

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

Отчет о лабораторной работе 5

«Работа с БД в СУБД MongoDB»

по дисциплине «Проектирование и реализация баз данных»

Автор: Королева Екатерина Максимовна

Факультет: ИКТ

Группа: К32422

Преподаватель: Говорова М. М.

Дата: 28.06.2023

ИТМО

Санкт-Петербург 2023

Цель: овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

Оборудование: компьютерный класс.

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

Практическое задание:

Практическое задание 8.1.1:

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

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 грае и leton, 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:*

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 Аупа 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 Аурога: 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 едиорогов.*

4) *Содержание коллекции единорогов unicorns:*

Практическое задание 8.3.2:

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

Содержание коллекции единорогов unicorns:

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 numbres.*
- 6) *Выполните запрос 2.*
- 7) *Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?*
- 8) *Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?*

Выполнение:

Практическое задание 8.1.1:

Создайте базу данных *learn*.

```
Atlas atlas-g3ik2e-shard-0 [primary] learn>
```

Заполните коллекцию единорогов *unicorns*:

```
db.unicorns.find({ gender: "m" }).sort({ name: 1 })
{ "_id" : ObjectId("64767898e2d5f758ace3db93"), name : "Dunx", loves : [ "grape", "watermelon" ], weight : 704, gender : "m", vampires : 63 },
{ "_id" : ObjectId("647676bd39ef05001120b95b"), name : "Horny", loves : [ "carrot", "papaya" ], weight : 600, gender : "m", vampires : 43 },
{ "_id" : ObjectId("647676bd39ef05001120b961"), name : "Kenny", loves : [ "grape", "lemon" ], weight : 690, gender : "m", vampires : 182 },
{ "_id" : ObjectId("647676bd39ef05001120b964"), name : "Pilot", loves : [ "apple", "watermelon" ], weight : 650, gender : "m", vampires : 99 },
{ "_id" : ObjectId("647676bd39ef05001120b962"), name : "Raleigh", loves : [ "apple", "sugar" ], weight : 421, gender : "m", vampires : 80 },
{ "_id" : ObjectId("647676bd39ef05001120b95e"), name : "Rooooooodles", loves : [ "apple" ], weight : 575, gender : "m", vampires : 40 },
{ "_id" : ObjectId("647676bd39ef05001120b95d"), name : "Unicrom", loves : [ "energon", "redbull" ], weight : 984, gender : "m", vampires : 39 },
{ "_id" : ObjectId("647676bd39ef05001120b95c"), name : "Aurora", loves : [ "carrot", "grape" ], weight : 450, gender : "f", vampires : 2 },
{ "_id" : ObjectId("647676bd39ef05001120b960"), name : "Ayna", loves : [ "strawberry", "lemon" ], weight : 733, gender : "f", vampires : 33 },
{ "_id" : ObjectId("647676bd39ef05001120b963"), name : "Leia", loves : [ "apple", "watermelon" ], weight : 601, gender : "f", vampires : 540 },
```

Используя второй способ, вставьте в коллекцию единорогов документ:

```
> document = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
< {
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
> db.unicorns.insertOne(document)
< {
  acknowledged: true,
  insertedId: ObjectId("649c020bd38b3b375224f9a8")
}
```

Проверьте содержимое коллекции с помощью метода `find`.

```
> db.unicorns.find()
< {
  _id: ObjectId("649bf60fd38b3b375224f98b"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("649bf73ad38b3b375224f98c"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```



```
{
  _id: ObjectId("649bf73bd38b3b375224f98d"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  _id: ObjectId("649bf822d38b3b375224f98e"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("649bf823d38b3b375224f98f"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
```

```
{
  _id: ObjectId("649bffe0d38b3b375224f992"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  _id: ObjectId("649bfff2d38b3b375224f993"),
  name: 'Unicrom',
  loves: [
    'energon',
    'redbull'
  ],
  weight: 984,
  gender: 'm',
  vampires: 182
}
{
  _id: ObjectId("649bfffcd38b3b375224f994"),
  name: 'Rooooooodles',
  loves: [
    'apple'
  ],
  weight: 1000,
  gender: 'f',
  vampires: 1000
}
```

