

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

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

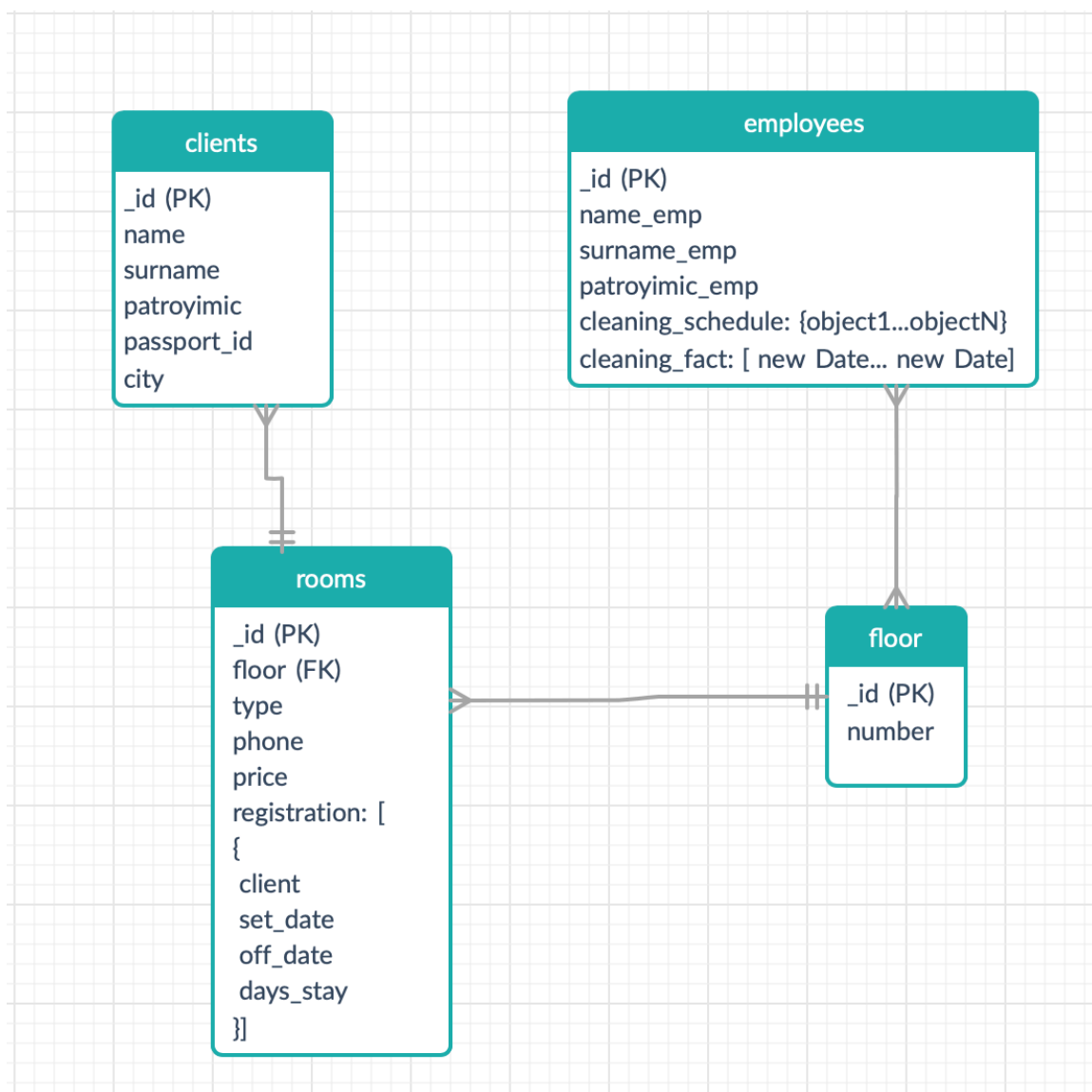
Лабораторная работа № 6
Знакомство с MongoDB

Группа: К3243
Студентка: Грицай Арина

Цель работы:

Овладеть практическими навыками реализации баз данных в MongoDB.

