

**Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7  
«Знакомство с mongodb»**

Специальность 09.02.07 «Информационные системы и программирование»

Дисциплина «Основы проектирования баз данных»

Преподаватель:

Говоров А.И. \_\_\_\_\_

«28» февраля 2020 г.

Оценка \_\_\_\_\_

Выполнил:

студент группы Y2337

Пукки К. А.

Санкт-Петербург  
2019/2020

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

Индивидуальное задание:

*Создать программную систему, предназначенную для учебной части колледжа.*

*Она должна обеспечивать хранение сведений о каждом преподавателе, о дисциплинах, которые он преподает, номере закрепленного за ним кабинета, о расписании занятий. Существуют преподаватели, которые не имеют собственного кабинета.*

*О студентах должны храниться следующие сведения: фамилия и имя, в какой группе учится, какую оценку имеет в текущем семестре по каждой дисциплине.*

*Замдекана должен иметь возможность добавить сведения о новом преподавателе или студенте, внести в базу данных семестровые оценки студентов каждой группы по каждой дисциплине, удалить данные об уволившемся преподавателе и отчисленном из колледжа студенте, внести изменения в данные об преподавателях и студентах, в том числе поменять оценку студента по той или иной дисциплине.*

*В задачу диспетчера учебной части входит составление расписания.*

*Замдекана могут потребоваться следующие сведения:*

- *Какой предмет будет в заданной группе в заданный день недели на заданном уроке?*
- *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?*

*Необходимо предусмотреть возможность получения документа, представляющего собой сводные ведомости успеваемости за семестр по каждой группе. В ведомости необходимо предусмотреть сведения о среднем балле группы за семестр.*

1. Попытаться уменьшить размер модели реляционной базы данных, с которой вы работали.
2. Реализовать вашу модель в mongoDB.
3. Заполнить модель данными, насколько, чтобы все запросы реализуемые вами в пункте 2.d выдавали непустые строки.
4. Реализовать минимум 40% ваших запросов, сделанных в работе по реляционным данным в MongoDB.

## Новая модель данных:

Коллекция Cabinets:

```
{  
    _id кабинета  
}
```

Коллекция disciplines:

```
{  
    _id дисциплины  
    Название  
}
```

Коллекция Groups:

```
{  
    _id группы  
}
```

Коллекция Students

```
{  
    _id студента  
    Имя  
    Фамилия  
    Возраст  
    Пол  
    Группа  
}
```

Коллекция Teacher:

```
{  
    _id учителя  
    Имя  
    Фамилия  
    Возраст  
    Пол  
    Расписание: {  
        Предмет  
        Дата  
    }  
}
```

Коллекция Exam:

```
{  
    _id экзамена  
    Учитель  
    Студент  
    Предмет  
}
```

	Дата	Оценка
}		

# Логи процесса реализации модели в MongoDB

Use college

Cabinets:

```
db.cabinets.insert({_id: 103});
```

```
db.cabinets.insert({_id: 210});
```

```
db.cabinets.insert({_id: 213});
```

```
> db.cabinets.insert({_id: 103});
WriteResult({ "nInserted" : 1 })
> db.cabinets.insert({_id: 210});
WriteResult({ "nInserted" : 1 })
> db.cabinets.insert({_id: 213});
WriteResult({ "nInserted" : 1 })
> db.cabinets.find()
{ "_id" : 103 }
{ "_id" : 210 }
{ "_id" : 213 }
>
```

Disciplines:

```
db.disciplines.insert({_id: 1, name: 'Химия'});
```

```
db.disciplines.insert({_id: 2, name: 'Математика'});
```

```
db.disciplines.insert({_id: 3, name: 'История'});
```

```
> db.disciplines.insert({_id: 1, name: 'Химия'});
WriteResult({ "nInserted" : 1 })
> db.disciplines.insert({_id: 2, name: 'Математика'});
WriteResult({ "nInserted" : 1 })
> db.disciplines.insert({_id: 3, name: 'История'});
WriteResult({ "nInserted" : 1 })
> db.disciplines.find()
{ "_id" : 1, "name" : "Химия" }
{ "_id" : 2, "name" : "Математика" }
{ "_id" : 3, "name" : "История" }
```

Groups:

```
db.groups.insert({_id: 2337});
```

```
db.groups.insert({_id: 2336});
```

```
db.groups.insert({_id: 2335});
```

```
> db.groups.insert({_id: 2337});
WriteResult({ "nInserted" : 1 })
> db.groups.insert({_id: 2336});
WriteResult({ "nInserted" : 1 })
> db.groups.insert({_id: 2335});
WriteResult({ "nInserted" : 1 })
> db.groups.find()
{ "_id" : 2337 }
{ "_id" : 2336 }
{ "_id" : 2335 }
```

