

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»**

Факультет инфокоммуникационных технологий

Дисциплина:
«Базы данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
«РАБОТА В БД С СУБД MongoDB»**

Выполнила:
студентка группы К32421
Панкова Кристина
Сергеевна

(подпись)

Проверила:
Говорова Марина Михайловна

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

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 8.1.1:

1) Создайте базу данных *learn*.

2) Заполните коллекцию единорогов *unicorns*:

```
db.unicorns.insert({name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});

db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
```

```
db.unicorns.insert({name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63});
db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
db.unicorns.insert({name: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64746e352bf5b5fdb2cb60337")
  }
}
test>|
```

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

```
{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(doc);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64746ea42bf5bfdb2cb60338")
  }
}
test>

```

4) Проверьте содержимое коллекции с помощью метода *find*.

```

> db.unicorns.find();
< {
  _id: ObjectId("64746e352bf5bfdb2cb6032d"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("64746e352bf5bfdb2cb6032e"),

```

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: 'f'}, {name: 1, _id: 0}).sort({name: 1}).limit(3).toArray();
< [ { name: 'Aurora' }, { name: 'Ayna' }, { name: 'Leia' } ]
test>

```

```

> db.unicorns.find({gender: 'm'}, {name: 1, _id: 0}).sort({name: 1}).toArray();
< [
  { name: 'Dunx' },
  { name: 'Horny' },
  { name: 'Kenny' },
  { name: 'Pilot' },
  { name: 'Raleigh' },
  { name: 'Roooooodles' },
  { name: 'Unicrom' }
]
test>

```

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

```

> db.unicorns.find({gender: 'f', loves: 'carrot'}, {name: 1});
< {
  _id: ObjectId("64746e352bf5bfdb2cb6032e"),
  name: 'Aurora'
}
{
  _id: ObjectId("64746e352bf5bfdb2cb60331"),
  name: 'Solnara'
}
{
  _id: ObjectId("64746e352bf5bfdb2cb60337"),
  name: 'Nimue'
}
test>

```

```

> db.unicorns.findOne({gender: 'f', loves: 'carrot'}).name;
< Aurora

```

```

> db.unicorns.find({gender: 'f', loves: 'carrot'}, {name: 1}).limit(1);
< {
  _id: ObjectId("64746e352bf5bfdb2cb6032e"),
  name: 'Aurora'
}

```

Практическое 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, _id: 0}).toArray();
< [
  { name: 'Horny', weight: 600, vampires: 63 },
  { name: 'Unicrom', weight: 984, vampires: 182 },
  { name: 'Roooooodles', weight: 575, vampires: 99 },
  { name: 'Kenny', weight: 690, vampires: 39 },
  { name: 'Raleigh', weight: 421, vampires: 2 },
  { name: 'Pilot', weight: 650, vampires: 54 },
  { name: 'Dunx', weight: 704, vampires: 165 }
]
test>
```

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

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

Практическое задание 8.1.5:

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

```
> db.unicorns.find({}, {name: 1, _id:0, loves: {$slice: 1}}).sort({ $natural: -1 }).toArray();
< [
  { name: 'Dunx', loves: [ 'grape' ] },
  { name: 'Nimue', loves: [ 'grape' ] },
  { name: 'Pilot', loves: [ 'apple' ] },
  { name: 'Leia', loves: [ 'apple' ] },
  { name: 'Raleigh', loves: [ 'apple' ] },
  { name: 'Kenny', loves: [ 'grape' ] },
  { name: 'Ayna', loves: [ 'strawberry' ] },
  { name: 'Solnara', loves: [ 'apple' ] },
  { name: 'Rooooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] },
  { name: 'Aurora', loves: [ 'carrot' ] },
  { name: 'Horny', loves: [ 'carrot' ] }
]
```

Практическое задание 8.1.6:

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

```
> db.unicorns.find({gender: 'f', weight: {$gte:500, $lte:700}}, { _id:0, loves: 0}).toArray();
< [
  { name: 'Solnara', weight: 550, gender: 'f', vampires: 80 },
  { name: 'Leia', weight: 601, gender: 'f', vampires: 33 },
  { name: 'Nimue', weight: 540, gender: 'f' }
]
test>
```

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find({vampires: {$exists: false}}).toArray();
< [
  {
    _id: ObjectId("64746e352bf5bfdb2cb60337"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
test>
```

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, _id:0, loves: {$slice: 1}}).sort({ name: 1 }).toArray();
< [
  { 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' ] }
]
test>
```

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"}}
```

- 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"}, {_id: 0, name:1, mayor:1});
< {
  name: 'New York',
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
test>
```

- 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}}, {_id: 0, name:1, mayor:1});
< {
  name: 'Punxsutawney ',
  mayor: {
    name: 'Jim Wehrle'
  }
}
test>
```

Практическое задание 8.2.2:

- 1) Сформировать функцию для вывода списка самцов единорогов.

