

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

**Отчет**  
**ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ**  
**«Лабораторная работа №5»**

Автор: Митурский Богдан Антонович

Факультет: ИКТ

Группа: К32392

Преподаватель: Говорова М. М.

**ИТМО**  
Санкт-Петербург 2023

## Задание 8.1.1

- 1) Создайте базу данных learn.
- 2) Заполните коллекцию единорогов unicorns.

### Результат:

```
{
  _id: ObjectId("6493cb555b6d6d11ae59242b"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{
  _id: ObjectId("6493cb555b6d6d11ae59242c"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
},
{
  _id: ObjectId("6493cba35b6d6d11ae59242d"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
```

## Задание 8.1.2

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

### Решение:

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

### Результат:

```
learn> db.unicorns.find({gender: 'f'}).sort({name: 1})
[
  {
    _id: ObjectId("6493cb545b6d6d11ae592423"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592427"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae59242a"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae59242c"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592426"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  }
]

learn> db.unicorns.find({gender: 'm'}).limit(3).sort({name: 1})
[
  {
    _id: ObjectId("6493cba35b6d6d11ae59242d"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("6493cb545b6d6d11ae592422"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592428"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

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

### Решение:

```
db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1)
db.unicorns.findOne({gender: 'f', loves: 'carrot'})
```

### Результат:

```
learn> db.unicorns.findOne({gender: 'f', loves: 'carrot'})
{
  _id: ObjectId("6493cb545b6d6d11ae592423"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

### Задание 8.1.3

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

#### Решение:

```
db.unicorns.find({gender: 'm'}, {loves: false, gender: false})
```

#### Результат:

```
[learn> db.unicorns.find({gender: 'm'}, {loves: false, gender: false})
[
  {
    _id: ObjectId("6493cb545b6d6d11ae592422"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592424"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592425"),
    name: 'Rooooooodles',
    weight: 575,
    vampires: 99
  },
]
```

### Задание 8.1.4

1) Вывести список единорогов в обратном порядке добавления.

**Решение:**

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

**Результат:**

```
[learn> db.unicorns.find().sort({$natural: -1})
[
  {
    _id: ObjectId("6493cba35b6d6d11ae59242d"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae59242c"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae59242b"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },

```

## Задание 8.1.5

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

### Решение:

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

### Результат:

```
[learn> db.unicorns.find({}, {_id: false, 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
  },
]
```

## Задание 8.1.6

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

### Решение:

```
db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 700}}, {_id: false})
```

### Результат:

```
learn> db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 700}}, {_id: false})
[
  {
    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'
  }
]
```



## Задание 8.1.7

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

### Решение:

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

### Результат:

```
[learn> db.unicorns.find({gender: 'm', weight: {$gt: 500}, loves: {$all: ['grape', 'lemon']}}, {_id: 0})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
learn> █
```



### Задание 8.1.8

1) Найти всех единорогов, не имеющих ключ vampires.

**Решение:**

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

**Результат:**

```
[learn> db.unicorns.find({vampires: {$exists: false}})
[
  {
    _id: ObjectId("6493cb555b6d6d11ae59242c"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
learn> █
```

### Задание 8.1.9

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

**Решение:**

```
db.unicorns.find({gender: 'm'}, { _id: 0, vampires: 0, weight: 0, loves:{$slice: 1}, gender: 0}).sort({name: 1})
```

**Результат:**

```
[
  { 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' ] }
]
```

## Задание 8.2.1

- 1) Создайте коллекцию towns.
- 2) Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I") . Вывести только название города и информацию о мэре.

### Решение:

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

### Результат:

```
towns> db.towns.find({"mayor.party": "I"}, {"mayor": 1, "name": 1})
[
  {
    _id: ObjectId("6493d3c72fa12f36d08da7b7"),
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
```

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

### Решение:

```
db.towns.find({"mayor.party": {$exists:false}}, {"mayor": 1, name: 1})
```

### Результат:

```
[
  {
    _id: ObjectId("6493d3c72fa12f36d08da7b6"),
    name: 'Punxsutawney ',
    mayor: { name: 'Jim Wehrle' }
  }
]
```

## 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.

### Решение:

```
unicorns_male = function() {return this.gender === 'm'}  
var cursor = db.unicorns.find({"$where": unicorns_male}).limit(2).sort({name: 1}); null;  
cursor.forEach(function(obj) {print(obj.name);})
```

### Результат:

```
learn> cursor.forEach(function(obj) {print(obj.name);})  
Dunx  
Horny
```

### Задание 8.2.3

1) Вывести количество самок единорогов весом от полутонны до 600 кг.

**Решение:**

```
db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 600}}).count()
```

**Результат:**

```
learn> db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 600}}).count()  
2
```

## Задание 8.2.4

1) Вывести список предпочтений.

