Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №6 «Работа с БД в СУБД MongoDB»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Красюк К.А.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы

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

Практическое задание

Практическое задание 2.1.1:

1) Создайте базу данных learn.

```
27017> use learn
switched to db learn
```

2) Заполните коллекцию единорогов unicorns

```
learn> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e4') }
}
learn> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e5') }
}
learn> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e6') }
}
learn> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e7') }
}
learn> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e8') }
}
learn> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8e9') }
}
learn> db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8ea') }
}
learn>
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8eb') }
}
learn> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8ec') }
}
```

```
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8ed') }
}
learn> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6585e12dace7a5a36953c8ee') }
}
```

3) Используя второй способ, вставьте в коллекцию единорогов документ

```
learn> document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
{
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
learn> db.unicorns.insert(document)
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582d431a0b6d8b88cc22489') }
}
```

4) Проверьте содержимое коллекции с помощью метода find

```
learn> db.unicorns.find()
[
  {
    _id: ObjectId('6582cc86a0b6d8b88cc2247e'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6582cd16a0b6d8b88cc2247f'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6582cd77a0b6d8b88cc22480'),
    name: 'Unicorm',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6582ce81a0b6d8b88cc22482'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Anya',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'm',
    vampires: 40
```

Практическое задание 2.2.1:

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

Запрос для получения всех самцов:

```
learn> db.unicorns.find({gender: 'm'}).sort({name: 1});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e4'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ea'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8eb'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e7'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e6'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
```

Запрос для получения всех самок:

```
learn> db.unicorns.find({gender: 'f'}).sort({name: 1});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e5'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e9'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ec'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ee'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e8'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
]
```

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

Запрос с помощью функции findOne:

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'});
{
  _id: ObjectId('6585e12dace7a5a36953c8e5'),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

Запрос с помощью функции limit:

```
learn> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e5'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Практическое задание 2.2.2:

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

```
learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0}).sort({name: 1});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e4'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ea'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8eb'),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e7'),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e6'),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  }
]
```

Практическое задание 2.2.3:

Вывести список единорогов в обратном порядке добавления.

```
learn> db.unicorns.find().sort({$natural: -1});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8ee'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ec'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8eb'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ea'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e9'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
```

```
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e8'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e7'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e6'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e5'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e4'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]
```

Практическое задание 2.1.4:

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

```
learn> 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' }
]
```

Практическое задание 2.3.1:

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

```
learn> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 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 2.3.2:

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

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

Практическое задание 2.3.3:

Найти всех единорогов, не имеющих ключ vampires.

```
learn> db.unicorns.find({ vampires: { $exists:false } }, { _id: 0 });
[
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Практическое задание 2.3.4:

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

```
learn> db.unicorns.find({gender: 'm'}, {loves: {$slice: 1}}).sort({name: 1});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e4'),
    name: 'Horny',
    loves: [ 'carrot' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ea'),
    name: 'Kenny',
    loves: [ 'grape' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
```

```
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    loves: [ 'apple' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8eb'),
    name: 'Raleigh',
    loves: [ 'apple' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e7'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e6'),
    name: 'Unicrom',
    loves: [ 'energon' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
]
```

Практическое задание 3.1.1:

1) Создайте коллекцию towns, включающую следующие документы:

```
learn> db.createCollection('towns')
{ ok: 1 }
learn> db.towns.insertMany([{
... name: 'Punxsutawney',
... population: 6200,
... last_sensus: ISODate('2008-01-31'),
... famous_for: [''],
... mayor: {name: 'Jim Wehrle'}
... },
... {
... name: 'New York',
... population: 22200000,
... last_sensus: ISODate('2009-07-31'),
... famous_for: ['status of liberty', 'food'],
... mayor: {name: 'Michael Bloomberg', party: 'I
... },
... {
... name: 'Portland',
... population: 528000,
... last_sensus: ISODate('2009-07-20'),
... famous_for: ['beer', 'food'],
... mayor: {name: 'Sam Adams', party: 'D'}
... }]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6583e2485d5a5a6dbd0eee7d'),
    '1': ObjectId('6583e2485d5a5a6dbd0eee7e'),
    '2': ObjectId('6583e2485d5a5a6dbd0eee7f')
  }
}
```

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

```
learn> db.towns.find({'mayor.party': 'I'}, {name: 1,'mayor.name': 1, _id: 0})
[ { name: 'New York', mayor: { name: 'Michael Bloomberg' } } ]
```

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

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

Практическое задание 3.1.2:

1. Сформировать функцию для вывода списка самцов единорогов.
2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя forEach.

