

**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 Тимаков Егор Павлович

Факультет прикладной информатики

Группа К3241

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

**Санкт-Петербург
2024/2025**

Введение

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. Создать базу данных learn.
2. Заполнить коллекцию единорогов unicorns
3. Использовать второй способ, вставьте в коллекцию единорогов
4. Проверьте содержимое коллекции с помощью метода find.

Скрипт который создает бд learn и заполняет коллекцию unicorns:

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: 'Roooooodles', 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'});
```

Содержимое коллекции unicorns:

```
[  
  {  
    "_id": {"$oid": "684347cba554a4bcf450eb67"},  
    "gender": "m",  
    "loves": ["carrot", "papaya"],  
    "name": "Horny",  
    "vampires": 63,  
    "weight": 600  
  },  
  {  
    "_id": {"$oid": "68434856febf7818232556c1"},  
    "gender": "m",  
    "loves": ["carrot", "papaya"],  
    "name": "Horny",  
    "vampires": 63,  
    "weight": 600  
  },  
  {  
    "_id": {"$oid": "68434856febf7818232556c2"},  
    "gender": "f",  
    "loves": ["carrot", "grape"],  
    "name": "Aurora",  
    "vampires": 43,  
    "weight": 450  
  },  
  {  
    "_id": {"$oid": "68434857febf7818232556c3"},  
    "gender": "m",  
    "loves": ["energon", "redbull"],  
    "name": "Unicrom",  
    "vampires": 182,  
  }  
]
```

```
"weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 99,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 40,
  "weight": 733
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 39,
  "weight": 690
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar"],
  "name": "Raleigh",
  "vampires": 2,
  "weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},

```

```

    "gender": "f",
    "loves": ["apple", "watermelon"],
    "name": "Leia",
    "vampires": 33,
    "weight": 601
  },
  {
    "_id": {"$oid": "6843485afebf7818232556ca"},
    "gender": "m",
    "loves": ["apple", "watermelon"],
    "name": "Pilot",
    "vampires": 54,
    "weight": 650
  },
  {
    "_id": {"$oid": "6843485afebf7818232556cb"},
    "gender": "f",
    "loves": ["grape", "carrot"],
    "name": "Nimue",
    "weight": 540
  },
  {
    "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
    "gender": "m",
    "loves": ["grape", "watermelon"],
    "name": "Dunx",
    "vampires": 165,
    "weight": 704
  }
]

```

Практическое задание 2.2.1:

1. Сформулируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте список по имени.
2. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций `findOne` и `limit`.

Запрос который выводит список самцов:

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

Результат выполнения запроса:

```
[
```

```
{
  "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
  "gender": "m",
  "loves": ["grape", "watermelon"],
  "name": "Dunx",
  "vampires": 165,
  "weight": 704
},
{
  "_id": {"$oid": "684347cba554a4bcf450eb67"},
  "gender": "m",
  "loves": ["carrot", "papaya"],
  "name": "Horny",
  "vampires": 63,
  "weight": 600
},
{
  "_id": {"$oid": "68434856febf7818232556c1"},
  "gender": "m",
  "loves": ["carrot", "papaya"],
  "name": "Horny",
  "vampires": 63,
  "weight": 600
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 39,
  "weight": 690
},
{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "loves": ["apple", "watermelon"],
  "name": "Pilot",
  "vampires": 54,
  "weight": 650
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar"],
```

```

    "name": "Raleigh",
    "vampires": 2,
    "weight": 421
  },
  {
    "_id": {"$oid": "68434857febf7818232556c4"},
    "gender": "m",
    "loves": ["apple"],
    "name": "Rooooooodles",
    "vampires": 99,
    "weight": 575
  },
  {
    "_id": {"$oid": "68434857febf7818232556c3"},
    "gender": "m",
    "loves": ["energon", "redbull"],
    "name": "Unicrom",
    "vampires": 182,
    "weight": 984
  }
]

```

Запрос который выводит самок, с ограничением первых трех особей:

```
db.unicorns.find( { gender: "f" } ) .sort( { "name": 1 } ) .limit(3)
```

Результат выполнения запроса:

```

[
  {
    "_id": {"$oid": "68434856febf7818232556c2"},
    "gender": "f",
    "loves": ["carrot", "grape"],
    "name": "Aurora",
    "vampires": 43,
    "weight": 450
  },
  {
    "_id": {"$oid": "68434858febf7818232556c6"},
    "gender": "f",
    "loves": ["strawberry", "lemon"],
    "name": "Ayna",
    "vampires": 40,
    "weight": 733
  },
  {

```

