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

О Т Ч Е Т

по лабораторной работе «Работа с БД в СУБД MongoDB»

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

Автор: Бакшилова Анастасия Денисовна
Факультет: ИКТ
Группа: К33391
Преподаватель: Говорова М. М.
Дата: 23.10.2023

Санкт-Петербург
2023

Цель: овладеть практическими навыками установки СУБД MongoDB.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 4+, 6.0.6 (текущая)

Выполнение:

1. Задание 8.1.1

    1) Создайте базу данных learn.
    2) Заполните коллекцию единорогов unicorns
    3) Используя второй способ, вставьте в коллекцию единорогов документ
    4) Проверьте содержимое коллекции с помощью метода find.

    a. Вставка через консоль:

```
> db.unicorns.insertOne({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});
< {
    acknowledged: true,
    insertedId: ObjectId("6536352c064ed20d66d3331b")
  }
> db.unicorns.insertOne({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
< {
    acknowledged: true,
    insertedId: ObjectId("65363539064ed20d66d3331c")
  }
> db.unicorns.insertOne({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
< {
    acknowledged: true,
    insertedId: ObjectId("65363541064ed20d66d3331d")
  }
```

```
> db.unicorns.insertOne({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
< {
    acknowledged: true,
    insertedId: ObjectId("65363363064ed20d66d33313")
  }
> db.unicorns.insertOne({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});
< {
    acknowledged: true,
    insertedId: ObjectId("65363376064ed20d66d33314")
  }
> db.unicorns.insertOne({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
< {
    acknowledged: true,
    insertedId: ObjectId("6536338b064ed20d66d33315")
  }
> db.unicorns.insertOne({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39});
< {
    acknowledged: true,
    insertedId: ObjectId("6536339f064ed20d66d33316")
  }
> db.unicorns.insertOne({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
< {
    acknowledged: true,
    insertedId: ObjectId("653633b2064ed20d66d33317")
  }
```

```
> db.unicorns.insertOne({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
< {
    acknowledged: true,
    insertedId: ObjectId("653633c3064ed20d66d33318")
  }
> db.unicorns.insertOne({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
< {
    acknowledged: true,
    insertedId: ObjectId("653633d1064ed20d66d33319")
  }
> db.unicorns.insertOne({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
< {
    acknowledged: true,
    insertedId: ObjectId("653633de064ed20d66d3331a")
  }
```

Вставка через документ:

```
> document=({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});
< {
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
```

Результат find

```
> db.unicorns.find().count()
< 12
```

2. Выборка из баз данных 8.1.2

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

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

1)

Самки:

```
> db.unicorns.find ({gender : 'f'}).sort({name : 1}).limit(3)
< {
    _id: ObjectId("65363539064ed20d66d3331c"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
  {
    _id: ObjectId("6536338b064ed20d66d33315"),
    name: 'Ayna',
    loves: [
      'strawberry',
      'lemon'
    ],
    weight: 733,
    gender: 'f',
    vampires: 40
  }
```

```
{
  _id: ObjectId("653633c3064ed20d66d33318"),
  name: 'Leia',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33
}
```

Самцы:

```
> db.unicorns.find ({gender : 'm'}).sort({name : 1}).limit(3)
< {
    _id: ObjectId("65363aabb5bf0bc956b971c7"),
    name: 'Dunx',
    loves: 'grape, watermelon',
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  {
    _id: ObjectId("6536352c064ed20d66d3331b"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
```

```
{
  _id: ObjectId("6536339f064ed20d66d33316"),
  name: 'Kenny',
  loves: [
    'grape',
    'lemon'
  ],
  weight: 690,
  gender: 'm',
  vampires: 39
}
```

2)

```
> db.unicorns.find({gender: 'f', loves: 'carrot'}, {_id: 0, name: 1}).sort({name :1}).limit(1)
< {
    name: 'Aurora'
  }
```

Задание 8.1.3:

