

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
Факультет инфокоммуникационных технологий

ОТЧЁТ
О ЛАБОРАТОРНОЙ РАБОТЕ №5.2
Работа с БД в СУБД MongoDB
по дисциплине:
ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ

Выполнил: Пятыго Д. А.

Группа: К32421

Проверил: Говорова М. М.

Санкт-Петербург

2023

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'});
```

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

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

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

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

Практическое задание 8.1.3:

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

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

Практическое задание 8.1.5:

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

Практическое задание 8.1.6:

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

Практическое задание 8.1.7:

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ `vampires`.

Практическое задание 8.1.9:

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

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

Практическое 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.
- 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: 'Rooooooodles', 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.

Практическое задание 8.2.4:

Вывести список предпочтений.

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

Практическое задание 8.2.6:

1) Выполнить команду:

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

2) Проверить содержимое коллекции unicorns.

Практическое задание 8.2.7:

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

2) Проверить содержимое коллекции unicorns.

Практическое задание 8.2.8:

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

2) Проверить содержимое коллекции unicorns.

Практическое задание 8.2.9:

1) Всем самцам единорогов увеличить количество убитых вампиров на 5.

2) Проверить содержимое коллекции unicorns.

Практическое задание 8.2.10:

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

2) Проверить содержимое коллекции towns.

Практическое задание 8.2.11:

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

2) Проверить содержимое коллекции unicorns.

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.

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.
- 4) Очистите коллекцию.
- 5) Просмотрите список доступных коллекций.

Практическое 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.
- 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: 'Rooooooodles', 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 8.3.2:

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

- 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 .
- 2) Удалите все индексы, кроме индекса для идентификатора.
- 3) Попытайтесь удалить индекс для идентификатора.

Практическое 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 numbers.
- 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?

Выполнение:

Практическое задание 8.1.1.

Формулировка:

- 1) Создайте базу данных learn.
- 2) Заполните коллекцию единорогов unicorns (данные для заполнения указаны выше).
- 3) Используя второй способ, вставьте в коллекцию единорогов документ (данные для вставки указаны выше)
- 4) Проверьте содержимое коллекции с помощью метода find.

Команда:

use learn;

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

Результат выполнения представлен на рисунке 1.

```
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b2"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b3"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
]
learn> █
```

Рисунок 1. Результат выполнения практического задания 8.1.1

Практическое задание 8.1.2.

Формулировка:

- 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.

Команда:

```
db.unicorns.find({gender: 'm'}).sort({name: 1});  
db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3);  
db.unicorns.findOne({gender: 'f', loves: 'carrot'});  
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);
```

Результат выполнения команды приведён на рисунке 2.

```
{  
  _id: ObjectId("6464bf37680a1521211f34ab"),  
  name: 'Roooooodles',  
  loves: [ 'apple' ],  
  weight: 575,  
  gender: 'm',  
  vampires: 99  
},  
{  
  _id: ObjectId("6464bf37680a1521211f34aa"),  
  name: 'Unicrom',  
  loves: [ 'energon', 'redbull' ],  
  weight: 984,  
  gender: 'm',  
  vampires: 182  
}  
]  
learn> db.unicorns.find({gender: 'f'}).sort({name: 1}).limit(3)  
[  
  {  
    _id: ObjectId("6464bf37680a1521211f34a9"),  
    name: 'Aurora',  
    loves: [ 'carrot', 'grape' ],  
    weight: 450,  
    gender: 'f',  
    vampires: 43  
  },  
  {  
    _id: ObjectId("6464bf38680a1521211f34ad"),  
    name: 'Ayna',  
    loves: [ 'strawberry', 'lemon' ],  
    weight: 733,  
    gender: 'f',  
    vampires: 40  
  },  
  {  
    _id: ObjectId("6464bf38680a1521211f34b0"),  
    name: 'Leia',  
    loves: [ 'apple', 'watermelon' ],  
    weight: 601,  
    gender: 'f',  
    vampires: 33  
  }  
]  
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})  
{  
  _id: ObjectId("6464bf37680a1521211f34a9"),  
  name: 'Aurora',  
  loves: [ 'carrot', 'grape' ],  
  weight: 450,  
  gender: 'f',  
  vampires: 43  
}  
learn> db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);  
[  
  {  
    _id: ObjectId("6464bf37680a1521211f34a9"),  
    name: 'Aurora',  
    loves: [ 'carrot', 'grape' ],  
    weight: 450,  
    gender: 'f',  
    vampires: 43  
  }  
]
```

Рисунок 2. Результат выполнения практического задания 8.1.2

Практическое задание 8.1.3.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 3.

