

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

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

О Т Ч Е Т

Тема задания: Лабораторная работа №6 – Знакомство с MongoDB

Выполнил:

Студент Джапуа Е.А
(Фамилия И.О.)

К3243
номер группы

Проверил:

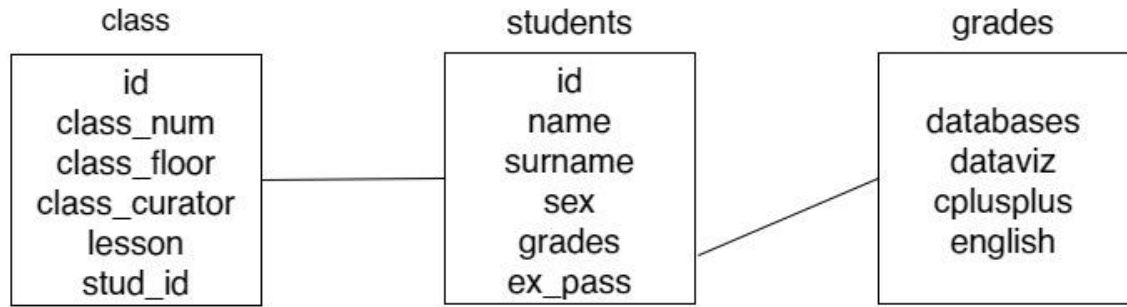
Преподаватель Говоров А.И
(Фамилия И.О.)

**Санкт-Петербург
2020**

Вариант №3

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

Измененная модель базы данных:



Ход выполнения лабораторной работы:

Создаем базу данных labwork:

```
> use labwork
switched to db labwork
>
```

Создаем таблицу students и заполняем её данными:

```
> db.students.insert({"name": "Jorje", "surname": "Masvidal", "sex": "male", "ex_pas": "Yes", "grades": {"databases": 5, "dataviz": 4, "cpl
plus": 4, "english": 3}})
WriteResult({"nInserted": 1 })
> db.students.insert({"name": "Ben", "surname": "Askren", "sex": "male", "ex_pas": "Yes", "grades": {"databases": 2, "dataviz": 2, "cplu
plus": 5, "english": 5}})
WriteResult({"nInserted": 1 })
> db.students.insert({"name": "Ronda", "surname": "Rousey", "sex": "female", "ex_pas": "No", "grades": {"databases": 5, "dataviz": 5, "cplu
plus": 2, "english": 2}})
WriteResult({"nInserted": 1 })
> db.students.insert({"name": "Amanda", "surname": "Nunes", "sex": "female", "ex_pas": "No", "grades": {"databases": 3, "dataviz": 3, "cplu
plus": 3, "english": 5}})
WriteResult({"nInserted": 1 })
> db.students.insert({"name": "Canelo", "surname": "Alvarez", "sex": "male", "ex_pas": "Yes", "grades": {"databases": 5, "dataviz": 5, "cpl
plus": 5, "english": 5}})
WriteResult({"nInserted": 1 })
```

Также создаем таблицу class:

```
> db.class.insert({"class_num": 234, "class_floor": 2, "lesson": "databases", "class_curator": "Deontay Wilder", "student_
id": [ObjectId("5f031da32f4d97aceb18d049"), ObjectId("5f031db62f4d97aceb18d04a")]})
WriteResult({"nInserted": 1 })
> db.class.insert({"class_num": 116, "class_floor": 1, "lesson": "databases", "class_curator": "Anthony Joshua", "student_
id": ObjectId("5f031dbb2f4d97aceb18d04b")})
WriteResult({"nInserted": 1 })
> db.class.insert({"class_num": 201, "class_floor": 2, "lesson": "math", "class_curator": "Tyson Fury", "student_id": Objec
tId("5f031dc12f4d97aceb18d04c")})
WriteResult({"nInserted": 1 })
> db.class.insert({"class_num": 274, "class_floor": 2, "lesson": "physics", "class_curator": "Tyson Fury", "student_id": O
bjectId("5f031dc62f4d97aceb18d04d")})
WriteResult({"nInserted": 1 })
>
```

