4. Проверяем содержимое бд

db.unicorns.find();

```

> db.unicorns.find();
{"_id": ObjectId("64778de4c5a920ab0069502b"), "name": "Horny", "loves": [ "carrot", "papaya" ], "weight": 600, "gender": "m", "vampires": 63 }
{"_id": ObjectId("64778de4c5a920ab0069502c"), "name": "Aurora", "loves": [ "carrot", "grape" ], "weight": 450, "gender": "f", "vampires": 43 }
{"_id": ObjectId("64778de4c5a920ab0069502d"), "name": "Unicrom", "loves": [ "energon", "redbull" ], "weight": 984, "gender": "m", "vampires": 182 }
{"_id": ObjectId("64778de4c5a920ab0069502e"), "name": "Rooooooodles", "loves": [ "apple" ], "weight": 575, "gender": "m", "vampires": 99 }
{"_id": ObjectId("64778de4c5a920ab0069502f"), "name": "Solnara", "loves": [ "apple", "carrot", "chocolate" ], "weight": 550, "gender": "f", "vampires": 80 }
{"_id": ObjectId("64778de4c5a920ab00695030"), "name": "Ayna", "loves": [ "strawberry", "lemon" ], "weight": 733, "gender": "f", "vampires": 40 }
{"_id": ObjectId("64778de4c5a920ab00695031"), "name": "Kenny", "loves": [ "grape", "lemon" ], "weight": 690, "gender": "m", "vampires": 39 }
{"_id": ObjectId("64778de4c5a920ab00695032"), "name": "Raleigh", "loves": [ "apple", "sugar" ], "weight": 421, "gender": "m", "vampires": 2 }
{"_id": ObjectId("64778de4c5a920ab00695033"), "name": "Leila", "loves": [ "apple", "watermelon" ], "weight": 601, "gender": "f", "vampires": 33 }
{"_id": ObjectId("64778de4c5a920ab00695034"), "name": "Pilot", "loves": [ "apple", "watermelon" ], "weight": 650, "gender": "m", "vampires": 54 }
{"_id": ObjectId("64778de4c5a920ab00695035"), "name": "Nimue", "loves": [ "grape", "carrot" ], "weight": 540, "gender": "f" }
{"_id": ObjectId("64778f76c5a920ab00695036"), "name": "Dunx", "loves": [ "grape", "watermelon" ], "weight": 704, "gender": "m", "vampires": 165 }
>
> db.unicorns.find();
{"_id": ObjectId("64778de4c5a920ab0069502b"), "name": "Horny", "loves": [ "carrot", "papaya" ], "weight": 600, "gender": "m", "vampires": 63 }
{"_id": ObjectId("64778de4c5a920ab0069502c"), "name": "Aurora", "loves": [ "carrot", "grape" ], "weight": 450, "gender": "f", "vampires": 43 }
{"_id": ObjectId("64778de4c5a920ab0069502d"), "name": "Unicrom", "loves": [ "energon", "redbull" ], "weight": 984, "gender": "m", "vampires": 182 }
{"_id": ObjectId("64778de4c5a920ab0069502e"), "name": "Rooooooodles", "loves": [ "apple" ], "weight": 575, "gender": "m", "vampires": 99 }
{"_id": ObjectId("64778de4c5a920ab0069502f"), "name": "Solnara", "loves": [ "apple", "carrot", "chocolate" ], "weight": 550, "gender": "f", "vampires": 80 }
{"_id": ObjectId("64778de4c5a920ab00695030"), "name": "Ayna", "loves": [ "strawberry", "lemon" ], "weight": 733, "gender": "f", "vampires": 40 }
{"_id": ObjectId("64778de4c5a920ab00695031"), "name": "Kenny", "loves": [ "grape", "lemon" ], "weight": 690, "gender": "m", "vampires": 39 }
{"_id": ObjectId("64778de4c5a920ab00695032"), "name": "Raleigh", "loves": [ "apple", "sugar" ], "weight": 421, "gender": "m", "vampires": 2 }
{"_id": ObjectId("64778de4c5a920ab00695033"), "name": "Leila", "loves": [ "apple", "watermelon" ], "weight": 601, "gender": "f", "vampires": 33 }
{"_id": ObjectId("64778de4c5a920ab00695034"), "name": "Pilot", "loves": [ "apple", "watermelon" ], "weight": 650, "gender": "m", "vampires": 54 }
{"_id": ObjectId("64778de4c5a920ab00695035"), "name": "Nimue", "loves": [ "grape", "carrot" ], "weight": 540, "gender": "f" }
{"_id": ObjectId("64778f76c5a920ab00695036"), "name": "Dunx", "loves": [ "grape", "watermelon" ], "weight": 704, "gender": "m", "vampires": 165 }
>