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

1)

```
> db.unicorns.find({ gender: 'm' }, { _id: 0, name: 1, weight: 1, vampires: 1 }).sort({ name: 1 })
< {
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
  {
    name: 'Horny',
    weight: 600,
    vampires: 63
  }
  {
    name: 'Kenny',
    weight: 690,
    vampires: 39
  }
  {
    name: 'Pilot',
    weight: 650,
    vampires: 54
  }
```

```
{
  name: 'Raleigh',
  weight: 421,
  vampires: 2
}
{
  name: 'Roooooodles',
  weight: 575,
  vampires: 99
}
{
  name: 'Unicrom',
  weight: 984,
  vampires: 182
}
```

Задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find({}).sort({$natural: -1})
< {
    _id: ObjectId("65363aabb5bf0bc956b971c7"),
    name: 'Dunx',
    loves: 'grape, watermelon',
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  {
    _id: ObjectId("65363541064ed20d66d3331d"),
    name: 'Unicrom',
    loves: [
      'energon',
      'redbull'
    ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
```

```
{
  _id: ObjectId("65363539064ed20d66d3331c"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  _id: ObjectId("6536352c064ed20d66d3331b"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

```
{
  _id: ObjectId("653633de064ed20d66d3331a"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  _id: ObjectId("653633d1064ed20d66d33319"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 650,
  gender: 'm',
  vampires: 54
}
```

```
{
  _id: ObjectId("653633c3064ed20d66d33318"),
  name: 'Leia',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33
}
{
  _id: ObjectId("653633b2064ed20d66d33317"),
  name: 'Raleigh',
  loves: [
    'apple',
    'sugar'
  ],
  weight: 421,
  gender: 'm',
  vampires: 2
}
```

```
{
  _id: ObjectId("6536339f064ed20d66d33316"),
  name: 'Kenny',
  loves: [
    'grape',
    'lemon'
  ],
  weight: 690,
  gender: 'm',
  vampires: 39
}
{
  _id: ObjectId("6536338b064ed20d66d33315"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40
}
```

```
{
  _id: ObjectId("65363376064ed20d66d33314"),
  name: 'Solnara',
  loves: [
    'apple',
    'carrot',
    'chocolate'
  ],
  weight: 550,
  gender: 'f',
  vampires: 80
}
{
  _id: ObjectId("65363363064ed20d66d33313"),
  name: 'Roooooodles',
  loves: [
    'apple'
  ],
  weight: 575,
  gender: 'm',
  vampires: 99
}
```

Задание 8.1.5:

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

```
> db.unicorns.find({}, {_id: 0, loves: {$slice : 1}});
< {
    name: 'Roooooodles',
    loves: [
      'apple'
    ],
    weight: 575,
    gender: 'm',
    vampires: 104,
    habitat: DBRef("habitats", 'forest')
  }
  {
    name: 'Solnara',
    loves: [
      'apple'
    ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
```

```
  {
    name: 'Ayna',
    loves: [
      'strawberry'
    ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
  {
    name: 'Kenny',
    loves: [
      'grape'
    ],
    weight: 690,
    gender: 'm',
    vampires: 44
  }
```

```
{
  name: 'Leia',
  loves: [
    'apple'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33,
  habitat: DBRef("habitats", 'mountain')
}
{
  name: 'Pilot',
  loves: [
    'apple'
  ],
  weight: 650,
  gender: 'm',
  vampires: 59,
  habitat: DBRef("habitats", 'meadow')
}
```

```
{
  name: 'Nimue',
  loves: [
    'grape'
  ],
  weight: 540,
  gender: 'f'
}
{
  name: 'Horny',
  loves: [
    'carrot'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68,
  habitat: DBRef("habitats", 'meadow')
}
```

```
{
  name: 'Aurora',
  loves: [
    'carrot'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  name: 'Unicrom',
  loves: [
    'energon'
  ],
  weight: 984,
  gender: 'm',
  vampires: 187
}
```

```
{
  name: 'Barny',
  loves: [
    'grape'
  ],
  weight: 340,
  gender: 'm',
  vampires: 5
}
```

Задание 8.1.6:

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

```
> db.unicorns.find({gender: 'f', weight: {$gt:500, $lt:700}}, {_id:0})
< {
    name: 'Solnara',
    loves: [
      'apple',
      'carrot',
      'chocolate'
    ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
  {
    name: 'Leia',
    loves: [
      'apple',
      'watermelon'
    ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
```

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

Задание 8.1.7:

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

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

Задание 8.1.8:

Найти всех единорогов, не имеющих ключ `vampires`.

```
> db.unicorns.find({gender:'f', vampires:{$exists:false}}, {_id:0, name:1})
< {
    name: 'Nimue'
  }
```

Задание 8.1.9:

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

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

```
{
  name: 'Raleigh',
  loves: [
    'apple'
  ]
}
{
  name: 'Roooooodles',
  loves: [
    'apple'
  ]
}
{
  name: 'Unicrom',
  loves: [
    'energon'
  ]
}
```

Задание 8.2.1:

1. Создайте коллекцию towns, включающую следующие документы

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

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

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

Задание 8.2.2:

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.

```
> function males() {
      var cursor = db.unicorns.find({ gender: 'm' }).sort({ name: 1 }).limit(2);
      cursor.forEach(function(unicorn) {
          print(unicorn.name);
      });
  }
  males();
< Dunx
< Horny
```

Задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> var count = db.unicorns.countDocuments({ gender: 'f', weight: { $gte: 500, $lte: 600 } });
  print(count);
< 2
```

Задание 8.2.4:

Вывести список предпочтений.

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

Задание 8.2.5:

Подсчитать количество особей единорогов обоих полов

```
> var result = db.unicorns.aggregate([
      {
          $group: {
              _id: "$gender",
              count: { $sum: 1 }
          }
      },
      {
          $project: {
              gender: "$_id",
              count: 1,
              _id: 0
          }
      }
  ]);
  result.forEach(function(doc) {
      print("Пол: " + doc.gender + ", Кол-во: " + doc.count);
  });

< Пол: m, Кол-во: 7
< Пол: f, Кол-во: 5
```

Задание 8.2.6:

1. Выполнить команду:

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

2. Проверить содержимое коллекции unicorns.

```
> db.unicorns.insert({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
< DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("6537da8976781612ce012df2")
    }
  }
```

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', gender: 'f' }, {$set: {weight: 800,vampires: 51}})

< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.find({ name: 'Ayna' })
< {
    _id: ObjectId("6536338b064ed20d66d33315"),
    name: 'Ayna',
    loves: [
      'strawberry',
      'lemon'
    ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
```

Задание 8.2.8:

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

```
> db.unicorns.updateOne({ name: 'Raleigh', gender: 'm' }, {$push:{loves: 'redbull'}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.find({ name: 'Raleigh' })
< {
    _id: ObjectId("653633b2064ed20d66d33317"),
    name: 'Raleigh',
    loves: [
      'apple',
      'sugar',
      'redbull'
    ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
```

Задание 8.2.9:

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

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

```
> db.unicorns.find({ gender: 'm' })
< {
    _id: ObjectId("65363363064ed20d66d33313"),
    name: 'Roooooodles',
    loves: [
      'apple'
    ],
    weight: 575,
    gender: 'm',
    vampires: 104
  }
  {
    _id: ObjectId("6536339f064ed20d66d33316"),
    name: 'Kenny',
    loves: [
      'grape',
      'lemon'
    ],
    weight: 690,
    gender: 'm',
    vampires: 44
  }
```

```
{
  _id: ObjectId("653633b2064ed20d66d33317"),
  name: 'Raleigh',
  loves: [
    'apple',
    'sugar',
    'redbull'
  ],
  weight: 421,
  gender: 'm',
  vampires: 7
}
{
  _id: ObjectId("653633d1064ed20d66d33319"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 650,
  gender: 'm',
  vampires: 59
}
```

```
{
  _id: ObjectId("6536352c064ed20d66d3331b"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68
}
{
  _id: ObjectId("65363541064ed20d66d3331d"),
  name: 'Unicrom',
  loves: [
    'energon',
    'redbull'
  ],
  weight: 984,
  gender: 'm',
  vampires: 187
}
```

```
{
  _id: ObjectId("65363aabb5bf0bc956b971c7"),
  name: 'Dunx',
  loves: 'grape, watermelon',
  weight: 704,
  gender: 'm',
  vampires: 170
}
{
  _id: ObjectId("6537da8976781612ce012df2"),
  name: 'Barny',
  loves: [
    'grape'
  ],
  weight: 340,
  gender: 'm',
  vampires: 5
}
```

