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: Гнеушев В. А.

Факультет: ИКТ

Группа: К3239

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.

Вставка документов в коллекцию

Практическое задание 2.1.1:

1) Создайте базу данных learn.
2) Заполните коллекцию единорогов unicorns

```
db> use learn
switched to db learn
learn> db.createCollection('unicorns');
{ ok: 1 }
learn> db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d492063fd6525f90fae94') }
}
learn> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d492463fd6525f90fae95') }
}
learn> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d492763fd6525f90fae96') }
}
learn> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d492a63fd6525f90fae97') }
}
learn> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d492e63fd6525f90fae98') }
}
learn> db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493163fd6525f90fae99') }
}
learn> db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493463fd6525f90fae9a') }
}
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493763fd6525f90fae9b') }
}
learn> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493a63fd6525f90fae9c') }
}
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493d63fd6525f90fae9d') }
}
learn> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d493f63fd6525f90fae9e') }
}
```

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.insertOne(document);
{
  acknowledged: true,
  insertedId: ObjectId('657d49a263fd6525f90fae9f')
}
```

4) Проверьте содержимое коллекции с помощью метода find.

```
learn> db.unicorns.find({});
[
  {
    _id: ObjectId('657d492063fd6525f90fae94'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('657d492463fd6525f90fae95'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('657d492763fd6525f90fae96'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('657d492a63fd6525f90fae97'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('657d492e63fd6525f90fae98'),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId('657d493163fd6525f90fae99'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('657d493463fd6525f90fae9a'),
    name: 'Kenny',
```

Выборка данных из БД

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.

Список самцов:

```
learn> db.unicorns.find({gender: "m"}).sort({name: 1});
[
  {
    _id: ObjectId('657d49a263fd6525f90fae9f'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('657d492063fd6525f90fae94'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('657d493463fd6525f90fae9a'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('657d493d63fd6525f90fae9d'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('657d493763fd6525f90fae9b'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('657d492a63fd6525f90fae97'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('657d492763fd6525f90fae96'),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
]
```

Список самок:

```
learn> db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3);
[
  {
    _id: ObjectId('657d492463fd6525f90fae95'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('657d493163fd6525f90fae99'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId('657d493a63fd6525f90fae9c'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

