

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

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

О Т Ч Е Т

ЛАБОРАТОРНАЯ РАБОТА №6

Знакомство с MongoDB

Выполнил:

Студент Смоленская Т.Д. К3243
(Фамилия И.О.) номер группы

Проверил:

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

Санкт-Петербург
2020

Индивидуальное задание:

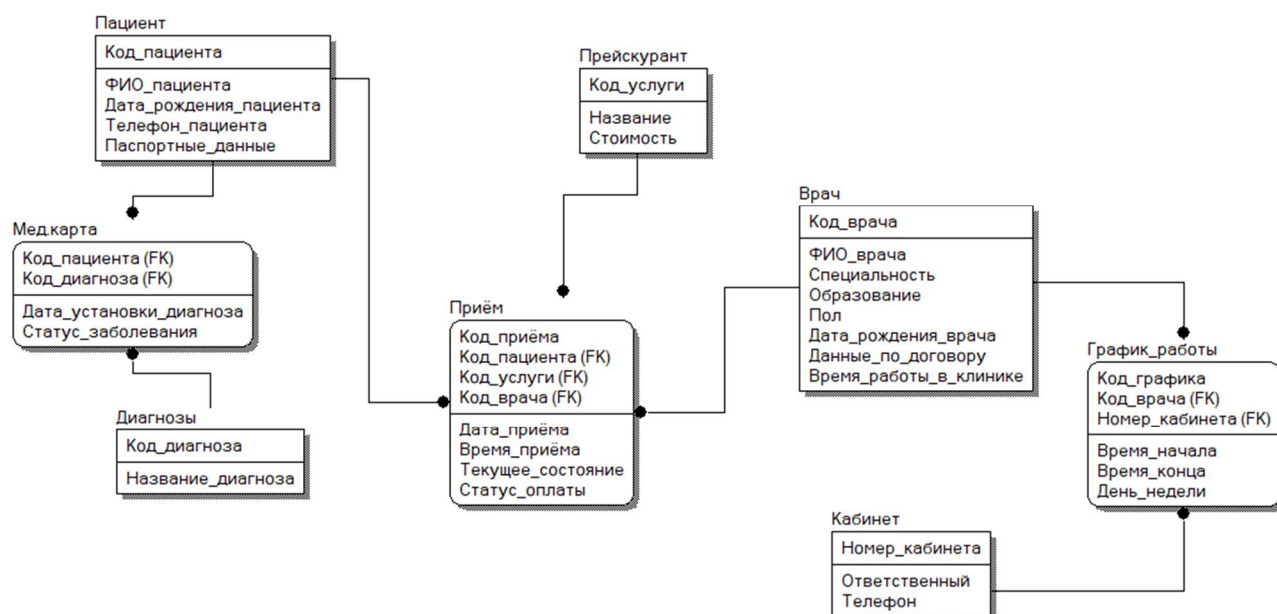
Создать программную систему, предназначенную для администратора лечебной клиники.

Прием пациентов ведут несколько врачей различных специализаций. На каждого пациента клиники заводится медицинская карта, в которой отражается вся информация по личным данным больного и истории его заболеваний (диагнозы). При очередном посещении врача в карте отражается дата и время приема, диагноз, текущее состояние больного, рекомендации по лечению. Так как прием ведется только на коммерческой основе, после очередного посещения пациент должен оплатить медицинские услуги (каждый прием оплачивается отдельно). Расчет стоимости посещения определяется врачом согласно прейскуранту по клинике.

