

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

Отчет
«Лабораторная работа № 5»

Автор: Злотников Карина Александровна

Факультет: ИКТ

Группа: K32392

Санкт-Петербург 2023

Лабораторная работ БД №5

Практическое задание 8.1.2

```
db.unicorns.insertOne({name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm',
  vampires: 63});
db.unicorns.insertOne({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f',
  vampires: 43});
db.unicorns.insertOne({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender:
  'm', vampires: 182});
db.unicorns.insertOne({name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vamp
  ires: 99});
db.unicorns.insertOne({name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550,
  gender:'f', vampires:80});
db.unicorns.insertOne({name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender:
  'f', vampires: 40});
db.unicorns.insertOne({name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', v
  ampaires: 39});
db.unicorns.insertOne({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm',
  vampires: 2});
db.unicorns.insertOne({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender:
  'f', vampires: 33});
db.unicorns.insertOne({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender:
  'm', vampires: 54});
db.unicorns.insertOne({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender:
  'f'});

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

db.unicorns.find()
```

	{_id	gender	loves	name	vampires	weight
1	6476340d21920c2380cbe718	m	["carrot", "papaya"]	Horny	63	600
2	6476340e21920c2380cbe719	f	["carrot", "grape"]	Aurora	43	450
3	6476340e21920c2380cbe71a	m	["energon", "redbull"]	Unicrom	182	984
4	6476340e21920c2380cbe71b	m	["apple"]	Roooooodles	99	575
5	6476340e21920c2380cbe71c	f	["apple", "carrot", "chocolate"]	Solnara	80	550
6	6476340f21920c2380cbe71d	f	["strawberry", "lemon"]	Ayna	40	733
7	6476340f21920c2380cbe71e	m	["grape", "lemon"]	Kenny	39	690
8	6476340f21920c2380cbe71f	m	["apple", "sugar"]	Raleigh	2	421
9	6476341021920c2380cbe720	f	["apple", "watermelon"]	Leia	33	601
10	6476341021920c2380cbe721	m	["apple", "watermelon"]	Pilot	54	650
11	6476341021920c2380cbe722	f	["grape", "carrot"]	Nimue	<unset>	540
12	6476654bd3b310316bdb2e13	m	["grape", "watermelon"]	Dunx	165	704

Результат работы 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.

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

	_id	gender	loves	name	vampires	weight
1	6476340e21920c2380cbe719	f	["carrot", "grape"]	Aurora	43	450
2	6476340f21920c2380cbe71d	f	["strawberry", "lemon"]	Ayna	40	733
3	6476341021920c2380cbe720	f	["apple", "watermelon"]	Leia	33	601

	_id	gender	loves	name	vampires	weight
1	6476654bd3b310316bdb2e13	m	["grape", "watermelon"]	Dunx	165	704
2	6476340d21920c2380cbe718	m	["carrot", "papaya"]	Horny	63	600
3	6476340f21920c2380cbe71e	m	["grape", "lemon"]	Kenny	39	690

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

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

	_id	gender	loves	name	vampires	weight
1	6476340e21920c2380cbe719	f	["carrot", "grape"]	Aurora	43	450

