

федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

**Лабораторная работа №5
«Работа с БД в СУБД MongoDB» по
дисциплине «Проектирование и
реализация баз данных»**

Выполнил:
студент 2 курса ФИКТ
группы K3241
Тимофеев Н.А.

Проверил:
Говорова М.В.

Санкт-Петербург
2022

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 5.0.8.

Выполнение заданий:

1) Создание, заполнение и вывод данных

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

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

ib.unicorns.find({})
```

	{ _id	÷ { gender	÷ { loves	÷ { name	÷ { vampires	÷ { weight
1	633cc70c1aa6dd367410ff0e	m	["carrot", "papaya"]	Horny	63	600
2	633cc70d1aa6dd367410ff0f	f	["carrot", "grape"]	Aurora	43	450
3	633cc70d1aa6dd367410ff10	m	["energon", "redbull"]	Unicrom	182	984
4	633cc70d1aa6dd367410ff11	m	["apple"]	Roooooodles	99	575
5	633cc70d1aa6dd367410ff12	f	["apple", "carrot", "chocolate"]	Solnara	80	550
6	633cc70d1aa6dd367410ff13	f	["strawberry", "lemon"]	Ayna	40	733
7	633cc70e1aa6dd367410ff14	m	["grape", "lemon"]	Kenny	39	690
8	633cc70e1aa6dd367410ff15	m	["apple", "sugar"]	Raleigh	2	421
9	633cc70e1aa6dd367410ff16	f	["apple", "watermelon"]	Leia	33	601
10	633cc70e1aa6dd367410ff17	m	["apple", "watermelon"]	Pilot	54	650
11	633cc70e1aa6dd367410ff18	f	["grape", "carrot"]	Nimue	<unset>	540
12	633cc70f1aa6dd367410ff19	m	["grape", "watermelon"]	Dunx	165	704

2) Запросы

```
// Сформируйте запросы для вывода списков самцов и самок единорогов.
// Ограничьте список самок первыми тремя особями.
// Отсортируйте списки по имени.
db.unicorns.find({gender: 'm'}).sort({name: 1})
db.unicorns.find({gender: 'f'}).limit(3).sort({name: 1})
```

	{ _id	÷ { gender	÷ { loves	÷ { name	÷ { vampires	÷ { weight
1	633cc70f1aa6dd367410ff19	m	["grape", "watermelon"]	Dunx	165	704
2	633cc70c1aa6dd367410ff0e	m	["carrot", "papaya"]	Horny	63	600
3	633cc70e1aa6dd367410ff14	m	["grape", "lemon"]	Kenny	39	690
4	633cc70e1aa6dd367410ff17	m	["apple", "watermelon"]	Pilot	54	650
5	633cc70e1aa6dd367410ff15	m	["apple", "sugar"]	Raleigh	2	421
6	633cc70d1aa6dd367410ff11	m	["apple"]	Roooooodles	99	575
7	633cc70d1aa6dd367410ff10	m	["energon", "redbull"]	Unicrom	182	984

	{ _id	÷ { gender	÷ { loves	÷ { name	÷ { vampires	÷ { weight
1	633cc70d1aa6dd367410ff0f	f	["carrot", "grape"]	Aurora	43	450
2	633cc70d1aa6dd367410ff13	f	["strawberry", "lemon"]	Ayna	40	733
3	633cc70e1aa6dd367410ff16	f	["apple", "watermelon"]	Leia	33	601

```
// 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: 0})
```

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

```
// Модифицируйте запрос для вывода списков самцов единорогов,
// исключив из результата информацию о предпочтениях и поле.
db.unicorns.find({gender: 'm'}, {loves: 0, gender: 0})
```

	_id	name	vampires	weight
1	633cc70c1aa6dd367410ff0e	Horny	63	600
2	633cc70d1aa6dd367410ff10	Unicrom	182	984
3	633cc70d1aa6dd367410ff11	Rooodooles	99	575
4	633cc70e1aa6dd367410ff14	Kenny	39	690
5	633cc70e1aa6dd367410ff15	Raleigh	2	421
6	633cc70e1aa6dd367410ff17	Pilot	54	650
7	633cc70f1aa6dd367410ff19	Dunx	165	704