```
"_id": {"$oid": "68434859febf7818232556c9"},
"gender": "f",
"loves": ["apple", "watermelon"],
"name": "Leia",
"vampires": 33,
"weight": 601
}
]
```

Запрос, который выводит всех самок, которые carrot. Список ограничен первой особью:

```
db.unicorns.find({loves: "carrot"}).sort().limit(1)
```

Результат выполнения запроса:

```
[
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 63,
    "weight": 600
  }
]
```

Практическое задание 2.2.2:

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

Запрос, который выводит список самцов без полей о предпочтениях:

```
db.unicorns.find({gender: "m"}, {loves: 0, _id: 0});
```

Результат выполнения запроса:

```
[
  {
    "gender": "m",
    "name": "Horny",
    "vampires": 63,
    "weight": 600
  },
  {
    "gender": "m",
    "name": "Horny",
    "vampires": 63,
  }
]
```



```
    "weight": 600
  },
  {
    "gender": "m",
    "name": "Unicrom",
    "vampires": 182,
    "weight": 984
  },
  {
    "gender": "m",
    "name": "Rooooooodles",
    "vampires": 99,
    "weight": 575
  },
  {
    "gender": "m",
    "name": "Kenny",
    "vampires": 39,
    "weight": 690
  },
  {
    "gender": "m",
    "name": "Raleigh",
    "vampires": 2,
    "weight": 421
  },
  {
    "gender": "m",
    "name": "Pilot",
    "vampires": 54,
    "weight": 650
  },
  {
    "gender": "m",
    "name": "Dunx",
    "vampires": 165,
    "weight": 704
  }
]
```

Практическое задание 2.2.3:

Вывести список единорогов в обратном порядке добавления

Запрос, который выводит список единорогов в обратном порядке добавления:

```
db.unicorns.find().sort({ _id: -1 })
```

Результат выполнения запроса:

```
[
  {
    "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
    "gender": "m",
    "loves": ["grape", "watermelon"],
    "name": "Dunx",
    "vampires": 165,
    "weight": 704
  },
  {
    "_id": {"$oid": "6843485afebf7818232556cb"},
    "gender": "f",
    "loves": ["grape", "carrot"],
    "name": "Nimue",
    "weight": 540
  },
  {
    "_id": {"$oid": "6843485afebf7818232556ca"},
    "gender": "m",
    "loves": ["apple", "watermelon"],
    "name": "Pilot",
    "vampires": 54,
    "weight": 650
  },
  {
    "_id": {"$oid": "68434859febf7818232556c9"},
    "gender": "f",
    "loves": ["apple", "watermelon"],
    "name": "Leia",
    "vampires": 33,
    "weight": 601
  },
  {
    "_id": {"$oid": "68434859febf7818232556c8"},
    "gender": "m",
    "loves": ["apple", "sugar"],
    "name": "Raleigh",
    "vampires": 2,
    "weight": 421
  },
  {

```

```
"_id": {"$oid": "68434859febf7818232556c7"},
"gender": "m",
"loves": ["grape", "lemon"],
"name": "Kenny",
"vampires": 39,
"weight": 690
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 40,
  "weight": 733
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 99,
  "weight": 575
},
{
  "_id": {"$oid": "68434857febf7818232556c3"},
  "gender": "m",
  "loves": ["energon", "redbull"],
  "name": "Unicrom",
  "vampires": 182,
  "weight": 984
},
{
  "_id": {"$oid": "68434856febf7818232556c2"},
  "gender": "f",
  "loves": ["carrot", "grape"],
  "name": "Aurora",
```

```

    "vampires": 43,
    "weight": 450
  },
  {
    "_id": {"$oid": "68434856febf7818232556c1"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 63,
    "weight": 600
  },
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 63,
    "weight": 600
  }
]

```

Практическое задание 2.1.4:

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

Запрос, который выводит список единорогов с название первого любимого блюда, без идентификатора:

```

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

```

Результат выполнения запроса:

```

[
  {
    "loves": ["carrot"],
    "name": "Horny"
  },

```

```
{
  "loves": ["carrot"],
  "name": "Horny"
},
{
  "loves": ["carrot"],
  "name": "Aurora"
},
{
  "loves": ["energon"],
  "name": "Unicrom"
},
{
  "loves": ["apple"],
  "name": "Rooooooodles"
},
{
  "loves": ["apple"],
  "name": "Solnara"
},
{
  "loves": ["strawberry"],
  "name": "Ayna"
},
{
  "loves": ["grape"],
  "name": "Kenny"
},
{
  "loves": ["apple"],
  "name": "Raleigh"
},
{
  "loves": ["apple"],
  "name": "Leia"
},
{
  "loves": ["apple"],
  "name": "Pilot"
},
{
  "loves": ["grape"],
  "name": "Nimue"
},
```

```
{
  "loves": ["grape"],
  "name": "Dunx"
}
]
```

Практическое задание 2.3.1:

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

Запрос, который выводит список самок единорогов, весом от 500 кг до 700 кг, исключив вывод идентификатора:

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

Результат выполнения запроса:

```
[
  {
    "gender": "f",
    "loves": ["apple", "carrot", "chocolate"],
    "name": "Solnara",
    "vampires": 80,
    "weight": 550
  },
  {
    "gender": "f",
    "loves": ["apple", "watermelon"],
    "name": "Leia",
    "vampires": 33,
    "weight": 601
  },
  {
    "gender": "f",
    "loves": ["grape", "carrot"],
    "name": "Nimue",
    "weight": 540
  }
]
```

Практическое задание 2.3.2:

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

Запрос, который выводит список самцов весом от 500 кг и предпочитающих grape и lemon, без идентификатора:

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

Результат выполнения запроса:

```
[
  {
    "gender": "m",
    "loves": ["grape", "lemon"],
    "name": "Kenny",
    "vampires": 39,
    "weight": 690
  }
]
```

Практическое задание 2.3.3:

Найти всех единорогов, не имеющих ключ vampires.

Запрос, который выводит единорогов не имеющих ключ vampires:

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

Результат выполнения запроса:

```
[
  {
    "_id": {"$oid": "6843485afebf7818232556cb"},
    "gender": "f",
    "loves": ["grape", "carrot"],
    "name": "Nimue",
  }
]
```

```
    "weight": 540
  }
]
```

Практическое задание 2.3.4:

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

Запрос, который выводит 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 запроса:

```
[
  {
    "loves": ["grape"],
    "name": "Dunx"
  },
  {
    "loves": ["carrot"],
    "name": "Horny"
  },
  {
    "loves": ["carrot"],
    "name": "Horny"
  },
  {
    "loves": ["grape"],
    "name": "Kenny"
  },
  {
    "loves": ["apple"],
    "name": "Pilot"
  },
  {
    "loves": ["apple"],
    "name": "Raleigh"
  }
]
```



```

    },
    {
      "loves": ["apple"],
      "name": "Rooooooodles"
    },
    {
      "loves": ["energon"],
      "name": "Unicrom"
    }
  ]
}

```

Практическое задание 3.1.1:

- 1) Создать коллекцию towns, включающую следующие документы:


```

      {name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""]},
      {name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty", "food"]},
      {name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"]},
      {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.

Запрос который создает коллекцию towns:

```

db.towns.insertOne({
  name: "Punxsutawney",
  populatiuon: 6200,
  last_sensus: ISODate("2008-01-31"),
  famous_for: [""],
  mayor: {
    name: "Jim Wehrle"
  }
})

db.towns.insertOne({
  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.insertOne({
  name: "Portland",
  populatiuon: 528000,
  last_sensus: ISODate("2009-07-20"),
  famous_for: ["beer", "food"],
  mayor: {
    name: "Sam Adams",
    party: "D"
  }
})

```

Запрос, который выводит список городов, с независимыми мэрами

```

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

```

Результат выполнения запроса:

```

[
  {
    "mayor": {
      "name": "Michael Bloomberg",
      "party": "I"
    },
    "name": "New York"
  }
]

```

Запрос который выводит список городов, где беспартийные мэры:

```

[
  {
    "mayor": {
      "name": "Jim Wehrle"
    },
    "name": "Punxsutawney"
  }
]

```

Результат выполнения запроса:

```
[
  {
    "mayor": {
      "name": "Jim Wehrle"
    },
    "name": "Punxsutawney"
  }
]
```

Практическое 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`.
- 4) Содержание коллекции единорогов `unicorns`:

Запрос, который создает функцию для вывода списка самцов единорогов:

```
function getMaleUnicorns() {
  return db.unicorns.find({ gender: "m" }).toArray();
}
```