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

```
learn> db.unicorns.find({gender: "f", loves: "carrot"}).limit(1);
[
  {
    _id: ObjectId('657d492463fd6525f90fae95'),
    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 предпочтениях и поле.

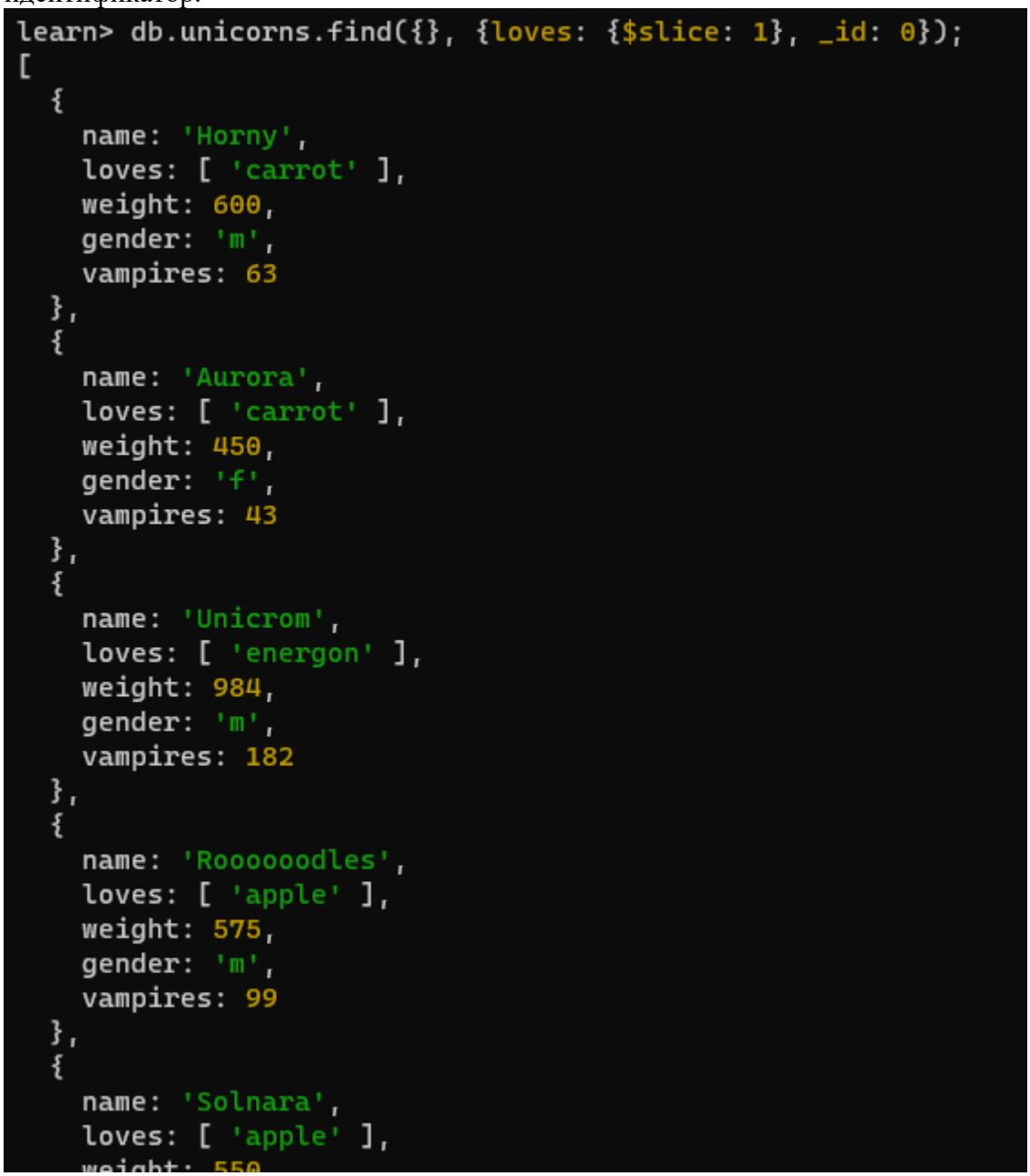
```
learn> db.unicorns.find({gender: "m"}, {gender: 0, loves: 0}).sort({name: 1});
[
  {
    _id: ObjectId('657d49a263fd6525f90fae9f'),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  },
  {
    _id: ObjectId('657d492063fd6525f90fae94'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId('657d493463fd6525f90fae9a'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId('657d493d63fd6525f90fae9d'),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId('657d493763fd6525f90fae9b'),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId('657d492a63fd6525f90fae97'),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId('657d492763fd6525f90fae96'),
    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('657d49a263fd6525f90fae9f'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('657d493f63fd6525f90fae9e'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('657d493d63fd6525f90fae9d'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('657d493a63fd6525f90fae9c'),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('657d493763fd6525f90fae9b'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('657d493463fd6525f90fae9a'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
```

Практическое задание 2.1.4:
Вывести список единорогов с названием первого любимого предпочтения, исключив

идентификатор.



```
learn> db.unicorns.find({}, {loves: {$slice: 1}, _id: 0});
[
  {
    name: 'Horny',
    loves: [ 'carrot' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Aurora',
    loves: [ 'carrot' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    name: 'Unicrom',
    loves: [ 'energon' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple' ],
    weight: 550
```

Логические операторы

Практическое задание 2.3.1:
Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора.

```
learn> db.unicorns.find({weight: {$gt: 500, $lt: 700}}, {_id: 0});
[
  {
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    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'
  }
]
```

Практическое задание 2.3.2:
Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.