Students:

```
db.students.insert({_id: 1, name: 'Костя', surname: 'Пукки', Age: 19, gender: 'M', group: 2337});
```

```
db.students.insert({_id: 2, name: 'Аня', surname: 'Петрова', Age: 18, gender: 'Ж', group: 2335});
```

```
db.students.insert({_id: 3, name: 'Женя', surname: 'Антонов', Age: 20, gender: 'M', group: 2336});
```

```
db.students.insert({_id: 4, name: 'Максим', surname: 'Воротынский', Age: 19, gender: 'M', group: 2337});
```

```

> db.students.insert({_id: 1, name: 'Костя', surname: 'Пукки', Age: 19, gender: 'М', group: 2337});
WriteResult({ "nInserted" : 1 })
> db.students.insert({_id: 2, name: 'Аня', surname: 'Петрова', Age: 18, gender: 'Ж', group: 2335});
WriteResult({ "nInserted" : 1 })
> db.students.insert({_id: 3, name: 'Женя', surname: 'Антонов', Age: 20, gender: 'М', group: 2336});
WriteResult({ "nInserted" : 1 })
> db.students.insert({_id: 4, name: 'Максим', surname: 'Воротынский', Age: 19, gender: 'М', group: 2337});
WriteResult({ "nInserted" : 1 })
> db.students.find()
{ "_id" : 1, "name" : "Костя", "surname" : "Пукки", "Age" : 19, "gender" : "М", "group" : 2337 }
{ "_id" : 2, "name" : "Аня", "surname" : "Петрова", "Age" : 18, "gender" : "Ж", "group" : 2335 }
{ "_id" : 3, "name" : "Женя", "surname" : "Антонов", "Age" : 20, "gender" : "М", "group" : 2336 }
{ "_id" : 4, "name" : "Максим", "surname" : "Воротынский", "Age" : 19, "gender" : "М", "group" : 2337 }

```

Teacher:

```

db.teachers.insert({_id: 1, name: 'Мария', surname: 'Павлова', age: 40, gender: 'Ж', cabinet: 103, schedule: [{discipline: 1, date: new Date(2020, 9, 10)}]});
db.teachers.insert({_id: 2, name: 'Татьяна', surname: 'Проккиева', age: 50, gender: 'Ж', cabinet: 210, schedule: [{discipline: 3, date: new Date(2020, 9, 10)}, {discipline: 3, date: new Date(2020, 9, 11)}]});
db.teachers.insert({_id: 3, name: 'Эдуард', surname: 'Ошурок', age: 39, gender: 'М', schedule: [{discipline: 1, date: new Date(2020, 9, 11)}, {discipline: 2, date: new Date(2020, 9, 11)}]});
db.teachers.insert({_id: 4, name: 'Рината', surname: 'Николаева', age: 24, gender: 'Ж', schedule: [{discipline: 3, date: new Date(2020, 9, 11)}, {discipline: 2, date: new Date(2020, 9, 12)}]});

```

```

> db.teachers.insert({_id: 1, name: 'Мария', surname: 'Павлова', age: 40, gender: 'Ж', cabinet: 103, schedule: [{discipline: 1, date: new Date(2020, 9, 10)}]});
WriteResult({ "nInserted" : 1 })
> db.teachers.insert({_id: 2, name: 'Татьяна', surname: 'Проккиева', age: 50, gender: 'Ж', cabinet: 210, schedule: [{discipline: 3, date: new Date(2020, 9, 10)}, {discipline: 3, date: new Date(2020, 9, 11)}]});
WriteResult({ "nInserted" : 1 })
> db.teachers.insert({_id: 3, name: 'Эдуард', surname: 'Ошурок', age: 39, gender: 'М', schedule: [{discipline: 1, date: new Date(2020, 9, 11)}, {discipline: 2, date: new Date(2020, 9, 11)}]});
WriteResult({ "nInserted" : 1 })
> db.teachers.insert({_id: 4, name: 'Рината', surname: 'Николаева', age: 24, gender: 'Ж', schedule: [{discipline: 3, date: new Date(2020, 9, 11)}, {discipline: 2, date: new Date(2020, 9, 12)}]});
WriteResult({ "nInserted" : 1 })
> db.teachers.find()
{ "_id" : 1, "name" : "Мария", "surname" : "Павлова", "age" : 40, "gender" : "Ж", "cabinet" : 103, "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-09T21:00:00Z") } ] }
{ "_id" : 2, "name" : "Татьяна", "surname" : "Проккиева", "age" : 50, "gender" : "Ж", "cabinet" : 210, "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-09T21:00:00Z") }, { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 3, "name" : "Эдуард", "surname" : "Ошурок", "age" : 39, "gender" : "М", "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 4, "name" : "Рината", "surname" : "Николаева", "age" : 24, "gender" : "Ж", "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-11T21:00:00Z") } ] }
>

