

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №6.2
«Работа с БД в СУБД MongoDB»

Выполнил: Александр Белов
Факультет: ИКТ
Группа: K3239
Преподаватель: М. М. Говорова

ИТМО

г. Санкт-Петербург, 2023

ВВЕДЕНИЕ

Практическое задание 2.1.1

1. Создайте базу данных *learn*

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

2. Заполните коллекцию единорогов

2.1. Создание коллекции

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

2.2. Вставка данных

```
> 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

```
> db.unicorns.insert({name: 'Dunx', loves: ['grape', 'watermelon'],
weight: 704, gender: 'm', vampires: 165})
```

4. Проверьте содержимое коллекции с помощью метода *find*

```
> db.unicorns.find()
< {
  _id: ObjectId("65743741fbb66b9682e00e42"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
...
{
  _id: ObjectId("6574380ffbb66b9682e00e4d"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ]
}
```

```

    ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }

```

Практическое задание 2.2.1

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

```

> db.unicorns.find({gender: "f"}).sort("name").limit(3)
< {
  _id: ObjectId("65743741fbb66b9682e00e43"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  _id: ObjectId("65743741fbb66b9682e00e47"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40
}
{
  _id: ObjectId("65743741fbb66b9682e00e4a"),
  name: 'Leia',
  loves: [
    'apple',
    'watermelon'
  ],
  weight: 601,
  gender: 'f',
  vampires: 33
}

```

Аналогично для самцов, в запросе «f» заменяется на «m»

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

```

> db.unicorns.find({loves: {$in: ["carrot"]}}).limit(1)
[OR]
> db.unicorns.findOne({loves: {$in: ["carrot"]}})
< {
  _id: ObjectId("65743741fbb66b9682e00e42"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ]
}

```

```

    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }

```

Практическое задание 2.2.2

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

```

> db.unicorns.find({gender: "m"}, {loves: 0})
< {
  _id: ObjectId("65743741fbb66b9682e00e42"),
  name: 'Horny',
  weight: 600,
  gender: 'm',
  vampires: 63
}
...
{
  _id: ObjectId("6574380ffbb66b9682e00e4d"),
  name: 'Dunx',
  weight: 704,
  gender: 'm',
  vampires: 165
}

```

Практическое задание 2.2.3

Вывести список единорогов в обратном порядке добавления.

```

> db.unicorns.find().sort({$natural: -1})
< {
  _id: ObjectId("6574380ffbb66b9682e00e4d"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
...
{
  _id: ObjectId("65743741fbb66b9682e00e42"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}

```

Практическое задание 2.2.3

Вывести список единорогов в обратном порядке добавления

```

> db.unicorns.find().sort({ _id: -1 });
< {
  _id: ObjectId("6574380ffbb66b9682e00e4d"),
  name: 'Dunx',

```

```

    loves: [
      'grape',
      'watermelon'
    ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  ...
  {
    _id: ObjectId("65743741fbb66b9682e00e42"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }

```

Практическое задание 2.1.4

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

```

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

```

```

{
  name: 'Pilot',
  loves: [
    'apple'
  ]
}
{
  name: 'Nimue',
  loves: [
    'grape'
  ]
}
{
  name: 'Dunx',
  loves: [
    'grape'
  ]
}

```

Практическое задание 2.3.1

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

```
{
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  name: 'Solnara',
  loves: [
    'apple',
    'carrot',
    'chocolate'
  ],
  weight: 550,
  gender: 'f',
  vampires: 80
}
```

Практическое задание 2.3.2

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

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

Практическое задание 2.3.3

```
< {  
  _id: ObjectId("65743741fbb66b9682e00e4c"),  
  name: 'Nimue',  
  loves: [  
    'grape',  
    'carrot'  
  ],  
  weight: 540,  
  gender: 'f'  
}
```

Практическое задание 2.3.4

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

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

Практическое задание 3.1.1