```

## Задание 8.1.2

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

```
db.unicorns.find({gender: 'm'}).limit(3).sort({name: 1});
```

```
db.unicorns.find({gender: 'f'}).limit(3);
```

```

> db.unicorns.find({gender: 'm'}).limit(3).sort({name: 1});
{"_id": ObjectId("64778f76c5a920ab00695036"), "name": "Dunx", "loves": [ "grape", "watermelon" ], "weight": 704, "gender": "m", "vampires": 165 }
{"_id": ObjectId("64778de4c5a920ab0069502b"), "name": "Horny", "loves": [ "carrot", "papaya" ], "weight": 600, "gender": "m", "vampires": 63 }
{"_id": ObjectId("64778de4c5a920ab00695031"), "name": "Kenny", "loves": [ "grape", "lemon" ], "weight": 690, "gender": "m", "vampires": 39 }
>
> db.unicorns.find({gender: 'f'}).limit(3);
{"_id": ObjectId("64778de4c5a920ab0069502c"), "name": "Aurora", "loves": [ "carrot", "grape" ], "weight": 450, "gender": "f", "vampires": 43 }
{"_id": ObjectId("64778de4c5a920ab0069502f"), "name": "Solnara", "loves": [ "apple", "carrot", "chocolate" ], "weight": 550, "gender": "f", "vampires": 80 }
{"_id": ObjectId("64778de4c5a920ab00695030"), "name": "Ayna", "loves": [ "strawberry", "lemon" ], "weight": 733, "gender": "f", "vampires": 40 }
>
> db.unicorns.find({gender: 'f'}).limit(3);
{"_id": ObjectId("64778de4c5a920ab0069502c"), "name": "Aurora", "loves": [ "carrot", "grape" ], "weight": 450, "gender": "f", "vampires": 43 }
{"_id": ObjectId("64778de4c5a920ab0069502f"), "name": "Solnara", "loves": [ "apple", "carrot", "chocolate" ], "weight": 550, "gender": "f", "vampires": 80 }
{"_id": ObjectId("64778de4c5a920ab00695030"), "name": "Ayna", "loves": [ "strawberry", "lemon" ], "weight": 733, "gender": "f", "vampires": 40 }
>

```

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

## Задание 8.1.3

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

```
db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0});
```

```

> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0});
{"_id": ObjectId("64778de4c5a920ab0069502b"), "name": "Horny", "weight": 600, "vampires": 63 }
{"_id": ObjectId("64778de4c5a920ab0069502d"), "name": "Unicrom", "weight": 984, "vampires": 182 }
{"_id": ObjectId("64778de4c5a920ab0069502e"), "name": "Rooooooodles", "weight": 575, "vampires": 99 }
{"_id": ObjectId("64778de4c5a920ab00695031"), "name": "Kenny", "weight": 690, "vampires": 39 }
{"_id": ObjectId("64778de4c5a920ab00695032"), "name": "Raleigh", "weight": 421, "vampires": 2 }
{"_id": ObjectId("64778de4c5a920ab00695034"), "name": "Pilot", "weight": 650, "vampires": 54 }
{"_id": ObjectId("64778f76c5a920ab00695036"), "name": "Dunx", "weight": 704, "vampires": 165 }
>

```

## Задание 8.1.4

- Вывести список единорогов в обратном порядке добавления.

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

```

db.unicorns.find().sort({$natural: -1});
{ "_id" : ObjectId("64778f76c5a920ab00695036"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("64778de4c5a920ab00695035"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
{ "_id" : ObjectId("64778de4c5a920ab00695034"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("64778de4c5a920ab00695033"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("64778de4c5a920ab00695032"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("64778de4c5a920ab00695031"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("64778de4c5a920ab00695030"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("64778de4c5a920ab0069502f"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "weight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("64778de4c5a920ab0069502e"), "name" : "Roooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 99 }
{ "_id" : ObjectId("64778de4c5a920ab0069502d"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("64778de4c5a920ab0069502c"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450, "gender" : "f", "vampires" : 43 }
{ "_id" : ObjectId("64778de4c5a920ab0069502b"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 63 }