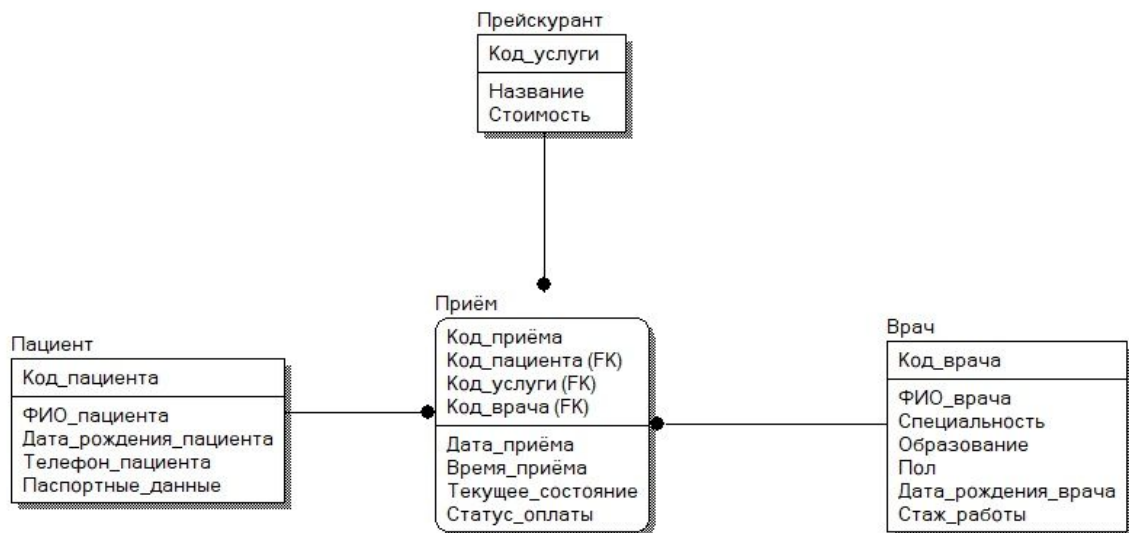
Для ведения внутренней отчетности необходима следующая информация о врач: фамилия, имя, отчество, специальность, образование, пол, дата рождения и дата начала и окончания работы в клинике, данные по трудовому договору. Для каждого врача составляется график работы с указанием рабочих и выходных дней.

Прием пациентов врачи могут вести в разных кабинетах. Каждый кабинет имеет определенный режим работы, ответственного и внутренний телефон.

Модель БД из второй лабораторной:



Уберём из нашей БД ненужные сущности:



Были сокращены сущности «Кабинет», «Мед карта» и «Диагнозы», «График работы».

Реализуем нашу модель в MongoDB с оставшимися сущностями «Пациент», «Врач», «Приём», «Прейскурант»:

Создадим таблицу «Пациент», наполним её значениями:

```

> use hospital
switched to db hospital
> db.patient.insert({"fio_patient": "Ivanov I.I.", "birthday_date_patient": "1993-12-03", "phone_patient": "+7 921 3734509", "passport_patient": "4010 565656"
})
WriteResult({"nInserted" : 1 })
> db.patient.insert({"fio_patient": "Petrov D.A.", "birthday_date_patient": "1976-09-01", "phone_patient": "+7 911 3635649", "passport_patient": "4014 347855"
})
WriteResult({"nInserted" : 1 })
> db.patient.insert({"fio_patient": "Mikhailov S.U.", "birthday_date_patient": "1980-04-05", "phone_patient": "+7 931 6394079", "passport_patient": "4118 5298
51"})
WriteResult({"nInserted" : 1 })
> db.patient.insert({"fio_patient": "Yurieva N.B.", "birthday_date_patient": "1965-04-05", "phone_patient": "+7 911 5350532", "passport_patient": "4510 656082
"})
WriteResult({"nInserted" : 1 })
> db.patient.insert({"fio_patient": "Antonov I.D.", "birthday_date_patient": "1956-11-25", "phone_patient": "+7 921 5019527", "passport_patient": "4117 958592
"})
WriteResult({"nInserted" : 1 })
> db.patient.insert({"fio_patient": "Valichev O.E.", "birthday_date_patient": "1978-05-31", "phone_patient": "+7 911 9545267", "passport_patient": "4016 99376
2"})
WriteResult({"nInserted" : 1 })
> db.patient.find()
{ "_id" : ObjectId("5f0605a574258a37a8ce30f7"), "fio_patient" : "Ivanov I.I.", "birthday_date_patient" : "1993-12-03", "phone_patient" : "+7 921 3734509", "p
assport_patient" : "4010 565656" }
{ "_id" : ObjectId("5f0605df74258a37a8ce30f8"), "fio_patient" : "Petrov D.A.", "birthday_date_patient" : "1976-09-01", "phone_patient" : "+7 911 3635649", "p
assport_patient" : "4014 347855" }
{ "_id" : ObjectId("5f0605f074258a37a8ce30f9"), "fio_patient" : "Mikhailov S.U.", "birthday_date_patient" : "1980-04-05", "phone_patient" : "+7 931 6394079",
"passport_patient" : "4118 529851" }
{ "_id" : ObjectId("5f06060274258a37a8ce30fa"), "fio_patient" : "Yurieva N.B.", "birthday_date_patient" : "1965-04-05", "phone_patient" : "+7 911 5350532", "
passport_patient" : "4510 656082" }
{ "_id" : ObjectId("5f06061174258a37a8ce30fb"), "fio_patient" : "Antonov I.D.", "birthday_date_patient" : "1956-11-25", "phone_patient" : "+7 921 5019527", "
passport_patient" : "4117 958592" }
{ "_id" : ObjectId("5f06061a74258a37a8ce30fc"), "fio_patient" : "Valichev O.E.", "birthday_date_patient" : "1978-05-31", "phone_patient" : "+7 911 9545267",
"passport_patient" : "4016 993762" }
    
```

