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

Обучающийся Саид Наваф

Факультет прикладной информатики

Группа K3241

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург

2025-2026

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 2.1.1:

1. Создайте базу данных learn.

```
>_MONGOSH
> use learn
< switched to db learn
> db
< learn
```

2. Заполните коллекцию единорогов unicorns.

```
> db.unicorns.insertMany([
    { name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63 },
    { name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43 },
    { name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182 },
    { name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99 },
    { name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80 },
    { name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40 },
    { name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39 },
    { name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2 },
    { 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' }
  ])
```

3. Используя второй способ, вставьте в коллекцию единорогов документ.

```
> let unicornDocument = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165};
  db.unicorns.insertOne(unicornDocument);
< {
    acknowledged: true,
    insertedId: ObjectId('6834efc53cd779a6989d185d')
  }
learn>
```

4.Проверьте содержимое коллекции с помощью метода find.

```
> db.unicorns.find().pretty()
{
    _id: ObjectId('6834ea4dfe765bfa7000a5ee')
}
{
    _id: ObjectId('6834eb6afe765bfa7000a5ef')
}
{
    _id: ObjectId('6834edb63cd779a6989d1852'),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
}
{
    _id: ObjectId('6834edb63cd779a6989d1853'),
    name: 'Aurora',
    loves: [
```

```
>_MONGOSH
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1854'),
    name: 'Unicrom',
    loves: [
      'energon',
      'redbull'
    ],
    weight: 984,
    gender: 'm',
    vampires: 182
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1855'),
    name: 'Roooooodles',
    loves: [
```

Практическое задание 2.2.1:

1. Сформируйте запросы для вывода списков самцов и самок единорогов.

```
>_MONGOSH
> db.unicorns.find(
      {gender: 'm'},
      {_id: 1, name: 1, weight: 1, vampires: 1}
  ).sort({name: 1}).pretty()
< {
    _id: ObjectId('6834efc53cd779a6989d185d'),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1852'),
    name: 'Horny',
    weight: 600,
    vampires: 63
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1858'),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  }
```

2.Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени

```
>_MONGOSH
> db.unicorns.find({gender: 'f'}, {_id: 0}).sort({name: 1}).limit(3)
< {
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
  {
    name: 'Ayna',
    loves: [
      'strawberry',
      'lemon'
    ],
    weight: 733,
    gender: 'f',
    vampires: 40
  }
  {
```

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

```
> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
< {
    _id: ObjectId('6834edb63cd779a6989d1853'),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
learn >
```

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

```
>_MONGOSH
> db.unicorns.find(
      {gender: 'm'},
      {loves: 0, _id: 0}
  )
< {
    name: 'Horny',
    weight: 600,
    gender: 'm',
    vampires: 63
  }
  {
    name: 'Unicrom',
    weight: 984,
    gender: 'm',
    vampires: 182
  }
  {
    name: 'Roooooodles',
    weight: 575,
    gender: 'm',
    vampires: 99
  }
```

```
>_MONGOSH
    gender: 'm',
    vampires: 39
  }
  {
    name: 'Raleigh',
    weight: 421,
    gender: 'm',
    vampires: 2
  }
  {
    name: 'Pilot',
    weight: 650,
    gender: 'm',
    vampires: 54
  }
  {
    name: 'Dunx',
    weight: 704,
    gender: 'm',
    vampires: 165
  }
learn >
```

5.Вывести список единорогов в обратном порядке добавления.

```
>_MONGOSH
> db.unicorns.find().sort({$natural: -1})
< {
    _id: ObjectId('6834efc53cd779a6989d185d'),
    name: 'Dunx',
    loves: [
      'grape',
      'watermelon'
    ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  {
    _id: ObjectId('6834edb63cd779a6989d185c'),
    name: 'Nimue',
    loves: [
      'grape',
      'carrot'
    ],
    weight: 540,
    gender: 'f'
  }
```

```
>_MONGOSH
    weight: 450,
    gender: 'f',
    vampires: 43
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1852'),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
  {
    _id: ObjectId('6834eb6afe765bfa7000a5ef')
  }
  {
    _id: ObjectId('6834ea4dfe765bfa7000a5ee')
  }
learn >
```