```
learn> db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0}).sort({name: 1});
[
  {
    _id: ObjectId("6464bf38680a1521211f34b3"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  },
  {
    _id: ObjectId("6464bf37680a1521211f34a8"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("6464bf38680a1521211f34ae"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b1"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("6464bf38680a1521211f34af"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("6464bf37680a1521211f34ab"),
    name: 'Rooooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("6464bf37680a1521211f34aa"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  }
]
```

Рисунок 3. Результат выполнения практического задания 8.1.3

Практическое задание 8.1.4.

Формулировка:

Вывести список единорогов в обратном порядке добавления.

Команда:

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

Результат выполнения команды представлен на рисунке 4.

```

[learn> db.unicorns.find().sort({$natural: -1});
[
  {
    _id: ObjectId("6464bf38680a1521211f34b3"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b2"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b1"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    _id: ObjectId("6464bf38680a1521211f34b0"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId("6464bf38680a1521211f34af"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    _id: ObjectId("6464bf38680a1521211f34ae"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
]

```

Рисунок 4. Результат выполнения практического задания 8.1.4

Практическое задание 8.1.5.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 5.

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

Рисунок 5. Результат выполнения практического задания 8.1.5

Практическое задание 8.1.6.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 6.

```
[learn> db.unicorns.find({gender: 'f', weight: {$gte: 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'
  }
]
```

Рисунок 6. Результат выполнения практического задания 8.1.6

Практическое задание 8.1.7.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 7.

```
learn> 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
  }
]
```

Рисунок 7. Результат выполнения практического задания 8.1.7

Практическое задание 8.1.8.

Формулировка:

Найти всех единорогов, не имеющих ключ vampires.

Команда:

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

Результат выполнения команды представлен на рисунке 8.

```
learn> db.unicorns.find({vampires: {$exists: false}});
[
  {
    _id: ObjectId("6464bf38680a1521211f34b2"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Рисунок 8. Результат выполнения практического задания 8.1.8

Практическое задание 8.1.9.

Формулировка:

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});
```

Результат выполнения команды представлен на рисунке 9.

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

Рисунок 9. Результат выполнения практического задания 8.1.9

Практическое задание 8.2.1.

Формулировка:

- 1) Создайте коллекцию towns, включающую следующие документы (приведены выше).
- 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.

Команда:

```
db.towns.insert({name: "Punxsutawney", populatiuon: 6200, last_sensus:
ISODate("2008-01-31"), famous_for: [""], mayor: {name: "Jim Wehrle"}});
db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus:
ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: {name:
"Michael Bloomberg", party: "I"}});
db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus:
ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: {name: "Sam
Adams", party: "D"}});
db.towns.find({"mayor.party": "I"}, {_id: 0, name: 1, mayor: 1});
db.towns.find({"mayor.party": {$exists: false}}, {_id: 0, name: 1, mayor: 1});
```

Результат выполнения команды представлен на рисунке 10.

```

towns> db.towns.find({"mayor.party": "I"}, {_id: 0, name: 1, mayor: 1});
[
  {
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
towns> db.towns.find({"mayor.party": { $exists: false }}, {_id: 0, name: 1, mayor: 1});
[ { name: 'Punxsutawney', mayor: { name: 'Jim Wehrle' } } ]

```

Рисунок 10. Результат выполнения практического задания 8.2.1

Практическое задание 8.2.2.

Формулировка:

- 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.
- 4) Содержание коллекции единорогов unicorns (приведено выше).

Команда:

```

function getMaleUnicorns() {return db.unicorns.find({gender: 'm'})};
var cursor = getMaleUnicorns().sort({name: 1}).limit(2);
cursor.forEach(function(obj) {print(obj)});

```

Результат выполнения команды представлен на рисунке 11.