Новая модель базы данных:



Создание и заполнение базы данных:

Создаем базу данных:

```
> use guesthouse  
switched to db guesthouse
```

Создаем поле Этажи:

```
floor1 = ({number: 1})  
floor2 = ({number: 2})  
floor3 = ({number: 3})  
db.floors.save(floor1)  
db.floors.save(floor2)  
db.floors.save(floor3)
```

Проверяем:

```
[> db.floors.find();  
{ "_id" : ObjectId("5f07873db11a7cdce3ca16d9"), "number" : 1 }  
{ "_id" : ObjectId("5f07873db11a7cdce3ca16da"), "number" : 2 }  
{ "_id" : ObjectId("5f078740b11a7cdce3ca16db"), "number" : 3 }
```

Создаем поле Клиенты:

```
client1 = ({name: "Ivan", surname: "Mayveev", patronymic: "Olegivich", passport_id: 171819, city: "Moscow"})  
client2 = ({name: "Alena", surname: "Selezneva", patronymic: "Victorovna", passport_id: 242628, city: "Saint-Petersburg"})  
client3 = ({name: "Maria", surname: "Loginova", patronymic: "Semenovna", passport_id: 353739, city: "Bryansk"})  
client4 = ({name: "Vladimir", surname: "Mironov", patronymic: "Vladislavovich", passport_id: "464840", city: "Samara"})  
db.clients.save(client1)  
db.clients.save(client2)  
db.clients.save(client3)  
db.clients.save(client4)
```

Проверяем:

```
> db.clients.find()  
{ "_id" : ObjectId("5f07876eb11a7cdce3ca16dc"), "name" : "Ivan", "surname" : "Mayveev", "patronymic" :  
"Olegivich", "passport_id" : 171819, "city" : "Moscow" }  
{ "_id" : ObjectId("5f07876eb11a7cdce3ca16dd"), "name" : "Alena", "surname" : "Selezneva", "patronymic"  
: "Victorovna", "passport_id" : 242628, "city" : "Saint-Petersburg" }  
{ "_id" : ObjectId("5f07876eb11a7cdce3ca16de"), "name" : "Maria", "surname" : "Loginova", "patronymic"  
: "Semenovna", "passport_id" : 353739, "city" : "Bryansk" }  
{ "_id" : ObjectId("5f078772b11a7cdce3ca16df"), "name" : "Vladimir", "surname" : "Mironov", "patronymic"  
: "Vladislavovich", "passport_id" : "464840", "city" : "Samara" }
```

Создаем поле Номер:

```
db.room.insertMany([
{
  number: 101,
  "type": "single",
  phone: 612111,
  floor: floor1,
  price: 1000,
  registration: [
    {
      client: client1,
      set_date: new Date ('2020-04-04'),
      off_date: new Date ('2020-04-24'),
      days_stay: 20,
    },
  ],
},
{
  number: 202,
  type: "double",
  phone: 612222,
  floor: floor2,
  price: "2000",
  registration: [
    {
      client: client2,
      set_date: new Date ('2020-05-01'),
      off_date: new Date ('2020-05-10'),
      days_stay: 9,
    },
  ],
},
{
  number: 303,
  type: "triple",
  phone: 612333,
  floor: floor3,
  price: 3000,
  registration: [
    {
      client: client3,
      set_date: new Date ('2020-06-01'),
      off_date: new Date ('2020-06-20'),
      days_stay: 19,
    },
  ],
},
{
  number: 303,
  type: "triple",
  floor: floor3,
  price: 3000,
  registration: [
    {
      client: client4,
      set_date: new Date ('2020-06-01'),
      off_date: new Date ('2020-06-20'),
      days_stay: 19,
    },
  ],
},
])
```