```

### Задание 8.1.5

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

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

```

> 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 }

```

### 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})
```

```

> 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" }

```

### Задание 8.1.7

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

```

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

```

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

### Задание 8.1.8

1. Найти всех единорогов, не имеющих ключ vampires.

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

```
> db.unicorns.find({vampires: {$exists: false}})
{ "_id" : ObjectId("64778de4c5a920ab00695035"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "gender" : "f" }
>
```

### Задание 8.1.9

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

```
db.unicorns.find({gender: 'm'}, {_id: 0, name: 1, loves: {$slice: 1}}).sort({name: 1});
```

```
> db.unicorns.find({gender: 'm'}, {_id: 0, name: 1, loves: {$slice: 1}}).sort({name: 1});
{ "name" : "Dunx", "loves" : [ "grape" ] }
{ "name" : "Horny", "loves" : [ "carrot" ] }
{ "name" : "Kenny", "loves" : [ "grape" ] }
{ "name" : "Pilot", "loves" : [ "apple" ] }
{ "name" : "Raleigh", "loves" : [ "apple" ] }
{ "name" : "Rooooooodles", "loves" : [ "apple" ] }
{ "name" : "Unicrom", "loves" : [ "energon" ] }
>
```

Часть 2. Вложенные запросы. Курсоры. Агрегированные запросы. Изменение данных.

### Задание 8.2.1

1. Сформировать коллекцию towns

```
db.towns.insertMany([
```

```
  {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.insertMany([
...   {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"}}
...   ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("64779befc5a920ab00695037"),
    ObjectId("64779befc5a920ab00695038"),
    ObjectId("64779befc5a920ab00695039")
  ]
}

```

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'}, {mayor: 1, name: 1});
{ "_id" : ObjectId("64779befc5a920ab00695038"), "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}}, {mayor: 1, name: 1});
{ "_id" : ObjectId("64779befc5a920ab00695037"), "name" : "Punxsutawney ", "mayor" : { "name" : "Jim Wehrle" } }
```

### Задание 8.2.2

1. Сформировать функцию для вывода списка самцов единорогов.

```
male = 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(male); null;
```

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

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

3. Вывести результат, используя forEach.

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

```
> cursor.forEach(function (obj){print(obj.name);});
Dunx
Horny
```

### Задание 8.2.3

1. Вывести количество самок единорогов весом от полутонны до 600 кг.

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

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

### Задание 8.2.4

1. Вывести список предпочтений.

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



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

### Задание 8.2.5

1. Посчитать количество особей единорогов обоих полов.

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

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

### Задание 8.2.6

1. Добавление объекта с помощью save

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

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

2. Проверка вставки

```
db.unicorns.find({name: 'Barney'});
```

```
> db.unicorns.find({name: 'Barney'});
{ "_id" : ObjectId("6477a2d2c5a920ab00695046"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m" }
```

### Задание 8.2.7

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

```
db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}}, {upsert: false});
```

```
> db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}}, {upsert: false});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

2. Проверить содержимое коллекции unicorns.

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

```
> db.unicorns.find({name: 'Ayna'});
{ "_id" : ObjectId("64779cd2c5a920ab0069503f"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 800, "gender" : "f", "vampires" : 51 }
```

### Задание 8.2.8

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

```
db.unicorns.updateOne({gender: 'm', name: 'Raleigh'}, {$set:{loves: ["redbull"]}}, {upsert: false});
```

```
> db.unicorns.updateOne({gender: 'm', name: 'Raleigh'}, {$set:{loves: ["redbull"]}}, {upsert: false});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

2. Проверить результат

```
db.unicorns.find({gender: 'm', name: 'Raleigh'});
```

```
> db.unicorns.find({gender: 'm', name: 'Raleigh'});
{ "_id" : ObjectId("64779cd2c5a920ab00695041"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 2 }
> -
```

### Задание 8.2.9

1. Всем самцам единорогов увеличить количество убитых вампиров на 5.