```

[learn> cursor.forEach(function(obj) { print(obj); });
{
  _id: ObjectId("6464bf38680a1521211f34b3"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("6464bf37680a1521211f34a8"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}

```

Рисунок 11. Результат выполнения практического задания 8.2.2

Практическое задание 8.2.3.

Формулировка:

Вывести количество самок единорогов весом от полутонны до 600 кг.

Команда:

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

Результат выполнения команды представлен на рисунке 12.

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

Рисунок 12. Результат выполнения практического задания 8.2.3

Практическое задание 8.2.4.

Формулировка:

Вывести список предпочтений.

Команда:

```
db.unicorns.distinct("loves");
```

Результат выполнения команды представлен на рисунке 13.

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

Рисунок 13. Результат выполнения практического задания 8.2.4

Практическое задание 8.2.5.

Посчитать количество особей единорогов обоих полов.

Команда:

```
db.unicorns.aggregate([{$group: {_id: "$gender", count: {$sum: 1}}}]
```

Результат выполнения команды представлен на рисунке 14.

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

Рисунок 14. Результат выполнения практического задания 8.2.5

Практическое задание 8.2.6.

Формулировка:

1) Выполнить команду:

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

2) Проверить содержимое коллекции unicorns.

Команда:

```
db.unicorns.updateOne({name: 'Barney'}, {$set: {name: 'Barney', loves: ['grape'],  
weight: 340, gender: 'm'}}), {upsert: true});  
db.unicorns.find();
```

Результат выполнения команды представлен на рисунке 15.

Комментарий: поскольку в версии 6.0.6 команда save не поддерживается, необходимо использовать аналогичную команду updateOne.

```
  {  
    _id: ObjectId("6464bf38680a1521211f34b3"),  
    name: 'Dunx',  
    loves: [ 'grape', 'watermelon' ],  
    weight: 704,  
    gender: 'm',  
    vampires: 165  
  },  
  {  
    _id: ObjectId("6467afb2df9a252e6c7424fa"),  
    name: 'Barney',  
    loves: [ 'grape' ],  
    weight: 340,  
    gender: 'm'  
  }  
]
```

Рисунок 15. Результат выполнения практического задания 8.2.6

Практическое задание 8.2.7.

Формулировка:

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

Команда:

```
db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}});  
db.unicorns.find();
```

Результат выполнения команды представлен на рисунке 16.

```
{  
  _id: ObjectId("6464bf38680a1521211f34ad"),  
  name: 'Ayna',  
  loves: [ 'strawberry', 'lemon' ],  
  weight: 800,  
  gender: 'f',  
  vampires: 51  
},
```

Рисунок 16. Результат выполнения практического задания 8.2.7

Практическое задание 8.2.8.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 17.

```
{  
  _id: ObjectId("6464bf38680a1521211f34af"),  
  name: 'Raleigh',  
  loves: [ 'redbull' ],  
  weight: 421,  
  gender: 'm',  
  vampires: 7  
},
```

Рисунок 17. Результат выполнения практического задания 8.2.8

Практическое задание 8.2.9.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 18.

```
learn> db.unicorns.updateMany({gender: 'm'}, {$inc: {vampires: 5}});  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 8,  
  modifiedCount: 8,  
  upsertedCount: 0  
}  
learn> db.unicorns.find();  
[  
  {  
    _id: ObjectId("6464bf37680a1521211f34a8"),  
    name: 'Horny',  
    loves: [ 'carrot', 'papaya' ],  
    weight: 600,  
    gender: 'm',  
    vampires: 68  
  },
```

Рисунок 18. Результат выполнения практического задания 8.2.9

Практическое задание 8.2.10.

Формулировка:

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

Команда:

```
db.towns.updateOne({name: 'Portland'}, {$unset: {'mayor.party': ''}});  
db.towns.find();
```

Результат выполнения команды представлен на рисунке 19.

```
{  
  _id: ObjectId("64676645df9a252e6c7424f9"),  
  name: 'Portland',  
  populatiuon: 528000,  
  last_sensus: ISODate("2009-07-20T00:00:00.000Z"),  
  famous_for: [ 'beer', 'food' ],  
  mayor: { name: 'Sam Adams' }  
}
```

Рисунок 19. Результат выполнения практического задания 8.2.10

Практическое задание 8.2.11.

Формулировка:

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

Команда:

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

Результат выполнения команды представлен на рисунке 20.