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

```
>_MONGOSH
> db.unicorns.find({}, {name: 1, loves: {$slice: 1}, _id: 0})
< {}
  {}
  {
    name: 'Horny',
    loves: [
      'carrot'
    ]
  }
  {
    name: 'Aurora',
    loves: [
      'carrot'
    ]
  }
  {
    name: 'Unicrom',
    loves: [
      'energon'
    ]
  }
  {
```

```
>_MONGOSH
    name: 'Roooooodles',
    loves: [
      'apple'
    ]
  }
  {
    name: 'Solnara',
    loves: [
      'apple'
    ]
  }
  {
    name: 'Ayna',
    loves: [
      'strawberry'
    ]
  }
  {
    name: 'Kenny',
    loves: [
      'grape'
    ]
```

```
{
  name: 'Raleigh',
  loves: [
    'apple'
  ]
}
{
  name: 'Leia',
  loves: [
    'apple'
  ]
}
{
  name: 'Pilot',
  loves: [
    'apple'
  ]
}
{
  name: 'Nimue',
  loves: [
    'grape'
```

Практическое задание 2.3.1:

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

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

2. 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}
    )
< {
    name: 'Kenny',
    loves: [
      'grape',
      'lemon'
    ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
learn >
```

3. Найти всех единорогов, не имеющих ключ vampires.

```
> db.unicorns.find(
      {vampires: {$exists: false}}
  )
< {
    _id: ObjectId('6834ea4dfe765bfa7000a5ee')
  }
  {
    _id: ObjectId('6834eb6afe765bfa7000a5ef')
  }
  {
    _id: ObjectId('6834edb63cd779a6989d185c'),
    name: 'Nimue',
    loves: [
      'grape',
      'carrot'
    ],
    weight: 540,
    gender: 'f'
  }
learn >
```

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

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

Практическое задание 3.1.1:

1.Создайте коллекцию towns, включающую следующие документы.

```
>_MONGOSH
> db.towns.insertMany([
      {
          name: "New York",
          population: 22200000,
          last_census: new Date("2004-03-19"),
          famous_for: ["statue of liberty", "food"],
          mayor: {
              name: "Michael Bloomberg",
              party: "I"
          }
      },
      {
          name: "Portland",
          population: 582000,
          last_census: new Date("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"}, {name: 1, mayor: 1, _id: 0})
< {
    name: 'New York',
    mayor: {
      name: 'Michael Bloomberg',
      party: 'I'
    }
  }
learn>
```

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

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

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

```
> var maleUnicornsCursor = db.unicorns.find({$where: filterMales})
                                    .sort({name: 1})
                                    .limit(2);
> maleUnicornsCursor.forEach(function(unicorn) {
    print("Имя: " + unicorn.name +
          ", Вес: " + unicorn.weight + "кг" +
          ", Любит: " + unicorn.loves.join(', '));
 });
< Имя: Dunx, Вес: 704кг, Любит: grape, watermelon
< Имя: Horny, Вес: 600кг, Любит: carrot, papaya
learn>
```

Практическое задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг

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

2.Вывести список предпочтений.

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

3.Посчитать количество особей единорогов обоих полов

```
> db.unicorns.aggregate([
      {
          $group: {
              _id: "$gender",
              count: {$sum: 1}
          }
      }
  ])
< {
    _id: 'm',
    count: 7
  }
  {
    _id: 'f',
    count: 5
  }
  {
    _id: null,
    count: 2
  }
learn >
```

Практическое задание 3.3.1:

1.Выполнить команду. Проверить содержимое коллекции unicorns.

