

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

ОТЧЕТ

по лабораторной работе «Работа с БД в СУБД MongoDB»

по дисциплине «Проектирование и реализация баз данных»

Отчет выполнил:

Студент: Никоноров Максим

Группа: К32422

Факультет: ИКТ

Преподаватель: Говорова Марина Михайловна



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

Практическое задание

Практическое задание 8.1.1

Задача: Заполните и Проверьте содержимое коллекции с помощью метода find.

```
> db.unicorns.find()
< {
  _id: ObjectId("64761ea47f91e75dbb71caa7"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("64761ea47f91e75dbb71caa8"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
```

```
> db.unicorns.find().count()
< 12
```

Практическое задание 8.2.1

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

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

```
< {
  _id: ObjectId("64761feb7f91e75dbb71cab4"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("6474a1138dbb846783f4f83d"),
  name: 'Horny',
  loves: [
    'carrot',
    'papaya'
  ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
{
  _id: ObjectId("6474a1138dbb846783f4f843"),
  name: 'Kenny',
  loves: [
    'grape'
```

Были выведены единороги Dunx, Horny и Kenny

6) db.unicorns.find({gender: "f"}).limit(3).sort({name: 1})

```
< {
  _id: ObjectId("6474a1138dbb846783f4f83e"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
{
  _id: ObjectId("6474a1138dbb846783f4f842"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 733,
  gender: 'f',
  vampires: 40
}
{
  _id: ObjectId("6474a1138dbb846783f4f845"),
  name: 'Leia',
  loves: [
```

Были выведены единороги Aurora, Ayna и Leia

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

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

```
< {
  _id: ObjectId("6474a1138dbb846783f4f83e"),
  name: 'Aurora',
  loves: [
    'carrot',
    'grape'
  ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
```

Вывод тот же самый

Практическое задание 8.1.3

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

```
db.unicorns.find({gender: "m"}, {loves: false, gender:
false}).limit(3).sort({name: 1})
```

```
< {
  _id: ObjectId("64761feb7f91e75dbb71cab4"),
  name: 'Dunx',
  weight: 704,
  vampires: 165
}
{
  _id: ObjectId("6474a1138dbb846783f4f83d"),
  name: 'Horny',
  weight: 600,
  vampires: 63
}
{
  _id: ObjectId("6474a1138dbb846783f4f843"),
  name: 'Kenny',
  weight: 690,
  vampires: 39
}
```

Практическое задание 8.1.4

Задача: Вывести список единорогов в обратном порядке добавления.

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

```
< {
  _id: ObjectId("64761feb7f91e75dbb71cab4"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("6474a1138dbb846783f4f847"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
{
  _id: ObjectId("6474a1138dbb846783f4f846"),
  name: 'Pilot',
```

Практическое задание 8.1.5:

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

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

```
< {
  name: 'Horny',
  loves: [
    'carrot'
  ]
}
{
  name: 'Aurora',
  loves: [
    'carrot'
  ]
}
{
  name: 'Unicrom',
  loves: [
    'energon'
  ]
}
{
  name: 'Rooooooodles',
  loves: [
    'apple'
  ]
}
```

Практическое задание 8.1.6

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

```
db.unicorns.find({weight: {$gt : 500, $lt: 700}, gender: "f"}, {"_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,
```


Практическое задание 8.1.7

Задача: 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": false})
```

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

Практическое задание 8.1.8

Задача: Найти всех единорогов, не имеющих ключ vampires.

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

```
{
  _id: ObjectId("6474a1138dbb846783f4f847"),
  name: 'Nimue',
  loves: [
    'grape',
    'carrot'
  ],
  weight: 540,
  gender: 'f'
}
```

Практическое задание 8.1.9

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

```
db.unicorns.find({gender: "m"}, {name: true, "_id": false, loves: {$slice :  
1}}).sort({name: 1})
```

```
{  
  name: 'Dunx',  
  loves: [  
    'grape'  
  ]  
}  
{  
  name: 'Horny',  
  loves: [  
    'carrot'  
  ]  
}  
{  
  name: 'Kenny',  
  loves: [  
    'grape'  
  ]  
}  
{  
  name: 'Pilot',  
  loves: [  
    'apple'  
  ]  
}  
{
```

Практическое задание 8.2.1

Задача 1: Создать коллекцию и 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, name: 1, mayor: 1})
```

```
{  
  name: 'New York',  
  mayor: {  
    name: 'Michael Bloomberg',  
    party: 'I'  
  }  
}
```

Задача 2: 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, name: 1, mayor:  
1})
```

```
{  
  name: 'Punxsutawney ',  
  mayor: {  
    name: 'Jim Wehrle'  
  }  
}
```

Практическое задание 8.2.2

Задача 1: Сформировать функцию для вывода списка самцов единорогов.