```
apple
],
weight: 575,
gender: 'm',
vampires: 99
}
{
  _id: ObjectId("649c0007d38b3b375224f995"),
  name: 'Solnara',
  loves: [
    'apple',
    'carrot',
    'chocolate'
  ],
  weight: 550,
  gender: 'f',
  vampires: 80
}
{
  _id: ObjectId("649c0011d38b3b375224f996"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40
}
```

И Т Д

Практическое задание 8.1.2:

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

```
> db.unicorns.find({gender: 'm'}, {_id : 0, name : 1}).sort({name : 1}).limit(3).toArray()
< [ { name: 'Dunx' }, { name: 'Horny' }, { name: 'Horny' } ]
```

```
db.unicorns.find({ gender: "m" }).sort({ name: 1 })
[
  {
    _id: ObjectId("649c020bd38b3b375224f9a8"),
    name: 'Dunx',
    loves: [
      'grape',
      'watermelon'
    ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  {
    _id: ObjectId("649bf60fd38b3b375224f98b"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
  {
    _id: ObjectId("649bf73ad38b3b375224f98c"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
]
```

```
{
  _id: ObjectId("649c01c0d38b3b375224f9a3"),
  name: 'Kenny',
  loves: [
    'grape',
    'lemon'
  ],
  weight: 690,
  gender: 'm',
  vampires: 39
}
{
  _id: ObjectId("649c0036d38b3b375224f99a"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 650,
  gender: 'm',
  vampires: 54
}
{
  _id: ObjectId("649c01d7d38b3b375224f9a6"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon'
  ],

```

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("649bf73bd38b3b375224f98d"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
```

Практическое задание 8.1.3:

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

```
> db.unicorns.find({ gender: "m" }, { gender: false, loves: false })
< {
  _id: ObjectId("649bf60fd38b3b375224f98b"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("649bf73ad38b3b375224f98c"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("649bf822d38b3b375224f98e"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("649bfabfd38b3b375224f990"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
```



```
{
  _id: ObjectId("649bffc38b3b375224f991"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("649bfff2d38b3b375224f993"),
  name: 'Unicrom',
  weight: 984,
  vampires: 182
}
{
  _id: ObjectId("649bfffcd38b3b375224f994"),
  name: 'Rooooooodles',
  weight: 575,
  vampires: 99
}
{
  _id: ObjectId("649c001bd38b3b375224f997"),
  name: 'Kenny',
  weight: 690,
  vampires: 39
}
```

и т д

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```
> db.unicorns.find({}, {_id: false, name: true}).sort({ $natural: -1 })
< {
  name: 'Dunx'
}
{
  name: 'Nimue'
}
{
  name: 'Pilot'
}
{
  name: 'Leia'
}
{
  name: 'Raleigh'
}
{
  name: 'Kenny'
}
{
  name: 'Ayna'
}
{
  name: 'Solnara'
}
```

```
    name: 'Aurora'
  }
  {
    name: 'Horny'
  }
  {
    name: 'Nimue'
  }
  {
    name: 'Pilot'
  }
  {
    name: 'Leia'
  }
  {
    name: 'Raleigh'
  }
  {
    name: 'Kenny'
  }
  {
    name: 'Ayna'
  }
  {
    name: 'Solnara'
  }
Type "it" for more
```

Практическое задание 8.1.5:

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

```
db.unicorns.find({}, {_id: false, loves: { $slice: 1 }})
{
  name: 'Horny',
  loves: [
    'carrot'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

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

```
{
  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
}
```

Практическое задание 8.1.6:

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

```
db.unicorns.find({ gender: "f", weight: { $gte: 500, $lt: 700 } }, {_id: false})
{
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
```

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


Практическое задание 8.1.7:

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

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

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

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

Практическое задание 8.1.9:

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

```
> db.unicorns.find({ gender: "m"}, {_id: false, loves: { $slice: 1 }})
< {
  name: 'Horny',
  loves: [
    'carrot'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

```
{
  name: 'Unicrom',
  loves: [
    'energon'
  ],
  weight: 984,
  gender: 'm',
  vampires: 182
}
{
  name: 'Rooooooodles',
  loves: [
    'apple'
  ],
  weight: 575,
  gender: 'm',
  vampires: 99
}
{
  name: 'Kenny',
  loves: [
    'grape'
  ],
  weight: 690,
  gender: 'm',
  vampires: 39
}
```

```
{
  name: 'Pilot',
  loves: [
    'apple'
  ],
  weight: 650,
  gender: 'm',
  vampires: 54
}
{
  name: 'Dunx',
  loves: [
    'grape'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
```

Практическое задание 8.2.1:

Создайте коллекцию *towns*, включающую следующие документы:

```
> db.createCollection("towns")
< { ok: 1 }
```

```
> db.towns.insert({ name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"), famous_for: [""], })
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c0e55d38b3b375224f9a9")
  }
}
```

```
db.towns.insert({ name: "Jim Wehrle"})

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c0e8ed38b3b375224f9aa")
  }
}
```

```
db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"), famous_for: ["status of liberty"]})

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c0ef8d38b3b375224f9ab")
  }
}
```

```
> db.towns.insert({name: "Portland", populatiuon: 528000, last_sensus: ISODate("2009-07-20"), famous_for: ["beer", "food"], ma

< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c0f4ad38b3b375224f9ac")
  }
}
```

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" }, { _id: false, mayor: true, name: true })
{
  name: 'New York',
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
```

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

```
db.towns.find({ "mayor.party": { $exists: false } }, { _id: false, mayor: true, name: true })
{
  name: 'Punxsutawney '
}
{
  name: 'Jim Wehrle'
}
```

Практическое задание 8.2.2:

Сформировать функцию для вывода списка самцов единорогов.

db.unicorns.find({ gender: "m" })

Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

```
> var cursor = db.unicorns.find().sort({ name: 1 }).limit(2); null;
null
< null
```

Вывести результат, используя forEach.

```
cursor.forEach(function(doc) {print(doc.name) })
Aurora
Aurora
```

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
> db.unicorns.count({ gender: "f", weight: { $gte: 500, $lt: 600 }})
< DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
< 6
```

Практическое задание 8.2.4:

Вывести список предпочтений.

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

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

```
> db.unicorns.aggregate({ $group: { _id: "$gender", counta: { $sum: 1 } } })
< {
  _id: 'f',
  counta: 13
}
{
  _id: 'm',
  counta: 17
}
```

Практическое задание 8.2.6:

Выполнить команду:

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


```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("649c1136d38b3b375224f9ad")
  }
}
```

```
> db.unicorns.find({}, { _id: false })
```

```
< {
  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: 'Rooooooodles',
  loves: [
    'apple'
  ],
  weight: 575,
  gender: 'm',
  vampires: 99
```

и т д

Практическое задание 8.2.7:

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

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

Практическое задание 8.2.8:

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

Проверить содержимое коллекции unicorns.

```
<
> db.unicorns.find({ name: "Raleigh"}, {$set^ {loves : ['redbull']}})
✖ ▶ Error: clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<
```

Практическое задание 8.2.9:

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

```
db.unicorns.find({ gender: "m" }, { _id: false })  
  
{  
  name: 'Horny',  
  loves: [  
    'carrot',  
    'papaya'  
  ],  
  weight: 600,  
  gender: 'm',  
  vampires: 63  
}
```

```
> db.unicorns.updateMany({ gender: "m" }, { $inc: { vampires: 5 } })  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 18,  
  modifiedCount: 18,  
  upsertedCount: 0  
}  
> db.unicorns.find({ gender: "m" }, { _id: false })  
< {  
  name: 'Horny',  
  loves: [  
    'carrot',  
    'papaya'  
  ],  
  weight: 600,  
  gender: 'm',  
  vampires: 68  
}
```

Практическое задание 8.2.10:

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

Проверить содержимое коллекции towns.