**Решение:**

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

**Результат:**

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

## Задание 8.2.5

1) Посчитать количество особей единорогов обоих полов.

**Решение:**

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

**Результат:**

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



## Задание 8.2.6

1) Выполнить команду:

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

**Решение:**

```
db.unicorns.insertOne({name: 'Barney', loves: ['grape'],  
weight: 340, gender: 'm'})  
(save уже убрали из новых версий)
```

**Результат:**

```
{  
  acknowledged: true,  
  insertedId: ObjectId("6493d643397336f2be65f84f")  
}
```

### Задание 8.2.7

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

#### Решение:

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

#### Результат:

```
learn> db.unicorns.find({name: 'Ayna'})
[
  {
    _id: ObjectId("6493cb555b6d6d11ae592427")
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]
```

### Задание 8.2.8

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

**Решение:**

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

**Результат:**

```
{
  _id: ObjectId("6493cb555b6d6d11ae592429"),
  name: 'Raleigh',
  loves: [ 'redbull' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
```

## Задание 8.2.9

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

### Решение:

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

### Результат:

```
[learn> db.unicorns.find({gender: "m"})
[
  {
    _id: ObjectId("6493cb545b6d6d11ae592422"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 68
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592424"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("6493cb555b6d6d11ae592425"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  [

```

## Задание 8.2.10

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

### Решение:

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

### Результат:

```
[{"_id": "6493d3c72fa12f36d08da7b8",  
  name: "Portland",  
  population: 528000,  
  last_census: ISODate("2009-07-20T00:00:00.000Z"),  
  famous_for: [ "beer", "food" ],  
  mayor: { name: "Sam Adams" }  
}]
```

## Задание 8.2.11

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

### Решение:

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

### Результат:

```
[
  {
    _id: ObjectId("6493cb555b6d6d11ae59242b"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  }
]
```

## Задание 8.2.12

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

### Решение:

```
db.unicorns.updateOne( {name: 'Aurora'}, {$push: {loves: {$each: ['sugar', 'lemons']}}} )
```

### Результат:

```
[
  {
    _id: ObjectId("6493cb545b6d6d11ae592423"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```



### Задание 8.2.13

1) Удалите документы с беспартийными мэрами.

**Решение:**

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

**Результат:**

```
[
  {
    _id: ObjectId("6493d3c72fa12f36d08da7b7"),
    name: 'New York',
    populatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
```

### Задание 8.3.1

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

#### Решение:

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

#### Результат:

```
{  
  _id: ObjectId("6493cb545b6d6d11ae592422"),  
  name: 'Horny',  
  loves: [ 'carrot', 'papaya' ],  
  weight: 600,  
  gender: 'm',  
  vampires: 68,  
  habitat: DBRef("habitats", 'FF')  
},  
{  
  _id: ObjectId("6493cb545b6d6d11ae592423"),  
  name: 'Aurora',  
  loves: [ 'carrot', 'grape', 'sugar', 'lemons' ],  
  weight: 450,  
  gender: 'f',  
  vampires: 43,  
  habitat: DBRef("habitats", 'CC')  
},  
{
```

### Задание 8.3.2

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

**Решение:**

```
db.unicorns.ensureIndex({"name": 1}, {"unique": true})
```

**Результат:**

```
learn> db.unicorns.ensureIndex({"name": 1}, {"unique": true})
[ 'name_1' ]
```

### Задание 8.3.3

1) Получите информацию о всех индексах коллекции unicorns.

**Решение:**

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

**Результат:**

```
{ v: 2, key: { _id: 1 }, name: '_id_' },  
{ v: 2, key: { name: 1 }, name: 'name_1', unique: true }
```

2) Удалите все индексы, кроме индекса для идентификатора.

**Решение:**

```
db.unicorns.dropIndex('name_1')
```

**Результат:**

```
learn> db.unicorns.dropIndex('name_1')  
nIndexesWas: 2, ok: 1 }
```

3) Попробуйте удалить индекс для идентификатора.

```
learn> db.unicorns.dropIndex('_id')  
{  
  ok: 0,  
  errmsg: 'index not found with name [_id]',  
  code: 27,  
  codeName: 'IndexNotFound'  
}
```

### Задание 8.3.4

- 1) Создайте объемную коллекцию numbers, задействовав курсор.
- 2) Время выполнения запроса последних 4х документов без индексации – 82мс.
- 3) Создаем индекс:

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

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

```
learn> db.numbers.getIndexes();  
[  
  { v: 2, key: { _id: 1 }, name: '_id_' },  
  { v: 2, key: { value: 1 }, name: 'value_1' }  
]
```

- 4) Время выполнения того же запроса после индексации – 24мс.

#### Запрос:

```
db.numbers.find({}).sort({value:1}).limit(4).explain("executionStats").executionStats.executionTimeMillis;
```