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'}, {loves: false, gender: false})
```

	🔗_id	👤 name	🧛 vampires	🏠 weight
1	6476340d21920c2380cbe718	Horny	63	600
2	6476340e21920c2380cbe71a	Unicrom	182	984
3	6476340e21920c2380cbe71b	Rooooooodles	99	575
4	6476340f21920c2380cbe71e	Kenny	39	690
5	6476340f21920c2380cbe71f	Raleigh	2	421
6	6476341021920c2380cbe721	Pilot	54	650
7	6476654bd3b310316bdb2e13	Dunx	165	704

Практическое задание 8.1.4

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

	🔗_id	👤 gender	🍎 loves	👤 name	🧛 vampires	🏠 weight
1	6476654bd3b310316bdb2e13	m	["grape", "watermelon"]	Dunx	165	704
2	6476341021920c2380cbe722	f	["grape", "carrot"]	Nimue	<unset>	540
3	6476341021920c2380cbe721	m	["apple", "watermelon"]	Pilot	54	650
4	6476341021920c2380cbe720	f	["apple", "watermelon"]	Leia	33	601
5	6476340f21920c2380cbe71f	m	["apple", "sugar"]	Raleigh	2	421
6	6476340f21920c2380cbe71e	m	["grape", "lemon"]	Kenny	39	690
7	6476340f21920c2380cbe71d	f	["strawberry", "lemon"]	Ayna	40	733
8	6476340e21920c2380cbe71c	f	["apple", "carrot", "chocolate"]	Solnara	80	550
9	6476340e21920c2380cbe71b	m	["apple"]	Rooooooodles	99	575
10	6476340e21920c2380cbe71a	m	["energon", "redbull"]	Unicrom	182	984
11	6476340e21920c2380cbe719	f	["carrot", "grape"]	Aurora	43	450
12	6476340d21920c2380cbe718	m	["carrot", "papaya"]	Horny	63	600

Практическое задание 8.1.5

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

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

	gender	loves	name	vampires	weight
1	m	["carrot"]	Horny	63	600
2	f	["carrot"]	Aurora	43	450
3	m	["energon"]	Unicrom	182	984
4	m	["apple"]	Rooooooodles	99	575
5	f	["apple"]	Solnara	80	550
6	f	["strawberry"]	Ayna	40	733
7	m	["grape"]	Kenny	39	690
8	m	["apple"]	Raleigh	2	421
9	f	["apple"]	Leia	33	601
10	m	["apple"]	Pilot	54	650
11	f	["grape"]	Nimue	<unset>	540
12	m	["grape"]	Dunx	165	704

Практическое задание 8.1.6

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

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

	gender	loves	name	vampires	weight
1	f	["apple", "carrot", "chocolate"]	Solnara	80	550
2	f	["apple", "watermelon"]	Leia	33	601
3	f	["grape", "carrot"]	Nimue	<unset>	540

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: {$gt: 500}, loves:{$all: ['grape', 'lemon']}}, {_id: 0})
```

	gender	loves	name	vampires	weight
1	m	["grape", "lemon"]	Kenny	39	690

Практическое задание 8.1.8

Найти всех единорогов, не имеющих ключ vampires.

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

	_id	gender	loves	name	weight
1	6476341021920c2380cbe722	f	["grape", "carrot"]	Nimue	540

Практическое задание 8.1.9

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

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

	gender	loves	name
1	m	["grape"]	Dunx
2	m	["carrot"]	Horny
3	m	["grape"]	Kenny
4	m	["apple"]	Pilot
5	m	["apple"]	Raleigh
6	m	["apple"]	Rooooooodles
7	m	["energon"]	Unicrom

Практическое задание 8.2.1

1. Создайте коллекцию towns, включающую следующие документы:

```
db.towns.insertMany([
  {
```

```

    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.

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

	 _id	 mayor	 name
1	64778b45e7e63d40bca7149d	{"name": "Michael Bloomberg", "party": "I"}	New York

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

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

	 _id	 mayor	 name
1	64778b45e7e63d40bca7149c	{"name": "Jim Wehrle"}	Punxsutawney

Практическое задание 8.2.2

1. Сформировать функцию для вывода списка самцов единорогов.

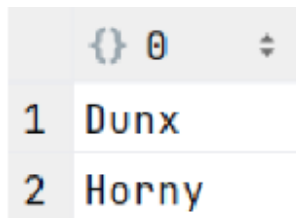
```
get_unicorns_male = function() {return this.gender == 'm'}
```

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

```
var cursor = db.unicorns.find({"$where": unicorns_male}).limit(2).sort({name: 1}); null;
```

3. Вывести результат, используя forEach.