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' },{$set: {party: ''}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
> db.towns.find({ name: 'Portland' })
< {
    _id: ObjectId("653659ccb5bf0bc956b971d1"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: '2009-07-20',
    famous_for: [
      'beer',
      'food'
    ],
    mayor: {
      name: 'Sam Adams',
      party: 'D'
    },
    party: ''
  }
```

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' },{$push: {loves: 'chocolate'}})

< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.find({ name: 'Pilot' })
< {
    _id: ObjectId("653633d1064ed20d66d33319"),
    name: 'Pilot',
    loves: [
      'apple',
      'watermelon',
      'chocolate'
    ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
```

Задание 8.2.12:

1. Изменить информацию о самке единорога `Aurora`: теперь она любит еще и сахар, и лимоны.
2. Проверить содержимое коллекции `unicorns`.

```
> db.unicorns.updateOne({ name: 'Aurora' },{$push: {loves: { $each: ['sugar', 'lemons'] }}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.find({ name: 'Aurora' })
< {
    _id: ObjectId("65363539064ed20d66d3331c"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape',
      'sugar',
      'lemons'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
```

Задание 8.2.13:

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

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

```
> db.towns.find()
< {
    _id: ObjectId("653659ccb5bf0bc956b971d0"),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: '2009-07-31',
    famous_for: [
      'status of liberty',
      'food'
    ],
    mayor: {
      name: 'Michael Bloomberg',
      party: 'I'
    }
  }
```

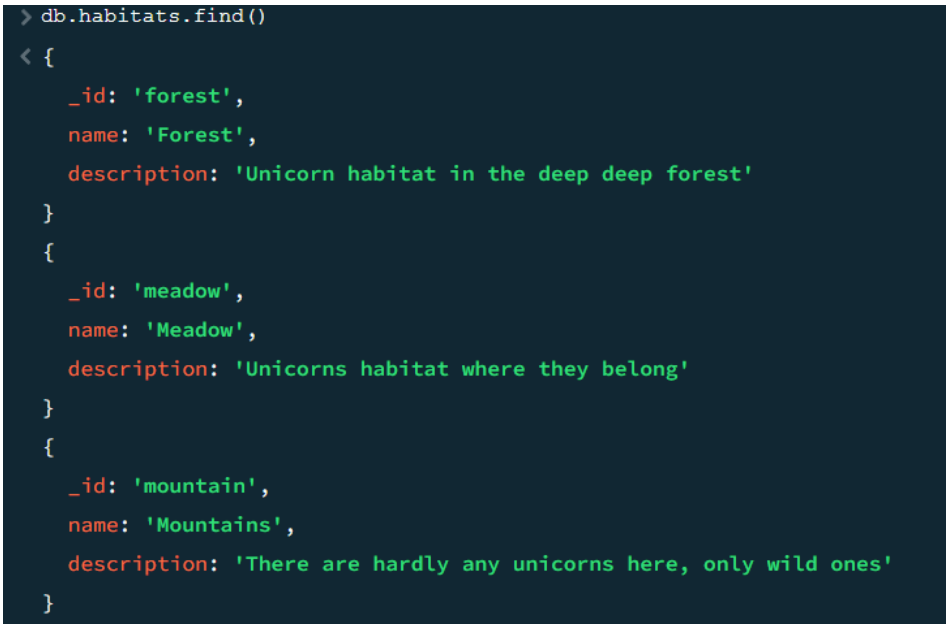
```
  {
    _id: ObjectId("653659ccb5bf0bc956b971d1"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: '2009-07-20',
    famous_for: [
      'beer',
      'food'
    ],
    mayor: {
      name: 'Sam Adams',
      party: 'D'
    },
```

```
> db.towns.drop()
< true
> show collections
< unicorns
```

