

Федеральное государственное автономное образовательное учреждение высшего образования "Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики"

## **Отчетный лист по предмету "Базы данных"**

Лабораторная работа № 2

АНАЛИЗ ДАННЫХ. ПОСТРОЕНИЕ ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ БД

Группа: K3242

Студент: Здоров Филипп Кириллович

# Оглавление

[Цель работы](#)

[Задание](#)

[Вывод](#)

# Цель работы

Овладеть практическими навыками проведения анализа данных системы и построения инфологической модели данных БД.

## Задание

*Вариант: 1*

Название БД: “Организация работы гостиницы”

1. Попытаться уменьшить размер модели реляционной базы данных, с которой работали.
2. Реализовать модель в MongoDB.
3. Заполнить модель данными
4. Реализовать минимум 40% Ваших запросов, сделанных в работе по реляционным базам данных в MongoDB.

# Ход выполнения

Создаем БД "hotel"

```
> use hotel
switched to db hotel
```

Создаем коллекцию "rooms" и заполняем ее данными.

```
>
> db.rooms.insert({'number': 204, 'floor': 9})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 205, 'floor': 9})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 207, 'floor': 9})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 104, 'floor': 8})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 206, 'floor': 8})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 207, 'floor': 8})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 208, 'floor': 8})
WriteResult({ "nInserted" : 1 })
> db.rooms.insert({'number': 210, 'floor': 8})
WriteResult({ "nInserted" : 1 })
>
```

Проверяем добавление данных

```
> db.rooms.find()
{ "_id" : ObjectId("5f8728735087dc6755069c67"), "number" : 204, "floor" : 9 }
{ "_id" : ObjectId("5f8728765087dc6755069c68"), "number" : 205, "floor" : 9 }
{ "_id" : ObjectId("5f87287a5087dc6755069c69"), "number" : 207, "floor" : 9 }
{ "_id" : ObjectId("5f87287e5087dc6755069c6a"), "number" : 104, "floor" : 8 }
{ "_id" : ObjectId("5f8728805087dc6755069c6b"), "number" : 206, "floor" : 8 }
{ "_id" : ObjectId("5f8728845087dc6755069c6c"), "number" : 207, "floor" : 8 }
{ "_id" : ObjectId("5f8728875087dc6755069c6d"), "number" : 208, "floor" : 8 }
{ "_id" : ObjectId("5f8728895087dc6755069c6e"), "number" : 210, "floor" : 8 }
> |
```

Найдем все комнаты на этаже номер 9

```
> db.rooms.find({"floor": 9}, {"_id":0, floor: 0})
{ "number" : 204 }
{ "number" : 205 }
{ "number" : 207 }
> |
```

Создадим коллекцию “guests” и добавим данные

```
> db.guests.insert({'first_name': 'Антон', 'last_name': 'Говоров', 'passport': '777777'})
WriteResult({ "nInserted" : 1 })
> db.guests.insert({'first_name': 'Филипп', 'last_name': 'Здоров', 'passport': '111111'})
WriteResult({ "nInserted" : 1 })
> db.guests.insert({'first_name': 'Антон', 'last_name': 'Лапенко', 'passport': '000000'})
WriteResult({ "nInserted" : 1 })
>
>
> db.guests.find()
{ "_id" : ObjectId("5f872a315087dc6755069c6f"), "first_name" : "Антон", "last_name" : "Говоров", "passport" : "777777" }
{ "_id" : ObjectId("5f872a375087dc6755069c70"), "first_name" : "Филипп", "last_name" : "Здоров", "passport" : "111111" }
{ "_id" : ObjectId("5f872a3b5087dc6755069c71"), "first_name" : "Антон", "last_name" : "Лапенко", "passport" : "000000" }
```

Создадим коллекцию “livings” и добавим в нее данные

```
>
> db.livings.insert({'guest_id': ObjectId('5f872a3b5087dc6755069c71'), 'from': new Date("2020-06-05"), 'to': new Date("2020-10-14"), 'room': 10})
WriteResult({ "nInserted" : 1 })
> db.livings.insert({'guest_id': ObjectId('5f872a375087dc6755069c70'), 'from': new Date("2020-09-01"), 'to': new Date("2020-10-15"), 'room': 20})
WriteResult({ "nInserted" : 1 })
> db.livings.insert({'guest_id': ObjectId('5f872a315087dc6755069c6f'), 'from': new Date("2020-10-01"), 'to': new Date("2020-10-15"), 'room': 20})
WriteResult({ "nInserted" : 1 })
>
> db.livings.find()
{ "_id" : ObjectId("5f873bda5087dc6755069c75"), "guest_id" : ObjectId("5f872a3b5087dc6755069c71"), "from" : ISODate("2020-06-05T00:00:00Z"), "to" : ISODate("2020-10-14T00:00:00Z"), "room" : 10 }
{ "_id" : ObjectId("5f873c205087dc6755069c76"), "guest_id" : ObjectId("5f872a375087dc6755069c70"), "from" : ISODate("2020-09-01T00:00:00Z"), "to" : ISODate("2020-10-15T00:00:00Z"), "room" : 20 }
{ "_id" : ObjectId("5f873c395087dc6755069c77"), "guest_id" : ObjectId("5f872a315087dc6755069c6f"), "from" : ISODate("2020-10-01T00:00:00Z"), "to" : ISODate("2020-10-15T00:00:00Z"), "room" : 20 }
> |
```

Выведем даты выселения людей, которые заселились позже 05.07.2020

```
>
> db.livings.find({'from': {'$gt': new ISODate('2020-07-05')}} , {'_id': 0, 'to': 1})
{ "to" : ISODate("2020-10-15T00:00:00Z") }
{ "to" : ISODate("2020-10-15T00:00:00Z") }
>
>
```

## Вывод

В ходе лабораторной работы был изучен MongoDB, создана база данных, которую наполнили данными и сделан запрос.