Запрос, который создает курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке:

```
function getMaleUnicornsCursor() {
  return db.unicorns
    .find({ gender: "m" })
    .sort({ name: 1 })
    .limit(2);
}
```

Запрос, выводящий список с использованием `forEach`:

```
const cursor = getMaleUnicornsCursor()
cursor.forEach(doc => printjson(doc))
```

Результат выполнения запроса:

```
{
  _id: ObjectId('684daf32c37cd51e31e42c5d'),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ]
}
```

```

],
weight: 704,
gender: 'm',
vampires: 165
}
{
  _id: ObjectId('684347cba554a4bcf450eb67'),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}

```

Практическое задание 3.2.1:

Вывести количество самок единорогов весом от полутонны до 600 кг.

Запрос, который выводит количество самок единорогов весом от 500 кг до 600 кг:

```

db.unicorns.countDocuments({
  gender: "f",
  weight: { $gte: 500, $lte: 600 }
})

```

Результат выполнения запроса:

2

Практическое задание 3.2.2:

Вывести список предпочтений.

Запрос который выводит список предпочтений:

```

db.unicorns.distinct("loves")

```

Результат выполнения запроса:

```

[
  {
    "result": "apple"
  },
  {
    "result": "carrot"
  },
  {
    "result": "chocolate"
  },
]

```

```
[
  {
    "result": "energon"
  },
  {
    "result": "grape"
  },
  {
    "result": "lemon"
  },
  {
    "result": "papaya"
  },
  {
    "result": "redbull"
  },
  {
    "result": "strawberry"
  },
  {
    "result": "sugar"
  },
  {
    "result": "watermelon"
  }
]
```

Практическое задание 3.2.3:

Посчитать количество особей единорогов обоих полов.

Запрос который подсчитывает количество особей единорогов обоих полов:

db.unicorns.aggregate ([

```
{
  $group: {
    _id: "$gender",
    count: { $sum: 1 }
  }
}
] )
```

Результат выполнения запроса:

```
[
  {
```

```
"_id": "f",
"count": 5
},
{
  "_id": "m",
  "count": 8
}
]
```

Практическое задание 3.3.1:

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

Добавляем документ в коллекцию(метод .save уже не поддерживается):

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

Проверим содержимое коллекции unicorns:

```
db.unicorns.find() ;
```

Результат выполнения запроса:

```
[
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 63,
    "weight": 600
  },
  {
    "_id": {"$oid": "68434856febf7818232556c1"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 63,
  }
]
```

```
"weight": 600
},
{
  "_id": {"$oid": "68434856febf7818232556c2"},
  "gender": "f",
  "loves": ["carrot", "grape"],
  "name": "Aurora",
  "vampires": 43,
  "weight": 450
},
{
  "_id": {"$oid": "68434857febf7818232556c3"},
  "gender": "m",
  "loves": ["energon", "redbull"],
  "name": "Unicrom",
  "vampires": 182,
  "weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 99,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
```

```
"loves": ["strawberry", "lemon"],
"name": "Ayna",
"vampires": 40,
"weight": 733
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 39,
  "weight": 690
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar"],
  "name": "Raleigh",
  "vampires": 2,
  "weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},
  "gender": "f",
  "loves": ["apple", "watermelon"],
  "name": "Leia",
  "vampires": 33,
  "weight": 601
},
{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "loves": ["apple", "watermelon"],
  "name": "Pilot",
  "vampires": 54,
  "weight": 650
},
}
```



```

{
  "_id": {"$oid": "6843485afebf7818232556cb"},
  "gender": "f",
  "loves": ["grape", "carrot"],
  "name": "Nimue",
  "weight": 540
},
{
  "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
  "gender": "m",
  "loves": ["grape", "watermelon"],
  "name": "Dunx",
  "vampires": 165,
  "weight": 704
},
{
  "_id": {"$oid": "684e1c3bc37cd51e31e42c66"},
  "gender": "m",
  "loves": ["grape"],
  "name": "Barney",
  "weight": 340
}
]

```

Единорог с именем Barney был добавлен в коллекцию.

