Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Инструменты для хранения и обработки больших данных

**Лабораторная работа №2.1**

**Тема:**

**«**Изучение методов хранения данных на основе NoSQL.**»**

Выполнила: Сергеева А. И., группа: АДЭУ-211

Преподаватель: Босенко Т. М.

Москва

2024

**Цель работы:** изучить и освоить методы хранения и работы с данными в NoSQL базах данных MongoDB, Redis и Neo4j. Научиться загружать данные из CSV файлов в указанные СУБД и выполнять базовые операции по работе с данными.

**Оборудование и ПО:**

- Операционная система Ubuntu.

- Установленные пакеты для работы с NoSQL базами данных: MongoDB, Redis, Neo4j.

- Язык программирования Python (с библиотеками pymongo, redis, neo4j).

- CSV файл с данными.

**Ход работы**

**Шаг 1. Работа с MongoDB, реализация примера и задания.**

MongoDB — это документоориентированная система управления базами данных с открытым исходным кодом. Она считается одним из классических примеров NoSQL-систем.

Особенности MongoDB:

* Данные хранятся в виде коллекций и документов. У коллекций не обязательно должна быть схожая структура. У одного документа может быть один набор полей, в то время как у другого документа — совершенно другой (как тип, так и количество полей).
* Информация отформатирована в BSON — двоичной кодировке JSON-подобных документов. Это позволяет поддерживать данные типа Date и двоичных файлов, что невозможно в JSON.

В рамках примера необходимо было настроить подключение к MongoDB (рисунок 1).

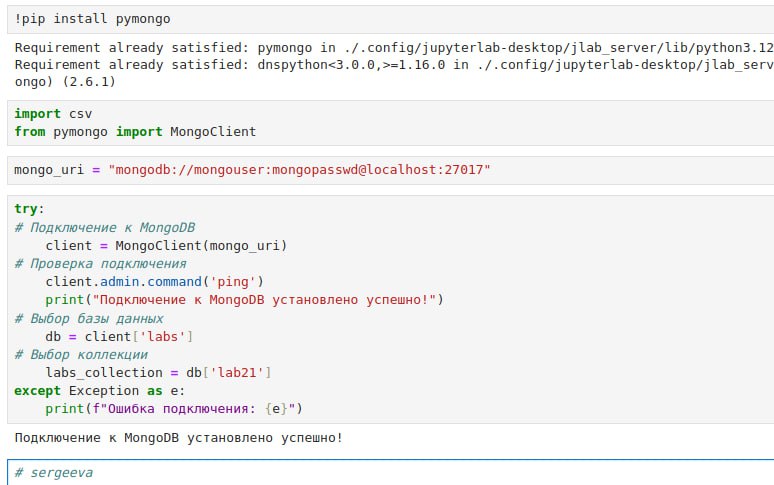


Рисунок 1 – Подключение к MongoDB

В рамках индивидуального задания был взят csv-файл, который хранит информацию о ФИО студентов, их специальностей и курсов на 112 записей. С помощью функции DictReader() каждая строка в CSV-файле преобразуется в словарь, где заголовки столбцов являются ключами, а значения строк — значениями. Далее каждый документ (данные о каждом студенте) с помощью insert\_one() по одному документу (их перебор идет через цикл for) вставляется в коллекцию MongoDB и далее выводится имя коллекции, куда осуществилась вставка. С помощью функции update\_one происходит обновление данных, а именно поле «студент» со значением Мальцев Константин Александрович, заменяется на Мальцев Константин Владимирович. Далее данная строчка удаляется (рисунки 2-5).



