

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: Полухин А.В.

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Цель: 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.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 4+, 7.0.1 (текущая), MongoShell.

Выполнение

Задание 2.1.1:

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

```
database> 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});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cc86a0b6d8b88cc2247e') }
}
learn> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cd16a0b6d8b88cc2247f') }
}
learn> db.unicorns.insert({name: 'Unicorn', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cd77a0b6d8b88cc22480') }
}
learn> db.unicorns.insert({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582ce3ba0b6d8b88cc22481') }
}
learn> db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582ce81a0b6d8b88cc22482') }
}
learn> db.unicorns.insert({name: 'Anya', loves: ['strawberry', 'lemon'], weight: 733, gender: 'm', vampires: 40});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cee2a0b6d8b88cc22483') }
}
learn> db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cf58a0b6d8b88cc22484') }
}
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582cfa4a0b6d8b88cc22485') }
}
```

```
learn> db.unicorns.insert({name: 'Leila', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582d0f0a0b6d8b88cc22486') }
}
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582d136a0b6d8b88cc22487') }
}
learn> db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('6582d169a0b6d8b88cc22488') }
}
```

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: 'Unicorn',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Rooooodles',
    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: 'Any',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'm',
    vampires: 40
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6582d0f0a0b6d8b88cc22486'),
    name: 'Leila',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6582d169a0b6d8b88cc22488'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6582d431a0b6d8b88cc22489'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
]
```

Задание 2.2.1:

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

Листинг запроса для получения всех самцов:

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

```
learn> db.unicorns.find({gender: 'm'}).sort({name: 1});
[
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Anya',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'm',
    vampires: 40
  },
  {
    _id: ObjectId('6582d431a0b6d8b88cc22489'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('6582cc86a0b6d8b88cc2247e'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Roooooodles',
    loves: [ 'apple' ],

```

Листинг запроса для получения всех самок:

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

```
learn> db.unicorns.find({gender: 'f'}).sort({name: 1});
[
  {
    _id: ObjectId('6582cd16a0b6d8b88cc2247f'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId('6582d0f0a0b6d8b88cc22486'),
    name: 'Leila',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6582d169a0b6d8b88cc22488'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6582ce81a0b6d8b88cc22482'),
    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`.

Листинг запроса с использованием `limit`:

```
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);
```

```
learn> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);
[
  {
    _id: ObjectId('6582cd16a0b6d8b88cc2247f'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Листинг запроса через `findOne`:

```
db.unicorns.findOne({gender: 'f', loves: 'carrot'});
```

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'});
{
  _id: ObjectId('6582cd16a0b6d8b88cc2247f'),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

Задание 2.2.2:

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

Листинг запроса:

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

```
learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0}).sort({name: 1});
[
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Anyu',
    weight: 733,
    vampires: 40
  },
  {
    _id: ObjectId('6582d431a0b6d8b88cc22489'),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  },
  {
    _id: ObjectId('6582cc86a0b6d8b88cc2247e'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Unicorn',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId('6582cd77a0b6d8b88cc22480'),
    name: 'Unicorn',
    weight: 984,
    vampires: 182
  }
]
```

Задание 2.2.3:

Вывести список единорогов в обратном порядке добавления.

Листинг запроса:

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

```
learn> db.unicorns.find().sort({$natural: -1});
[
  {
    _id: ObjectId('6582d431a0b6d8b88cc22489'),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('6582d169a0b6d8b88cc22488'),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6582d0f0a0b6d8b88cc22486'),
    name: 'Leila',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
]
```

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

Задание 2.1.4:

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

Листинг запроса:


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

```
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: 'Unicorn',
    loves: [ 'energon' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    name: 'Rooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Any',
    loves: [ 'apple' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Anya',
    loves: [ 'strawberry' ],
    weight: 733,
    gender: 'm',
    vampires: 40
  },
  {
    name: 'Kenny',
    loves: [ 'grape' ],
    weight: 690,
    gender: 'm',
    vampires: 165
  }
]
```