Создадим таблицу «Врач», наполним её значениями:

```
> db.doctor.insert({"fio_doctor": "Bykov V.D.", "speciality": "Отоларинголог", "education": "Высшее", "gender": "Мужской", "birthday_date_doctor": "1961-03-17", "work_time_in_clinic": 19})
WriteResult({ "nInserted" : 1 })
> db.doctor.insert({"fio_doctor": "Frank A.L.", "speciality": "Аллерголог", "education": "Высшее", "gender": "Мужской", "birthday_date_doctor": "1974-11-09", "work_time_in_clinic": 26})
WriteResult({ "nInserted" : 1 })
> db.doctor.insert({"fio_doctor": "Vanchuk M.T.", "speciality": "Главный врач", "education": "Высшее", "gender": "Женский", "birthday_date_doctor": "1965-01-29", "work_time_in_clinic": 31})
WriteResult({ "nInserted" : 1 })
> db.doctor.insert({"fio_doctor": "Tygymbek A.V.", "speciality": "Главный врач", "education": "Высшее", "gender": "Мужской", "birthday_date_doctor": "1976-06-01", "work_time_in_clinic": 12})
WriteResult({ "nInserted" : 1 })
> db.doctor.insert({"fio_doctor": "Bibekova V.M.", "speciality": "Пульмонолог", "education": "Высшее", "gender": "Женский", "birthday_date_doctor": "1973-04-11", "work_time_in_clinic": 24})
WriteResult({ "nInserted" : 1 })
> db.doctor.find()
{ "_id" : ObjectId("5f0720e674258a37a8ce310d"), "fio_doctor" : "Bykov V.D.", "speciality" : "Отоларинголог", "education" : "Высшее", "gender" : "Мужской", "birthday_date_doctor" : "1961-03-17", "work_time_in_clinic" : 19 }
{ "_id" : ObjectId("5f0720e674258a37a8ce310e"), "fio_doctor" : "Frank A.L.", "speciality" : "Аллерголог", "education" : "Высшее", "gender" : "Мужской", "birthday_date_doctor" : "1974-11-09", "work_time_in_clinic" : 26 }
{ "_id" : ObjectId("5f0720e674258a37a8ce310f"), "fio_doctor" : "Vanchuk M.T.", "speciality" : "Главный врач", "education" : "Высшее", "gender" : "Женский", "birthday_date_doctor" : "1965-01-29", "work_time_in_clinic" : 31 }
{ "_id" : ObjectId("5f0720e674258a37a8ce3110"), "fio_doctor" : "Tygymbek A.V.", "speciality" : "Главный врач", "education" : "Высшее", "gender" : "Мужской", "birthday_date_doctor" : "1976-06-01", "work_time_in_clinic" : 12 }
{ "_id" : ObjectId("5f0720e674258a37a8ce3111"), "fio_doctor" : "Bibekova V.M.", "speciality" : "Пульмонолог", "education" : "Высшее", "gender" : "Женский", "birthday_date_doctor" : "1973-04-11", "work_time_in_clinic" : 24 }
```