```
// Вывести список единорогов в обратном порядке добавления.
db.unicorns.find({}).sort({$natural: -1})
```

	_id	gender	loves	name	vampires	weight
1	633cc70f1aa6dd367410ff19	m	["grape", "watermelon"]	Dunx	165	704
2	633cc70e1aa6dd367410ff18	f	["grape", "carrot"]	Nimue	<unset>	540
3	633cc70e1aa6dd367410ff17	m	["apple", "watermelon"]	Pilot	54	650
4	633cc70e1aa6dd367410ff16	f	["apple", "watermelon"]	Leia	33	601
5	633cc70e1aa6dd367410ff15	m	["apple", "sugar"]	Raleigh	2	421
6	633cc70e1aa6dd367410ff14	m	["grape", "lemon"]	Kenny	39	690
7	633cc70d1aa6dd367410ff13	f	["strawberry", "lemon"]	Ayna	40	733
8	633cc70d1aa6dd367410ff12	f	["apple", "carrot", "chocolate"]	Solnara	80	550
9	633cc70d1aa6dd367410ff11	m	["apple"]	Rooodooles	99	575
10	633cc70d1aa6dd367410ff10	m	["energon", "redbull"]	Unicrom	182	984
11	633cc70d1aa6dd367410ff0f	f	["carrot", "grape"]	Aurora	43	450
12	633cc70c1aa6dd367410ff0e	m	["carrot", "papaya"]	Horny	63	600

```
// Вывести список единорогов с названием первого любимого предпочтения,
// исключив идентификатор.
db.unicorns.find({}, {_id: 0, loves: {$slice: -1}})
```

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

```
// Вывести список самок единорогов весом от полутонны до 700 кг,
// исключив вывод идентификатора.
db.unicorns.find({weight: {$gt: 500, $lt: 700}}, {_id: 0})
```

	gender	loves	name	vampires	weight
1	m	["carrot", "papaya"]	Horny	63	600
2	m	["apple"]	Rooodoodles	99	575
3	f	["apple", "carrot", "chocolate"]	Solnara	80	550
4	m	["grape", "lemon"]	Kenny	39	690
5	f	["apple", "watermelon"]	Leia	33	601
6	m	["apple", "watermelon"]	Pilot	54	650
7	f	["grape", "carrot"]	Nimue	<unset>	540

```
// Вывести список самцов единорогов весом от полутонны
// и предпочитающих граде 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

```
// Найти всех единорогов, не имеющих ключ vampires.
db.unicorns.find({vampires: {$exists: 0}}, {_id: 0})
```

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

```
// Вывести список упорядоченный список
// имен самцов единорогов с информацией об их первом предпочтении.
db.unicorns.find({gender: 'm'}, {name: 1, loves: {$slice: 1}, _id: 0}).sort({name: 1})
```

	{ loves }	{ name }
1	["grape"]	Dunx
2	["carrot"]	Horny
3	["grape"]	Kenny
4	["apple"]	Pilot
5	["apple"]	Raleigh
6	["apple"]	Roooooodles
7	["energon"]	Unicrom

3) Запросы ко вложенным объектам

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

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

```
db.towns.find({'mayor.party': {$exists: false}}, {name: 1, mayor: 1, _id: 0})
```

	{ mayor }	{ name }
1	{"name": "Jim Wehrle"}	Punxsutawney

4) ИСПОЛЬЗОВАНИЕ JS И КУРСОРОВ

```
var cursor = db.unicorns.find({gender: 'm'}).sort({name: 1}).limit(2);
cursor.forEach(function (obj) {
  print(obj.name);
})
```

	{ 0 }
1	Dunx
2	Horny

5) АГРЕГИРОВАННЫЕ ЗАПРОСЫ

```
// Вывести количество самок единорогов весом от полутонны до 600 кг.  
db.unicorns.find({gender: 'f', weight: {$gte: 500, $lt: 600}}).count()
```

{ } result	
1	2

```
// 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

```
// Подсчитать количество особей единорогов обоих полов.  
db.unicorns.aggregate({"$group": {_id: "$gender", count: {$sum: 1}}})
```

2 rows		
	{ } _id	{ } count
1	m	7
2	f	5

