



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ПОСТРЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Отчёт по лабораторной работе № 6

«Работа с документной NOSQL БД на примере MongoDB»

Выполнил:
студент группы ИУ5 – 23М

Крутов Т. Ю.

Преподаватель:

Виноградова М. В.

2020г.

Цель работы:

1. Изучить модель представления данных способы работы с документными БД NoSql.
2. Освоить методы создания документной БД и языки запросов к ним.
3. Получить навыки работы с документной БД MongoDB.

Задание 1. Создание БД (базовое)

Создать в среде MongoDB свою БД по теме ДЗ, содержащую не менее двух коллекций.

Добавить в коллекции БД объекты сложной структуры, содержащие вложенные структуры и массивы.

Создание коллекций:

```
db.createCollection("users")
db.createCollection("accounts")
db.createCollection("orders")
```

Добавление объектов в коллекции:

```
db.users.save({_id:1, fio:{firstName:"Иванов", lastName:"Александр",
middleName:"Иванович"}, sex:"М", birthday:"1990.12.21",
address:{city:"Москва", street:"Малая Бронная", building:15,
flat:74}, email:"ivanov@ya.ru", phone:["89264469800","89160879898"]})
```

```
db.users.save({_id:2, fio:{firstName:"Иванов", lastName:"Пётр",
middleName:"Михайлович"}, sex:"М", birthday:"1987.05.13",
address:{city:"Москва", street:"Новослободская",
building:34, flat:16}, email:"ivanpetr@gmail.com", phone:["89257790212"]})
```

```
>db.users.save({_id:2, fio:{firstName:"Иванов", lastName:"Пётр",
middleName:"Михайлович"}, sex:"М", birthday:"1987.05.13",
address:{city:"Москва", street:"Новослободская",
building:34, flat:16}, email:"ivanpetr@gmail.com", phone:["89257790212"]})
```

```
> db.users.save({_id:3, fio:{firstName:"Антонина", middleName:"Тригорьевна",
lastName:"Соловьева"}, sex:"Ж", birthday:"1997.04.16",
address:{city:"Москва",
street:"Серпуховской вал", building:24, flat:43}, email:"sol174@gmail.com",
phone:["89031743636","79168790574"]})
```

```
> db.users.save({_id:4, fio:{firstName:"Василий", middleName:"Сергеевич",
lastName:"Вакуленко"}, sex:"М", birthday:"1998.07.18",
address:{city:"Москва", street:"Ленинградский проспект", building:71,
flat:29}, email:"vvc2000@gmail.com", phone:["89156394032"]})
```

```
> db.users.save({_id:5, fio:{firstName:"Виктор", middleName:"Ахматович",
lastName:"Борисов"}, sex:"М", birthday:"1970.05.10",
address:{city:"Москва", street:"Большая Садовая", building:302, flat:50},
email:"borisovva@outlook.com",
phone:["89038880538","89035557535","89267779507"]})
```

```
> db.users.save({_id:6, fio:{firstName:"Пелагия",
middleName:"Александровна", lastName:"Кабанова"}, sex:"Ж",
birthday:"1990.08.09",
address:{city:"Москва", street:"Профсоюзная", building:"134 к.4", flat:127},
email:"kabanovapelag@gmail.com", phone:["89775472389"]})
```

```
> db.users.save({_id:7, fio:{firstName:"Акакий", middleName:"Акакиевич",
lastName:"Петрушевич"}, sex:"М", birthday:"1954.08.27",
address:{city:"Москва", street:"Удальцова", building:"89", flat:449},
email:"akakiy@mail.ru", phone:["89991892530"]})
```

Добавление аккаунтов:

```
> db.accounts.insert({_id:1,userID:1,login:"ivanovivanov", password:"73-
mpasdf-56"})
> db.accounts.insert({_id:2,userID:2,login:"ivanovpm", password:"psalfa982"})
> db.accounts.insert({_id:4,userID:4,login:"bbc98", password:"@740-Heo4&#1"})
> db.accounts.insert({_id:5,userID:5,login:"borisov-138mcc-16",
password:"$#910HSDor-3472@&6$9832bc"})
> db.accounts.insert({_id:6,userID:6,login:"kabanovap",
password:"rjnbrbghbdtb"})
> db.accounts.insert({_id:7,userID:7,login:"akakiy13", password:"123qwerty"})
> db.accounts.insert({_id:8,userID:7,login:"akakiyCorporation",
password:"3632qw73@73"})
> db.accounts.insert({_id:3,userID:3,login:"solov97", password:"password"})
```