```
> db.unicorns.insertOne({
      name: 'Barny',
      loves: ['grape'],
      weight: 340,
      gender: 'm'
  });
< {
    acknowledged: true,
    insertedId: ObjectId('68351ba23cd779a6989d1860')
  }
learn >
```

```
> db.unicorns.find({name: 'Barny'});
< {
    _id: ObjectId('68351ba23cd779a6989d1860'),
    name: 'Barny',
    loves: [
      'grape'
    ],
    weight: 340,
    gender: 'm'
  }
learn >
```

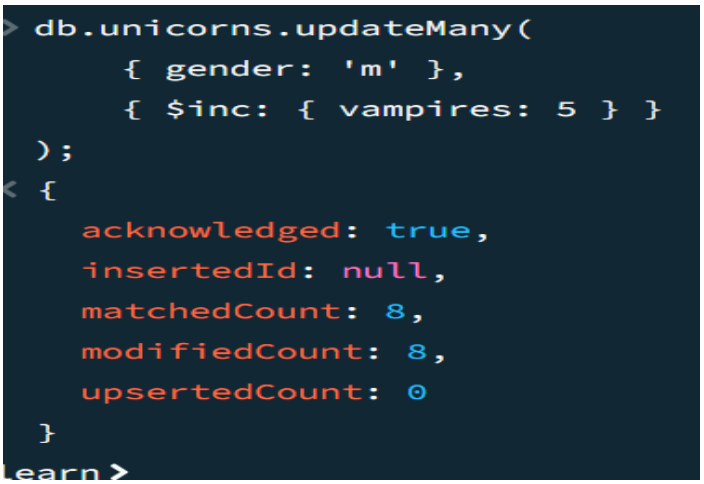
2. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.

```
> db.unicorns.updateOne(
      { name: 'Ayna', gender: 'f' },
      {
          $set: {
              weight: 800,
              vampires: 51
          }
      }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
learn >
```

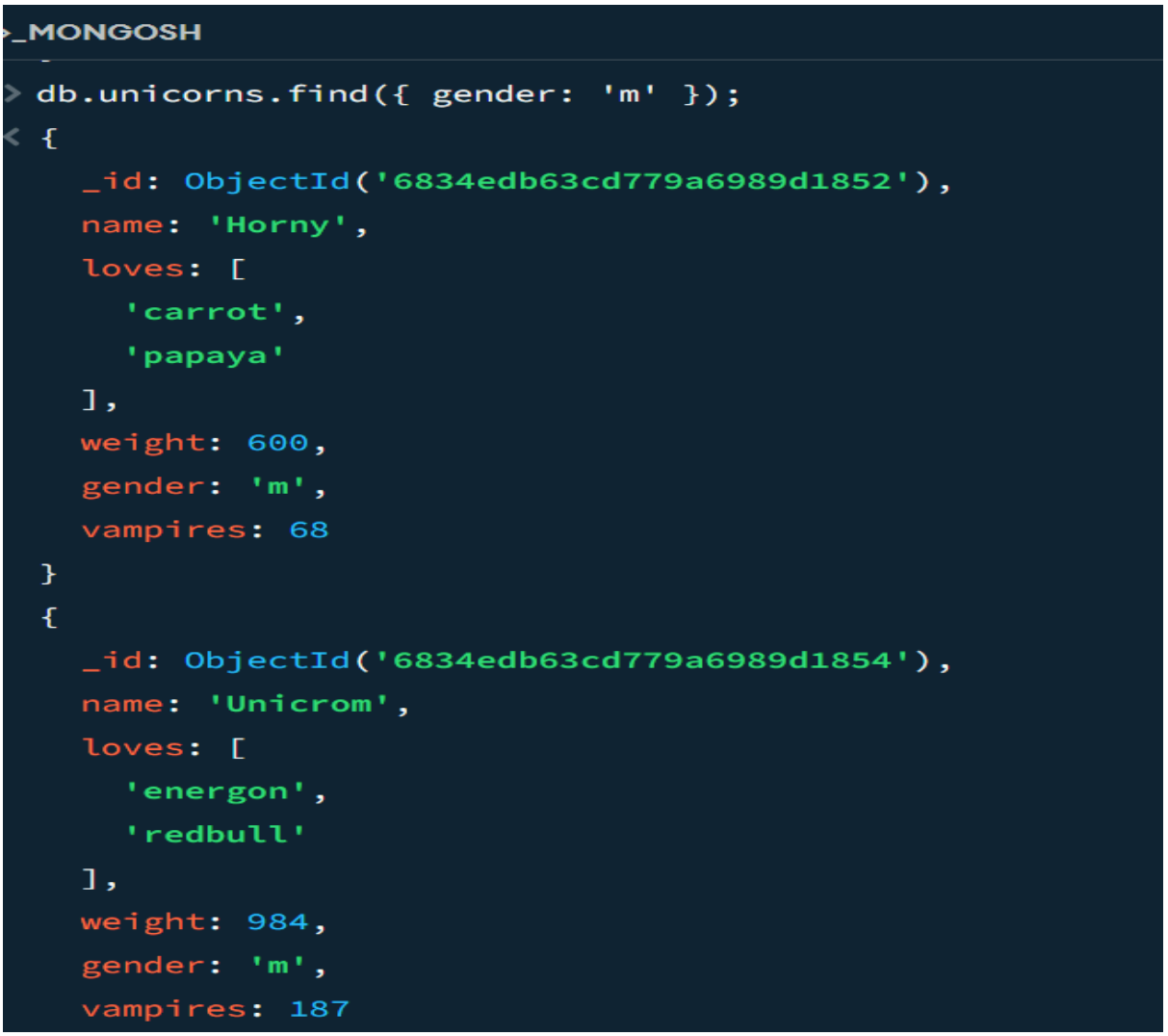
3.Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({ name: 'Ayna' });
< {
    _id: ObjectId('6834edb63cd779a6989d1857'),
    name: 'Ayna',
    loves: [
      'strawberry',
      'lemon'
    ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
learn >
```