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

```
> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
{ "acknowledged" : true, "matchedCount" : 9, "modifiedCount" : 9 }
>
```

2. Проверить результат

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

```
> db.unicorns.find({gender: 'm'});
{ "_id" : ObjectId("64779cd2c5a920ab0069503a"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600, "gender" : "m", "vampires" : 68 }
{ "_id" : ObjectId("64779cd2c5a920ab0069503c"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 984, "gender" : "m", "vampires" : 187 }
{ "_id" : ObjectId("64779cd2c5a920ab0069503d"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gender" : "m", "vampires" : 104 }
{ "_id" : ObjectId("64779cd2c5a920ab00695040"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "gender" : "m", "vampires" : 44 }
{ "_id" : ObjectId("64779cd2c5a920ab00695041"), "name" : "Raleigh", "loves" : [ "redbull" ], "weight" : 421, "gender" : "m", "vampires" : 7 }
{ "_id" : ObjectId("64779cd2c5a920ab00695043"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 650, "gender" : "m", "vampires" : 59 }
{ "_id" : ObjectId("64779cd2c5a920ab00695045"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704, "gender" : "m", "vampires" : 170 }
{ "_id" : ObjectId("6477a2d2c5a920ab00695046"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
{ "_id" : ObjectId("6477a2f0c5a920ab00695047"), "name" : "Barney", "loves" : [ "grape" ], "weight" : 340, "gender" : "m", "vampires" : 5 }
```

### Задание 8.2.10

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

```
db.towns.updateOne({name: 'Portland'}, {$unset: {"mayor.party": 1}});
```

```
> db.towns.updateOne({name: 'Portland'}, {$unset: {"mayor.party": 1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

2. Проверить результат

```
db.towns.find({name: 'Portland'}, {mayor: 1});
```

```
> db.towns.find({name: 'Portland'}, {mayor: 1});
{ "_id" : ObjectId("64779befc5a920ab00695039"), "mayor" : { "name" : "Sam Adams" } }
>
```

### Задание 8.2.11

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

```
db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
```

```
> db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

## 2. Проверить результат

```
> db.unicorns.find({name: 'Pilot'});
{ "_id" : ObjectId("64779cd2c5a920ab00695043"), "name" : "Pilot", "loves" : [ "apple", "watermelon", "chocolate" ],
  "weight" : 650, "gender" : "m", "vampires" : 59 }
```

### Задание 8.2.12

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

```
db.unicorns.updateOne(
  {name: 'Aurora'},
  {$push: {loves: {$each: ['sugar', 'lemons']}}});
```

```
> db.unicorns.updateOne(
...   {name: 'Aurora'},
...   {$push: {loves: {$each: ['sugar', 'lemons']}}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.unicorns.find({name: 'Aurora'})
```

## 2. Проверить результат

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

```
> db.unicorns.find({name: 'Aurora'});
{ "_id" : ObjectId("64779cd2c5a920ab0069503b"), "name" : "Aurora", "loves" : [ "carrot", "grape", "sugar", "lemons" ],
  "weight" : 450, "gender" : "f", "vampires" : 43 }
>
```

### Задание 8.2.13

1. Создайте коллекцию towns:

```
db.towns.insertMany([
  {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.insertMany([
...   {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"}}
...   ]);
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6477a73dc5a920ab00695048"),
    ObjectId("6477a73dc5a920ab00695049"),
    ObjectId("6477a73dc5a920ab0069504a")
  ]
}
```

2. Удалите документы с беспартийными мэрами

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

```
> db.towns.deleteMany({"mayor.party": {$exists: false}});
{ "acknowledged" : true, "deletedCount" : 1 }
```

3. Проверьте содержание коллекции.

```
db.towns.find({}, {mayor: 1});
```

```
> db.towns.find({}, {mayor: 1});
{ "_id" : ObjectId("6477a73dc5a920ab00695049"), "mayor" : { "name" : "Michael Bloomberg", "party" : "I" } }
{ "_id" : ObjectId("6477a73dc5a920ab0069504a"), "mayor" : { "name" : "Sam Adams", "party" : "D" } }
>
```

4. Очистите коллекцию.

```
db.towns.drop();
```

```
> db.towns.drop();
true
```

5. Просмотрите список доступных коллекций.

```
show collections
```

```
> show collections
unicorns
```

### Часть 3. Ссылки и работа с индексами

#### Задание 8.3.1

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

```
db.habitats.insertMany([
  { _id: 'ru', name: 'Russia', description: 'description '},
  { _id: 'us', name: 'America', description: 'description '},
  { _id: 'uk', name: 'United kingdom', description: 'description ' }
]);
```

```
> db.habitats.insertMany([
...   { _id: 'ru', name: 'Russia', description: 'description '},
...   { _id: 'us', name: 'America', description: 'description '},
...   { _id: 'uk', name: 'United kingdom', description: 'description ' }
... ]);
{ "acknowledged" : true, "insertedIds" : [ "ru", "us", "uk" ] }
>
```

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

```
db.unicorns.updateOne({name: 'Horny'},
  {$set: {habitat: {$ref:"habitats", $id: 'ru'}}});
```

```
db.unicorns.updateOne({name: 'Aurora'},
  {$set: {habitat: {$ref:"habitats", $id: 'us'}}});
```

```
> db.unicorns.updateOne({name: 'Aurora'},
...   {$set: {habitat: {$ref:"habitats", $id: 'us'}}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

3. Проверьте содержание коллекции единорогов.