Добавление заказов:

```
> db.orders.insert({_id:1, accountID:3, orderDate:"2020.04.30",
orderType:"Изображение", orderOwner:"Соловьева А.Г.", phone:"89031743636",
email:"sol174@gmail.com"})
> db.orders.insert({_id:2, accountID:3, orderDate:"2020.05.01",
orderType:"Текст", orderOwner:"Соловьева А.Г.", phone:"89031743636",
email:"sol174@gmail.com"})
```

Для вывода используем команду:

```
db.users.find().pretty()
```

Задание 2. Изменение данных (базовое)

- Добавление элемента объекта

```
db.users.update({_id:1}, {$set :{name:"Саня"}})
```

- Изменение элемента объекта

```
db.users.update({_id:1}, {$set :{name:"Санёк"}})
```

- Удаление элемента объекта

```
db.users.update({_id:1}, {$unset :{name:1}})
```

- Замена всего объекта

```
db.users.update({_id:1}, {fio:{firstName:"Сидоров",
lastName:"Афанасий", middleName:"Искандерович"}, sex:"М",
birthday:"1990.12.21",
address:{city:"Тверь", street:"Театральный пр.", building:7, flat:64},
email:"ivanov@ya.ru", phone:["89264469800"]},{upsert : true})
```

- Удаление объекта

```
db.users.remove({_id:1})
```

Задание 3. Запросы к БД (базовое).

- **Вывод всех элементов коллекции**

```
db.users.find().pretty()  
db.accounts.find().pretty()  
db.orders.find().pretty()
```
- **Вывод с фильтрацией (условия с И)**

```
db.users.find({$and : [{sex:"М"}, {"address.city":"Тверь"}] }).pretty()
```
- **Проекция вывода (вывод части полей)**

```
db.users.find({}, {sex:1, address:1, fio:1}).pretty() db.users.find({$and :  
[ {sex:"М"}, {"address.city":"Тверь"}] }).pretty() ()
```
- **Сортировка**

```
db.users.find().sort({"fio.lastName":1}).pretty()
```
- **Условия на поля вложенных структур**

```
db.users.find({"fio.lastName":"Иванов"})  
  
db.users.find({"fio.lastName":{"$ne":"Иванов"}})  
db.users.find({"fio.lastName":{"$exists":true}, "$expr"  
:{"$lt":[{"$strLenCP": "$fio.lastName" }, 8]}}).pretty()
```
- **Поиск по элементам массива объекта**

```
db.orders.find({}, {tags:{$slice:1}}).pretty() -- вывод записи только с  
первым эл-м массива  
db.orders.find({tags:{$all:["коттики"]}}).pretty()  
db.orders.find({tags:{$all:["реферат"]}}).pretty()
```

Задание 4. Расширенные возможности (хорошо)

- **Вывод с фильтрацией (условия И, ИЛИ, операции сравнения)**

```
db.users.find({$or : [{"address.flat": {$eq:43}}, {"fio.firstName" :  
{$eq: "Пётр"}}]}).pretty() -- ИЛИ  
db.users.find({birthday :{$lt : "1989.01.01"}}).pretty() -- отдельно  
сравнение
```
- **Проекция вывода (вывод конкретных полей; вывод без идентификатора)**

```
db.orders.find({}, {_id:null , orderOwner:1, orderDate : 1 ,  
email:1}).pretty()
```
- **С условием на наличие поля**

```
db.users.find({name:{"$exists":true}}).pretty()
```
- **Вывести один элемент коллекции**

```
db.users.find().limit(1).pretty()
```
- **Выполнение операторов count, distinct, limit**

```
db.users.distinct("address.city") -- число разных городов
db.users.count() -- количество элементов коллекции
db.users.find({"address.city":"Москва"}).count() -- число
пользователей проживающих в Москве
db.users.find().limit(3).pretty()
```

- **поиск по вложенным коллекциям объекта**

```
db.users.find({"address.flat": {$eq : 50}}).pretty()
```

- **добавление, удаление, изменение элементов массива**

```
db.users.updateOne({"_id":5}, {$push : {"phone" : "89994441102"}}) ---
добавление
```

```
db.users.update({"_id":5}, {$push : {phone:{$each:
["86666888","55555555"], $position : 3}}}) ----- изменение
```

```
db.users.update({"_id" : 5},{$pop : {phone : 1}} ) -----удаление
```

- **изменение нескольких объектов коллекции в одном запросе**

```
db.users.updateMany({"address.city" : "Москва"}, {$set :
{"address.city": "Санкт-Петербург"}})
```

```
db.users.updateMany({"address.city" : "Санкт-Петербург"}, {$set :
{"address.city": "Москва"}})
```

Список используемой литературы:

1. Методические материалы по MongoDB <https://metanit.com/nosql/mongodb/2.1.php>
2. Официальный сайт <https://www.mongodb.com/>
3. Язык запросов MongoDB
http://softtime.info/view/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2_MongoDB