```
> db.towns.update({name: 'Portland'}, {$unset: {'mayor.party': 1}})
```

```
✖ Error: clone(t={}){const r=t.loc||{};return e(({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<
```

```
{
  _id: ObjectId("649c0f4ad38b3b375224f9ac"),
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  famous_for: [
    'beer',
    'food'
  ],
  mayor: {
    name: 'Sam Adams',
    party: 'D'
  }
}
```

Практическое задание 8.2.11:

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

```
> db.unicorns.update({gender: 'm', name: 'Pilot'}, {$addToSet: {loves: 'chocolate'}})
✖ Error: clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc.line,"column"in r?r.column:...<
Atlas Atlas 20k2e shard-0 [primary]-loose>
```

Практическое задание 8.2.12:

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

```
> db.unicorns.updateOne({ name: "Aurora" }, { $push: { loves: { $each: ["sugar", "lemon"] } } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Проверить содержимое коллекции unicorns.

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

Практическое задание 8.2.13:

Создайте коллекцию *towns*, включающую следующие документы:

```
> db.towns.deleteMany({ "mayor.party": { $exists: false } })
< {
  acknowledged: true,
  deletedCount: 0
}
```

```
> db.towns.find({}, { _id: false })
< {
  name: 'New York',
  populatiuon: 22200000,
  last_sensus: 2009-07-31T00:00:00.000Z,
  famous_for: [
    'status of liberty',
    'food'
  ],
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
{
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  famous_for: [
    'beer',
    'food'
  ],
  mayor: {
    name: 'Sam Adams',
    party: 'D'
  }
}
```

Удалите документы с беспартийными мэрами.

```
> db.towns.drop()  
< true
```

Проверьте содержание коллекции

Очистите коллекцию.

Просмотрите список доступных коллекций.

```
> db.getCollectionNames()  
< [ 'unicorns' ]
```


Практическое задание 8.3.1:

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

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

Проверьте содержание коллекции единорогов.

```
> db.towns.insert({ "_id" : "1", "name" : "zone_1" } )
< {
  acknowledged: true,
  insertedIds: {
    '0': '1'
  }
}
> db.towns.insert({ "_id" : "2", "name" : "zone_2" })
< {
  acknowledged: true,
  insertedIds: {
    '0': '2'
  }
}
}
```

```
> db.unicorns.updateOne({ name: "Dunx" }, { $set: { zone: { $ref: "zones", $id: "1" } } })
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.unicorns.find({}, { _id: false })
< {
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68
}
```

```
{
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68
}
{
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
```

Практическое задание 8.3.2:

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

```
> db.unicorns.ensureIndex({ name: 1 }, { unique: true })
* ► MongoServerError: Index build failed: cf8e1c87-74e2-451d-9f3f-1aba0f0e6bd5: Collection learn.unicorns ( 0eaeffd1-30eb-49c7-
```

Практическое задание 8.3.3:

Получите информацию о всех индексах коллекции `unicorns`.

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

Удалите все индексы, кроме индекса для идентификатора.

```
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]
> db.unicorns.dropIndex("name_1")
< {
  ok: 0,
  errmsg: 'index not found with name [name_1]',
  code: 27,
  codeName: 'IndexNotFound'
}
```

Попытайтесь удалить индекс для идентификатора.

```
> db.unicorns.dropIndex("_id_")
* ► MongoServerError: cannot drop _id index
```

Практическое задание 8.3.4:

Создайте объемную коллекцию *numbers*, задействовав курсор:

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

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

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

(долго думает)

Выберите последних четыре документа.

(все еще думает)...

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

Создайте индекс для ключа *value*.

Получите информацию о всех индексах коллекции *numbers*.

Выполните запрос 2.

Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

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