Рисунок 2 – Загрузка данных из csv файла

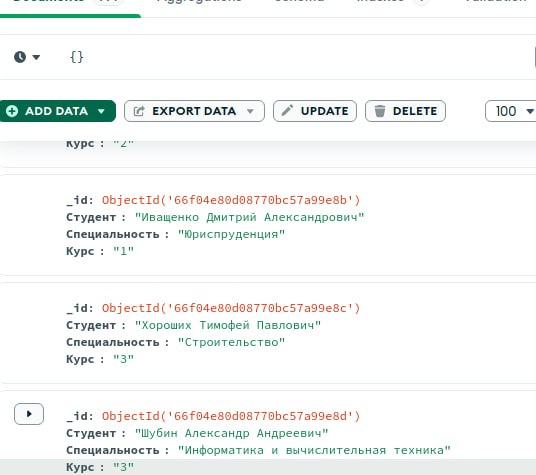


Рисунок 3 – Выгруженные документы в MongoDB

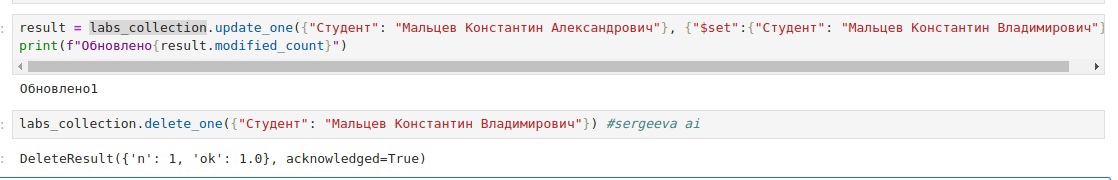


Рисунок 4 – Обновление и удаление данных

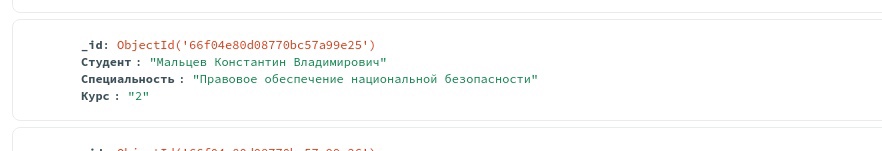


Рисунок 5 – Обновленная строка

**Шаг 2. Работа с Resis, реализация примера и задания.**

Redis (от англ. remote dictionary server) — нереляционная система управления базами данных (СУБД) класса NoSQL. Работает со структурами данных типа «ключ — значение».

Redis используется как для баз данных, так и для реализации кэшей, брокеров сообщений.

Для начала работы запустим докер к Redis (рисунок 6).

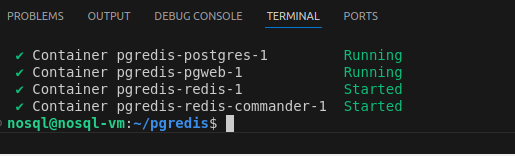


Рисунок 6 – Запуск докера

Далее в рамках примера и для дальнейшей работы необходимо было подключиться к Redis (рисунок 7).



Рисунок 7 – Подключение к Redis

В рамках примера было сгенерировано и создано 10 записей (рисунки 7-9).