```
> db.towns.find({"mayor.party": "I"}, {mayor: 1, name: 1})
< {
  _id: ObjectId("658047a7fbb66b9682e00e52"),
  name: 'New York',
  mayor: {
    name: 'Michael Bloomberg',
    party: 'I'
  }
}
```

```
> db.towns.find({"mayor.party": {$exists: 0}}, {mayor: 1, name: 1})
< {
  _id: ObjectId("658047a7fbb66b9682e00e51"),
  name: 'Punxsutawney ',
  mayor: {
    name: 'Jim Wehrle'
  }
}
learn> |
```

Практическое задание 3.1.2

```
> const maleUnicorns = () => {
  return db.unicorns.find({"gender": "m"})
}
```

```
var cursor = maleUnicorns().sort({name: 1}).limit(2)
cursor.forEach(unicorn => {
  print(unicorn)
})
{
  _id: ObjectId("6574380ffbb66b9682e00e4d"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("65743741fbb66b9682e00e42"),
  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.

```
> db.unicorns.find({"gender": "f", weight: {$gte: 500, $lte: 600}}).count()  
< 2
```

Практическое задание 3.2.2

Вывести список предпочтений.

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

Практическое задание 3.2.3

Посчитать количество особей единорогов обоих полов.

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

Практическое задание 3.3.1

Выполнить команду

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

Проверить содержимое коллекции unicorns

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

✖ ▶ TypeError: db.unicorns.save is not a function

Практическое задание 3.3.2

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

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

Практическое задание 3.3.3

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

```
> db.unicorns.update({ name: 'Raleigh', gender: 'm' },{$set: {loves: ['redbull']}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Практическое задание 3.3.4

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

```
> db.unicorns.updateMany({gender: "m"}, {$inc: {vampires: 5}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 7,
  modifiedCount: 7,
  upsertedCount: 0
}
```

Практическое задание 3.3.5

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

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

Практическое задание 3.3.6

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

```
> db.unicorns.updateOne({ name: "Pilot", gender: "m" },{$push: {loves: "шоколад"}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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.

```
> db.unicorns.update({ name: "Aurora", gender: "f" },{$push: {loves: { $each:
["sugar", "lemons"] }}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
learn> |
```

Практическое задание 3.4.1

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

```
> db.towns.remove({ "mayor.party": null })
< DeprecationWarning: Collection.remove() is deprecated
< {
  acknowledged: true,
  deletedCount: 2
}
```

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

```
{
  _id: ObjectId("657346c5a649c816b644a528"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 68,
  habitatZone: DBRef("HabitatZone", 'us')
}
{
  _id: ObjectId("657346f6a649c816b644a529"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape',
    'sugar',
    'lemons'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43,
  habitatZone: DBRef("HabitatZone", 'ru')
}
```

Практическое задание 4.2.1

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

Содержание коллекции единорогов unicorns

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

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

Практическое задание 4.4.1

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})}
```

Выберите последних четыре документа.

```
> db.numbers.find().sort({ _id: -1 }).limit(4)
< {
  _id: ObjectId("6580505dfbb66b9682e194f3"),
  value: 99999
}
{
  _id: ObjectId("6580505dfbb66b9682e194f2"),
  value: 99998
}
{
  _id: ObjectId("6580505dfbb66b9682e194f1"),
  value: 99997
}
{
  _id: ObjectId("6580505dfbb66b9682e194f0"),
  value: 99996
}
```

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 7,
  totalKeysExamined: 4,
  totalDocsExamined: 4,
  executionStages: {
    stage: 'limit',
    planNodeId: 3,
    nReturned: 4,
    executionTimeMillisEstimate: 0,
    opens: 1,
    closes: 1,
```

```
executionStats: {
  executionSuccess: true,
  nReturned: 4,
  executionTimeMillis: 5,
  totalKeysExamined: 4,
  totalDocsExamined: 4,
  executionStages: {
```

Выборка данных с индексом быстрее на 2 мс.