```

  {
    name: 'Raleigh',
    loves: [ 'apple' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    name: 'Leila',
    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 2.3.1:

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

Листинг запроса:

```
db.unicorns.find({gender: 'f', weight: {$gte: 500, $lte: 700}}, {_id: 0});
```

```
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: 'Leila',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Задание 2.3.2:

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

Листинг запроса:

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

```
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
  }
]
```

Задание 2.3.3:

Найти всех единорогов, не имеющих ключ vampires.

Листинг запроса:

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

```
learn> db.unicorns.find({vampires: {exists: false}});
learn> _
```

Задание 2.3.4:

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

Листинг запроса:

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

```
learn> db.unicorns.find({gender: 'm'}, {loves: {$slice: 1}}).sort({name: 1});
[
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Anya',
    loves: [ 'strawberry' ],
    weight: 733,
    gender: 'm',
    vampires: 40
  },
  {
    _id: ObjectId('6582d431a0b6d8b88cc22489'),
    name: 'Dunx',
    loves: [ 'grape' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId('6582cc86a0b6d8b88cc2247e'),
    name: 'Horny',
    loves: [ 'carrot' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    loves: [ 'grape' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    _id: ObjectId('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    loves: [ 'apple' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    loves: [ 'apple' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Rooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId('6582cd77a0b6d8b88cc22480'),
    name: 'Unicorn',
    loves: [ 'energon' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
]
```

Задание 3.1.1:

- 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"}}

```

```

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')
  }
}

```

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

Листинг запроса:

```
db.towns.find({'mayor.party': 'I'}, {name: 1, 'mayor.name': 1, _id: 0})
```

```

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

```

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

Листинг запроса:

```
db.towns.find({'mayor.party': {'$exists': false}}, {name: 1, 'mayor.name': 1, _id: 0})
```

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

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.

Листинг запроса:

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

```
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()
Anyu
Dunx
```

Задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг.

Листинг запроса:

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

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

Задание 3.2.2:

Вывести список предпочтений.

Листинг запроса:

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

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

Задание 3.2.3:

Посчитать количество особей единорогов обоих полов.

Листинг запроса:

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

```
learn> db.unicorns.aggregate({'$group': {'_id': '$gender', count: {'$sum': 1}}})
[ { _id: 'f', count: 4 }, { _id: 'm', count: 8 } ]
```

Задание 3.3.1:

- Выполнить команду:

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

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

Задание 3.3.2:

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

Листинг запроса:

```
db.unicorns.update({name: 'Anya'}, {'$set: {weight: 800, vampires: 51}})
```

```
learn> db.unicorns.update({name: 'Anya'}, {'$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
}
```

2. Проверить содержимое коллекции unicorns.

```

    vampires: 80
  },
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Anya',
    loves: [ 'strawberry', 'lemon' ],
    weight: 80,
    gender: 'm',
    vampires: 51
  },
  {

```

Задание 3.3.3:

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

Листинг запроса:

```
db.unicorns.update({name: 'Raleigh'}, {$set: {loves: ['redbull']}})
```

```

learn> db.unicorns.update({name: 'Raleigh'}, {$set: {loves: ['redbull']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

2. Проверить содержимое коллекции unicorns.