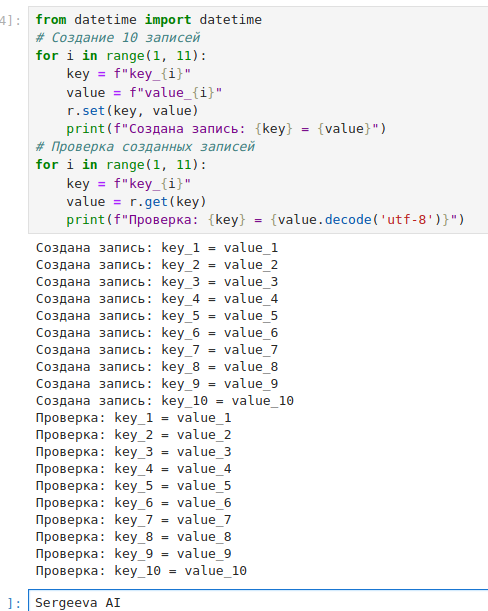


Рисунок 8 – Создание 10 записей

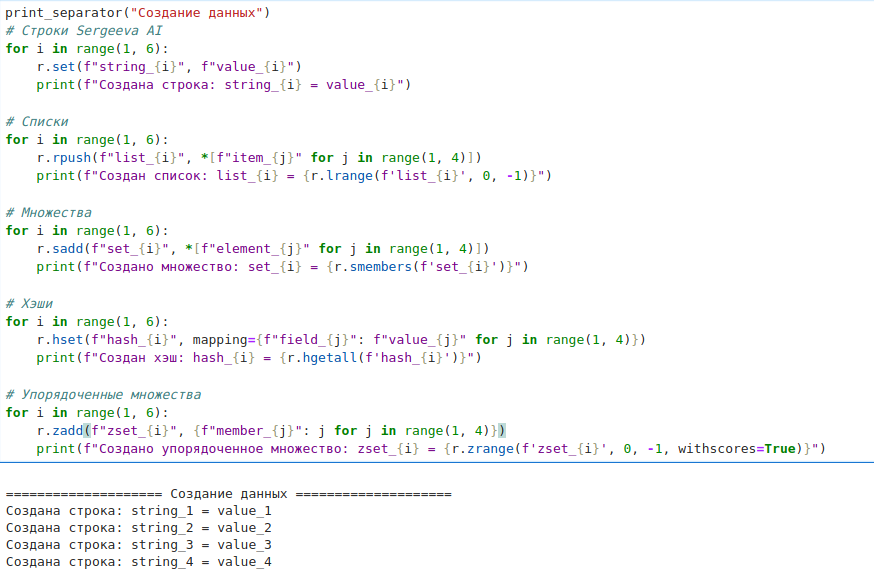


Рисунок 9 – создание разных типов данных

За получение данных отвечают функции get, lrange (возвращает элементы или значения, хранящиеся по ключу, в указанном диапазоне списков), smembers (возвращает массив со всеми элементами набора), hgetall (возвращает все поля и значения хэша, хранящегося по указанному ключу), zrange (возвращает значения элементов из отсортированного набора) (рисунок 10).

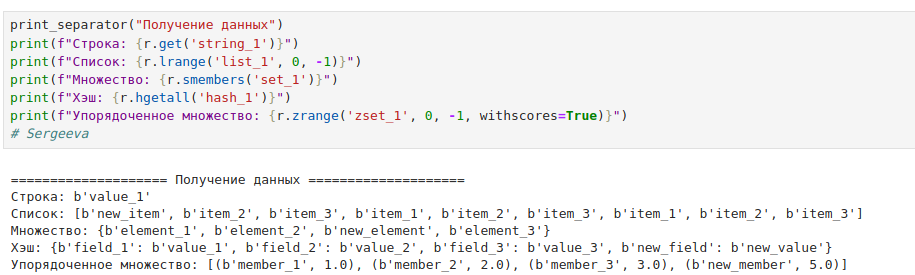


Рисунок 10 – Получение данных

Также были проведены обновления и удаления данных (рисунки 11-13).