```
fn = function() { return this.gender=="m"; }  
db.unicorns.find(fn)
```

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

```
var cursor = db.unicorns.find({gender: "m"}).limit(2);null;
```

Задача 3: Вывести результат, используя forEach.

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

```
> var cursor = db.unicorns.find({gender: "m"}).limit(2);null;  
< null  
> cursor.forEach(function(obj) {  
  print(obj.name);  
  })  
< Horny  
< Unicrom
```

Практическое задание 8.2.3

Задача: Вывести количество самок единорогов весом от полутонны до 600 кг.

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

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

Практическое задание 8.2.4

Задача: Вывести список предпочтений.

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

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

Практическое задание 8.2.5

Задача: Посчитать количество особей единорогов обоих полов.

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

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

Практическое задание 8.2.6

Задача 1: Выполнить команду:

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

```
db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
```

Документация говорит, что ф-ия save устарела и предлагает использовать взамен insertOne()

Задание 2: Проверить содержимое коллекции unicorns.

```
{
  _id: ObjectId("647750bf2ed6b367f644793f"),
  name: 'Barny',
  loves: [
    'grape'
  ],
  weight: 340,
  gender: 'm'
}
```

Функция вставила запись в конец коллекции

Практическое задание 8.2.7

Задание 1: 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}})
```

Документация говорит, что ф-ия save устарела и предлагает использовать взамен updateOne()

Задание 2: Проверить содержимое коллекции unicorns.

```
db.unicorns.find({name : "Ayna"})
{
  _id: ObjectId("6474a1138dbb846783f4f842"),
  name: 'Ayna',
  loves: [
    'strawberry',
    'lemon'
  ],
  weight: 800,
  gender: 'f',
  vampires: 51
}
```

Практическое задание 8.2.8

Задание 1: Для самца единорога Raleigh внести изменения в БД: теперь он любит ред булл.

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

Задание 2: Проверить содержимое коллекции unicorns.

```
db.unicorns.find({name : "Raleigh"})
{
  _id: ObjectId("6474a1138dbb846783f4f844"),
  name: 'Raleigh',
  loves: 'redbull',
  weight: 421,
  gender: 'm',
  vampires: 2
}
```


Практическое задание 8.2.9

Задание 1: Всем самцам единорогов увеличить количество убитых вампиров на 5.

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

```
db.unicorns.updateMany({gender: "m"}, {$inc: {vampires: 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
```

Задание 2: Проверить содержимое коллекции unicorns.

```
{
  _id: ObjectId("6474a1138dbb846783f4f844"),
  name: 'Raleigh',
  loves: 'redbull',
  weight: 421,
  gender: 'm',
  vampires: 7
}
```

Практическое задание 8.2.10

Задание 1: Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.

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

Задание 2: Проверить содержимое коллекции towns.

```
{
  _id: ObjectId("647634fb7f91e75dbb71cab7"),
  name: 'Portland',
  populatiuon: 528000,
  last_sensus: 2009-07-20T00:00:00.000Z,
  famous_for: [
    'beer',
    'food'
  ],
  mayor: {
    name: 'Sam Adams'
  }
}
```

Практическое задание 8.2.11

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

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

Задание 2: Проверить содержимое коллекции unicorns.

```
{
  _id: ObjectId("6474a1138dbb846783f4f846"),
  name: 'Pilot',
  loves: [
    'apple',
    'watermelon',
    'chocolate'
  ],
  weight: 650,
  gender: 'm',
  vampires: 59
}
```

Практическое задание 8.2.12

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

```
db.unicorns.updateOne({name : "Aurora"}, {$addToSet: {loves: {$each: ["sugar", "lemon"]}}})
```

Задание 2: Проверить содержимое коллекции unicorns.

```
{
  _id: ObjectId("6474a1138dbb846783f4f83e"),
  name: 'Aurora',
  weight: 450,
  gender: 'f',
  vampires: 43,
  loves: [
    'carrot',
    'grape',
    'sugar',
    'lemon'
  ]
}
```

Практическое задание 8.2.13:

1. Создать коллекцию towns и удалить документы с беспартийными мэрами.