```
learn> db.unicorns.find({weight: {$gt: 500}, gender: "m", loves: ["grape", "lemon"]}, {_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: ObjectId('657d493f63fd6525f90fae9e'),
    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"}, {_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: 'Roooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

Запрос к вложенным объектам

Практическое задание 3.1.1:

1) Создайте коллекцию towns, включающую следующие документы:

```
learn> db.towns.insert({name: "Punxsutawney ",  populatiuon: 6200,  last_sensus: ISODate("2008-01-31"),  famous_for: [""],  mayor: {name: "Jim Wehrle"}});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d50f663fd6525f90faea5') }
}
learn> db.towns.insert({name: "New York", populatiuon: 22200000,  last_sensus: ISODate("2009-07-31"),  famous_for: ["status of liberty", "food"], mayor: {name: "Michael Bloomberg"}, party: "I"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d50fa63fd6525f90faea6') }
}
learn> db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: {name: "Sam Adams"}, party: "D"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657d50fd63fd6525f90faea7') }
}
```

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

```
learn> db.towns.find({party: "I"}, {name: 1, mayor: 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({party: {$exists: false}}, {name: 1, mayor: 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 task3_1_2(){
...     var cursor = db.unicorns.find({gender: "m"}); null;
...     cursor.sort({name: 1}).limit(2); null;
...     cursor.forEach(function(unicorn) {
...             print(unicorn.name);
...     });
... }
[Function: task3_1_2]
learn> task3_1_2();
Dunx
Horny
```

Агрегированные запросы

Практическое задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
learn> db.unicorns.find({gender: "f", weight: {$gt: 500, $lt: 600}}).count();
2
```

Практическое задание 3.2.2:
Вывести список предпочтений.

```
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: 'f', count: 5 }, { _id: 'm', count: 7 } ]
```

Редактирование данных

Практическое задание 3.3.1:

1) Выполнить команду:
   > db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
2) Проверить содержимое коллекции unicorns.

```
  },
  {
    _id: ObjectId('657d7c7863fd6525f90faea8'),
    name: 'Barny',
    loves: [ 'grape' ],
    weight: 340,
    gender: 'm'
  }
]
```

Практическое задание 3.3.2:

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

```
learn> db.unicorns.update({name: "Ayna"}, {$set: {weight: 800, vampires: 51}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Ayna"});
[
  {
    _id: ObjectId('657e9742d8b61504518dec4c'),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]
```

Практическое задание 3.3.3:

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

```
learn> db.unicorns.update({name: "Raleigh"}, {$set: {loves: ["redbull"]}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Raleigh"});
[
  {
    _id: ObjectId('657e9748d8b61504518dec4e'),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
```

Практическое задание 3.3.4:

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

```
learn> db.unicorns.update({gender: "m"}, {$inc: {vampires: 5}}, {multi: true});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
```

Практическое задание 3.3.5:

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

```
learn> db.towns.updateOne({name: "Portland"}, {$unset: {"mayor.party": 1}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
learn> db.towns.find();
[
  {
    _id: ObjectId('657d50f663fd6525f90faea5'),
    name: 'Punxsutawney ',
    populatiuon: 6200,
    last_sensus: ISODate('2008-01-31T00:00:00.000Z'),
    famous_for: [ '' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId('657d50fa63fd6525f90faea6'),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate('2009-07-31T00:00:00.000Z'),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg' },
    party: 'I'
  },
  {
    _id: ObjectId('657d50fd63fd6525f90faea7'),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' },
    party: 'D'
  }
]
```

Практическое задание 3.3.6:

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

```
learn> db.unicorns.updateOne({name: "Pilot" }, {$addToSet: {loves: "chocolate"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Pilot"});
[
  {
    _id: ObjectId('657d493d63fd6525f90fae9d'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    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.
2) Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.updateOne({name: "Aurora" }, {$addToSet: {loves: {$each: ["sugar", "lemons"]}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.find({name: "Aurora"});
[
  {
    _id: ObjectId('657d492463fd6525f90fae95'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Удаление данных из коллекции

Практическое задание 3.4.1:

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

```
learn> db.towns.insert({name: "Punxsutawney ",
... popujatiuon: 6200,
... last_sensus: ISODate("2008-01-31"),
... famous_for: ["phil the groundhog"],
... mayor: {
...     name: "Jim Wehrle"
... }});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657e9d17d8b61504518dec56') }
}
learn> db.towns.insert({name: "New York",
... popujatiuon: 22200000,
... last_sensus: ISODate("2009-07-31"),
... famous_for: ["status of liberty", "food"],
... mayor: {
...     name: "Michael Bloomberg",
... party: "I"}});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657e9d24d8b61504518dec57') }
}
learn> db.towns.insert({name: "Portland",
... popujatiuon: 528000,
... last_sensus: ISODate("2009-07-20"),
... famous_for: ["beer", "food"],
... mayor: {
...     name: "Sam Adams",
... party: "D"}});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('657e9d2ed8b61504518dec58') }
}
learn> db.towns.remove({"mayor.party": {$exists: false}});
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.find();
[
  {
    _id: ObjectId('657e9d24d8b61504518dec57'),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate('2009-07-31T00:00:00.000Z'),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId('657e9d2ed8b61504518dec58'),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate('2009-07-20T00:00:00.000Z'),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
```

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

learn> show collections
towns
unicorns
learn>
```

Ссылки в БД

Практическое задание 4.1.1:

1) Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2) Включите для нескольких единорогов в документы ссылку на зону обитания, использую второй способ автоматического связывания.
3) Проверьте содержание коллекции едиорогов.

```
learn> db.habitats.insert({_id: "USA", name: "United States of America", description: "US country"});
{ acknowledged: true, insertedIds: { '0': 'USA' } }
learn> db.habitats.insert({_id: "RU", name: "Russia", desciption: "Russian Federation country"});
{ acknowledged: true, insertedIds: { '0': 'RU' } }
learn> db.habitats.insert({_id: "UK", name: "United Kingdom", desciption: "United Kingdom of Great Britain country"});
{ acknowledged: true, insertedIds: { '0': 'UK' } }
```

```
learn> db.unicorns.update({name: "Pilot"}, {$set: {habitat: {$ref: "habitats", $id: "USA"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.update({name: "Dunx"}, {$set: {habitat: {$ref: "habitats", $id: "RU"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.update({name: "Nimue"}, {$set: {habitat: {$ref: "habitats", $id: "UK"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
  {
    _id: ObjectId('657e9e6ed8b61504518dec62'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54,
    habitat: DBRef('habitats', 'USA')
  },
  {
    _id: ObjectId('657e9e71d8b61504518dec63'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f',
    habitat: DBRef('habitats', 'UK')
  },
  {
    _id: ObjectId('657e9e75d8b61504518dec64'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165,
    habitat: DBRef('habitats', 'RU')
  }
]
```

Настройка индексов

Практическое задание 4.2.1:
Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.

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

Управление индексами

Практическое задание 4.3.1:

1) Получите информацию о всех индексах коллекции unicorns
2) Удалите все индексы, кроме индекса для идентификатора.
3) Попытайтесь удалить индекс для идентификатора.

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

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

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

План запроса

Практическое задание 4.4.1:

1) Создайте объемную коллекцию numbers, задействовав курсор:
   for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
2) Выберите последних четыре документа.
3) Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
4) Создайте индекс для ключа value.
5) Получите информацию о всех индексах коллекции numbres.
6) Выполните запрос 2.
7) Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8) Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

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

```
executionSuccess: true,
nReturned: 4,
executionTimeMillis: 50,
totalKeysExamined: 0,
totalDocsExamined: 100000,
```

```
learn> db.numbers.ensureIndex({"value": 1}, {"unique": true});
[ 'value_1' ]
learn> db.numbers.find().sort({value: -1}).limit(4).explain("executionStats");
```

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 1,
  totalKeysExamined: 4,
  totalDocsExamined: 4,
```

Без индекса запрос выполнился за 50 миллисекунд, а с индексом – за 1. Время выполнения уменьшилось в 50 раз.

Вывод

В ходе данной лабораторной работы я научился работать в СУБД MongoDB. Я овладел практическими навыками работы с CRUD-операциями, со вложенными объектами в коллекции, с агрегациями и изменениями данных, с ссылками и индексами в базе данных MongoDB.