```

Exam:

```

db.exams.insert({_id: 1, teacher: 1, student: 1, discipline: 1, date: new Date(2020, 9, 20), score: 3});
db.exams.insert({_id: 2, teacher: 1, student: 2, discipline: 1, date: new Date(2020, 9, 21), score: 4});
db.exams.insert({_id: 3, teacher: 2, student: 1, discipline: 3, date: new Date(2020, 9, 21), score: 5});
db.exams.insert({_id: 4, teacher: 3, student: 2, discipline: 3, date: new Date(2020, 9, 21), score: 3});

```

```

> db.exams.insert({_id: 1, teacher: 1, student: 1, discipline: 1, date: new Date(2020, 9, 20), score: 3});
WriteResult({ "nInserted" : 1 })
> db.exams.insert({_id: 2, teacher: 1, student: 2, discipline: 1, date: new Date(2020, 9, 21), score: 4});
WriteResult({ "nInserted" : 1 })
> db.exams.insert({_id: 3, teacher: 2, student: 1, discipline: 3, date: new Date(2020, 9, 21), score: 5});
WriteResult({ "nInserted" : 1 })
> db.exams.insert({_id: 4, teacher: 3, student: 2, discipline: 3, date: new Date(2020, 9, 21), score: 3});
WriteResult({ "nInserted" : 1 })
> db.exams.find()
{ "_id" : 1, "teacher" : 1, "student" : 1, "discipline" : 1, "date" : ISODate("2020-10-19T21:00:00Z"), "score" : 3 }
{ "_id" : 2, "teacher" : 1, "student" : 2, "discipline" : 1, "date" : ISODate("2020-10-20T21:00:00Z"), "score" : 4 }
{ "_id" : 3, "teacher" : 2, "student" : 1, "discipline" : 3, "date" : ISODate("2020-10-20T21:00:00Z"), "score" : 5 }
{ "_id" : 4, "teacher" : 3, "student" : 2, "discipline" : 3, "date" : ISODate("2020-10-20T21:00:00Z"), "score" : 3 }
>

```

# Запросы

- 1) Результаты по химии с оценкой выше 3.

```
> db.exams.find({discipline: 1, score: {$gt: 3}})
{ "_id" : 2, "teacher" : 1, "student" : 2, "discipline" : 1, "date" : ISODate("2020-10-20T21:00:00Z"), "score" : 4 }
>
```

- 2) Учителя и их кабинеты (с сортировкой по кабинету).

```
> db.teachers.find().sort({cabinet: 1})
{ "_id" : 3, "name" : "Эдуард", "surname" : "Ошурок", "age" : 39, "gender" : "М", "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 4, "name" : "Рината", "surname" : "Николаева", "age" : 24, "gender" : "Ж", "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-11T21:00:00Z") } ] }
{ "_id" : 1, "name" : "Мария", "surname" : "Павлова", "age" : 40, "gender" : "Ж", "cabinet" : 103, "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-09T21:00:00Z") } ] }
{ "_id" : 2, "name" : "Татьяна", "surname" : "Проккиева", "age" : 50, "gender" : "Ж", "cabinet" : 210, "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-09T21:00:00Z") }, { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
>
```

- 3) Максимальный возраст среди учителей.

```
> db.teachers.find({}, {age: 1}).sort({age: 1}).limit(1)
{ "_id" : 4, "age" : 24 }
>
```

- 4) Учителя младше 40 лет.

```
> db.teachers.find({age: {$lt: 40}})
{ "_id" : 3, "name" : "Эдуард", "surname" : "Ошурок", "age" : 39, "gender" : "М", "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 4, "name" : "Рината", "surname" : "Николаева", "age" : 24, "gender" : "Ж", "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-11T21:00:00Z") } ] }
>
```

- 5) Расписание учителей.

```
> db.teachers.find({}, {name: 1, surname: 1, schedule: 1})
{ "_id" : 1, "name" : "Мария", "surname" : "Павлова", "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-09T21:00:00Z") } ] }
{ "_id" : 2, "name" : "Татьяна", "surname" : "Проккиева", "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-09T21:00:00Z") }, { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 3, "name" : "Эдуард", "surname" : "Ошурок", "schedule" : [ { "discipline" : 1, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-10T21:00:00Z") } ] }
{ "_id" : 4, "name" : "Рината", "surname" : "Николаева", "schedule" : [ { "discipline" : 3, "date" : ISODate("2020-10-10T21:00:00Z") }, { "discipline" : 2, "date" : ISODate("2020-10-11T21:00:00Z") } ] }
>
```