Департамент образования города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

Инструменты для хранения и обработки больших данных

Лабораторная работа 2.1

Изучение методов хранения данных на основе NoSQL

Выполнила: студентка группы АДЭУ-221

Ильина Алина Сергеевна

Проверил:

доцент департамента информатики, управления и технологий

Босенко Тимур Муртазович

Москва

2025

Задание 1 (MongoDB)

Задание 1.1. Вариант 8. Медицинские карты пациентов

Разработайте систему учета пациентов с коллекцией patients:

Структура документа:

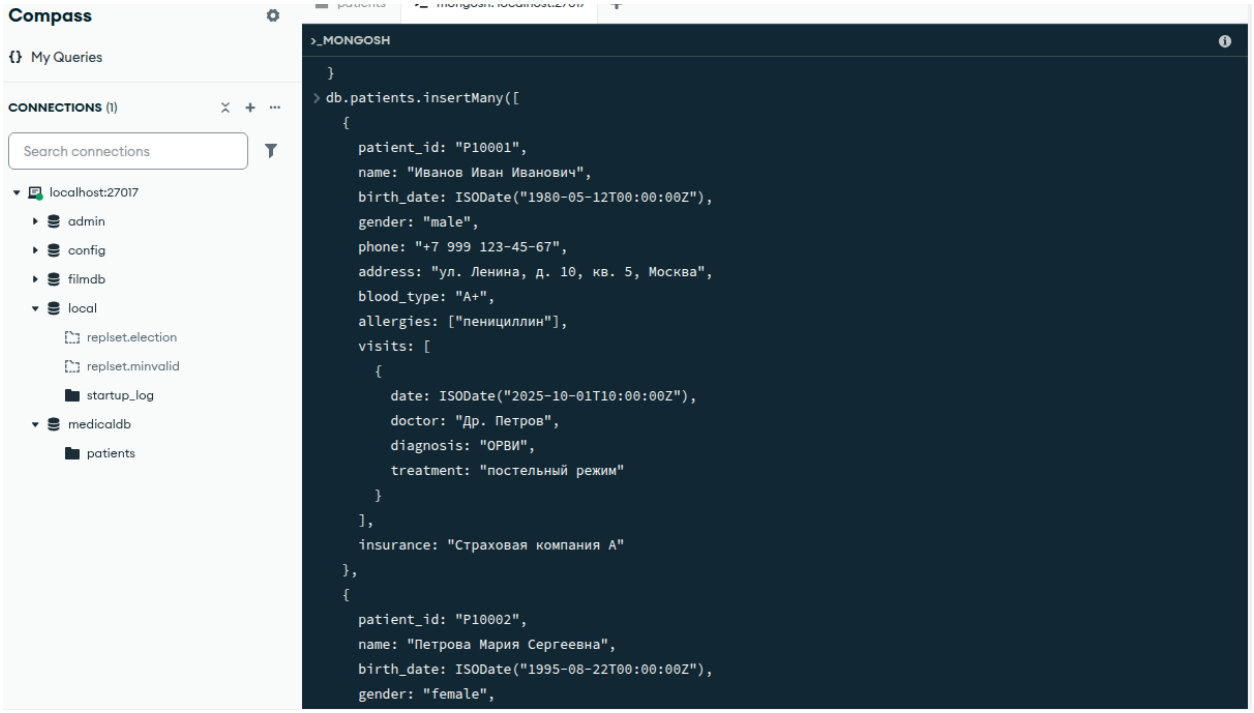
```
{
    "_id": ObjectId,
    "patient_id": str,      # ID пациента
    "name": str,            # ФИО
    "birth_date": datetime,# Дата рождения
    "gender": str,          # Пол
    "phone": str,           # Телефон
    "address": str,         # Адрес
    "blood_type": str,      # Группа крови
    "allergies": [str],     # Аллергии
    "visits": [             # Посещения
        {
            "date": datetime,
            "doctor": str,
            "diagnosis": str,
            "treatment": str
        }
    ],
    "insurance": str        # Страховка
}
```