```
learn> function printMaleUnicorns() {learn> function printMaleUnicorns()
... {
... var cursor = db.unicorns.find({gender: 'm'});
... null;
... cursor.sort({name: 1}).limit(2);
... cursor.forEach(function(u)
... {print(u.name);}
... );
... }
[Function: printMaleUnicorns]
learn> printMaleUnicorns()
Anya
Dunx
```

Практическое задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 600}}).count();
2
```

Практическое задание 3.2.2:

Вывести список предпочтений

- $group: агрегатор, который вернет новый документ
- _id: указывает на ключ, по которому надо проводить группировку ($+назване поля)
- $sum: оператор для вычисления.

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

Практическое задание 3.2.3:

Посчитать количество особей единорогов обоих полов.

```
learn> db.unicorns.aggregate({"$group":{_id:"$gender",count:{$sum:1}}});
[ { _id: 'm', count: 6 }, { _id: 'f', count: 5 } ]
```

Практическое задание 3.3.1:

1) Выполнить команду:

> db.unicorns.save({name: 'Barny', loves: ['grape'],

weight: 340, gender: 'm'})

2) Проверить содержимое коллекции unicorns.

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

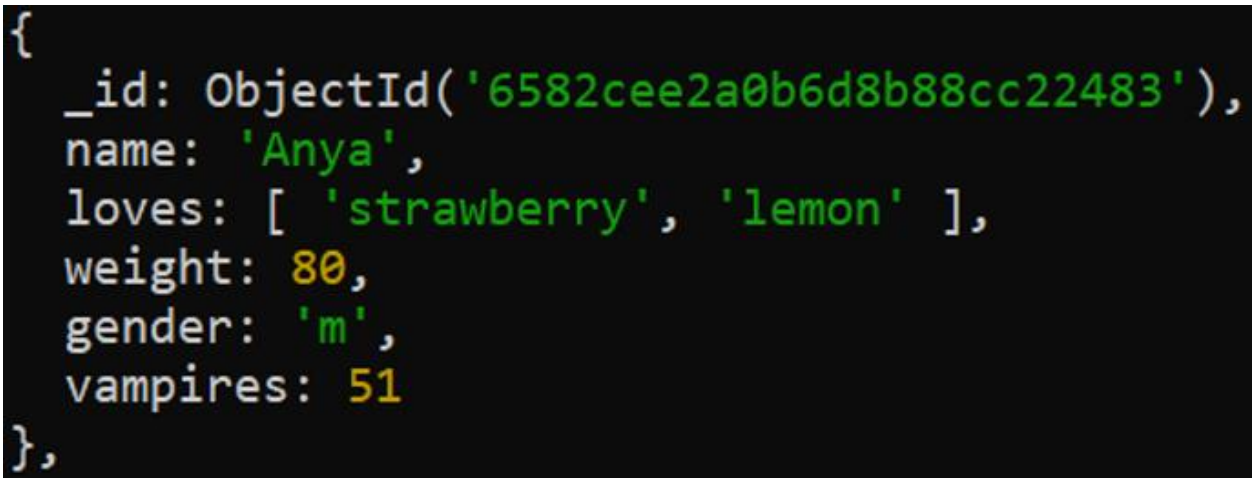
```
{
  _id: ObjectId('65855d1a0f23cd33f7b8a5eb'),
  name: 'Barny',
  loves: [ 'grape' ],
  weight: 340,
  gender: 'm'
}
```

Практическое задание 3.3.2:

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

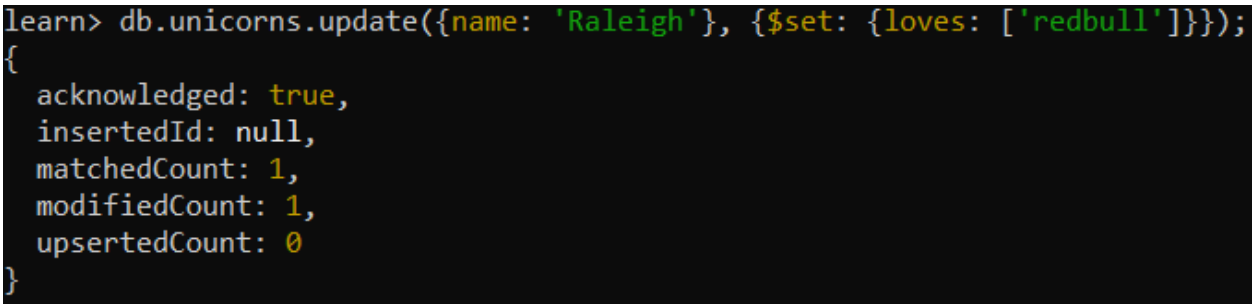
```
learn> db.unicorns.update({name: 'Anya'}, {$set: {weight: 80, vampires: 51}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2) Проверить содержимое коллекции unicorns.

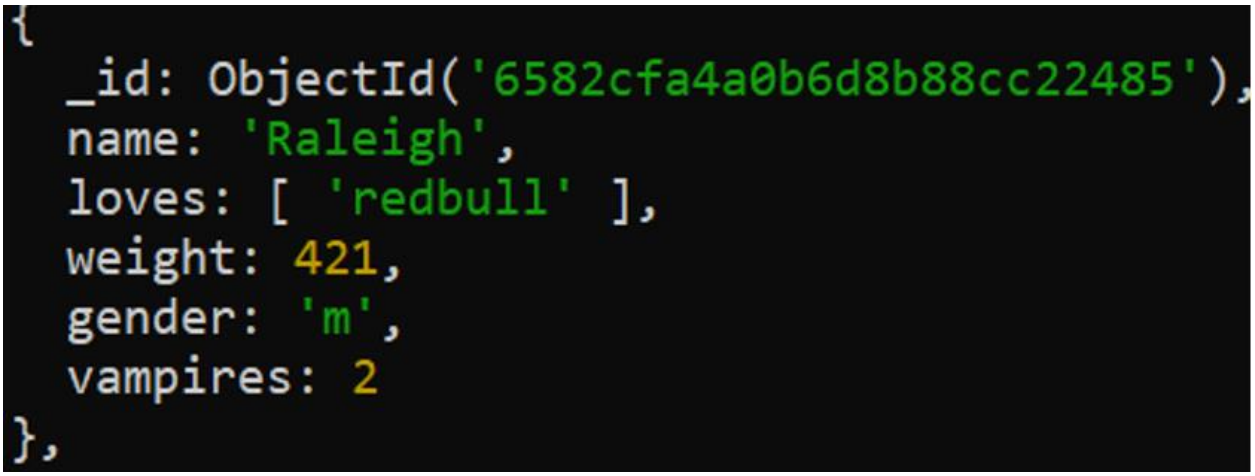


Практическое задание 3.3.3:

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



2) Проверить содержимое коллекции unicorns.



Практическое задание 3.3.4:

1) Всем самцам единорогов увеличить количество убитых вапмиров на 5.

```
learn> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
```

2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.find({gender: 'm'});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e4'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e6'),
    name: 'Unicrom',
learn>
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8e7'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ea'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 44
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8eb'),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 7
  },
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
```

Практическое задание 3.3.5:

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

```
learn> db.towns.update({name: 'Portland'}, {$set: {'mayor.party': undefined}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2) Проверить содержимое коллекции towns.

```
{
  _id: ObjectId('6583e2485d5a5a6dbd0eee7f'),
  name: 'Portland',
  population: 528000,
  last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
  famous_for: [ 'beer', 'food' ],
  mayor: { name: 'Sam Adams', party: null }
}
```

Практическое задание 3.3.6:

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

```
learn> db.unicorns.update({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.find({name: 'Pilot'});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8ed'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
```

Практическое задание 3.3.7:

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

```
learn> db.unicorns.update({name: 'Aurora'}, {$push: {loves: {$each: ['sugar', 'lemons']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.find({name: 'Aurora'});
[
  {
    _id: ObjectId('6585e12dace7a5a36953c8e5'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Практическое задание 3.4.1:

1) Создайте коллекцию towns, включающую следующие документы:
2) Удалите документы с беспартийными мэрами.

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

3) Проверьте содержание коллекции.

```
learn> db.towns.find()
[
  {
    _id: ObjectId('6583e2485d5a5a6dbd0eee7e'),
    name: 'New York',
    population: 22200000,
    last_sensus: ISODate('2009-07-31T00:00:00.000Z'),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId('6583e2485d5a5a6dbd0eee7f'),
    name: 'Portland',
    population: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: null }
  }
```

4) Очистите коллекцию.

```
learn> db.towns.deleteMany({})
{ acknowledged: true, deletedCount: 2 }
```

5) Просмотрите список доступных коллекций.

```
learn> show collections
towns
unicorns
```

Практическое задание 4.1.1:

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

```
learn> db.createCollection('areas')
{ ok: 1 }
learn> db.areas.insert({_id: 'eq', name: 'Equestria', description: 'Satrting base location'})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 'eq' } }
learn> db.areas.insert({_id: 'ef', name: 'Everfree Forest', description: 'Magic forest with colorfull plants'})
{ acknowledged: true, insertedIds: { '0': 'ef' } }
learn> db.areas.insert({_id: 'fn', name: 'Frozen North', description: 'Snowy mountains'})
{ acknowledged: true, insertedIds: { '0': 'fn' } }
learn> db.areas.insert({_id: 'sl', name: 'Saddle Lake', description: 'Big clean lake'})
{ acknowledged: true, insertedIds: { '0': 'sl' } }
```

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

```
learn> db.unicorns.update({name: 'Pilot'}, {$set: {area: {$ref: 'areas', $id: 'fn'}}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.update({name: 'Anya'}, {$set: {area: {$ref: 'areas', $id: 'sl'}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

3) Проверьте содержание коллекции едиорогов.

```
{
  _id: ObjectId('6582d136a0b6d8b88cc22487'),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon', 'chocolate' ],
  weight: 650,
  gender: 'm',
  vampires: 59,
  area: DBRef('areas', 'fn')
},
```

```
{
  _id: ObjectId('6582cee2a0b6d8b88cc22483'),
  name: 'Anya',
  loves: [ 'strawberry', 'lemon' ],
  weight: 80,
  gender: 'm',
  vampires: 56,
  area: DBRef('areas', 'sl')
},
```

Практическое задание 4.2.1:

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

```
learn> db.unicorns.ensureIndex({'name': 1}, {'unique': true});
[ 'name_1' ]
```

Практическое задание 4.3.1:

1) Получите информацию о всех индексах коллекции unicorns

```
learn> 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.

```
learn> db.unicorns.dropIndexes();
{
  nIndexesWas: 2,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
```

3) Попытайтесь удалить индекс для идентификатора.

```
learn> db.unicorns.dropIndex('_id_');
MongoServerError: cannot drop _id index
```

Практическое задание 4.4.1:

1) Создайте объемную коллекцию numbers, задействовав курсор:

```
learn> db.createCollection('numbers')
{ ok: 1 }
learn> for (i=0; i < 100000; i++) {db.numbers.insert({value: i})}
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('658558d30f23cd33f7b8a5ea') }
```

2) Выберите последних четыре документа.

```
learn> db.numbers.find().sort({value: -1}).limit(4).explain('executionStats')
```

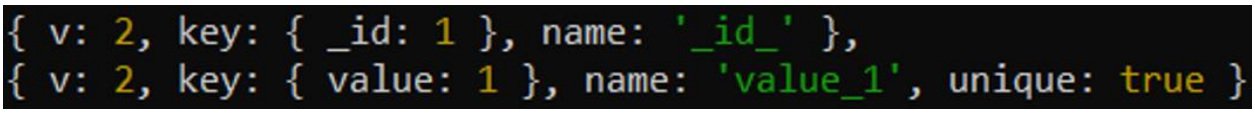
3) Проанализируйте план выполнения запроса. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)

```
  executionStats: {
    executionSuccess: true,
    nReturned: 4,
    executionTimeMillis: 92,
    totalKeysExamined: 0,
    totalDocsExamined: 100000,
    executionStages: {
```

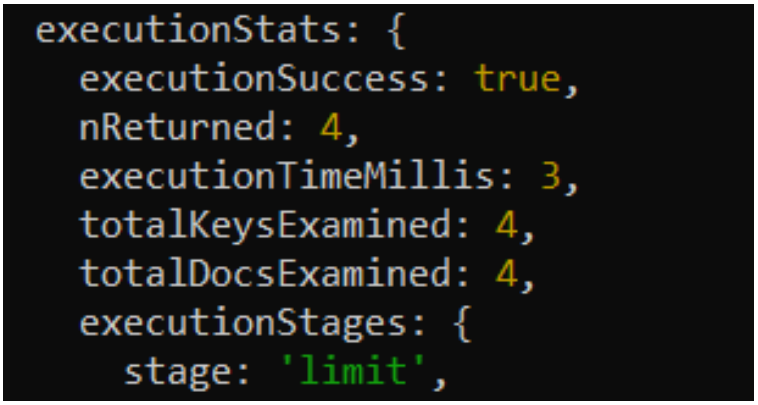
4) Создайте индекс для ключа value.

```
learn> db.numbers.ensureIndex({'value': 1}, {'unique': true})
[ 'value_1' ]
```

5) Получите информацию о всех индексах коллекции numbres.



6) Выполните запрос 2.



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

Без использования индексов для выполнения запроса потребовалось больше времени, чем с использованием индексов.

Вывод

В ходе данной лабораторной работы я овладела 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.