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

2. Проверьте содержание коллекции.

```
db.towns.find().count()  
2
```

3. Очистите коллекцию.

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

```
db.towns.deleteMany({})  
{  
  acknowledged: true,  
  deletedCount: 2  
}
```

4. Просмотрите список доступных коллекций.

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

Коллекция все еще есть, просто осталась пустой. Можно использовать `db.towns.drop()` для удаления коллекции

Практическое задание 8.3.1

Задание 1: Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.

```
db.land.find()
{
  _id: 's',
  land: 'sea',
  desc: 'Pretty wet'
}
{
  _id: 'e',
  land: 'earth',
  desc: 'Pretty dry'
}
{
  _id: 'h',
  land: 'hell',
  desc: 'Pretty hot'
}
{
  _id: 'm',
  land: 'mountain',
  desc: 'Pretty snowy'
}
```

Задание 2: Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания.

```
db.unicorns.update({_id : ObjectId("647750bf2ed6b367f644793f")} , {$set:
{land : {$ref : "land", $id : "e"}}})
db.unicorns.update({_id : ObjectId("64761feb7f91e75dbb71cab4")} , {$set:
{land : {$ref : "land", $id : "h"}}})
db.unicorns.update({_id : ObjectId("6474a1138dbb846783f4f847")} , {$set:
{land : {$ref : "land", $id : "s"}}})
db.unicorns.update({_id : ObjectId("6474a1138dbb846783f4f846")} , {$set:
{land : {$ref : "land", $id : "m"}}})
```

Задание 3: Проверьте содержание коллекции единорогов.

```
{
  _id: ObjectId("64761feb7f91e75dbb71cab4"),
  name: 'Dunx',
  loves: [
    'grape',
    'watermelon'
  ],
  weight: 704,
  gender: 'm',
  vampires: 170,
  land: DBRef("land", 'h')
}
{
  _id: ObjectId("647750bf2ed6b367f644793f"),
  name: 'Barny',
  loves: [
    'grape'
  ],
  weight: 340,
  gender: 'm',
  vampires: 5,
  land: DBRef("land", 'e')
}
```

Практическое задание 8.3.2

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

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

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

Можно, так как все names разные в коллекции.

Практическое задание 8.3.3

Задание 1: Получите информацию о всех индексах коллекции unicorns .

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

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

Задание 2: Удалите все индексы, кроме индекса для идентификатора.

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

```
db.unicorns.dropIndex("name_1")  
{ nIndexesWas: 2, ok: 1 }
```


Задание 3: Попробуйте удалить индекс для идентификатора.

```
db.unicorns.dropIndex("name_1")
{ nIndexesWas: 2, ok: 1 }
db.unicorns.dropIndex("_id_")
▸ MongoServerError: cannot drop _id index
```

Нельзя удалять индекс для идентификатора.

Практическое задание 8.3.4

Задание 1: Создайте объемную коллекцию numbers, задействовав курсор:
for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}

Задание 2: Выберите последних четыре документа.

```
db.numbers.find().skip(99996)
```

```
{
  _id: ObjectId("64776b89870b5989c43f85c4"),
  value: 99996
}
{
  _id: ObjectId("64776b89870b5989c43f85c5"),
  value: 99997
}
{
  _id: ObjectId("64776b89870b5989c43f85c6"),
  value: 99998
}
{
  _id: ObjectId("64776b89870b5989c43f85c7"),
  value: 99999
}
```

Задание 3: Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра `executionTimeMillis`)

```
db.numbers.find().skip(99996).explain("executionStats")
```

```
executionStats: {  
  executionSuccess: true,  
  nReturned: 4,  
  executionTimeMillis: 84,
```

Задание 4: Создайте индекс для ключа `value`.

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

Задание 5: Получите информацию о всех индексах коллекции `numbers`.

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

Задание 6: Выполните запрос 2 и проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?

```
executionStats: {  
  executionSuccess: true,  
  nReturned: 4,  
  executionTimeMillis: 81,
```

Задание 7: Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

Не могу сказать, что появление индекса сыграло свою роль. Разница в три миллисекунды слишком мала, чтобы делать какие-то основательные выводы. Но по идее, с индексом запрос должен работать быстрее.

Выводы

В процессе работы были получены навыки работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.