```

{
  _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
  name: 'Raleigh',
  loves: [ 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
{

```

Задание 3.3.4:

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

Листинг запроса:

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

```

learn> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}

```

2. Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.find({gender: 'm'})
[
  {
    _id: ObjectId('6582cc86a0b6d8b88cc2247e'),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 73
  },
  {
    _id: ObjectId('6582cd77a0b6d8b88cc22480'),
    name: 'Unicorn',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 187
  },
  {
    _id: ObjectId('6582ce3ba0b6d8b88cc22481'),
    name: 'Roooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 104
  },
  {
    _id: ObjectId('6582cee2a0b6d8b88cc22483'),
    name: 'Any',
    loves: [ 'strawberry', 'lemon' ],
    weight: 80,
    gender: 'm',
    vampires: 56
  },
  {
    _id: ObjectId('6582cf58a0b6d8b88cc22484'),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 44
  },
  {
    _id: ObjectId('6582cfa4a0b6d8b88cc22485'),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
  }
]
```

Задание 3.3.5:

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

Листинг запроса:

```
db.towns.update({name: 'Portland'}, {$set: {'mayor.party': undefined}})
```

```
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 }
}
```

Задание 3.3.6:

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

Листинг запроса:

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

```
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('6582d136a0b6d8b88cc22487'),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
```

Задание 3.3.7:

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

Листинг запроса:

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

```
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('6582cd16a0b6d8b88cc2247f'),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],
    weight: 450,
    gender: 'f',
  }
]
```

Задание 3.4.1:

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

Листинг запроса:

db.towns.deleteMany({'mayor.party': {\$exists: false}})

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

- 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 }
  }
]
```

- Очистите коллекцию.

Листинг запроса: *db.towns.deleteMany({})*

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

- Просмотрите список доступных коллекций.

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

Задание 4.1.1:

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

Листинг запроса:

```
db.createCollection('areas')

db.areas.insert({_id: 'eq', name: 'Equestria', description: 'Satrting base location'})

db.areas.insert({_id: 'ef', name: 'Everfree Forest', description: 'Magic forest with colorfull plants'})

db.areas.insert({_id: 'fn', name: 'Frozen North', description: 'Snowy mountains'})

db.areas.insert({_id: 'sl', name: 'Saddle Lake', description: 'Big clean lake'})
```

```
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, используя второй способ автоматического связывания.

Листинг запроса:

```
db.unicorns.update({name: 'Pilot'}, {$set: {area: {$ref: 'areas', $id: 'fn'}}})

db.unicorns.update({name: 'Anya'}, {$set: {area: {$ref: 'areas', $id: 'sl'}}})
```

```
learn> db.unicorns.update({name: 'Pilot'}, {$set: {area: {$ref: 'areas', $id: 'fn'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> db.unicorns.update({name: 'Anya'}, {$set: {area: {$ref: 'areas', $id: 'sl'}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

3. Проверьте содержание коллекции единорогов.

<pre>{ _id: ObjectId('6582d136a0b6d8b88cc22487'), name: 'Pilot', loves: ['apple', 'watermelon', 'chocolate'], weight: 650, gender: 'm', vampires: 59, area: DBRef('areas', 'fn') },</pre>	<pre>{ _id: ObjectId('6582cee2a0b6d8b88cc22483'), name: 'Anya', loves: ['strawberry', 'lemon'], weight: 80, gender: 'm', vampires: 56, area: DBRef('areas', 'sl') },</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Задание 4.2.1:

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

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

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

- Создайте индекс для ключа `value`.

Листинг запроса:

```
db.numbers.ensureIndex({'value': 1}, {'unique': true})
```

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

- Получите информацию о всех индексах коллекции `numbers`.

Листинг запроса:

```
db.numbers.getIndexes()
```

```
[  
  { v: 2, key: { _id: 1 }, name: '_id_' },  
  { v: 2, key: { value: 1 }, name: 'value_1', unique: true }  
]
```

- Выполните запрос 2.

```
executionStats: {  
  executionSuccess: true,  
  nReturned: 4,  
  executionTimeMillis: 3,  
  totalKeysExamined: 4,  
  totalDocsExamined: 4,  
  executionStages: {  
    stage: 'limit',
```

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

В первом случае (без использования индексов) для выполнения запроса потребовалось 92 миллисекунды. Во втором случае (с использованием индексов) потребовалось 3 миллисекунды.

Вывод: в ходе лабораторной работы овладели 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.