```
db.unicorns.find();
```

```
> db.unicorns.find();
{ "_id" : ObjectId("6477c603c5a920ab00695057"), "name" : "Horny", "loves" : [ "carrot", "papaya" ], "weight" : 600,
  "gender" : "m", "vampires" : 63, "habitat" : DBRef("habitats", "ru") }
{ "_id" : ObjectId("6477c603c5a920ab00695058"), "name" : "Aurora", "loves" : [ "carrot", "grape" ], "weight" : 450,
  "gender" : "f", "vampires" : 43, "habitat" : DBRef("habitats", "us") }
{ "_id" : ObjectId("6477c603c5a920ab00695059"), "name" : "Unicrom", "loves" : [ "energon", "redbull" ], "weight" : 9
4, "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("6477c603c5a920ab0069505a"), "name" : "Rooooooodles", "loves" : [ "apple" ], "weight" : 575, "gend
er" : "m", "vampires" : 99 }
{ "_id" : ObjectId("6477c603c5a920ab0069505b"), "name" : "Solnara", "loves" : [ "apple", "carrot", "chocolate" ], "w
eight" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("6477c603c5a920ab0069505c"), "name" : "Ayna", "loves" : [ "strawberry", "lemon" ], "weight" : 733
, "gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("6477c603c5a920ab0069505d"), "name" : "Kenny", "loves" : [ "grape", "lemon" ], "weight" : 690, "g
ender" : "m", "vampires" : 39 }
{ "_id" : ObjectId("6477c603c5a920ab0069505e"), "name" : "Raleigh", "loves" : [ "apple", "sugar" ], "weight" : 421,
  "gender" : "m", "vampires" : 2 }
{ "_id" : ObjectId("6477c603c5a920ab0069505f"), "name" : "Leia", "loves" : [ "apple", "watermelon" ], "weight" : 601
, "gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("6477c603c5a920ab00695060"), "name" : "Pilot", "loves" : [ "apple", "watermelon" ], "weight" : 65
, "gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("6477c603c5a920ab00695061"), "name" : "Nimue", "loves" : [ "grape", "carrot" ], "weight" : 540, "
gender" : "f" }
{ "_id" : ObjectId("6477c603c5a920ab00695062"), "name" : "Dunx", "loves" : [ "grape", "watermelon" ], "weight" : 704
, "gender" : "m", "vampires" : 165 }
>
```

### 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
}
```

### Задание 8.3.3

1. Получите информацию о всех индексах коллекции unicorns .

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

```

> 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.dropIndex('name_1');
```

```

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

```

3. Попробуйте удалить индекс для идентификатора.

```
db.unicorns.dropIndex('_id_');
```

```

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

```

#### Задание 8.3.4

1. Создайте объемную коллекцию numbers, задействовав курсор:

```
arr = []
```

```
for (i=0; i<100000; i++)
```

```
{arr.push({value: i})}
```

```
db.numbers.insertMany(arr);
```

2. Выберите последних четыре документа.

```
db.numbers.find({}).sort({value: -
```

```
1}).limit(4).explain("executionStats").executionStats.executionTimeMillis;
```

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

```
> print(measureMeanTime(5));  
203
```

4. Создайте индекс для ключа `value`.

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

5. Получите информацию о всех индексах коллекции `numbers`

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

6. Выполните запрос 2
7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

```
> print(measureMeanTime(5));  
11
```

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

Создание индекса значительно уменьшило время выполнения запроса.

### Вывод

В процессе выполнения лабораторной работы мы изучили основы работы с MongoDB, создание баз данных и коллекций, а также выполнение базовых операций CRUD (создание, чтение, обновление, удаление). Мы также получили практический опыт



работы с индексами в MongoDB, которые помогают улучшить производительность и эффективность запросов в базе данных без SQL.