Практическое задание 3.3.2:

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

Запрос для изменения в бд:

```

db.unicorns.updateOne(
  { name: "Ayna", gender: "f" },
  {
    $set: {
      weight: 800,

```

```
    vampires: 51
  }
}
)
```

Проверим изменение в коллекции:

```
[
  {
    "_id": {"$oid": "68434858febf7818232556c6"},
    "gender": "f",
    "loves": ["strawberry", "lemon"],
    "name": "Ayna",
    "vampires": 51,
    "weight": 800
  }
]
```

Практическое задание 3.3.3:

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

Запрос для изменения данных:

```
db.unicorns.updateOne(
  { name: "Raleigh", gender: "m" },
  {
    $addToSet: { loves: "redbull" }
  }
)
```

Проверим изменения в коллекции:

```
[
  {
    "_id": {"$oid": "68434859febf7818232556c8"},
    "gender": "m",
    "loves": ["apple", "sugar", "redbull"],
  }
]
```

```
"name": "Raleigh",
"vampires": 2,
"weight": 421
}
]
```

Практическое задание 3.3.4:

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

Запрос, который увеличивает все самцам количество убитых вампиров на 5:

```
db.unicorns.updateMany(
```

```
{ gender: "m" },
{ $inc: { vampires: 5 } }
)
```

Проверим содержимое коллекции unicorns:

```
[
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 68,
    "weight": 600
  },
  {
    "_id": {"$oid": "68434856febf7818232556c1"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 68,
    "weight": 600
  },
  {

```

```
"_id": {"$oid": "68434856febf7818232556c2"},
"gender": "f",
"loves": ["carrot", "grape"],
"name": "Aurora",
"vampires": 43,
"weight": 450
},
{
  "_id": {"$oid": "68434857febf7818232556c3"},
  "gender": "m",
  "loves": ["energon", "redbull"],
  "name": "Unicrom",
  "vampires": 187,
  "weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 104,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 51,
```

```
"weight": 800
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 44,
  "weight": 690
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar", "redbull"],
  "name": "Raleigh",
  "vampires": 7,
  "weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},
  "gender": "f",
  "loves": ["apple", "watermelon"],
  "name": "Leia",
  "vampires": 33,
  "weight": 601
},
{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "loves": ["apple", "watermelon"],
  "name": "Pilot",
  "vampires": 59,
  "weight": 650
},
{
  "_id": {"$oid": "6843485afebf7818232556cb"},
  "gender": "f",
```

```

    "loves": ["grape", "carrot"],
    "name": "Nimue",
    "weight": 540
  },
  {
    "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
    "gender": "m",
    "loves": ["grape", "watermelon"],
    "name": "Dunx",
    "vampires": 170,
    "weight": 704
  },
  {
    "_id": {"$oid": "684e1c3bc37cd51e31e42c66"},
    "gender": "m",
    "loves": ["grape"],
    "name": "Barney",
    "vampires": 5,
    "weight": 340
  }
]

```

Практическое задание 3.3.5:

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

Запрос, который изменяет информацию о городе портланд:

db.towns.updateOne (

```

{ name: "Portland" },
{
  $unset: { "mayor.party": "" }
}
)

```

Содержимое коллекции towns:

[

```

{
  "_id": {"$oid": "684dc6a7c37cd51e31e42c60"},
  "famous_for": [""],
  "last_sensus": {"$date": "2008-01-31T00:00:00.000Z"},
  "mayor": {
    "name": "Jim Wehrle"
  },
  "name": "Punxsutawney",
  "populatiuon": 6200
},
{
  "_id": {"$oid": "684dc6a8c37cd51e31e42c61"},
  "famous_for": ["status of liberty", "food"],
  "last_sensus": {"$date": "2009-07-31T00:00:00.000Z"},
  "mayor": {
    "name": "Michael Bloomberg",
    "party": "I"
  },
  "name": "New York",
  "populatiuon": 22200000
},
{
  "_id": {"$oid": "684dc6a8c37cd51e31e42c62"},
  "famous_for": ["beer", "food"],
  "last_sensus": {"$date": "2009-07-20T00:00:00.000Z"},
  "mayor": {
    "name": "Sam Adams"
  },
  "name": "Portland",
  "populatiuon": 528000
}
]

```

Практическое задание 3.3.6:

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

Запрос, который изменяет информацию о самце единорога Pilot:

```
db.unicorns.updateOne(  
  { name: "Pilot", gender: "m" },  
  {  
    $addToSet: { loves: "шоколад" }  
  }  
)
```