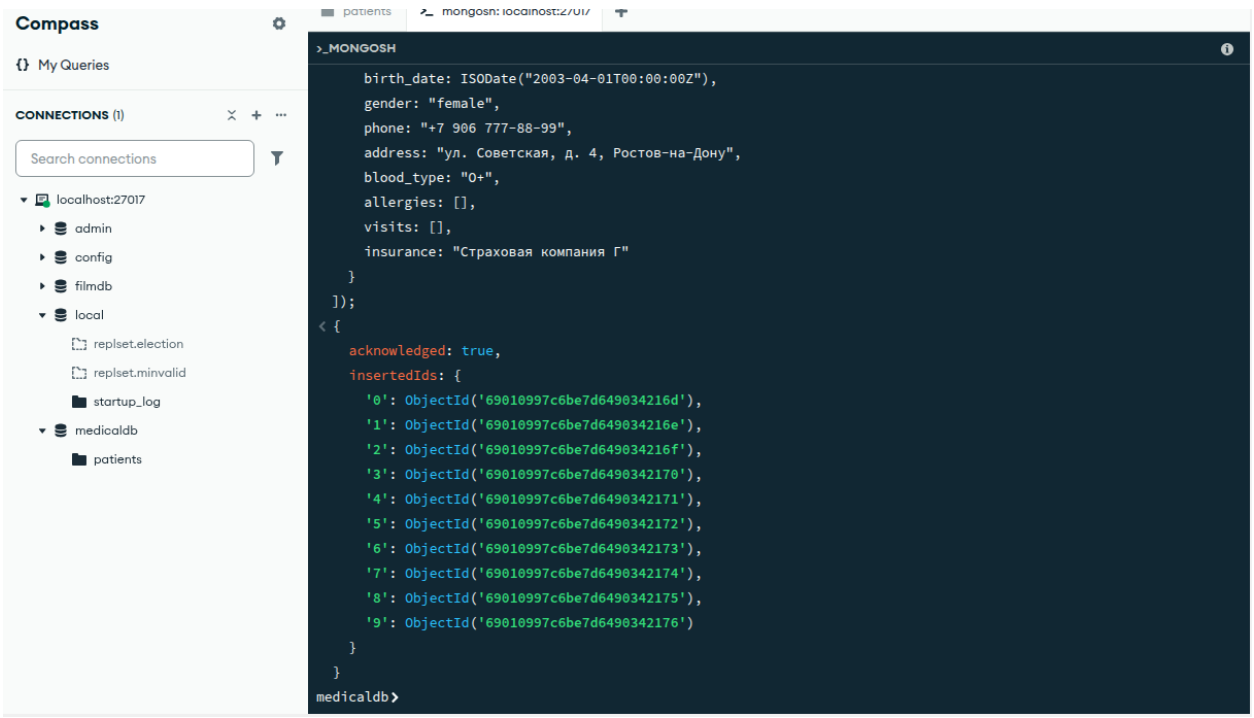
Функционал:

- Добавление пациента
- Запись визита к врачу
- Поиск пациентов по врачу
- Получение истории болезни
- Поиск по аллергиям
- Статистика по возрастным группам

1. Добавление пациента

Добавляем пациентов с помощью команды insertMany





Одного пациента можно добавить с помощью команды insertOne()

```
> db.patients.insertOne({
      patient_id: "P10011",
      name: "Петрова Вера Сергеевна",
      birth_date: ISODate("1994-09-23T00:00:00Z"),
      gender: "female",
      phone: "+7 911 245-59-78",
      address: "пр-т Мира, д. 5, Москва",
      blood_type: "B-",
      allergies: [],
      visits: [],
      insurance: "Страховая компания Б"
  });
< {
    acknowledged: true,
    insertedId: ObjectId('69010cddc6be7d6490342178')
  }
medicaldb>
```

2. Запись визита к врачу (Update)