Задание 8.3.1:

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, используя второй способ автоматического связывания.
3) Проверьте содержание коллекции единорогов.

db.createCollection("habitats") &&
db.habitats.insert([{_id:"forest",name:"Forest",description:"Unicorn habitat in the deep deep forest"},{_id:"meadow",name:"Meadow",description:"Unicorns habitat where they belong"},{_id:"mountain",name:"Mountains",description:"There are hardly any unicorns here, only wild ones"}])



```
> db.unicorns.updateOne({name : 'Horny'}, {$set: {habitat : {$ref: 'habitats', $id: 'meadow'}}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.updateOne({name : 'Leia'}, {$set: {habitat : {$ref: 'habitats', $id: 'mountain'}}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.updateOne({name : 'Pilot'}, {$set: {habitat : {$ref: 'habitats', $id: 'meadow'}}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
> db.unicorns.updateOne({name : 'Roooooodles'}, {$set: {habitat : {$ref: 'habitats', $id: 'forest'}}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

```
> db.unicorns.find()
< {
    _id: ObjectId("65363363064ed20d66d33313"),
    name: 'Roooooodles',
    loves: [
      'apple'
    ],
    weight: 575,
    gender: 'm',
    vampires: 104,
    habitat: DBRef("habitats", 'forest')
  }
```

```
{
  _id: ObjectId("653633c3064ed20d66d33318"),
  name: 'Leia',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33,
  habitat: DBRef("habitats", 'mountain')
}
```

```
{
  _id: ObjectId("653633d1064ed20d66d33319"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon',
    'chocolate'
  ],
  weight: 650,
  gender: 'm',
  vampires: 59,
  habitat: DBRef("habitats", 'meadow')
}
```

```
{
  _id: ObjectId("6536352c064ed20d66d3331b"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68,
  habitat: DBRef("habitats", 'meadow')
}
```

Задание 8.3.2:

Проверьте, можно ли задать для коллекции `unicorns` индекс для ключа `name` с флагом `unique`.

```
> db.unicorns.createIndex({ name: 1 }, { unique: true })
< name_1
```

Задание 8.3.3:

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

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

```
> db.unicorns.dropIndexes('_id_')
✖ ▸ MongoServerError: cannot drop _id index
```

Задание 8.3.4:

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?

```
> for (var i = 0; i < 100000; i++) {
      db.numbers.insert({ value: i });
  }
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("6538068d76781612ce02b492")
    }
  }
```

```
> var lastFourDocuments = db.numbers.find().sort({ _id: -1 }).limit(4);
> var queryExecution = lastFourDocuments.explain("executionStats");
  var executionTimeMillis = queryExecution.executionStats.executionTimeMillis;
  print("Время выполнения запроса без индекса: " + executionTimeMillis + " миллисекунд");
< Время выполнения запроса без индекса: 9 миллисекунд
> db.numbers.createIndex({ value: 1 });
< value_1
> var allIndexes = db.numbers.getIndexes();
  printjson(allIndexes);
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { value: 1 }, name: 'value_1' }
  ]
> var lastFourDocumentsWithIndex = db.numbers.find().sort({ _id: -1 }).limit(4).explain("executionStats");
  var executionTimeMillisWithIndex = lastFourDocumentsWithIndex.executionStats.executionTimeMillis;
  print("Время выполнения запроса с индексом: " + executionTimeMillisWithIndex + " миллисекунд");
< Время выполнения запроса с индексом: 1 миллисекунд
```

Вывод по заданию 8.3.4: запрос с индексом выполняется быстрее, так как они ускоряют запросы на поиск и доступ к данным. Индексы позволяют базе данных быстро определить, где находится искомая информация, и извлечь её без просмотра всей коллекции. Также у них сниженная нагрузка на CPU и память.

Общий вывод: В процессе работы были получены навыки работы с CRUD-операциями с вложенными объектами в коллекции бд MongoDB, агрегациями и изменениями данных, со ссылками и индексами в бд MongoDB.