Создадим таблицу «Прейскурант», наполним её значениями:

```
> db.pricelist.insert({"name_service": "Первичная консультация", "price_service": 1000})
WriteResult({ "nInserted" : 1 })
> db.pricelist.insert({"name_service": "Базовое обследование", "price_service": 1500})
WriteResult({ "nInserted" : 1 })
> db.pricelist.insert({"name_service": "Анализ крови", "price_service": 1300})
WriteResult({ "nInserted" : 1 })
> db.pricelist.insert({"name_service": "МРТ", "price_service": 1200})
WriteResult({ "nInserted" : 1 })
> db.pricelist.insert({"name_service": "Полное обследование", "price_service": 3200})
WriteResult({ "nInserted" : 1 })
> db.pricelist.insert({"name_service": "Консультация", "price_service": 800})
WriteResult({ "nInserted" : 1 })
> db.pricelist.find()
{ "_id" : ObjectId("5f061c3274258a37a8ce3102"), "name_service" : "Первичная консультация", "price_service" : 1000 }
{ "_id" : ObjectId("5f061c3e74258a37a8ce3103"), "name_service" : "Базовое обследование", "price_service" : 1500 }
{ "_id" : ObjectId("5f061c4974258a37a8ce3104"), "name_service" : "Анализ крови", "price_service" : 1300 }
{ "_id" : ObjectId("5f061c5474258a37a8ce3105"), "name_service" : "МРТ", "price_service" : 1200 }
{ "_id" : ObjectId("5f061c6274258a37a8ce3106"), "name_service" : "Полное обследование", "price_service" : 3200 }
{ "_id" : ObjectId("5f061c6c74258a37a8ce3107"), "name_service" : "Консультация", "price_service" : 800 }
```

Создадим таблицу «Приём», наполним её значениями:

```
> db.meet.insert({id_patient: ObjectId("5f0605a574258a37a8ce30f7"), id_service: ObjectId("5f061c3274258a37a8ce3102"), id_doctor: ObjectId("5f0720e674258a37a8ce310d"), "date_meet": "2019-12-16", "time_meet": "13:10:00", "current_state": "Составлен план лечения", "payment_state": true})
WriteResult({ "nInserted" : 1 })
> db.meet.insert({id_patient: ObjectId("5f0605df74258a37a8ce30f8"), id_service: ObjectId("5f061c4974258a37a8ce3104"), id_doctor: ObjectId("5f0720e674258a37a8ce310f"), "date_meet": "2020-05-16", "time_meet": "09:15:00", "current_state": "Анализ на наличие вируса", "payment_state": true})
WriteResult({ "nInserted" : 1 })
> db.meet.insert({id_patient: ObjectId("5f0605f074258a37a8ce30f9"), id_service: ObjectId("5f061c4974258a37a8ce3104"), id_doctor: ObjectId("5f0720e674258a37a8ce310f"), "date_meet": "2020-04-28", "time_meet": "16:55:00", "current_state": "Анализ на наличие вируса", "payment_state": true})
WriteResult({ "nInserted" : 1 })
> db.meet.insert({id_patient: ObjectId("5f06060274258a37a8ce30fa"), id_service: ObjectId("5f061c6c74258a37a8ce3107"), id_doctor: ObjectId("5f0720e674258a37a8ce3110"), "date_meet": "2020-05-16", "time_meet": "14:20:00", "current_state": "Вылечено", "payment_state": true})
WriteResult({ "nInserted" : 1 })
> db.meet.insert({id_patient: ObjectId("5f06061174258a37a8ce30fb"), id_service: ObjectId("5f061c3e74258a37a8ce3103"), id_doctor: ObjectId("5f0720e674258a37a8ce310d"), "date_meet": "2020-03-18", "time_meet": "17:30:00", "current_state": "Состояние стабильное", "payment_state": true})
WriteResult({ "nInserted" : 1 })
> db.meet.find()
{ "_id" : ObjectId("5f072b45cf3e034f2274a3"), "id_patient" : ObjectId("5f0605a574258a37a8ce30f7"), "id_service" : ObjectId("5f061c3274258a37a8ce3102"), "id_doctor" : ObjectId("5f0720e674258a37a8ce310d"), "date_meet" : "2019-12-16", "time_meet" : "13:10:00", "current_state" : "Составлен план лечения", "payment_state" : true }
{ "_id" : ObjectId("5f072b45cf3e034f2274a4"), "id_patient" : ObjectId("5f0605df74258a37a8ce30f8"), "id_service" : ObjectId("5f061c4974258a37a8ce3104"), "id_doctor" : ObjectId("5f0720e674258a37a8ce310f"), "date_meet" : "2020-05-16", "time_meet" : "09:15:00", "current_state" : "Анализ на наличие вируса", "payment_state" : true }
{ "_id" : ObjectId("5f072b46cf3e034f2274a5"), "id_patient" : ObjectId("5f0605f074258a37a8ce30f9"), "id_service" : ObjectId("5f061c4974258a37a8ce3104"), "id_doctor" : ObjectId("5f0720e674258a37a8ce310f"), "date_meet" : "2020-04-28", "time_meet" : "16:55:00", "current_state" : "Анализ на наличие вируса", "payment_state" : true }
{ "_id" : ObjectId("5f072b46cf3e034f2274a6"), "id_patient" : ObjectId("5f06060274258a37a8ce30fa"), "id_service" : ObjectId("5f061c6c74258a37a8ce3107"), "id_doctor" : ObjectId("5f0720e674258a37a8ce3110"), "date_meet" : "2020-05-16", "time_meet" : "14:20:00", "current_state" : "Вылечено", "payment_state" : true }
{ "_id" : ObjectId("5f072b48cf3e034f2274a7"), "id_patient" : ObjectId("5f06061174258a37a8ce30fb"), "id_service" : ObjectId("5f061c3e74258a37a8ce3103"), "id_doctor" : ObjectId("5f0720e674258a37a8ce310d"), "date_meet" : "2020-03-18", "time_meet" : "17:30:00", "current_state" : "Состояние стабильное", "payment_state" : true }
```