Чтобы добавить визит к врачу, нужно обновить массив visits, добавив новый элемент с помощью оператора $push, например, по patient_id

```
> db.patients.updateOne(
      { patient_id: "P10005" },
      {
          $push: {
              visits: {
                  date: ISODate("2025-10-15T14:00:00Z"),
                  doctor: "Др. Сидоров",
                  diagnosis: "Грипп",
                  treatment: "антивирусные препараты"
              }
          }
      }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
medicaldb>
```

- acknowledged: true - сервер подтвердил операцию
- upsertedId: null - операция не вставляла новый документ (upsert не применялся)

- matchedCount: 1 - найден один документ с patient_id "P10005"
- modifiedCount: 1 - этот документ был успешно обновлен (в массив visits добавлен новый визит)
- upsertCount: 0 - обновился существующий документ

3. Поиск пациентов по врачу (Query)

Находим всех пациентов, у которых были визиты к определённому врачу Сидорову

```
>_MONGOSH
> db.patients.find(
      { "visits.doctor": "Др. Сидоров" }
  );
< {
    _id: ObjectId('69010997c6be7d6490342171'),
    patient_id: 'P10005',
    name: 'Морозов Дмитрий Павлович',
    birth_date: 2000-03-18T00:00:00.000Z,
    gender: 'male',
    phone: '+7 901 222-33-44',
    address: 'ул. Ленина, д. 1, Казань',
    blood_type: 'A-',
    allergies: [],
    visits: [
      {
        date: 2025-08-25T09:30:00.000Z,
        doctor: 'Др. Смирнова',
        diagnosis: 'Грипп',
        treatment: 'постельный режим, жаропонижающее'
      },
      {
        date: 2025-10-10T11:00:00.000Z,
        doctor: 'Др. Смирнова',
        diagnosis: 'Бронхит',
        treatment: 'антибиотики'
      },
```

```
        diagnosis: 'Бронхит',
        treatment: 'антибиотики'
      },
      {
        date: 2025-10-15T14:00:00.000Z,
        doctor: 'Др. Сидоров',
        diagnosis: 'Грипп',
        treatment: 'антивирусные препараты'
      }
    ],
    insurance: 'Страховая компания Б'
  }
medicaldb>
```

Также можно получить только имена пациентов:

```
> db.patients.find(
    { "visits.doctor": "Др. Сидоров" },
    { name: 1, _id: 0 }
  );
< {
    name: 'Морозов Дмитрий Павлович'
  }
medicaldb>
```

4. Получение истории болезни (Query)

Получаем полную историю болезни пациента по его patient_id:

```
> db.patients.findOne(
    { patient_id: "P10005" },
    { name: 1, visits: 1, _id: 0 }
);
< {
  name: 'Морозов Дмитрий Павлович',
  visits: [
    {
      date: 2025-08-25T09:30:00.000Z,
      doctor: 'Др. Смирнова',
      diagnosis: 'Грипп',
      treatment: 'постельный режим, жаропонижающее'
    },
    {
      date: 2025-10-10T11:00:00.000Z,
      doctor: 'Др. Смирнова',
      diagnosis: 'Бронхит',
      treatment: 'антибиотики'
    },
    {
      date: 2025-10-15T14:00:00.000Z,
      doctor: 'Др. Сидоров',
      diagnosis: 'Грипп',
```

```
      date: 2025-08-25T09:30:00.000Z,
      doctor: 'Др. Смирнова',
      diagnosis: 'Грипп',
      treatment: 'постельный режим, жаропонижающее'
    },
    {
      date: 2025-10-10T11:00:00.000Z,
      doctor: 'Др. Смирнова',
      diagnosis: 'Бронхит',
      treatment: 'антибиотики'
    },
    {
      date: 2025-10-15T14:00:00.000Z,
      doctor: 'Др. Сидоров',
      diagnosis: 'Грипп',
      treatment: 'антивирусные препараты'
    }
  ]
}
medicaldb>
```

5. Поиск по аллергиям (Query)

