

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»
Факультет среднего профессионального образования

ОТЧЁТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7
по теме: Знакомство с MongoDB
по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Проверил:
_____ Говоров А. И.
Дата: «_____» _____ 2020г.
Оценка _____

Выполнил:
студент группы Y2336
_____ Наумов М.А.

Санкт-Петербург 2020

ЦЕЛЬ РАБОТЫ

Цель практической работы №7: овладеть практическими навыками и умениями реализации баз данных в MongoDB.

ЗАДАНИЕ

Для выполнения работы Вам потребуется:

- a. Попытаться уменьшить размер модели реляционной базы данных, с которой Вы работали.
- b. Реализовать Вашу модель в MongoDB.
- c. Заполнить модель данными, настолько, чтобы все запросы, реализуемые Вами в пункте 2.d, выдавали непустые строки.
- d. Реализовать минимум 40% Ваших запросов, сделанных в работе по реляционным базам данных в MongoDB.

ВЫПОЛНЕНИЕ

1) Схема физической модели БД, спроектированной с использованием СА Erwin Data Modeler представлена на рисунке №1.

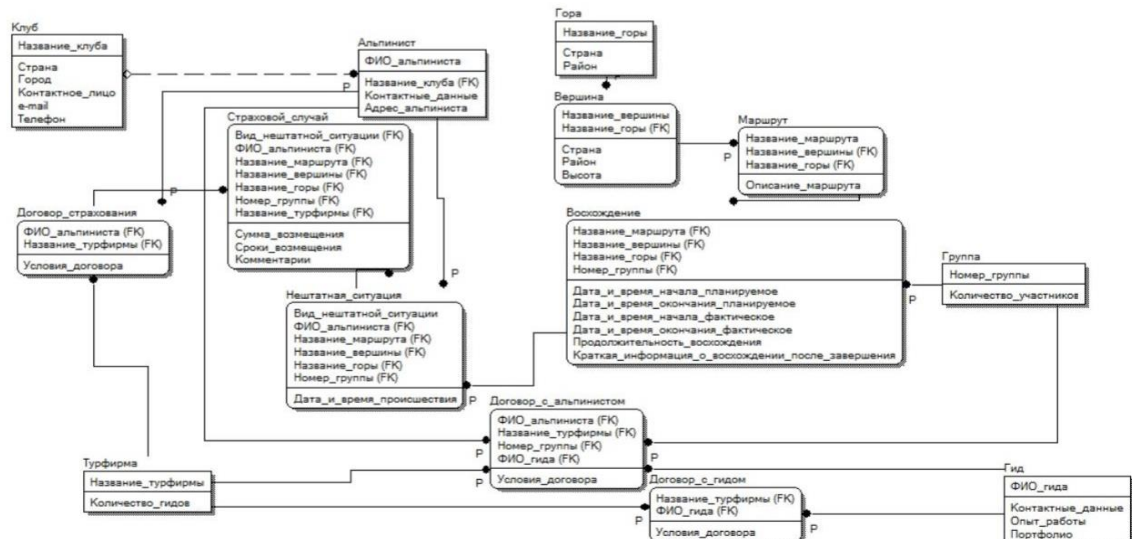


Рисунок 1 Модель базы данных

2) Логи реализации модели:

```
> db.createCollection('club');

> db.insert({id: 1, name: 'Hello club', country: 'Russia', email:
'abbracadabra@email.com'});

WriteResult({ "nInserted" : 1 })

> db.club.insert({id: 2, name: 'High fashion', country: 'Armenia', email:
'armenia.the.best@email.com'});

WriteResult({ "nInserted" : 1 })

> db.club.insert({id: 3, name: 'Shashlyk super', country: 'USA', email:
'shashlyk.ne.iz.sobaki@email.com'});

WriteResult({ "nInserted" : 1 })

> db.climber.insert({id: 1, name: 'Scott Travis', clubID: 1});

WriteResult({ "nInserted" : 1 })

> db.climber.insert({id: 2, name: 'Ksenya Sobchak', clubID: 1});

WriteResult({ "nInserted" : 1 })
```

```

> db.climber.insert({id: 3, name: 'Donald Trump', clubID: 2});
WriteResult({ "nInserted" : 1 })
> db.climber.insert({id: 4, name: 'Ivan Dorn', clubID: 1});
WriteResult({ "nInserted" : 1 })
> db.climber.insert({id: 5, name: 'Marusya Kotova', clubID: 3});
WriteResult({ "nInserted" : 1 })
> db.climber.insert({id: 6, name: 'Egor Tolkien', clubID: 3});
WriteResult({ "nInserted" : 1 })
> db.firm.insert({id: 1, name: 'Ne odnodnevka'});
WriteResult({ "nInserted" : 1 })
> db.firm.insert({id: 2, name: 'The worst suicide boys'});
WriteResult({ "nInserted" : 1 })
> db.firm.insert({id: 3, name: 'Unknown firm'});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 1, name: 'Vasya Petrov', firmID: 1, experience: 7});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 2, name: 'Irina Sharohova', firmID: 1, experience: 5});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 3, name: 'Maksim Nevzorov', firmID: 2, experience: 3});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 4, name: 'Nikita Abramov', firmID: 2, experience: 2});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 5, name: 'Polina Vlasova', firmID: 2, experience: 1});
WriteResult({ "nInserted" : 1 })
> db.guide.insert({id: 6, name: 'Pavel Vlasov', firmID: 3, experience: 2});
WriteResult({ "nInserted" : 1 })
> db.climb.insert({id: 1, guideID: 1, routeID: 1, data: new Date('2020-12-12'),
participants: [