Проверка содержимого коллекции unicorns:

```
[  
  {  
    "_id": {"$oid": "684347cba554a4bcf450eb67"},  
    "gender": "m",  
    "loves": ["carrot", "papaya"],  
    "name": "Horny",  
    "vampires": 68,  
    "weight": 600  
  },  
  {  
    "_id": {"$oid": "68434856febf7818232556c1"},  
    "gender": "m",  
    "loves": ["carrot", "papaya"],  
    "name": "Horny",  
    "vampires": 68,  
    "weight": 600  
  },  
  {  
    "_id": {"$oid": "68434856febf7818232556c2"},  
    "gender": "f",  
    "loves": ["carrot", "grape"],  
    "name": "Aurora",  
    "vampires": 43,  
    "weight": 450  
  },  
  {  
    "_id": {"$oid": "68434857febf7818232556c3"},  
    "gender": "m",
```



```
"loves": ["energon", "redbull"],
"name": "Unicrom",
"vampires": 187,
"weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 104,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 51,
  "weight": 800
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 44,
  "weight": 690
},
}
```

```
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar", "redbull"],
  "name": "Raleigh",
  "vampires": 7,
  "weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},
  "gender": "f",
  "loves": ["apple", "watermelon"],
  "name": "Leia",
  "vampires": 33,
  "weight": 601
},
{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "loves": ["apple", "watermelon", "chocolate"],
  "name": "Pilot",
  "vampires": 59,
  "weight": 650
},
{
  "_id": {"$oid": "6843485afebf7818232556cb"},
  "gender": "f",
  "loves": ["grape", "carrot"],
  "name": "Nimue",
  "weight": 540
},
{
  "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
  "gender": "m",
  "loves": ["grape", "watermelon"],
  "name": "Dunx",
  "vampires": 170,
```

```

    "weight": 704
  },
  {
    "_id": {"$oid": "684e1c3bc37cd51e31e42c66"},
    "gender": "m",
    "loves": ["grape"],
    "name": "Barney",
    "vampires": 5,
    "weight": 340
  }
]

```

Практическое задание 3.3.7:

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

Запрос, который изменяет информацию о самке единорога Aurora:

```

db.unicorns.updateOne(
  { name: "Aurora", gender: "f" },
  {
    $addToSet: {
      loves: {
        $each: ["sugar", "lemon"]
      }
    }
  }
)

```

Проверка содержимого коллекции:

```

[
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 68,
  }
]

```

```
"weight": 600
},
{
  "_id": {"$oid": "68434856febf7818232556c1"},
  "gender": "m",
  "loves": ["carrot", "papaya"],
  "name": "Horny",
  "vampires": 68,
  "weight": 600
},
{
  "_id": {"$oid": "68434856febf7818232556c2"},
  "gender": "f",
  "loves": ["carrot", "grape", "sugar", "lemon"],
  "name": "Aurora",
  "vampires": 43,
  "weight": 450
},
{
  "_id": {"$oid": "68434857febf7818232556c3"},
  "gender": "m",
  "loves": ["energon", "redbull"],
  "name": "Unicrom",
  "vampires": 187,
  "weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Rooooooodles",
  "vampires": 104,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
```

```
"loves": ["apple", "carrot", "chocolate"],
"name": "Solnara",
"vampires": 80,
"weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 51,
  "weight": 800
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 44,
  "weight": 690
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
  "loves": ["apple", "sugar", "redbull"],
  "name": "Raleigh",
  "vampires": 7,
  "weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},
  "gender": "f",
  "loves": ["apple", "watermelon"],
  "name": "Leia",
  "vampires": 33,
  "weight": 601
},
}
```

```

{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "loves": ["\apple\", \"watermelon\", \"chocolate\""],
  "name": "Pilot",
  "vampires": 59,
  "weight": 650
},
{
  "_id": {"$oid": "6843485afebf7818232556cb"},
  "gender": "f",
  "loves": ["grape", "carrot"],
  "name": "Nimue",
  "weight": 540
},
{
  "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
  "gender": "m",
  "loves": ["grape", "watermelon"],
  "name": "Dunx",
  "vampires": 170,
  "weight": 704
},
{
  "_id": {"$oid": "684e1c3bc37cd51e31e42c66"},
  "gender": "m",
  "loves": ["grape"],
  "name": "Barney",
  "vampires": 5,
  "weight": 340
}
]

```

Практическое задание 3.3.7:

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.