Запросы:

- 1) Показать код пациентов, дату и время приёма с определённым кодом доктора:

```
db.meet.find({id_doctor: ObjectId("5f06182c74258a37a8ce30ff")}, {date_meet: 1, _id: 0, id_patient: 1, time_meet: 1}).sort({date_meet: 1})
```

```
> db.meet.find({id_doctor: ObjectId("5f06182c74258a37a8ce30ff")}, {date_meet: 1, _id: 0, id_patient: 1, time_meet: 1}).sort({date_meet: 1})
{ "id_patient" : ObjectId("5f0605f074258a37a8ce30f9"), "date_meet" : "2020-04-28", "time_meet" : "16:55:00" }
{ "id_patient" : ObjectId("5f0605df74258a37a8ce30f8"), "date_meet" : "2020-05-16.", "time_meet" : "09:15:00" }
```

- 2) Показать названия услуг, их стоимость, где она равна больше 1000 с сортировкой по стоимости услуг:

```
db.pricelist.find({price_service: {$gt: 1000}}, {_id: 0, name_service: 1, price_service: 1}).sort({price_service: 1})
```

```
> db.pricelist.find({price_service: {$gt: 1000}}, {_id: 0, name_service: 1, price_service: 1}).sort({price_service: 1})
{ "name_service" : "МРТ", "price_service" : 1200 }
{ "name_service" : "Анализ крови", "price_service" : 1300 }
{ "name_service" : "Базовое обследование", "price_service" : 1500 }
{ "name_service" : "Полное обследование", "price_service" : 3200 }
```

- 3) Показать ФИО докторов с высшим образованием, их пол, специальность, стаж с сортировкой по стажу работы по убыванию:

```
db.doctor.find({education: "Высшее"}, {_id: 0, fio_doctor: 1, speciality: 1, gender: 1, work_time_in_clinic: 1}).sort({work_time_in_clinic: -1})
```

```
> db.doctor.find({education: "Высшее"}, {_id: 0, fio_doctor: 1, speciality: 1, gender: 1, work_time_in_clinic: 1}).sort({work_time_in_clinic: -1})
{ "fio_doctor" : "Vanchuk M.T.", "speciality" : "Главный врач", "gender" : "Женский", "work_time_in_clinic" : 31 }
{ "fio_doctor" : "Frank A.L.", "speciality" : "Аллерголог", "gender" : "Мужской", "work_time_in_clinic" : 26 }
{ "fio_doctor" : "Bibekova V.M.", "speciality" : "Пульмонолог", "gender" : "Женский", "work_time_in_clinic" : 24 }
{ "fio_doctor" : "Bykov V.D.", "speciality" : "Отоларинголог", "gender" : "Мужской", "work_time_in_clinic" : 19 }
{ "fio_doctor" : "Tygymbek A.V.", "speciality" : "Главный врач", "gender" : "Мужской", "work_time_in_clinic" : 12 }
```