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



5. Проверить содержимое коллекции unicorns.



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

```
> db.towns.updateOne(
      { name: 'Portland' },
      { $unset: { "mayor.party": "" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
learn >
```

7. Проверить содержимое коллекции towns.

```
> db.towns.find({name: 'Portland'});
< {
    _id: ObjectId('683503ec3cd779a6989d185f'),
    name: 'Portland',
    population: 582000,
    last_census: 2009-07-20T00:00:00.000Z,
    famous_for: [
      'beer',
      'food'
    ],
    mayor: {
      name: 'Sam Adams'
    }
  }
learn >
```

8. Изменить информацию о самце единорога Pilot: теперь он любит и шокола.

```
> db.unicorns.updateOne(
      { name: 'Pilot', gender: 'm' },
      { $addToSet: { loves: 'chocolate' } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
learn >
```

9. Проверить содержимое коллекции unicorns.

```
> db.unicorns.find({name: 'Pilot'});
< {
    _id: ObjectId('6834edb63cd779a6989d185b'),
    name: 'Pilot',
    loves: [
      'apple',
      'watermelon',
      'chocolate'
    ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
learn >
```

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

```
> db.unicorns.updateOne(
      { name: 'Aurora', gender: 'f' },{
          $addToSet: {
              loves: { $each: ['sugar', 'lemon']}}});
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
learn >
```

11.Проверить содержимое коллекции unicorns.

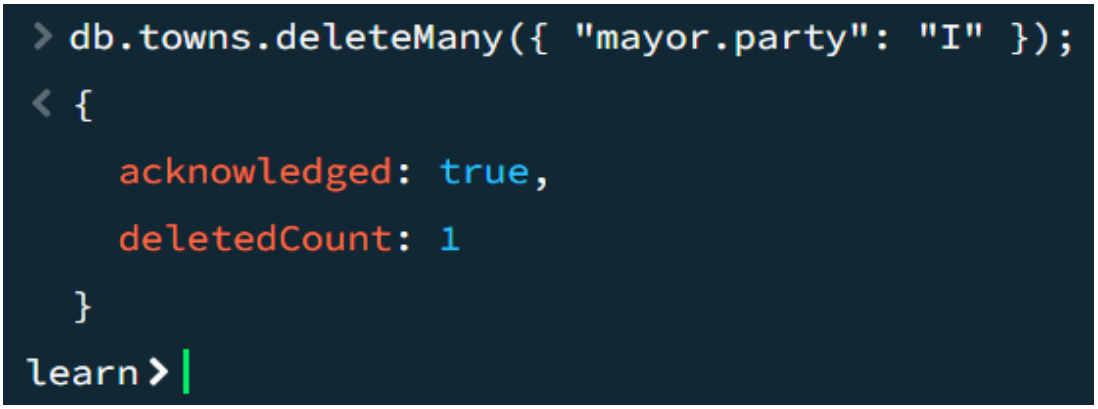
```
> db.unicorns.find({name: 'Aurora'});
< {
    _id: ObjectId('6834edb63cd779a6989d1853'),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape',
      'sugar',
      'lemon'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
learn >
```