```
> fn = function() {return this.gender == 'm'; }
< [Function: fn]
> db.unicorns.find(fn);
^ db.unicorns.find(fn);
```

- 2) Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

```
>_MONGOSH
```

```
test> var cursor = db.unicorns.find(fn).limit(2).sort({name: 1}); null;
```

- 3) Вывести результат, используя forEach.


```
test> cursor.ForEach(function(a) {print(a.name);})
```

4) Содержание коллекции единорогов unicorns:

```
db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

db.unicorns.insert ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> db.unicorns.find({gender: 'f', weight: {$gte:500, $lte:600}}, {name: 1, _id:0}).count(true);
< 2
test> |
```

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.find({gender: 'f'}).count(true);
< 5
> db.unicorns.find({gender: 'm'}).count(true);
< 7
test>
```

Практическое задание 8.2.6:

1. Выполнить команду:

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

```
✖ ▶ TypeError: db.unicorns.save is not a function

> db.unicorns.insertOne({name: 'Barney', loves: ['grape'],
weight: 340, gender: 'm'})
< {
  acknowledged: true,
  insertedId: ObjectId("6474867eb4d88ccd73119e52")
}
```

2. Проверить содержимое коллекции *unicorns*.

```

    },
    { name: 'Barney', loves: [ 'grape' ], weight: 340, gender: 'm' }
  ]
test> |

```

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.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

> db.unicorns.find({name: 'Ayna'})
< {
  _id: ObjectId("64746e352bf5bfdb2cb60332"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 800,
  gender: 'f',
  vampires: 51
}
test> |

```

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.updateOne({name: 'Raleigh'}, {$set: {loves: ['redbull']}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.unicorns.find({name: 'Raleigh'})
< {
  _id: ObjectId("64746e352bf5bfdb2cb60334"),
  name: 'Raleigh',
  loves: [
    'redbull'
  ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
test>

```

Практическое задание 8.2.9:

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

```

> db.unicorns.find({gender: 'm'}, {name: 1, vampires: 1, _id: 0}).toArray()
< [
  { name: 'Horny', vampires: 63 },
  { name: 'Unicrom', vampires: 182 },
  { name: 'Rooooooodles', vampires: 99 },
  { name: 'Kenny', vampires: 39 },
  { name: 'Raleigh', vampires: 2 },
  { name: 'Pilot', vampires: 54 },
  { name: 'Dunx', vampires: 165 },
  { name: 'Barney' }
]
test>

```

```

> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
> db.unicorns.find({gender: 'm'}, {name: 1, vampires: 1, _id: 0}).toArray()
< [
  { name: 'Horny', vampires: 68 },
  { name: 'Unicrom', vampires: 187 },
  { name: 'Rooooooodles', vampires: 104 },
  { name: 'Kenny', vampires: 44 },
  { name: 'Raleigh', vampires: 7 },
  { name: 'Pilot', vampires: 59 },
  { name: 'Dunx', vampires: 170 },
  { name: 'Barney', vampires: 5 }
]
test>

```

Практическое задание 8.2.10:

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

```

> db.towns.updateOne({name: 'Portland'}, {$unset: {'mayor.party': 1}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.towns.find({name: 'Portland'})
< {
  _id: ObjectId("647477e62bf5bfdb2cb6033b"),
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  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.updateOne({name: 'Pilot', gender: 'm'}, {$push: {loves: 'chocolate'}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.unicorns.find({name: 'Pilot', gender: 'm'}, {_id: 0}).toArray()
< [
  {
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
test>
```

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

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"}}

```

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

```

> db.towns.find({"mayor.party": {$exists: false}})
< {
  _id: ObjectId("647498a5b4d88ccd73119e53"),
  name: 'Punxsutawney ',
  popujatiuon: 6200,
  last_sensus: 2008-01-31T00:00:00.000Z,
  famous_for: [
    'phil the groundhog'
  ],
  mayor: {
    name: 'Jim Wehrle'
  }
}
> db.towns.deleteMany({"mayor.party": {$exists: false}})
< {
  acknowledged: true,
  deletedCount: 1
}
> db.towns.find({"mayor.party": {$exists: false}})
<
test>

```

- 4) Очистите коллекцию.

```

> db.towns.remove({})
< {
  acknowledged: true,
  deletedCount: 2
}
> db.towns.countDocuments()
< 0
test> |

```

5) Просмотрите список доступных коллекций.