Рисунок 11- Обновление данных

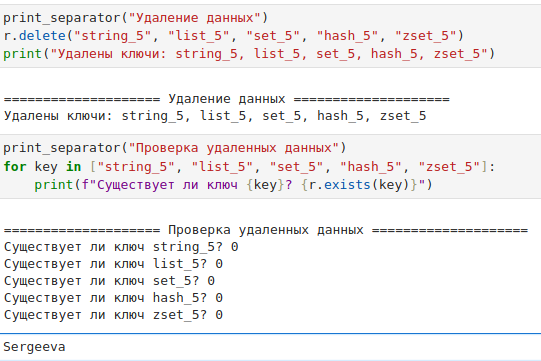


Рисунок 12 – Удаление данных

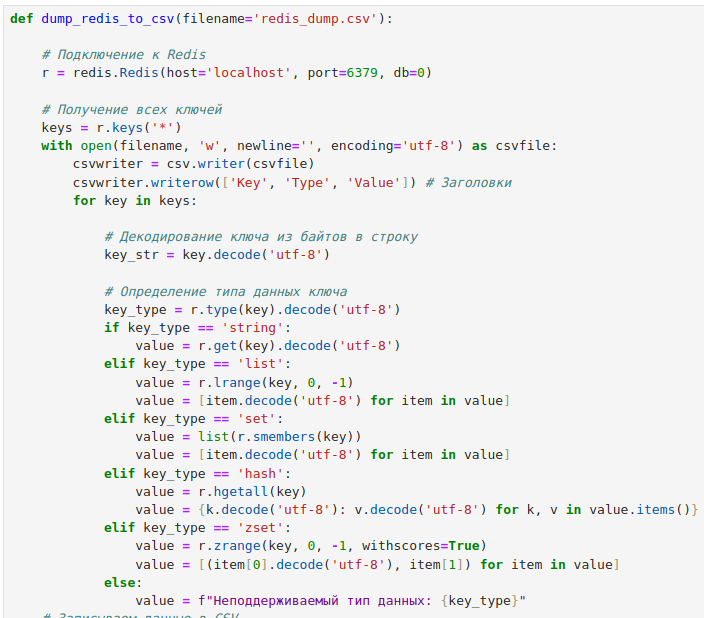


Рисунок 13 – Выгрузка данных в CSV

Далее загружалась таблица про студентов, используемая в примере выше для реализации выгрузки данных в Redis и применения базовых операций над ними (рисунок 14).

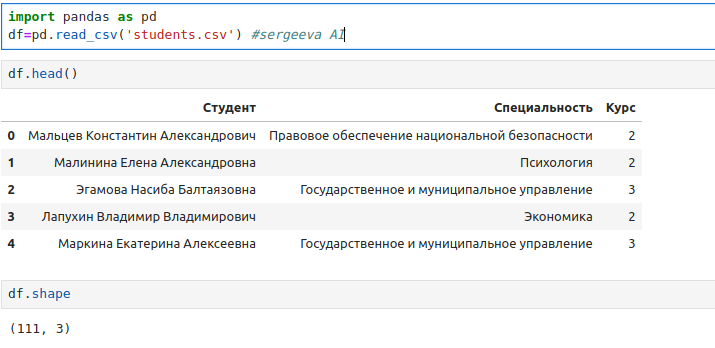


Рисунок 14 – Загрузка данных формата CSV

Далее ключи и значения добавлялись в Redis. В данном примере уже использовалась другая функция для построчного добавления данных, это itertuples, перебирающая строки в виде кортежей значений. Также была осуществлена проверка выгруженных данных, для каждого ключа выводилось его значение (рисунки 15-16).