Добавим в коллекцию towns документы:

```
db.towns.insertOne({  
  name: "Punxsutawney",  
  popujatiuon: 6200,  
  last_sensus: ISODate("2008-01-31"),  
  famous_for: ["phil the groundhog"],  
  mayor: {  
    name: "Jim Wehrle"  
  }  
})  
  
db.towns.insertOne({  
  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.insertOne({  
  name: "Portland",  
  popujatiuon: 528000,  
  last_sensus: ISODate("2009-07-20"),  
  famous_for: ["beer", "food"],  
  mayor: {
```

```
    name: "Sam Adams",  
    party: "D"  
  }  
})
```

Удаление документов с беспартийными мэрами:

```
db.towns.deleteMany({  
  "mayor.party": { $exists: false }  
})
```

Проверка содержания коллекции:

```
[  
  {  
    "_id": {"$oid": "684e2377c37cd51e31e42c6a"},  
    "famous_for": ["status of liberty", "food"],  
    "last_sensus": {"$date": "2009-07-31T00:00:00.000Z"},  
    "mayor": {  
      "name": "Michael Bloomberg",  
      "party": "I"  
    },  
    "name": "New York",  
    "popujatiuon": 22200000  
  },  
  {  
    "_id": {"$oid": "684e2379c37cd51e31e42c6c"},  
    "famous_for": ["beer", "food"],  
    "last_sensus": {"$date": "2009-07-20T00:00:00.000Z"},  
    "mayor": {  
      "name": "Sam Adams",  
      "party": "D"  
    },  
    "name": "Portland",  
    "popujatiuon": 528000  
  }  
]
```


Очищение коллекции:

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

Просмотр доступных коллекций:

```
show collections
```

Результат выполнения:

```
[
  {
    "badge": "",
    "name": "towns"
  }
]
```

Практическое задание 4.1.1:

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.

Создание коллекции зон обитания единорогов:

```
db.habitats.insertMany([
```

```
{
  _id: "nf",
  full_name: "Northern Forests",
  description: "Dense and misty forests in the far
north. Home to many magical creatures."
},
{
  _id: "ds",
```

```

    full_name: "Desert Sands",

    description: "Hot and dry desert zone where
unicorns survive by magic-fed oases."

},

{

    _id: "hg",

    full_name: "High Gardens",

    description: "Floating gardens in the sky where
only winged unicorns live."

}

] )

```

Добавление для 3 единорогов зон обитания:

```

db.unicorns.updateOne(

```

```

{ name: "Horny" },

{

    $set: {

        habitat: {

            $ref: "habitats",

            $id: "nf"

        }

    }

}

)

```

```
db.unicorns.updateOne(
  { name: "Aurora" },
  {
    $set: {
      habitat: {
        $ref: "habitats",
        $id: "hg"
      }
    }
  }
)
```

```
db.unicorns.updateOne(
  { name: "Pilot" },
  {
    $set: {
      habitat: {
        $ref: "habitats",
        $id: "ds"
      }
    }
  }
)
```

)

