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

Отчет

по лабораторной работе №6 «Работа с БД в СУБД MongoDB»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Казарян Т.Г.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

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 (текущая).

Выполнение

Практическое задание 2.1.1:

Запрос на создание базы данных «learn» и вставку нескольких данных в коллекцию «unicorns»:

```
> use learn
< switched to db learn
> db.createCollection("unicorns")
< { ok: 1 }
> db.unicorns.insertMany([
  {name: 'Horny', loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63},
  {name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},
  {name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182},
  {name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99},
  {name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80},
  {name:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},
  {name:'Kenny', loves: ['grape', 'lemon'], weight: 690,  gender: 'm', vampires: 39},
  {name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2},
  {name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33},
  {name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54},
  {name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'}
```

Результат:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("656f8cfe8f7f72992538c699"),
    '1': ObjectId("656f8cfe8f7f72992538c69a"),
    '2': ObjectId("656f8cfe8f7f72992538c69b"),
    '3': ObjectId("656f8cfe8f7f72992538c69c"),
    '4': ObjectId("656f8cfe8f7f72992538c69d"),
    '5': ObjectId("656f8cfe8f7f72992538c69e"),
    '6': ObjectId("656f8cfe8f7f72992538c69f"),
    '7': ObjectId("656f8cfe8f7f72992538c6a0"),
    '8': ObjectId("656f8cfe8f7f72992538c6a1"),
    '9': ObjectId("656f8cfe8f7f72992538c6a2"),
    '10': ObjectId("656f8cfe8f7f72992538c6a3")
  }
}
```

Запрос на вставку одного документа в коллекцию «unicorns»:

```
> document = ({name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165})
  db.unicorns.insertOne(document)
< {
    acknowledged: true,
    insertedId: ObjectId("656f8d938f7f72992538c6a4")
  }
```

Запрос на вывод всех документов в коллекции «unicorns»:

```
> db.unicorns.find({})
< {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
  {
    _id: ObjectId("656f8cfe8f7f72992538c69a"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
```

Практическое задание 2.2.1:

Запрос на вывод 3 самцов единорогов, отсортированных по имени:

Запрос на вывод 3 самок единорогов, отсортированных по имени:

Вывод первого попавшегося единорога, который любит морковку:

```
> db.unicorns.findOne({"loves" : "carrot"})
< {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
```

Вывод первого попавшегося единорога, который любит морковку при помощи limit:

```
> db.unicorns.find({"loves" : "carrot"}).limit(1)
< {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
```

Практическое задание 2.2.2:

Запрос для вывода списков самцов единорогов без информации о предпочтениях и поле:

```
> db.unicorns.find({"gender": "m"}, {"gender": 0, "loves": 0}).limit(3).sort("name")
< {
    _id: ObjectId("656f8d938f7f72992538c6a4"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
  {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  }
  {
    _id: ObjectId("656f8cfe8f7f72992538c69f"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  }
```

Практическое задание 2.2.3:

Запрос на вывод всех единорогов в обратном порядке:

```
> db.unicorns.find({}).sort({ "$natural": -1 })
< {
    _id: ObjectId("656f8d938f7f72992538c6a4"),
    name: 'Dunx',
    loves: [
      'grape',
      'watermelon'
    ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
  {
    _id: ObjectId("656f8cfe8f7f72992538c6a3"),
    name: 'Nimue',
    loves: [
      'grape',
      'carrot'
    ],
    weight: 540,
    gender: 'f'
```

Практическое задание 2.1.4:

Вывод списка единорогов с названием первого любимого предпочтения без идентификатора:

```
> db.unicorns.find({}, {"loves": { "$slice": 1 }, "_id": false}).limit(3)
< {
    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
```

Практическое задание 2.3.1:

Вывод списка самок единорогов весом от полутонов до 700 кг без идентификатора:

```
> db.unicorns.find({"gender": "f", "weight": {"$gte": 500, "$lte": 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
  }
```

Практическое задание 2.3.2:

Вывод всех самцов от полутоны и предпочитающих grape и lemon без идентификатора:

```
> db.unicorns.find({"gender": "m", "weight": {"$gte": 500}, "loves": {"$all": ["grape", "lemon"]}}, {"_id": false})
< {
    name: 'Kenny',
    loves: [
      'grape',
      'lemon'
    ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
```

Практическое задание 2.3.3:

Список всех единорогов без vampires:

```
> db.unicorns.find({"vampires": {"$exists": false}})
< {
    _id: ObjectId("656f8cfe8f7f72992538c6a3"),
    name: 'Nimue',
    loves: [
      'grape',
      'carrot'
    ],
    weight: 540,
    gender: 'f'
  }
```

Практическое задание 3.1.1:

Создание коллекции «towns»:

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

Вставка документов в коллекцию «towns»:

```
> documents = ([
  {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"}}
  ])
  db.towns.insertMany(documents)
< {
    acknowledged: true,
    insertedIds: {
```

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": true, "mayor": true})
< {
    name: 'New York',
    mayor: {
      name: 'Michael Bloomberg',
      party: 'I'
    }
  }
```

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": true, "mayor": true})
< {
    name: 'Punxsutawney ',
    mayor: {
      name: 'Jim Wehrle'
    }
  }
```

Практическое задание 3.1.2:

Вывод первых двух особей мужского пола с помощью использования JavaScript синтаксиса:

```
> const find_all_male_unicorns = () => db.unicorns.find({ "gender": "m" })
  const cursor = find_all_male_unicorns().limit(2).sort("name")
  cursor.forEach((document) => print(document))
< {
    _id: ObjectId("656f8d938f7f72992538c6a4"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
< {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  }
```

Практическое задание 3.2.1:

Вывод количества самок единорогов весов от полутоны до 600 кг:

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

Практическое задание 3.2.2:

Вывод списка предпочтений:

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

Практическое задание 3.2.3:

Вывод количества особей обоих полов:

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

Практическое задание 3.3.1:

Вставка 1 записи:

```
> db.unicorns.insertOne({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
< {
    acknowledged: true,
    insertedId: ObjectId("65709a1e990211857bc2389b")
  }
```

Практическое задание 3.3.2:

Обновление одного единорога:

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

Результат:

```
> db.unicorns.find({"name" : "Ayna"})
< {
    _id: ObjectId("656f8cfe8f7f72992538c69e"),
    name: 'Ayna',
    loves: [
      'strawberry',
      'lemon'
    ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
```

Практическое задание 3.3.3:

Обновление одной записи (добавление в массив элемента):

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

Результат:

```
> db.unicorns.find({"name" : "Raleigh"})
< {
    _id: ObjectId("656f8cfe8f7f72992538c6a0"),
    name: 'Raleigh',
    loves: [
      'apple',
      'sugar',
      'рэдбул'
    ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
```

Практическое задание 3.3.4:

Запрос на инкремент значения vampires для самцов единорогов:

```
> db.unicorns.updateMany({ "gender" : "m" }, { $inc : { "vampires" : 5 } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 8,
    modifiedCount: 8,
    upsertedCount: 0
  }
> db.unicorns.find({ "gender": "m" })
< {
    _id: ObjectId("656f8cfe8f7f72992538c699"),
    name: 'Horny',
    loves: [
      'carrot',
      'papaya'
    ],
    weight: 600,
    gender: 'm',
    vampires: 68
  }
  {
    _id: ObjectId("656f8cfe8f7f72992538c69b"),
    name: 'Unicrom',
```

Практическое задание 3.3.5:

Изменение информации о городе Портленд: мэр теперь беспартийный:

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

Результат:

```
> db.towns.findOne({ name : "Portland" })
< {
    _id: ObjectId("656fa9488f7f72992538c6a7"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: 2009-07-20T00:00:00.000Z,
    famous_for: [
      'beer',
      'food'
    ],
    mayor: {
      name: 'Sam Adams'
    }
  }
```

Практическое задание 3.3.6:

Изменение информации о самце единорога Pilot: теперь он любит и шоколад:

```
> db.unicorns.updateOne({ "name" : "Pilot" }, { $push : {"loves" : "chocolate"} })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.findOne({ "name" : "Pilot" })
< {
    _id: ObjectId("656f8cfe8f7f72992538c6a2"),
    name: 'Pilot',
    loves: [
      'apple',
      'watermelon',
      'chocolate'
    ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
```

Практическое задание 3.3.7:

Изменение информации о самке единорога Aurora: теперь она любит еще и сахар, и лимоны:

```
> db.unicorns.updateOne({ "name" : "Aurora" }, { $addToSet : { "loves" : { $each : ["sugar", "lemon"] } } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.unicorns.findOne({ "name" : "Aurora" })
< {
    _id: ObjectId("656f8cfe8f7f72992538c69a"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape',
      'sugar',
      'lemon'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
```

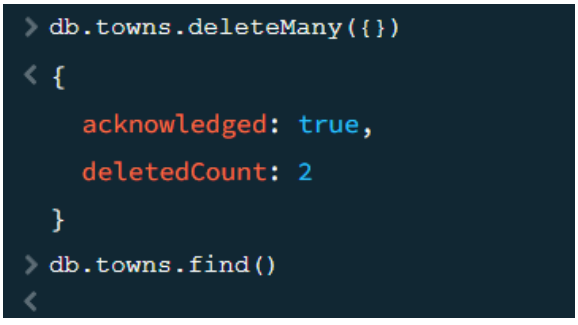
Практическое задание 3.4.1:

Удаление беспартийных мэров:

```
> db.towns.deleteMany({ "mayor.party" : null })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

Результат:

Удаление всех записей из коллекции «towns»:



Практическое задание 4.1.1:

Создание коллекции зон обитания единорогов:

```
db.habitat_area.insertMany([
  { "_id" : "rainbow", "name" : "rainbow_habitat_area", "description" : "Lorem..." },
  { "_id" : "arcobaleno", "name" : "arcobaleno_habitat_area", "description" : "Lorem..." },
  { "_id" : "forest", "name" : "forest_habitat_area", "description" : "Lorem..." }
])
```

Результат:

```
> db.habitat_area.find()
< {
    _id: 'rainbow',
    name: 'rainbow_habitat_area',
    description: 'Lorem...'
  }
  {
    _id: 'arcobaleno',
    name: 'arcobaleno_habitat_area',
    description: 'Lorem...'
  }
  {
    _id: 'forest',
    name: 'forest_habitat_area',
    description: 'Lorem...'
  }
```

Добавление ссылки на зону обитания для единорогов женского пола:

```
> const arcobaleno = db.habitat_area.findOne({ "_id" : "arcobaleno" })
  db.unicorns.updateMany({ "gender" : "f" }, { $set : { "area" : { "$ref" : "habitat_area", "$id" : arcobaleno._id } } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 5,
    modifiedCount: 5,
    upsertedCount: 0
  }
> db.unicorns.find({ "gender" : "f" })
< {
    _id: ObjectId("6570c48d990211857bc238ac"),
    name: 'Aurora',
    loves: [
      'carrot',
      'grape'
    ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    area: DBRef("habitat_area", 'arcobaleno')
  }
  {
    _id: ObjectId("6570c48d990211857bc238af"),
    name: 'Solnara',
    loves: [
      'apple',
```

Практическое задание 4.2.1:

Создание unique индекса для коллекции unicorns для поля name:

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

Практическое задание 4.3.1:

Вывод всех индексов для коллекции unicorns:

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

Удаление всех индексов в unicorns (кроме _id_):

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

Попытка удалить id индекс (неудачная):

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

Практическое задание 4.4.1:

Запрос на получение последних 4 записей (до добавления индекса):

```
> db.numbers.find({}).sort({value : -1}).limit(4)
< {
    _id: ObjectId("6570cb67072a0bbc99678e50"),
    value: 99999
  }
  {
    _id: ObjectId("6570cb67072a0bbc99678e4f"),
    value: 99998
  }
  {
    _id: ObjectId("6570cb67072a0bbc99678e4e"),
    value: 99997
  }
  {
    _id: ObjectId("6570cb67072a0bbc99678e4d"),
    value: 99996
  }
```

Получение времени выполнения запроса (до добавления индекса):

```
> db.numbers.explain("executionStats").find({}).sort({value : -1}).limit(4)
    executionTimeMillis: 95,
```

Запрос на добавления индекса по полю value и получение времени выполнения по следобавления индекса:

```
> db.numbers.createIndex({ "value" : 1 })
< value_1
> db.numbers.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { value: 1 }, name: 'value_1' }
  ]
> db.numbers.explain("executionStats").find({}).sort({value : -1}).limit(4)
    executionTimeMillis: 1,
```

Вывод

В ходе лабораторной работы была освоена работа с СУБД MongoDB. Были проведены практические работы с CRUD-операциями, вложенными объектами, агрегациями, изменениями данных, ссылками и индексами.