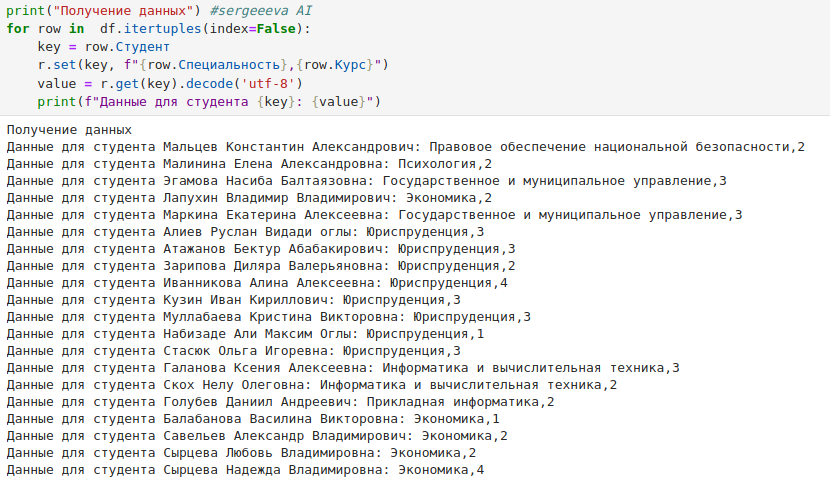


Рисунок 15 – Загрузка данных в Redis

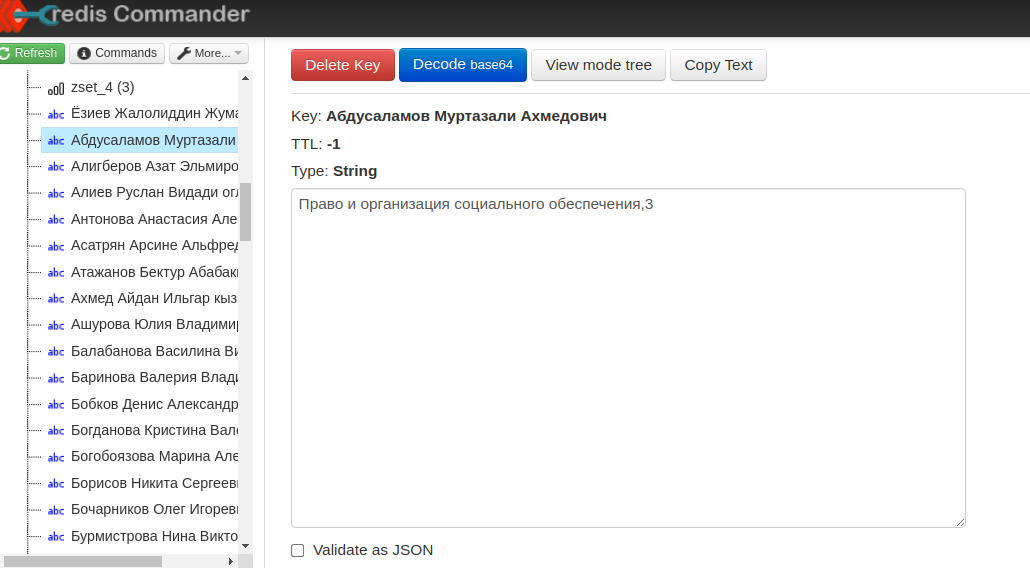


Рисунок 16 – Данные в Redis

Для Мальцева Константина Александровича я изменила курс со 2 на 1 (рисунок17).

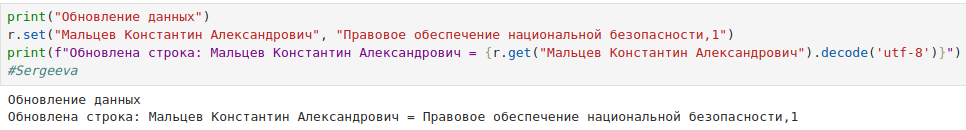


Рисунок 17 – Обновление данных

Далее данная строка была удалена (рисунок 18).

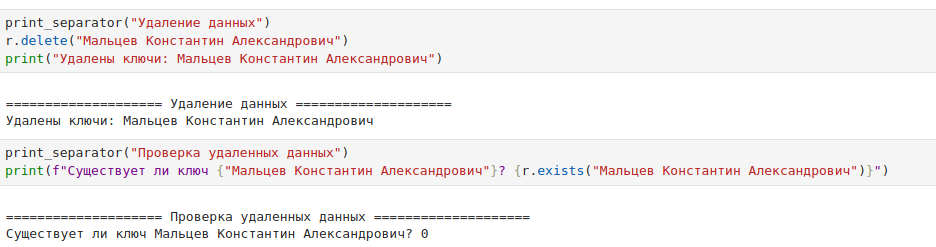


Рисунок 18 – Удаление данных

**Шаг 3. Запрос, чтобы получить список всех сотрудников, которые участвуют в создании хотя бы одного курса в роли автора, диктора или редактора на Neo4j.**

Neo4j — это графовая система управления базами данных с открытым исходным кодом, реализованная на Java.