```
{  
  _id: ObjectId("6464bfb38680a1521211f34b1"),  
  name: 'Pilot',  
  loves: [ 'apple', 'watermelon', 'chocolate' ],  
  weight: 650,  
  gender: 'm',  
  vampires: 59  
},
```

Рисунок 20. Результат выполнения практического задания 8.2.11

Практическое задание 8.2.12.

Формулировка:

- 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']}}});  
db.unicorns.find();
```

Результат выполнения команды представлен на рисунке 21.

```
{  
  _id: ObjectId("6464bf37680a1521211f34a9"),  
  name: 'Aurora',  
  loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],  
  weight: 450,  
  gender: 'f',  
  vampires: 43  
},
```

Рисунок 21. Результат выполнения практического задания 8.2.12

Практическое задание 8.2.13.

Формулировка:

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

Команда:

```
db.towns.insert({name: "Punxsutawney", popujatiuon: 6200, last_sensus:  
ISODate("2008-01-31"), famous_for: ["phil the groundhog"], mayor: {name: "Jim  
Wehrle"}});  
db.towns.insert({name: "New York", popujatiuon: 22200000, last_sensus:  
ISODate("2009-07-31"), famous_for: ["status of liberty", "food"], mayor: {name:  
"Michael Bloomberg", party: "I"}});  
db.towns.insert({name: "Portland", popujatiuon: 528000, last_sensus:  
ISODate("2009-07-20"), famous_for: ["beer", "food"], mayor: {name: "Sam  
Adams", party: "D"}});  
db.towns.deleteMany({'mayor.party': null});  
db.towns.find();
```



```
db.towns.deleteMany({});
```

```
show collections;
```

Результат выполнения команды представлен на рисунке 22.

```
learn> db.towns.deleteMany({'mayor.party': null});
{ acknowledged: true, deletedCount: 1 }
learn> db.towns.find();
[
  {
    _id: ObjectId("6467c37edf9a252e6c7424ff"),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("6467c37edf9a252e6c742500"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
learn> db.towns.deleteMany({});
{ acknowledged: true, deletedCount: 2 }
learn> show collections;
learn
towns
unicorns
```

Рисунок 22. Результат выполнения практического задания 8.2.13

Практическое задание 8.3.1.

Формулировка:

- 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.
- 4) Содержание коллекции единорогов unicorns (приведено выше).

Команда:

```
db.habitat_zones.insertMany([
```

```
  { _id: 'forest', full_name: 'Enchanted Forest', description: 'A magical forest filled with ancient trees and mystical creatures.' },
```

```
  { _id: 'mountain', full_name: 'Crystal Mountain', description: 'A sparkling mountain with crystal caves and snowy peaks.' },
```

```

    { _id: 'river', full_name: 'Rainbow River', description: 'A river that flows with pure,
clear water and reflects all colors of the rainbow.' },
    { _id: 'meadow', full_name: 'Sunlit Meadow', description: 'A sunlit meadow full of
vibrant flowers and soft, green grass.' }
]);

db.unicorns.updateOne({name: 'Horny'}, {$set: {habitat_zone: {$ref:
"habitat_zones", $id: "forest"}}});

db.unicorns.updateOne({name: 'Aurora'}, {$set: {habitat_zone: {$ref:
"habitat_zones", $id: "mountain"}}});

db.unicorns.updateOne({name: 'Unicrom'}, {$set: {habitat_zone: {$ref:
"habitat_zones", $id: "river"}}});

db.unicorns.updateOne({name: 'Rooooooodles'}, {$set: {habitat_zone: {$ref:
"habitat_zones", $id: "meadow"}}});

db.unicorns.find();

```

Результат выполнения команды представлен на рисунке 23.

```

learn> db.unicorns.find();
[
  {
    _id: ObjectId("6467c5e3df9a252e6c742501"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63,
    habitat_zone: DBRef("habitat_zones", 'forest')
  },
  {
    _id: ObjectId("6467c5e3df9a252e6c742502"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    habitat_zone: DBRef("habitat_zones", 'mountain')
  },
  {
    _id: ObjectId("6467c5e3df9a252e6c742503"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182,
    habitat_zone: DBRef("habitat_zones", 'river')
  },
]

```

Рисунок 23. Результат выполнения практического задания 8.3.1

Практическое задание 8.3.2.

Формулировка:

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

2) Содержание коллекции единорогов unicorns (приведено выше).