Практическое задание 3.4.1:

1. Создайте коллекцию towns, включающую следующие документы.

```
>_MONGOSH
> db.towns.drop();
< true
> db.towns.insertMany([
      {
          name: "Punxsutawney",
          population: 6200,
          last_sensus: new Date("2008-01-31"),
          famous_for: ["phil the groundhog"],
          mayor: {
              name: "Jim Wehrle"
          }
      },
      {
          name: "New York",
          population: 22200000,
          last_sensus: new Date("2009-07-31"),
          famous_for: ["statue of liberty", "food"],
          mayor: {
              name: "Michael Bloomberg",
              party: "I"
          }
      },
```

2. Удалите документы с беспартийными мэрами.



3. Проверьте содержание коллекции.

```
> db.towns.find().pretty()
< {
    _id: ObjectId('6835379c3cd779a6989d1861'),
    name: 'Punxsutawney',
    population: 6200,
    last_sensus: 2008-01-31T00:00:00.000Z,
    famous_for: [
      'phil the groundhog'
    ],
    mayor: {
      name: 'Jim Wehrle'
    }
  }
  {
    _id: ObjectId('6835379c3cd779a6989d1863'),
    name: 'Portland',
    population: 528000,
    last_sensus: 2009-07-20T00:00:00.000Z,
    famous_for: [
      'beer',
      'food'
    ],
```

4. Очистите коллекцию.

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

5. Просмотрите список доступных коллекций.

```
> show collections;
< towns
  unicorns
learn >
```

Практическое задание 4.1.1:

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

```
> db.habitats.insertMany([
      {
          _id: "forest_magic",
          name: "Magic Forest",
          description: "A forest full of glitter and ancient trees."},{
          _id: "mountain_crystal",
          name: "Crystal Mountain",
          description: "High peaks made of pure crystal, reflecting rainbows."},{
          _id: "meadow_dream",
          name: "Dreamy Meadow",
          description: "Endless fields of soft grass and sweet flowers."}]);
< {
    acknowledged: true,
    insertedIds: {
      '0': 'forest_magic',
      '1': 'mountain_crystal',
      '2': 'meadow_dream'
    }
  }
learn >
```

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

```
> db.unicorns.updateOne(
     { name: "Aurora" },{ $set: { habitat: { $ref: "habitats", $id: "forest_magic" } } });
  db.unicorns.updateOne(
     { name: "Pilot" }, { $set: { habitat: { $ref: "habitats", $id: "mountain_crystal" } } });
  db.unicorns.updateOne(
     { name: "Horny" }, { $set: { habitat: { $ref: "habitats", $id: "meadow_dream" } } });
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
learn >
```

3. Проверьте содержание коллекции едиорогов.

```
>_MONGOSH
> db.unicorns.find({ name: { $in: ["Aurora", "Pilot", "Horny"] } }).pretty
< {
    _id: ObjectId('6834edb63cd779a6989d1852'),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 68,
    habitat: DBRef('habitats', 'meadow_dream')
  }
  {
    _id: ObjectId('6834edb63cd779a6989d1853'),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape',
      'sugar',
      'lemon'
    ],
    weight: 450,
```

Практическое задание 4.2.1:

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

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

2. 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 }
  ]
learn >
```

3. Удалите все индексы, кроме индекса для идентификатора.