```

/🔥 Выполнить команду, проверить содержимое
db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})

```

```

TypeError: Proxy({_mongo: {_serviceProvider: JavaObject[com.mongodb.mongosh.service.JavaServiceProvider], _explicitEncryptionOnly: false,
_readPreferenceWasExplicitlyRequested: false, _cachedDatabaseNames: Array(4), _instanceState: {...}, _databases: {...}, _connectionInfo: {...}},
_database: Proxy({_mongo: {...}}, _name: "unicorns.save"), {get: (e,o)=>o in e?e[o]:"string"==typeof
o&&o.startsWith("_")&&(0,helpers_1.isValidCollectionName)(o)?t.getCollection(`${r}.${o}`):void 0)} is not a function
at user_script:34:62
at user_script:29:16
at user_script:1:1
at ShellEvaluator.innerEval (all-standalone.js:6436:17 ...)

```

P.S. Метод deprecated + save не является функцией 6) РЕДАКТИРОВАНИЕ ДАННЫХ

```

// Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51
// вампира, проверить содержимое коллекции unicorns.
db.unicorns.updateOne({name: 'Ayna'}, {$set:{weight: 800, vampires: 51}})
db.unicorns.find({})

```

4	633ccd6e5ec3ea70762b98d8	m	["apple"]	Rooodooles	99	575
5	633ccd6e5ec3ea70762b98d9	f	["apple", "carrot", "chocolate"]	Solnara	80	550
6	633ccd6e5ec3ea70762b98da	f	["strawberry", "lemon"]	Ayna	51	800
7	633ccd6e5ec3ea70762b98db	m	["grape", "lemon"]	Kenny	39	690
8	633ccd6e5ec3ea70762b98dc	m	["apple", "sugar"]	Raleigh	2	421

```

// Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.
db.unicorns.updateOne({name: 'Raleigh'}, {$set:{loves: ['Redbull']}})
db.unicorns.find()

```

7	633ccd6e5ec3ea70762b98db	m	["grape", "lemon"]	Kenny	39	690
8	633ccd6e5ec3ea70762b98dc	m	["Redbull"]	Raleigh	2	421
9	633ccd6e5ec3ea70762b98dd	f	["apple", "watermelon"]	Leia	33	601

	{_id	{_gender	{_loves	{_name	{_vampires
1	633ccd6d5ec3ea70762b98d5	m	["carrot", "papaya"]	Horny	68
2	633ccd6d5ec3ea70762b98d6	f	["carrot", "grape"]	Aurora	43
3	633ccd6d5ec3ea70762b98d7	m	["energon", "redbull"]	Unicrom	187
4	633ccd6e5ec3ea70762b98d8	m	["apple"]	Rooodooles	104
5	633ccd6e5ec3ea70762b98d9	f	["apple", "carrot", "chocolate"]	Solnara	80
6	633ccd6e5ec3ea70762b98da	f	["strawberry", "lemon"]	Ayna	51
7	633ccd6e5ec3ea70762b98db	m	["grape", "lemon"]	Kenny	44
8	633ccd6e5ec3ea70762b98dc	m	["Redbull"]	Raleigh	7
9	633ccd6e5ec3ea70762b98dd	f	["apple", "watermelon"]	Leia	33
10	633ccd6e5ec3ea70762b98de	m	["apple", "watermelon"]	Pilot	59
11	633ccd6f5ec3ea70762b98df	f	["grape", "carrot"]	Nimue	<unset>
12	633ccd6f5ec3ea70762b98e0	m	["grape", "watermelon"]	Dunx	170

```

// Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.
// Проверить содержимое коллекции towns.
db.towns.updateOne({name: 'Portland'}, {$unset: {'mayor.party': 1}})
db.towns.find({})

```


major	name	populat
{ "name": "Jim Wehrle" }	Punxsutawney	
{ "name": "Michael Bloomberg", "party": "I" }	New York	22
{ "name": "Sam Adams" }	Portland	