Модель Neo4j в основном состоит из следующих основных компонентов:

* Узлы (эквивалентны вершинам в теории графов): основные элементы данных, которые связаны отношениями. Узлы могут иметь метки и свойства.
* Отношения (эквивалентно ребрам в теории графов): описывает соединения между узлами и соединяет их вместе. Отношения могут иметь одно или несколько свойств.
* Метки: представляют роль узлов. Метки используются для группировки узлов. У каждого узла может быть несколько меток. Метки также индексируются для ускорения процесса поиска узлов в графе.
* Свойства: атрибуты узлов и отношений, включающих пары имен или значений.

Задание по выгрузке сотрудников, которые участвуют в создании хотя бы одного курса в роли автора, диктора или редактора, было выполнено на базе «Учебные курсы». Для начала необходимо было внести данные в базу (рисунок 19).

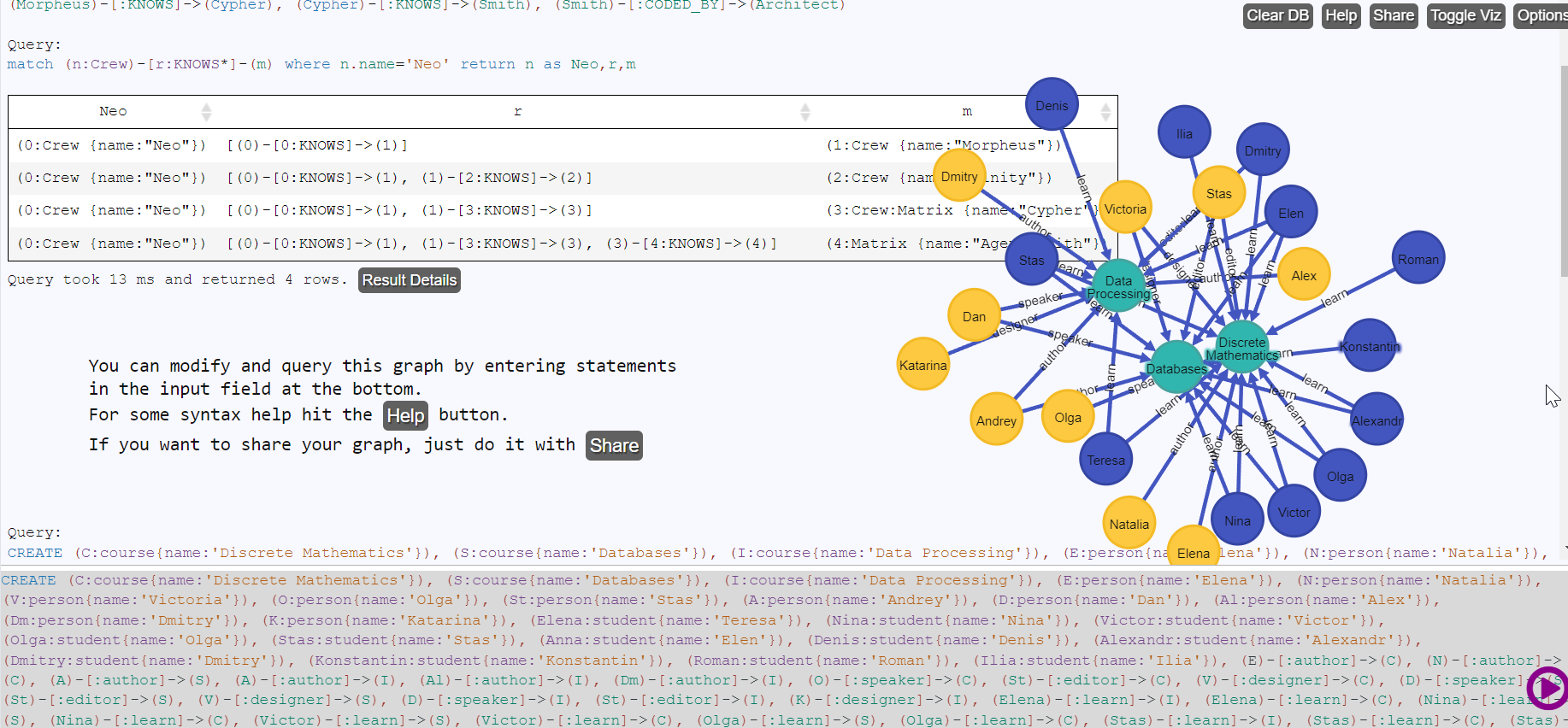


Рисунок 19 – Загрузка данных по курсам, студентам и сотрудникам

Затем был составлен нужный запрос. MATCH использовалось для создания выборки, переменным p, c были присвоены преподаватели и курсы, а в связи указывались роли, далее был задан вывод уникальных имен, ведь один преподаватель может быть связан с несколькими курсами (рисунки 20-22).