Проверяем:

```
> db.room.find()
{ "_id" : ObjectId("5f0787a5b11a7cdce3ca16e0"), "number" : 101, "type" : "single", "phone" : 612111, "floor" : { "number" : 1, "_id" : ObjectId("5f07873db11a7cdce3ca16d9") }, "price" : 1000, "registration" : [ { "client" : { "name" : "Ivan", "surname" : "Mayveev", "patronymic" : "Olegivich", "passport_id" : 171819, "city" : "Moscow", "_id" : ObjectId("5f07876eb11a7cdce3ca16dc") }, "set_date" : ISODate("2020-04-04T00:00:00Z"), "off_date" : ISODate("2020-04-24T00:00:00Z"), "days_stay" : 20 } ] }
{ "_id" : ObjectId("5f0787a5b11a7cdce3ca16e1"), "number" : 202, "type" : "double", "phone" : 612222, "floor" : { "number" : 2, "_id" : ObjectId("5f07873db11a7cdce3ca16da") }, "price" : 2000, "registration" : [ { "client" : { "name" : "Alena", "surname" : "Selezneva", "patronymic" : "Victorovna", "passport_id" : 242628, "city" : "Saint-Petersburg", "_id" : ObjectId("5f07876eb11a7cdce3ca16dd") }, "set_date" : ISODate("2020-05-01T00:00:00Z"), "off_date" : ISODate("2020-05-10T00:00:00Z"), "days_stay" : 9 } ] }
{ "_id" : ObjectId("5f0787a5b11a7cdce3ca16e2"), "number" : 303, "type" : "triple", "phone" : 612333, "floor" : { "number" : 3, "_id" : ObjectId("5f078740b11a7cdce3ca16db") }, "price" : 3000, "registration" : [ { "client" : { "name" : "Maria", "surname" : "Loginova", "patronymic" : "Semenovna", "passport_id" : 353739, "city" : "Bryansk", "_id" : ObjectId("5f07876eb11a7cdce3ca16de") }, "set_date" : ISODate("2020-06-01T00:00:00Z"), "off_date" : ISODate("2020-06-20T00:00:00Z"), "days_stay" : 19 } ] }
{ "_id" : ObjectId("5f0787a5b11a7cdce3ca16e3"), "number" : 303, "type" : "triple", "floor" : { "number" : 3, "_id" : ObjectId("5f078740b11a7cdce3ca16db") }, "price" : 3000, "registration" : [ { "client" : { "name" : "Vladimir", "surname" : "Mironov", "patronymic" : "Vladislavovich", "passport_id" : "464840", "city" : "Samara", "_id" : ObjectId("5f078772b11a7cdce3ca16df") }, "set_date" : ISODate("2020-06-01T00:00:00Z"), "off_date" : ISODate("2020-06-20T00:00:00Z"), "days_stay" : 19 } ] }
```

Создаем поле Работники:

```
db.employees.insertMany([
{
  name_emp: "Ludmila",
  surname_emp: "Samoilova",
  patronymic: "Andreevna",
  wage: 10000,
  cleaning_schedule: {monday: floor1},
  cleaning_date: [new Date('2020-04-06'), new Date('2020-04-20')]
},
{
  name_emp: "Anatoliy",
  surname_emp: "Grigorev",
  patronymic: "Ivanovich",
  wage: 20000,
  cleaning_schedule: {tuesday: floor2, wednesday: floor3},
  cleaning_date: [new Date('2020-05-05'), new Date('2020-06-10')]
},
{
  name_emp: "Vasiliy",
  surname_emp: "Petrov",
  patronymic: "Dmitrievich",
  wage: 20000,
  cleaning_schedule: {thursday: floor2, saturday: floor3},
  cleaning_date: [new Date('2020-05-14'), new Date('2020-06-13')]
},
])
```

Проверяем:

```
[> db.employees.find()
{ "_id" : ObjectId("5f0787c4b11a7cdce3ca16e4"), "name_emp" : "Ludmila", "surname_emp" : "Samoilova", "patronymic" : "Andreevna", "wage" : 10000, "cleaning_schedule" : { "monday" : { "number" : 1, "_id" : ObjectId("5f07873db11a7cdce3ca16d9") } }, "cleaning_date" : [ ISODate("2020-04-06T00:00:00Z"), ISODate("2020-04-20T00:00:00Z") ] }
{ "_id" : ObjectId("5f0787c4b11a7cdce3ca16e5"), "name_emp" : "Anatoliy", "surname_emp" : "Grigorev", "patronymic" : "Ivanovich", "wage" : 20000, "cleaning_schedule" : { "tuesday" : { "number" : 2, "_id" : ObjectId("5f07873db11a7cdce3ca16da") } }, "wednesday" : { "number" : 3, "_id" : ObjectId("5f078740b11a7cdce3ca16db") } }, "cleaning_date" : [ ISODate("2020-05-05T00:00:00Z"), ISODate("2020-06-10T00:00:00Z") ] }
{ "_id" : ObjectId("5f0787c4b11a7cdce3ca16e6"), "name_emp" : "Vasiliy", "surname_emp" : "Petrov", "patronymic" : "Dmitrievich", "wage" : 20000, "cleaning_schedule" : { "thursday" : { "number" : 2, "_id" : ObjectId("5f07873db11a7cdce3ca16da") } }, "saturday" : { "number" : 3, "_id" : ObjectId("5f078740b11a7cdce3ca16db") } }, "cleaning_date" : [ ISODate("2020-05-14T00:00:00Z"), ISODate("2020-06-13T00:00:00Z") ] }
```

Запросы к базе данных:

1) Клиент, который дольше всех жил в гостинице, номер, в котором он жил, даты проживания, количество дней:

```
db.room.find({}, {_id:0, "type":0, "phone":0, "price":0}).limit(1).sort({"days_stay": -1})
```

```
> db.room.find({}, {_id:0, "type":0, "phone":0, "price":0}).limit(1).sort({"days_stay": -1})
{ "number" : "101", "floor" : { "number" : 1, "_id" : ObjectId("5f077273b11a7cdce3ca16cb") }, "registration" : [ { "client" : { "name" : "Ivan", "surname" : "Mayveev", "patronymic" : "Olegivich", "passport_id" : "171819", "city" : "Moscow", "_id" : ObjectId("5f077415b11a7cdce3ca16ce") }, "set_date" : ISODate("2020-04-04T00:00:00Z"), "off_date" : ISODate("2020-04-24T00:00:00Z"), "days_stay" : 20 } ] }
```

2) Фамилия, имя, отчество клиента, проживавшего в номере 101 с 1 апреля 2020 года:

```
cursor = db.room.findOne({number: 101}, {registration: 1});
cursor.registration.forEach(function(log) {
    if (log.set_date >= ISODate('2020-04-01')) {
        print(log.client.surname, log.client.name, log.client.patronymic)
    }
})
> cursor.registration.forEach(function(log) {
... if (log.set_date >= ISODate('2020-04-01')) {
... print(log.client.surname, log.client.name, log.client.patronymic)
... }
[... ]
Mayveev Ivan Olegivich
```

3) Все данные по клиенту с фамилией Loginova:

```
db.clients.createIndex({surname: "text"})
db.clients.find({$text: {$search: "Loginova"}})
> db.clients.find({$text: {$search: "Loginova"}})
{ "_id" : ObjectId("5f07876eb11a7cdce3ca16de"), "name" : "Maria", "surname" : "Loginova", "patronymic" : "Semenovna", "passport_id" : 353739, "city" : "Bryansk" }
```


4) Данные по сотруднику, проводившему уборку в номере, где проживал клиент с passport_id=353739:

```
cursor = db.room.findOne(
{
  "registration.client.passport_id": 353739,
  "registration.set_date": {$lte: ISODate('2020-06-13')},
  $or: [
    {"registration.off_date": {$gte: ISODate('2020-06-13')}}],
  },
{
  floor: 2,
})
db.employees.findOne({"cleaning_schedule.saturday": cursor.floor})
> cursor = db.room.findOne(
... {
...   "registration.client.passport_id": 353739,
...   "registration.set_date": {$lte: ISODate('2020-06-13')},
...   $or: [
...     {"registration.off_date": {$gte: ISODate('2020-06-13')}}],
...   ]
... },
... {
...   floor: 2,
... })
{
  "_id" : ObjectId("5f0787a5b11a7cdce3ca16e2"),
  "floor" : {
    "number" : 3,
    "_id" : ObjectId("5f078740b11a7cdce3ca16db")
  }
}
[> db.employees.findOne({"cleaning_schedule.saturday": cursor.floor})
{
  "_id" : ObjectId("5f0787c4b11a7cdce3ca16e6"),
  "name_emp" : "Vasiliy",
  "surname_emp" : "Petrov",
  "patronymic" : "Dmitrievich",
  "wage" : 20000,
  "cleaning_schedule" : {
    "thursday" : {
      "number" : 2,
      "_id" : ObjectId("5f07873db11a7cdce3ca16da")
    },
    "saturday" : {
      "number" : 3,
      "_id" : ObjectId("5f078740b11a7cdce3ca16db")
    }
  },
  "cleaning_date" : [
    ISODate("2020-05-14T00:00:00Z"),
    ISODate("2020-06-13T00:00:00Z")
  ]
}
```

5) Количество и фамилии клиентов из разных городов:

```
db.clients.mapReduce(
  function() { emit(this.city, this.surname); },
  function(key, values) { return values.length; },
  {
    query: { city: {$exists: true} },
    out: "client_city"
  }
)
db.client_city.find()
> db.clients.mapReduce(
...   function() { emit(this.city, this.surname); },
...   function(key, values) { return values.length; },
...   {
...     query: { city: {$exists: true} },
...     out: "client_city"
...   }
... )
{
  "result" : "client_city",
  "timeMillis" : 86,
  "counts" : {
    "input" : 4,
    "emit" : 4,
    "reduce" : 0,
    "output" : 4
  },
  "ok" : 1
}
> db.client_city.find()
{ "_id" : "Bryansk", "value" : "Loginova" }
{ "_id" : "Moscow", "value" : "Mayveev" }
{ "_id" : "Saint-Petersburg", "value" : "Selezneva" }
{ "_id" : "Samara", "value" : "Mironov" }
```

Вывод: в результате выполнения лабораторной работы были получены навыки работы с нереляционными базами данных на примере MongoDB.