Находим всех пациентов с определённой аллергией на "пенициллин"

```
>_MONGOSH
> db.patients.find(
      { allergies: "пенициллин" }
  );
< {
    _id: ObjectId('69010997c6be7d649034216d'),
    patient_id: 'P10001',
    name: 'Иванов Иван Иванович',
    birth_date: 1980-05-12T00:00:00.000Z,
    gender: 'male',
    phone: '+7 999 123-45-67',
    address: 'ул. Ленина, д. 10, кв. 5, Москва',
    blood_type: 'A+',
    allergies: [
      'пенициллин'
    ],
    visits: [
      {
        date: 2025-10-01T10:00:00.000Z,
        doctor: 'Др. Петров',
        diagnosis: 'ОРВИ',
        treatment: 'постельный режим'
      }
    ],
    insurance: 'Страховая компания А'
  }
  {
```

```
>_MONGOSH
    gender: 'female',
    phone: '+7 900 111-22-33',
    address: 'ул. Тверская, д. 7, Москва',
    blood_type: 'O+',
    allergies: [
      'пенициллин'
    ],
    visits: [],
    insurance: 'Страховая компания А'
  }
  {
    _id: ObjectId('69010997c6be7d6490342174'),
    patient_id: 'P10008',
    name: 'Егорова Елена Сергеевна',
    birth_date: 1990-02-14T00:00:00.000Z,
    gender: 'female',
    phone: '+7 904 555-66-77',
    address: 'пр-т Ленина, д. 8, Волгоград',
    blood_type: 'AB-',
    allergies: [
      'пенициллин'
    ],
    visits: [],
    insurance: 'Страховая компания Г'
  }
medicaldb>
```

6. Статистика по возрастным группам (Aggregation)

Для этого нам необходимо вычислить возраст каждого пациента (по дате рождения) и сгруппировать их, например, по возрастным группам 0–17, 18–40, 41–60, 61+

```
> db.patients.aggregate([
    {
        $addFields: {
            age: {
                $dateDiff: {
                    startDate: "$birth_date",
                    endDate: new Date(),
                    unit: "year"
                }
            }
        }
    },
    {
        $bucket: {
            groupBy: "$age",
            boundaries: [0, 18, 41, 61, 200],
            default: "Other",
            output: {
                count: { $sum: 1 }
            }
        }
    }
]);
< {
    _id: 18,
```

```
]);
< {
    _id: 18,
    count: 7
  }
  {
    _id: 41,
    count: 4
  }
medicaldb>
```

Здесь _id — это нижняя граница возрастной группы, а count — число пациентов в этой группе

Задание 1.2. Найти все фильмы, у которых поле plotOutline не существует ($exists: false), и установить им это поле со значением "Description pending".

acknowledged: true — сервер подтвердил выполнение операции

insertedId: null означает, что в результате операции MongoDB не был вставлен новый документ, то есть никакого нового уникального _id не создано

matchedCount: 1 — количество документов, которые подошли под условие фильтра (у которых не существует поле plotOutline)

modifiedCount: 1 — сколько из них реально были обновлены (им было добавлено поле plotOutline со значением "Description pending").

upsertedCount: 0 показывает, что в операции upsert (обновление с возможной вставкой) не было создано новых документов. Это значит, что обновился уже существующий документ

Задание 2 (Neo4j)

Посчитать, в скольких фильмах снялся каждый актер, и вывести топ-5 актеров по количеству фильмов.