```
> db.unicorns.dropIndex("name_1");
< { nIndexesWas: 2, ok: 1 }
> db.unicorns.dropIndex("_id_");
⊗ ▸ MongoServerError[InvalidOptions]: cannot drop _id index
learn >
```

Практическое задание 4.4.1:

1. Создайте объемную коллекцию numbers, задействовав курсор.

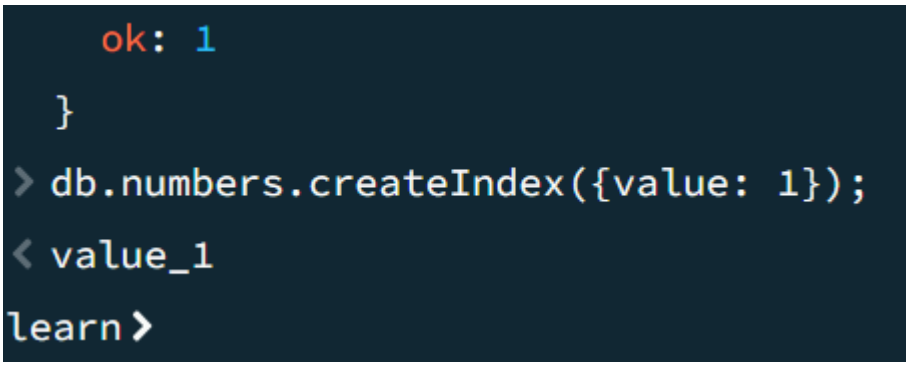
```
> for(let i=0; i<10000; i++) { db.numbers.insertOne({value: i}); }
< {
    acknowledged: true,
    insertedId: ObjectId('6835429c3cd779a6989d3f73')
  }
learn >
```

2. Выберите последних четыре документа.

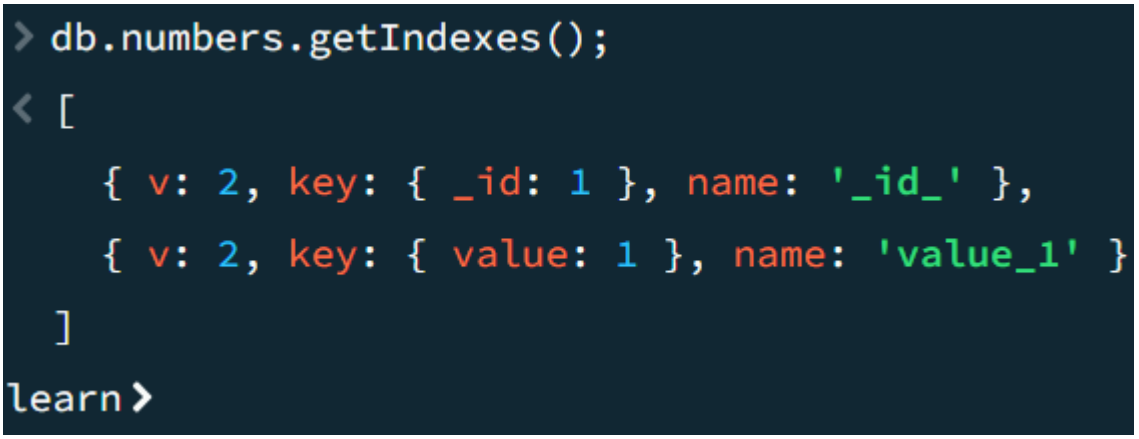
```
> db.numbers.find().sort({value: -1}).limit(4);
< {
    _id: ObjectId('6835429c3cd779a6989d3f73'),
    value: 9999
  }
  {
    _id: ObjectId('6835429c3cd779a6989d3f72'),
    value: 9998
  }
  {
    _id: ObjectId('6835429c3cd779a6989d3f71'),
    value: 9997
  }
  {
    _id: ObjectId('6835429c3cd779a6989d3f70'),
    value: 9996
  }
```

3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis) 107.

4. Создайте индекс для ключа value.



5. Получите информацию о всех индексах коллекции numbres.



Вывод

В ходе выполнение лабораторной работы я научился работать с MongoDB,

выполнять CRUD-операции, узнать про добавление, поиск, удаление, сортировки, операторы, как связывать таблицы и индексацию.