Проверка содержимого коллекции unicorns:

```
[
  {
    "_id": {"$oid": "684347cba554a4bcf450eb67"},
    "gender": "m",
    "habitat": "{ \"$ref\" : \"habitats\", \"$id\" : \"nf\" }",
    "habitat_id": "nf",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 68,
    "weight": 600
  },
  {
    "_id": {"$oid": "68434856febf7818232556c1"},
    "gender": "m",
    "loves": ["carrot", "papaya"],
    "name": "Horny",
    "vampires": 68,
    "weight": 600
  },
  {
    "_id": {"$oid": "68434856febf7818232556c2"},
    "gender": "f",
    "habitat": "{ \"$ref\" : \"habitats\", \"$id\" : \"hg\" }",
    "habitat_id": "hg",
    "loves": ["carrot", "grape", "sugar", "lemon"],
    "name": "Aurora",
    "vampires": 43,
    "weight": 450
  },
  {
    "_id": {"$oid": "68434857febf7818232556c3"},
    "gender": "m",
    "loves": ["energon", "redbull"],
    "name": "Unicrom",
    "vampires": 187,
```

```
"weight": 984
},
{
  "_id": {"$oid": "68434857febf7818232556c4"},
  "gender": "m",
  "loves": ["apple"],
  "name": "Roooooodles",
  "vampires": 104,
  "weight": 575
},
{
  "_id": {"$oid": "68434858febf7818232556c5"},
  "gender": "f",
  "loves": ["apple", "carrot", "chocolate"],
  "name": "Solnara",
  "vampires": 80,
  "weight": 550
},
{
  "_id": {"$oid": "68434858febf7818232556c6"},
  "gender": "f",
  "loves": ["strawberry", "lemon"],
  "name": "Ayna",
  "vampires": 51,
  "weight": 800
},
{
  "_id": {"$oid": "68434859febf7818232556c7"},
  "gender": "m",
  "loves": ["grape", "lemon"],
  "name": "Kenny",
  "vampires": 44,
  "weight": 690
},
{
  "_id": {"$oid": "68434859febf7818232556c8"},
  "gender": "m",
```

```
"loves": ["apple", "sugar", "redbull"],
"name": "Raleigh",
"vampires": 7,
"weight": 421
},
{
  "_id": {"$oid": "68434859febf7818232556c9"},
  "gender": "f",
  "loves": ["apple", "watermelon"],
  "name": "Leia",
  "vampires": 33,
  "weight": 601
},
{
  "_id": {"$oid": "6843485afebf7818232556ca"},
  "gender": "m",
  "habitat": "{ \"$ref\" : \"$habitats\", \"$id\" : \"$ds\" }",
  "habitat_id": "ds",
  "loves": ["apple", "watermelon", "chocolate"],
  "name": "Pilot",
  "vampires": 59,
  "weight": 650
},
{
  "_id": {"$oid": "6843485afebf7818232556cb"},
  "gender": "f",
  "loves": ["grape", "carrot"],
  "name": "Nimue",
  "weight": 540
},
{
  "_id": {"$oid": "684daf32c37cd51e31e42c5d"},
  "gender": "m",
  "loves": ["grape", "watermelon"],
  "name": "Dunx",
  "vampires": 170,
  "weight": 704
}
```

```

},
{
  "_id": {"$oid": "684e1c3bc37cd51e31e42c66"},
  "gender": "m",
  "loves": ["grape"],
  "name": "Barney",
  "vampires": 5,
  "weight": 340
}
]

```

Практическое задание 4.2.1:

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

создание индекса:

```

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

```

Проверка создания индексов:

```

[
  {
    "key": {
      "_id": 1
    },
    "name": "_id_",
    "v": 2
  },
  {
    "key": {
      "name": 1
    },
    "name": "name_1",
    "v": 2,
  }
]

```

```
"unique": true
}
]
```

Практическое задание 4.3.1:

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

Получим информацию о всех индексах:

```
db.unicorns.getIndexes()
```

Результат выполнения:

```
[
  {
    "key": {
      "_id": 1
    },
    "name": "_id_",
    "v": 2
  },
  {
    "key": {
      "name": 1
    },
    "name": "name_1",
    "v": 2,
    "unique": true
  }
]
```

Удалим все индексы, кроме индекса для идентификатора:

```
db.unicorns.dropIndexes()
```

Проверка индексов:

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



```
"name": "_id_",  
"v": 2  
}  
]
```

Попытаемся удалить системный индекс:

```
db.unicorns.dropIndex("_id_")
```

При удалении системного индекса возникает ошибка.

Практическое задание 4.4.1:

- 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?

Создадим объемную коллекцию:

```
for (let i = 0; i < 100000; i++) {  
  db.numbers.insertOne({ value: i });  
}
```

Выбираем последние четыре документа:

```
db.numbers.find().sort({ value: -1 }).limit(4)
```

Результат выполнения запроса:

```
[  
  {  
    "_id": {"$oid": "684e2b66c37cd51e31e5b324"},  
    "value": 99999  
  }  
]
```

```
},
{
  "_id": {"$oid": "684e2b66c37cd51e31e5b323"},
  "value": 99998
},
{
  "_id": {"$oid": "684e2b66c37cd51e31e5b322"},
  "value": 99997
},
{
  "_id": {"$oid": "684e2b66c37cd51e31e5b321"},
  "value": 99996
}
]
```

Время выполнения: 167 мс

Создание индекса для ключа value:

```
db.numbers.createIndex ({ value: 1 })
```

Просмотр всех индексов коллекции:

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

Результат выполнения:

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

```
"v": 2  
}  
]
```

Повторим второй запрос и сравним время выполнения:

Время выполнения с индексом 19 мс.

Из сравнения времени выполнения можно увидеть, что время поиска с индексом уменьшает время выполнения в разы.

Выводы.

При работе над лабораторной работой 6 я узнал, что такое MongoDB, в чем ее отличие от реляционных БД. Создал базу данных, познакомился с синтаксисом, выполнил множество различных запросов.