- 4) Показать коды пациентов, которых принимал доктор с определённым кодом доктора и у которых статус оплаты равен true:

```
db.meet.find({id_doctor: ObjectId("5f0720e674258a37a8ce310d"), payment_state: true}, {id_patient: 1, _id: 0})
```

```
> db.meet.find({id_doctor: ObjectId("5f0720e674258a37a8ce310d"), payment_state: true}, {id_patient: 1, _id: 0})
{ "id_patient" : ObjectId("5f0605a574258a37a8ce30f7") }
{ "id_patient" : ObjectId("5f06061174258a37a8ce30fb") }
```

- 5) Показать длину ФИО пациентов в символах:

```
db.patient.aggregate( [ { $project: { "fio_patient": 1, "length": { $strLenCP: "$fio_patient" } } } ] )
```

```
> db.patient.aggregate( [ { $project: { "fio_patient": 1, "length": { $strLenCP: "$fio_patient" } } } ] )
{ "_id" : ObjectId("5f0605a574258a37a8ce30f7"), "fio_patient" : "Ivanov I.I.", "length" : 11 }
{ "_id" : ObjectId("5f0605df74258a37a8ce30f8"), "fio_patient" : "Petrov D.A.", "length" : 11 }
{ "_id" : ObjectId("5f0605f074258a37a8ce30f9"), "fio_patient" : "Mikhailov S.U.", "length" : 14 }
{ "_id" : ObjectId("5f06060274258a37a8ce30fa"), "fio_patient" : "Yurieva N.B.", "length" : 12 }
{ "_id" : ObjectId("5f06061174258a37a8ce30fb"), "fio_patient" : "Antonov I.D.", "length" : 12 }
{ "_id" : ObjectId("5f06061a74258a37a8ce30fc"), "fio_patient" : "Valichev O.E.", "length" : 13 }
```

6) Агрегирование результатов с помощью MapReduce:

```
> function map(){ emit(this.fio_patient, 1); }  
> function reduce(key, values) { var sum = 0; for(var i in values) { sum += values[i]; } return sum; }  
sum; }
```

```
> db.patient.mapReduce(map, reduce, {out: "fio_patient"})
```

```
> function map(){ emit(this.fio_patient, 1); }  
> function reduce(key, values) { var sum = 0; for(var i in values) { sum += values[i]; } return sum; }  
> db.patient.mapReduce(map, reduce, {out: "fio_patient"})  
{  
  "result" : "fio_patient",  
  "timeMillis" : 70,  
  "counts" : {  
    "input" : 6,  
    "emit" : 6,  
    "reduce" : 0,  
    "output" : 6  
  },  
  "ok" : 1  
}
```

7) Показать ФИО пациентов и дату их рождения, если дата рождения меньше 1970-01-01:

```
db.patient.find({birthday_date_patient: {$lt: '1970-01-01'}}, {_id: 0, fio_patient: 1,  
birthday_date_patient: 1})
```

```
> db.patient.find({birthday_date_patient: {$lt: '1970-01-01'}}, {_id: 0, fio_patient: 1, birthday_date_patient: 1})  
{ "fio_patient" : "Yurieva N.B.", "birthday_date_patient" : "1965-04-05" }  
{ "fio_patient" : "Antonov I.D.", "birthday_date_patient" : "1956-11-25" }
```

8) Показать ФИО врачей мужского пола со специальностью «Главный врач»:

```
db.doctor.find({speciality: "Главный врач", gender: "Мужской"}, {fio_doctor: 1, _id: 0})
```

```
> db.doctor.find({speciality: "Главный врач", gender: "Мужской"}, {fio_doctor: 1, _id: 0})  
{ "fio_doctor" : "Tygymbek A.V." }
```

Выводы: В ходе данной лабораторной работы было совершено знакомство с MongoDB, а также осуществлено сокращение исходной базы данных и реализация её с помощью этой СУБД. Для демонстрации основных функций системы, было выполнено 8 запросов.