Команда:

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

Результат выполнения команды представлен на рисунке 24.

```
learn> db.unicorns.createIndex({name: 1}, {unique: true});
name_1_
```

Рисунок 24. Результат выполнения практического задания 8.3.2

Вывод: уникальный индекс для ключа name коллекции unicorns задать возможно.

Практическое задание 8.3.3.

Формулировка:

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

Команда:

```
db.unicorns.getIndexes();
var indexes = db.unicorns.getIndexes();
for (var i = 0; i < indexes.length; i++) {
    var index = indexes[i];
    if (index.name !== '_id_') {
        db.unicorns.dropIndex(index.name);
    }
};
db.unicorns.dropIndex('_id_');
```

Результат выполнения команды представлен на рисунке 25.

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

learn> for (var i = 0; i < indexes.length; i++) {
...   var index = indexes[i];
...   if (index.name !== '_id_') {
...     db.unicorns.dropIndex(index.name);
...   }
... };

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

Рисунок 25. Результат выполнения практического задания 8.3.3

Вывод: команда удаления индекса ' _id ' вызовет ошибку, потому что поле _id и его индекс используются для идентификации документов в коллекции.

Практическое задание 8.3.4.

Формулировка:

- 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 `numbers`.
- 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?

Команда:

```
for(let i = 0; i < 100000; i++){ db.numbers.insertOne({value: i}); };  
db.numbers.find().sort({value: -1}).limit(4);  
db.numbers.find().sort({value: -1}).limit(4).explain("executionStats");  
db.numbers.createIndex({value: 1});  
db.numbers.getIndexes();  
db.numbers.find().sort({value: -1}).limit(4);  
db.numbers.find().sort({value: -1}).limit(4).explain("executionStats");
```

Результат выполнения команды представлен на рисунке 26.

```
[learn> db.numbers.find().sort({value: -1}).limit(4);  
[  
  { _id: ObjectId("6467cd334d04bd0139ae02a4"), value: 99999 },  
  { _id: ObjectId("6467cd334d04bd0139ae02a3"), value: 99998 },  
  { _id: ObjectId("6467cd334d04bd0139ae02a2"), value: 99997 },  
  { _id: ObjectId("6467cd334d04bd0139ae02a1"), value: 99996 }  
]  
[learn> db.numbers.find().sort({value: -1}).limit(4).explain("executionStats");
```

```

executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 71,
learn> db.numbers.createIndex({value: 1});
value_1
learn> db.numbers.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1' }
]
learn> db.numbers.find().sort({value: -1}).limit(4);
[
  { _id: ObjectId("6467cd334d04bd0139ae02a4"), value: 99999 },
  { _id: ObjectId("6467cd334d04bd0139ae02a3"), value: 99998 },
  { _id: ObjectId("6467cd334d04bd0139ae02a2"), value: 99997 },
  { _id: ObjectId("6467cd334d04bd0139ae02a1"), value: 99996 }
]
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 1,

```

Рисунок 26. Результат выполнения практического задания 8.3.4

Вывод: запрос с индексами оказался более эффективен (executionTimeMillis: 1), чем запрос без индексов (executionTimeMillis: 71). Действительно, запросы с индексами быстрее, чем запросы без индексов, особенно на больших наборах данных. Это связано с тем, что индекс позволяет быстро найти и извлечь нужные документы, не просматривая каждый документ в коллекции.

Выводы: в данной лабораторной работе была освоена работа с MongoDB, а именно:

- 1) разработка запросов поиска с использованием различных операторов (например, \$gte, \$in, \$exists, \$slice) для отбора и ограничения результатов поиска в коллекции;
- 2) применение функций find(), findOne(), sort(), limit() и update() для выполнения различных операций с данными в коллекции;
- 3) обновление документов с использованием метода update() и различных операторов обновления (например, \$set и \$addToSet);
- 4) изменение структуры документов в коллекции, включая добавление и удаление полей;
- 5) создание, удаление и анализ индексов коллекций для оптимизации производительности запросов;
- 6) создание ссылок между документами различных коллекций;
- 7) понимание способов работы с курсорами в MongoDB, включая их использование для обхода документов коллекции;
- 8) освоение методов aggregate() и count() для агрегации и подсчета документов в коллекции соответственно.