

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»  
Направление подготовки «45.04.04 Интеллектуальные системы в гуманитарной  
среде»

**О Т Ч Е Т**

**Лабораторная работа №6,  
Базы данных**

**Тема  
задания:** Знакомство с MongoDB

---

**Выполнил:**

**Студент**

**Чанова С. Ю.**

Фамилия И.О.

**K3243**

номер группы

**Проверил:**

**Преподаватель**

**Говоров А. И.**

Фамилия И.О.

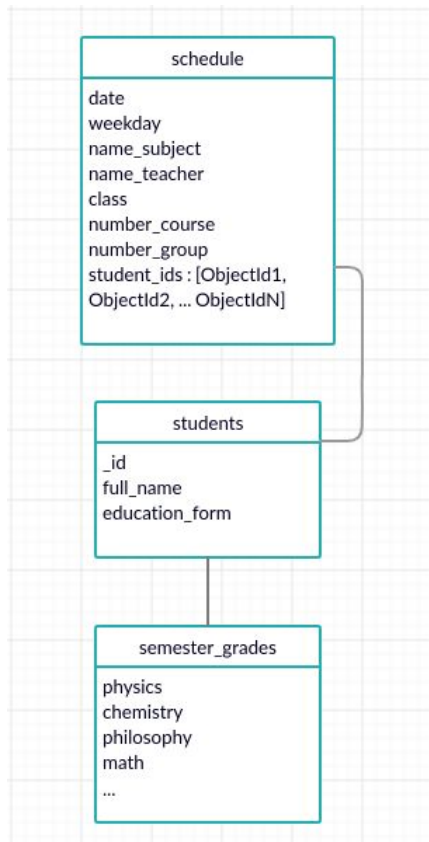
**Санкт-Петербург  
2020**

## Вариант 12

**Цель работы:** овладеть практическими навыками и умениями реализации баз данных в MongoDB

### Ход работы:

## I. Модель БД:



По принципу денормализации из большого количества таблиц в реляционной модели, “развернем” данные внутри одной основной коллекции - расписание. Основная коллекция выбрана, основываясь на списке требуемых запросов задания:

- Какой предмет будет в заданной группе в заданный день недели?
- Кто из преподавателей преподает в заданной группе?
- В каких группах преподает заданный предмет заданный преподаватель?
- Расписание на заданный день недели для указанной группы?
- Сколько студентов обучается на каждом курсе в указанном классе?

## II. Реализация модели в MongoDB:

### 1. Создание БД college:

```
> use college
switched to db college
```

2. Создание коллекции students и наполнение информацией о студентах (всего 12 студентов):

```
db.students.insertMany([
  {
    "full_name": "Lupov Comrad Aleksandrovitch",
    "education_form": "full-time",
    "semester_grades": {
      "biology": "A",
      "chemistry": "A",
      "philosophy": "A",
      "math": "A"
    }
  }
])
```

### 3. Убедимся, что данные вставлены корректно:

```
> db.students.find()
{ "_id" : ObjectId("5ee5cf2b03a69257b7c8f0b7"), "full_name" : "Pupov Ivan Vasilevitch", "education_form" : "full-time", "semester_grades" : { "biology" : "A", "chemistry" : "B", "philosophy" : "A", "math" : "C" } }
{ "_id" : ObjectId("5ee5cf7f03a69257b7c8f0b8"), "full_name" : "Novikova Lora Pavlovna", "education_form" : "part-time", "semester_grades" : { "biology" : "B", "chemistry" : "B", "philosophy" : "C", "math" : "B" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0b9"), "full_name" : "Lupov Comrad Aleksandrovitch", "education_form" : "full-time", "semester_grades" : { "biology" : "A", "chemistry" : "A", "philosophy" : "A", "math" : "A" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0ba"), "full_name" : "Smirnova Irina Igorevna", "education_form" : "part-time", "semester_grades" : { "biology" : "C", "chemistry" : "A", "philosophy" : "B", "math" : "A" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0bb"), "full_name" : "Lomov Evgeny Ildarovitch", "education_form" : "part-time", "semester_grades" : { "physics" : "C", "philosophy" : "B", "math" : "C" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0bc"), "full_name" : "Romov Nikita Alexandrovitch", "education_form" : "part-time", "semester_grades" : { "physics" : "B", "philosophy" : "C", "math" : "B" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0bd"), "full_name" : "Kotova Regina Evgenevna", "education_form" : "full-time", "semester_grades" : { "biology" : "A", "physics" : "C", "philosophy" : "A", "math" : "B" } }
{ "_id" : ObjectId("5ee5d03603a69257b7c8f0be"), "full_name" : "Tokareva Elena Yurevna", "education_form" : "full-time", "semester_grades" : { "physics" : "A", "chemistry" : "A", "philosophy" : "C", "math" : "A" } }
{ "_id" : ObjectId("5ee5f004feeab8a1954f718e"), "full_name" : "Bobov Sanya Mikhailovitch", "education_form" : "full-time", "semester_grades" : { "physics" : "A", "philosophy" : "B", "math" : "A" } }
{ "_id" : ObjectId("5ee5f004feeab8a1954f718f"), "full_name" : "Nikitin Anton Alexandrovitch", "education_form" : "part-time", "semester_grades" : { "physics" : "B", "philosophy" : "B", "math" : "B" } }
{ "_id" : ObjectId("5ee5f004feeab8a1954f7190"), "full_name" : "Borova Daria Evgenevna", "education_form" : "full-time", "semester_grades" : { "biology" : "A", "physics" : "A", "philosophy" : "A", "math" : "B" } }
{ "_id" : ObjectId("5ee5f004feeab8a1954f7191"), "full_name" : "Nirena Elizaveta Yurevna", "education_form" : "full-time", "semester_grades" : { "physics" : "A", "chemistry" : "A", "philosophy" : "A", "math" : "A" } }
```