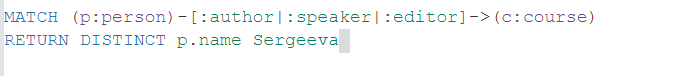


Рисунок 20 – Запрос на вывод авторов, дикторов или редакторов курсов

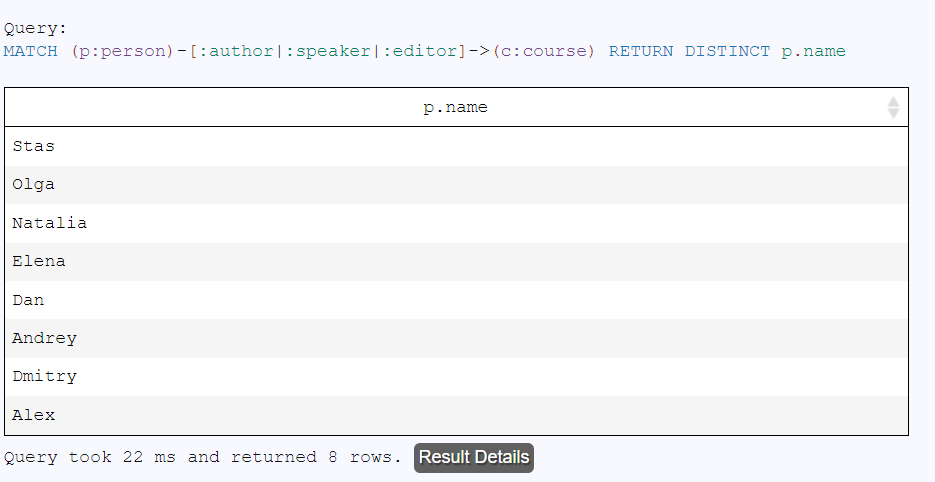


Рисунок 21 – Отображение результата запроса

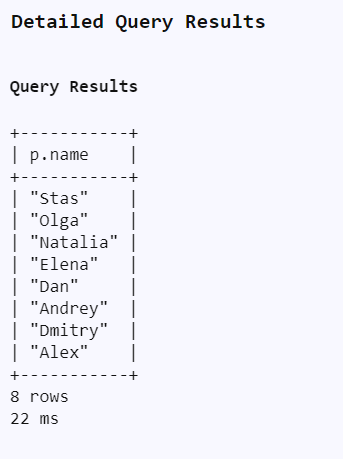


Рисунок 22 – Детальный отчет по результату

**Выводы**

1. Redis обладает высокой скоростью, т.к. это хранилище ключ-значение, но вместе с этим он хорош для хранения простых структур данных (строки, списки и т.д.), также объем хранимых данных ограничен доступным объемом оперативной памяти, что не позволит работать с большими данными. Хорош для кэширования.
2. В MongoDB данные хранятся в виде документов JSON, для хранения данных о студентах в рамках моего задания он лучше. Можно создавать схемы данных с различными полями для каждого студента отдельно и нет ограничений оперативной памятью для данных.
3. Neoj4 позволяет хранить данные с отношениями (например, соц. сети), что позволяет также быстро обращаться к данным через их связи. Также используется для запросов язык Cypher, несложный для понимания. Подходит для графовых структур.