```
cursor.forEach(function(obj) {print(obj.name)})
```



1	Dunx
2	Horny

Практическое задание 8.2.3

```
db.unicorns.deleteMany({})
```

```
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', 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
```



```
db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f',
  vampires: 33});
db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm',
  vampires: 54});
db.unicorns.insert ({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
db.unicorns.insert ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm',
  vampires: 165})
```

Вывести количество самок единорогов весом от полутонны до 600 кг.

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

	result	
1	2	

Практическое задание 8.2.4

Вывести список предпочтений.

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

	result	
1	apple	
2	carrot	
3	chocolate	
4	energon	
5	grape	
6	lemon	
7	papaya	
8	redbull	
9	strawberry	
10	sugar	
11	watermelon	

Практическое задание 8.2.5

Посчитать количество особей единорогов обоих полов.

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

	{ _id ÷	{ count ÷
1	m	7
2	f	5

Практическое задание 8.2.6

1. Выполнить команду:

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

Тут все ругается, говорит использовать `insertOne`

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

Практическое задание 8.2.7

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

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

```
db.unicorns.find({name: 'Ayna'}, {name: 1, vampires: 1, _id: 0})
```

	{ name ÷	{ vampires ÷
1	Ayna	51

Практическое задание 8.2.8:

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

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

```
db.unicorns.find({name: 'Raleigh'}, {name: 1, loves: 1, _id: 0})
```

	🔗 loves	÷	🔗 name	÷
1	["redbull"]		Raleigh	

Практическое задание 8.2.9

Всем самцам единорогов увеличить количество убитых вампиров на 5.

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

	🔗 name	÷	🔗 vampires	÷
1	Horny		68	
2	Unicrom		182	
3	Roooooodles		99	
4	Kenny		39	
5	Raleigh		2	
6	Pilot		54	
7	Dunx		165	

Практическое задание 8.2.10

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

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

	🔗 mayor	÷	🔗 name	÷
1	{"name": "Sam Adams"}		Portland	

Практическое задание 8.2.11

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

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

	🔗 loves	÷	🔗 name	÷
1	["apple", "watermelon", "chocolate"]		Pilot	

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', 'lemons']}}}
)
```

	loves	name
1	["carrot", "grape", "sugar", "lemons"]	Aurora

Практическое задание 8.2.13

Удалите документы с беспартийными мэрами.

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

	famous_for	last_sensus	mayor	name	popujatiuon
1	["status of liberty", "food"]	2009-07-31T00:00:00.000Z	{"name": "Michael Bloomberg", "party": "I"}	New York	22200000
2	["beer", "food"]	2009-07-20T00:00:00.000Z	{"name": "Sam Adams", "party": "D"}	Portland	528000

Очистите коллекцию.

```
db.towns.drop()
```

Просмотрите список доступных коллекций.

```
show collections
```

name
unicorns

Практическое задание 8.3.1

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

```
db.habitats.insertMany([
  {
    _id: "FF",
    name: "Fairy Forest",
    description: "Forest inhabited by unicorns and fairies."
  },
  {
    _id: "CC",
    name: "Crystal Cave",
    description: "A glittering crystal cave where unicorns meditate."
  },
  {
    _id: "rainbow_meadow",
    name: "Rainbow Meadow",
    description: "A meadow with a rainbow sky where unicorns joyfully play."
  },
  {
    _id: "SP",
    name: "Snowy Peak",
    description: "A majestic mountain where unicorns enjoy peace and insight."
  }
]);
```

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

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

	🔗 habitat	🔗 name
1	{ "\$ref" : "habitats", "\$id" : "SP" }	Aurora
2	{ "\$ref" : "habitats", "\$id" : "FF" }	Pilot

Практическое задание 8.3.2

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

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

	🔗 result
1	name_1

Практическое задание 8.3.3

Получите информацию о всех индексах коллекции unicorns .

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

	key	name	v	unique
1	{"_id": new NumberInt("1")}	_id_	2	<unset>
2	{"name": new NumberInt("1")}	name_1	2	• true

Удалите все индексы, кроме индекса для идентификатора.

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

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2

Попытайтесь удалить индекс для идентификатора.

```
'cannot drop _id index'
```

Практическое задание 8.3.4

Создайте объемную коллекцию numbers, задействовав курсор:

```
for (i=0; i<1000000; i++)  
{  
    db.numbers.insertOne({value: i})  
}
```

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

- Время выполнения запроса без индекса - 97 ms.

Создание индекса:

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

	key	name	v
1	{"_id": new NumberInt("1")}	_id_	2
2	{"value": new NumberInt("1")}	value_1	2

Время выполнения запроса после создание индекса - 1 ms

	result
1	1