### 4. Создание коллекции schedule и наполнение информацией о расписании. Данные о преподавателе, предмете, группе представляются, не используя другие коллекции; состав группы представлен с помощью сгенерированного \_id каждого студента в коллекции students:

```
> db.schedule.insert({
...   "date": "2020-06-15",
...   "weekday": "Monday",
...   "name_subject": "Math",
...   "name_teacher": "Lobachevsky Nikolai Ivanovitch",
...   "class": 10,
...   "number_group": 101,
...   "number_course": 1,
...   "students_ids": [
...     ObjectId("5ee5cf2b03a69257b7c8f0b7"),
...     ObjectId("5ee5cf7f03a69257b7c8f0b8"),
...     ObjectId("5ee5d03603a69257b7c8f0b9"),
...     ObjectId("5ee5d03603a69257b7c8f0ba")
...   ]
... })
```

Всего вставлено 3 записи для трех разных групп.

```
> db.schedule.find()
{ "_id" : ObjectId("5ee5d62b03a69257b7c8f0bf"), "date" : "2020-06-15", "weekday" : "Monday", "name_subject" : "Math", "name_teacher" : "Lobachevsky Nikolai Ivanovitch", "class" : 10, "number_group" : 101, "number_course" : 1, "student_ids" : [ ObjectId("5ee5cf2b03a69257b7c8f0b7"), ObjectId("5ee5cf7f03a69257b7c8f0b8"), ObjectId("5ee5d03603a69257b7c8f0b9"), ObjectId("5ee5d03603a69257b7c8f0ba") ] }
{ "_id" : ObjectId("5ee5d6cc03a69257b7c8f0c0"), "date" : "2020-06-16", "weekday" : "Tuesday", "name_subject" : "Math", "name_teacher" : "Lobachevsky Nikolai Ivanovitch", "class" : 10, "number_group" : 102, "number_course" : 1, "student_ids" : [ ObjectId("5ee5d03603a69257b7c8f0bb"), ObjectId("5ee5d03603a69257b7c8f0bc"), ObjectId("5ee5d03603a69257b7c8f0bd"), ObjectId("5ee5d03603a69257b7c8f0be") ] }
{ "_id" : ObjectId("5ee5f15dfeeab8a1954f7192"), "date" : "2020-06-17", "weekday" : "Wednesday", "name_subject" : "Philosophy", "name_teacher" : "Platonov Platon Platonovitch", "class" : 10, "number_group" : 202, "number_course" : 2, "student_ids" : [ ObjectId("5ee5f004feeab8a1954f718e"), ObjectId("5ee5f004feeab8a1954f718f"), ObjectId("5ee5f004feeab8a1954f7190"), ObjectId("5ee5f004feeab8a1954f7191") ] }
```

## III. Реализация запросов

### 1. Какой предмет будет в заданной группе в заданный день недели?

```
> db.schedule.find({"weekday":"Monday", "number_group":101}, {"name_subject":1, _id:0})
{ "name subject" : "Math" }
```



2. Кто из преподавателей преподает в заданной группе?

```
> db.schedule.find({"number_group":101}, {"name_teacher":1, _id:0})
{ "name_teacher" : "Lobachevsky Nikolai Ivanovitch" }
```

3. В каких группах преподает заданный предмет заданный преподаватель?

```
> db.schedule.find({"name_teacher":"Lobachevsky Nikolai Ivanovitch"}, {"number_group":1, _id:0})
{ "number_group" : 101 }
{ "number_group" : 102 }
```

4. Расписание на заданный день недели для указанной группы?

```
> db.schedule.find({"number_group":101, "weekday":"Monday"}, {_id:0, "student_ids":0})
{ "date" : "2020-06-15", "weekday" : "Monday", "name_subject" : "Math", "name_teacher" : "Lobachevsky Nikolai Ivanovitch", "class" : 10, "number_group" : 101, "number_course" : 1 }
```

5. Сколько студентов обучается на каждом курсе в указанном классе?

Для 1 курса:

```
> db.schedule.aggregate([{"$unwind": "$student_ids"}, {"$match": {"class" : 10, "number_course" : 1}}, {"$group": {"_id": "null", "total": {"$sum": 1}}}]
{ "_id" : "null", "total" : 8 }
```

Для 2 курса:

```
> db.schedule.aggregate([{"$unwind": "$student_ids"}, {"$match": {"class" : 10, "number_course" : 2}}, {"$group": {"_id": "null", "total": {"$sum": 1}}}]
{ "_id" : "null", "total" : 4 }
```

6. С помощью **MapReduce** посчитаем количество студентов, обучающихся очно.

```
> map = function() {emit(1, 1);}
function () {emit(1, 1);}
> reduce = function(key, values) {var count = 0; values.forEach(function(v) {count += v;}); return count;}
function (key, values) {var count = 0; values.forEach(function(v) {count += v;}); return count;}
> db.students.mapReduce(map, reduce, {out:"query_1", query:{education_form:"full-time"}});
{
  "result" : "query_1",
  "timeMillis" : 976,
  "counts" : {
    "input" : 6,
    "emit" : 6,
    "reduce" : 1,
    "output" : 1
  },
  "ok" : 1
}
> db.query_1.find()
{ "_id" : 1, "value" : 6 }
```

#### IV. Вывод:

В результате выполнения лабораторной работы были изучены принципы проектирования нереляционных (а именно - документ-ориентированных) баз данных с помощью MongoDB.