```
// Изменить информацию о самце единорога Pilot: теперь он любит и шоколад. Проверить
// содержимое коллекции unicorns.
db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
db.unicorns.find()
```

8	633ccd6e5ec3ea70762b98dc	m	["redbull"]	Kateigh	7	421
9	633ccd6e5ec3ea70762b98dd	f	["apple", "watermelon"]	Leia	33	601
10	633ccd6e5ec3ea70762b98de	m	["apple", "watermelon", "chocolate"]	Pilot	59	650
11	633ccd6f5ec3ea70762b98df	f	["grape", "carrot"]	Nimue	<unset>	540

```
// Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и
// лимоны. Проверить содержимое коллекции unicorns.
db.unicorns.updateOne({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
db.unicorns.find()
```

1	633ccd6d5ec3ea70762b98d5	m	["carrot", "papaya"]	Horny	68	600
2	633ccd6d5ec3ea70762b98d6	f	["carrot", "grape", "sugar", "lemon"]	Aurora	43	450
3	633ccd6d5ec3ea70762b98d7	m	["energon", "redbull"]	Unicrom	187	984
4	633ccd6e5ec3ea70762b98d8	m	["apple"]	RoooooodLes	104	575

7) УДАЛЕНИЕ ДАННЫХ ИЗ КОЛЛЕКЦИИ

```
// Создайте коллекцию towns. Удалите документы с беспартийными мэрами. Проверьте
// содержание коллекции. Очистите коллекцию. Просмотрите список доступных коллекций.
db.towns.findOneAndDelete({'major.party': {'$exists': false}})
```

1 row						
_id	famous_for	last_sensus	mayor	name	populatiuon	
1	633cce8f5ec3ea70762b98ee	[""]	2008-01-31T00:00:00.000Z	{"name": "Jim Wehrle"}	Punxsutawney	6200

8) ССЫЛКИ В БД

```
// 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.areas.insertMany([
  { _id: 'a', name: 'location a', desc: 'sunny walley at a' },
  { _id: 'b', name: 'location b', desc: 'sunny walley at b' },
  { _id: 'c', name: 'location c', desc: 'sunny walley at c' },
])

db.unicorns.updateOne({name: 'Horny'}, {$set: {areal: {$ref: 'areas', $id: 'a'}}})
db.unicorns.updateOne({name: 'Pilot'}, {$set: {areal: {$ref: 'areas', $id: 'b'}}})
db.unicorns.updateOne({name: 'Ayna'}, {$set: {areal: {$ref: 'areas', $id: 'c'}}})

db.unicorns.find({'name': 'Ayna'})
```


9) НАСТРОЙКА ИНДЕКСОВ

```
// Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом
// unique.
db.unicorns.ensureIndex({'name': 1}, {'unique': true})
```

Output Result 73 x Result 74 x ratings.unicorns 6 x ratings.unicorns 7 x

1 row

key
1 name_1

10) УПРАВЛЕНИЕ ИНДЕКСАМИ

```
// Получите информацию о всех индексах коллекции unicorns . Удалите все индексы, кроме
// индекса для идентификатора. Попробуйте удалить индекс для идентификатора
db.unicorns.getIndexes()
db.unicorns.dropIndex('name_1')
db.unicorns.dropIndex('_id_')
```

170 db.unicorns.dropIndex(' id')

Command failed with error 27 (IndexNotFound): 'index not found with name [_id]' on server localhost:27017. The full response is {"ok": 0.0, "errmsg": "index not found with name [_id]", "code": 27, "codeName": "IndexNotFound"}

Output Result 73 x Result 74 x ratings.unicorns 6 x ratings.unicorns 7 x unique. x индекса для иденти

2 rows

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

11) ПЛАН ЗАПРОСА

```
// 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?
db.numbers.explain('executionStats').find({}).sort({'value': -1}).limit(50)

db.numbers.ensureIndex({'value': 1})

db.numbers.explain('executionStats').find({}).sort({'value': -1}).limit(50)
```

Ожидание запроса без индекса: 49 мс, с индексом 5 мс.

Выводы: выполнена работа с CRUD-операциями (insert, insertOne, update, remove, find), с вложенными объектами, работа с индексами в NoSQL базе данных MongoDB.