```

```

    });
    db.climb.insert({id: 2, guideID: 2, routeID: 2, data: new Date('2020-12-12'),
    participants: [
        3, 4
    ]});
    WriteResult({ "nInserted" : 1 })
    db.climb.insert({id: 3, guideID: 3, routeID: 3, data: new Date('2020-01-01'),
    participants: [
        6, 6
    ]});
    WriteResult({ "nInserted" : 1 })
    db.route.insert({id: 1, name: 'Gimalai', durationKM: 40.5});
    WriteResult({ "nInserted" : 1 })
    db.route.insert({id: 2, name: 'Everest', durationKM: 60});
    WriteResult({ "nInserted" : 1 })
    db.route.insert({id: 3, name: 'Mountain on the corner', durationKM: 10});
    WriteResult({ "nInserted" : 1 })

```

3) Запросы:

1. Вывести все восхождения, которые были позже определенной даты

```
db.climb.find({data: {$gte: new Date('2020-05-05')}})
```

	{ _id	{ data	{ guideID	{ id	{ participants	{ routeID
1	5fa43d292690b81db533acf3	2020-12-12T00:00:00.000Z	1	1	[1, 2]	1
2	5fa43d342690b81db533acf4	2020-12-12T00:00:00.000Z	2	2	[3, 4]	2
3	5fa447af2690b81db533acff	2021-01-01T00:00:00.000Z	3	4	[1]	2

2. Вывести все восхождения за определенный период

```
db.climb.find({data: {$gte: new Date('2020-05-05'), $lte: new Date('2020-12-13')}})
```

	{ _id	{ data	{ guideID	{ id	{ participants	{ routeID
1	5fa43d292690b81db533acf3	2020-12-12T00:00:00.000Z	1	1	[1, 2]	1
2	5fa43d342690b81db533acf4	2020-12-12T00:00:00.000Z	2	2	[3, 4]	2

3. Для каждого восхождения вывести количество его участников

```
db.climb.aggregate([
  {
    $project: {
      numberOfPeople: {$size: "$participants"}
    }
  }
])
```

	{ _id	{ numberOfPeople
1	5fa43d292690b81db533acf3	2
2	5fa43d342690b81db533acf4	2
3	5fa43d442690b81db533acf5	2
4	5fa447af2690b81db533acff	1

4. Вывести длину самого большого маршрута

```
db.route.aggregate([
  {
    $group: {
      _id: 1,
      maxLength: {$max: "$durationKM"}
    }
  }
])
```

	{ _id	{ maxLength
1	1	60

5. Вывести маршруты, которые длинее 40км или имеют название Gimalai

```
db.route.find({$or: [{name: 'Gimalai'}, {durationKM: {$gte: 40}}]})
```

	{ _id	{ durationKM	{ id	{ name
1	5fa43d952690b81db533acf8	40.5	1	Gimalai
2	5fa43da72690b81db533acf9	60	2	Everest

6. Вывести турфирмы, у которых есть вебсайты

```
db.firm.find({website: {$exists: true}})
```

	{ _id	{ id	{ name	{ website
1	5fa444e82690b81db533acfe	4	Special one	example.com

ВЫВОД

В лабораторной работе №7 были получены практические навыки создания базы данных в MongoDB, заполнение их рабочими данными и выполнение запросов.