```

> show collections
< towns
unicorns
test> |

```

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.u_areas.insert({place: 'a field', desc: 'a very nice place'});
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64749ec2b4d88ccd73119e56")
  }
}
> db.u_areas.insert({place: 'Balakovo', desc: 'not a very nice place'});
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64749ef3b4d88ccd73119e57")
  }
}
test>

```

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


```

> db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40, area: {$ref: 'u_areas', $id: 1}});
db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2, area: {$ref: 'u_areas', $id: 2}});
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f', area: {$ref: 'u_areas', $id: 1}});
db.unicorns.insert({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});

< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6474a119b4d88ccd73119e63")
  }
}
test>

```

3) Проверьте содержание коллекции единорогов.

4) Содержание коллекции единорогов unicorns:

```
db.unicorns.insert({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
```

```
db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
```

```
db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
```

```
db.unicorns.insert({name: 'Roooooodles', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
```

```
db.unicorns.insert({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
```

```
db.unicorns.insert({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
```

```
db.unicorns.insert({name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
```

```
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
```

```
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
```

```
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
```

```
db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
```

```
db.unicorns.insert {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
```

```
> db.unicorns.find({}, {_id: 0, name: 1, area: 1}).toArray();
< [
  { name: 'Horny', area: DBRef("u_areas", 2) },
  { name: 'Aurora', area: DBRef("u_areas", 2) },
  { name: 'Unicrom' },
  { name: 'Roooooodles', area: DBRef("u_areas", 2) },
  { name: 'Solnara' },
  { name: 'Ayna', area: DBRef("u_areas", 1) },
  { name: 'Kenny' },
  { name: 'Raleigh', area: DBRef("u_areas", 2) },
  { name: 'Leia' },
  { name: 'Pilot' },
  { name: 'Nimue', area: DBRef("u_areas", 1) },
  { name: 'Dunx' }
]
```

Практическое задание 8.3.2:

1. Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique. - **МОЖНО**

```
db.unicorns.insert((name: 'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 650, gender: 'm', vampires: 39));
db.unicorns.insert((name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2));
db.unicorns.insert((name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33));
db.unicorns.insert((name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54));
db.unicorns.insert((name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'));
db.unicorns.insert((name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165));
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6474a1d5b4d88ccd73119e6f")
  }
}
> db.unicorns.ensureIndex((name: 1), {unique: true});
< [ 'name_1' ]
test>
```

2. Содержание коллекции единорогов unicorns:

```
db.unicorns.insert({name: 'Horny', dob: new Date(1992, 2, 13, 7, 47), loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63});
```

```
db.unicorns.insert({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
```

```
db.unicorns.insert({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
```

```
db.unicorns.insert({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
```

```
db.unicorns.insert({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves: ['apple',
```

```
'carrot',      'chocolate'],      weight:550,  
gender:'f', vampires:80});
```

```
db.unicorns.insert({name:'Ayna',  dob:  new  
Date(1998,  2,  7,  8,  30),  loves:  
['strawberry',  'lemon'],  weight:  733,  
gender: 'f', vampires: 40});
```

```
db.unicorns.insert({name:'Kenny',  dob:  new  
Date(1997,  6,  1,  10,  42),  loves: ['grape',  
'lemon'],  weight:  690,      gender:  'm',  
vampires: 39});
```

```
db.unicorns.insert({name:  'Raleigh',  dob:  
new  Date(2005,  4,  3,  0,  57),  loves:  
['apple',  'sugar'],  weight:  421,  gender:  
'm', vampires: 2});
```

```
db.unicorns.insert({name:  'Leia',  dob:  new  
Date(2001,  9,  8,  14,  53),  loves: ['apple',  
'watermelon'],  weight:  601,  gender:  'f',  
vampires: 33});
```

```
db.unicorns.insert({name:  'Pilot',  dob:  new  
Date(1997,  2,  1,  5,  3),  loves: ['apple',  
'watermelon'],  weight:  650,  gender:  'm',  
vampires: 54});
```

```
db.unicorns.insert  ({name:  'Nimue',  dob:  
new  Date(1999,  11,  20,  16,  15),  loves:  
['grape',  'carrot'],  weight:  540,  gender:  
'f'});
```

```
db.unicorns.insert {name:  'Dunx',  dob:  new  
Date(1976,  6,  18,  18,  18),  loves: ['grape',  
'watermelon'],  weight:  704,  gender:  'm',  
vampires: 165
```

Практическое задание 8.3.3:

1) *Получите информацию о всех индексах коллекции unicorns.*

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

2) *Удалите все индексы, кроме индекса для идентификатора.*

```
> db.unicorns.dropIndex( name_1 );  
< { nIndexesWas: 2, ok: 1 }  
> db.unicorns.getIndexes();  
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]  
test>
```

3) Попробуйте удалить индекс для идентификатора.

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

Вывод:

В процессе выполнения 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.