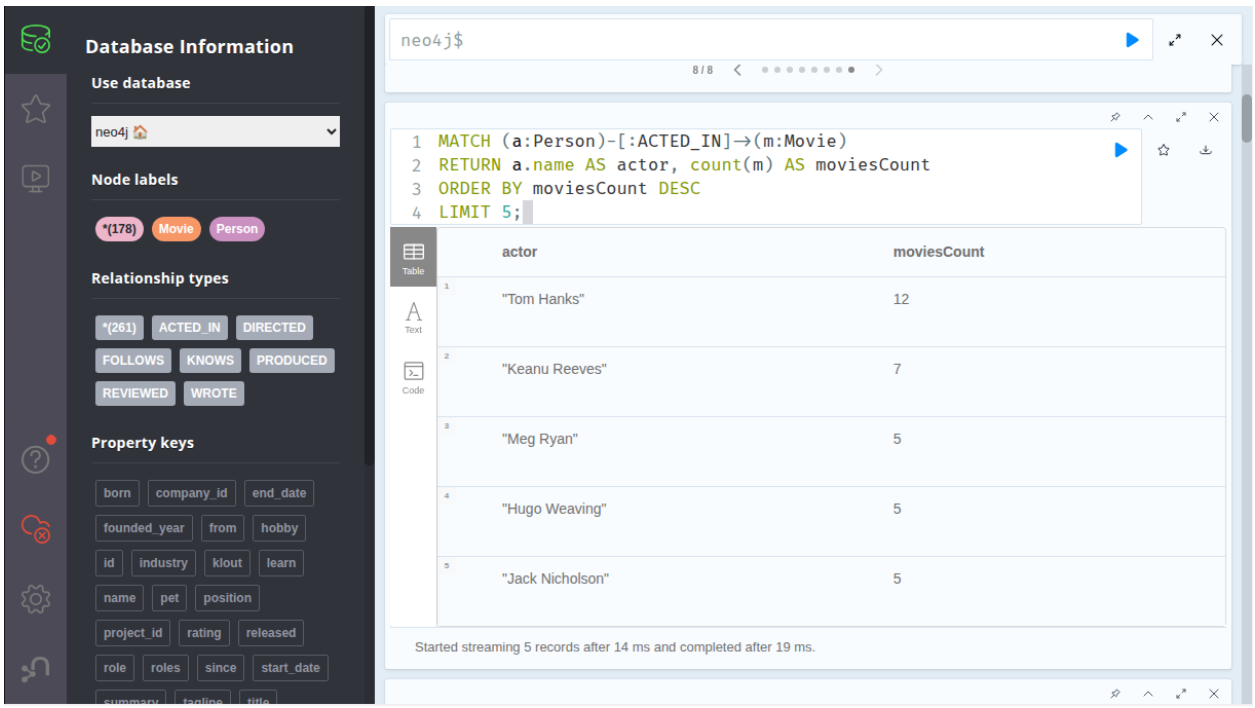
Предварительно была добавлена база данных по гайду с фильмами и актерами.

MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)

RETURN a.name AS actor, count(m) AS moviesCount

ORDER BY moviesCount DESC

LIMIT 5;



Задание 3 (Redis)

Установить ключ api_key:user7 со значением и сроком жизни 1 час (SET с EX). Проверить оставшееся время жизни ключа (TTL).

Запускаем Redis CLI

```
mgpu@mgpu-vm:~$ docker run -it --rm --network nosql-platform bitnami/redis redis
-cli -h redis-1 -p 6379
redis 22:01:14.73 INFO  ==>
redis 22:01:14.73 INFO  ==> Welcome to the Bitnami redis container
redis 22:01:14.74 INFO  ==> Subscribe to project updates by watching https://git
hub.com/bitnami/containers
redis 22:01:14.74 INFO  ==> NOTICE: Starting August 28th, 2025, only a limited s
ubset of images/charts will remain available for free. Backup will be available
for some time at the 'Bitnami Legacy' repository. More info at https://github.co
m/bitnami/containers/issues/83267
redis 22:01:14.75 INFO  ==>
```

Сначала устанавливаем новую пару «ключ-значение»

```
redis-1:6379> SET api_key:user7 "значение"
OK
```

А затем устанавливаем срок его действия в 3600 секунд с помощью команды EXPIRE

```
redis-1:6379> EXPIRE api_key:user7 3600
(integer) 1
```

Можем проверить, как долго ключ будет существовать, с помощью команды TTL. Она возвращает количество секунд до его удаления

```
redis-1:6379> TTL api_key:user7
(integer) 3582
redis-1:6379>
```

Или в 2 запроса

```
redis-1:6379> SET api_key:user7 значение_ключа EX 3600
OK
redis-1:6379> TTL api_key:user7
(integer) 3585
redis-1:6379>
```