Проверяем добавленные в таблицы данные:

students

```
> db.students.find()
{ "_id" : ObjectId("5f031da32f4d97aceb18d049"), "name" : "Jonje", "surname" : "Masvidal", "sex" : "male", "ex_pas" : "Yes", "grades" : { "databases" : 5, "dataviz" : 4, "cplusplus" : 4, "english" : 3 } }
{ "_id" : ObjectId("5f031db62f4d97aceb18d04a"), "name" : "Ben", "surname" : "Askren", "sex" : "male", "ex_pas" : "Yes", "grades" : { "databases" : 2, "dataviz" : 2, "cplusplus" : 5, "english" : 5 } }
{ "_id" : ObjectId("5f031dbb2f4d97aceb18d04b"), "name" : "Ronda", "surname" : "Rousey", "sex" : "female", "ex_pas" : "No", "grades" : { "databases" : 5, "dataviz" : 5, "cplusplus" : 2, "english" : 2 } }
{ "_id" : ObjectId("5f031dc12f4d97aceb18d04c"), "name" : "Amanda", "surname" : "Nunes", "sex" : "female", "ex_pas" : "No", "grades" : { "databases" : 3, "dataviz" : 3, "cplusplus" : 3, "english" : 5 } }
{ "_id" : ObjectId("5f031dc62f4d97aceb18d04d"), "name" : "Canelo", "surname" : "Alvarez", "sex" : "male", "ex_pas" : "Yes", "grades" : { "databases" : 5, "dataviz" : 5, "cplusplus" : 5, "english" : 5 } }
>
```

class

```
> db.class.find()
{ "_id" : ObjectId("5f031f7411f519357ec1186f"), "class_num" : 234, "class_floor" : 2, "lesson" : "databases", "class_curator" : "Deontay Wilder", "student_id" : [ ObjectId("5f031da32f4d97aceb18d049"), ObjectId("5f031db62f4d97aceb18d04a") ] }
{ "_id" : ObjectId("5f031f7b11f519357ec11870"), "class_num" : 116, "class_floor" : 1, "lesson" : "databases", "class_curator" : "Anthony Joshua", "student_id" : ObjectId("5f031dbb2f4d97aceb18d04b") }
{ "_id" : ObjectId("5f031f7f11f519357ec11871"), "class_num" : 201, "class_floor" : 2, "lesson" : "math", "class_curator" : "Tyson Fury", "student_id" : ObjectId("5f031dc12f4d97aceb18d04c") }
{ "_id" : ObjectId("5f031f8311f519357ec11872"), "class_num" : 274, "class_floor" : 2, "lesson" : "physics", "class_curator" : "Tyson Fury", "student_id" : ObjectId("5f031dc62f4d97aceb18d04d") }
>
```

Далее выполняем запросы:

Выводим номера всех классов на втором этаже, в которых преподает выбранный учитель:

```
> db.class.find({"class_floor":2, "class_curator":"Tyson Fury"},{"class_num":1, _id:0})
{ "class_num" : 201 }
{ "class_num" : 274 }
>
```

Выводим имена всех учителей, которые преподают выбранный предмет:

```
> db.class.find({"lesson":"databases"},{"class_curator":1, _id:0})
{ "class_curator" : "Deontay Wilder" }
{ "class_curator" : "Anthony Joshua" }
>
```

Выводим имена и фамилии учеников, у которых высший балл по выбранным предметам:

```
> db.students.find({"grades.dataviz":5, "grades.databases":5},{"name":1,"surname":2,_id:0})
{ "name" : "Ronda", "surname" : "Rousey" }
{ "name" : "Canelo", "surname" : "Alvarez" }
>
```

Считаем количество учеников, которые сдали и не сдали экзамены:

```
> db.students.aggregate([{$unwind: "$ex_pas"}, {$group: {_id: "$ex_pas", count: {$sum: 1}}}] );
{ "_id" : "Yes", "count" : 3 }
{ "_id" : "No", "count" : 2 }
>
```

Считаем количество девочек среди учеников, используя mapReduce:

```
> 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: {sex: "female"}});
{
  "result" : "query_1",
  "timeMillis" : 452,
  "counts" : {
    "input" : 2,
    "emit" : 2,
    "reduce" : 1,
    "output" : 1
  },
  "ok" : 1
}
> db.query_1.find()
{ "_id" : 1, "value" : 2 }
>
```

Вывод: в данной лабораторной работе мы изучили основы работы в MongoDB, а именно – проектирование нереляционных баз данных, их заполнение данными и выполнение запросов.