|  |
| --- |
| ИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| **«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |
| **(МОСКОВСКИЙ ПОЛИТЕХ)** |
| Факультет информационных технологий |

Кафедра «Прикладная информатика»

Форма обучения: очная

**Лабораторная работа №5**

**Neo4j**

**по дисциплине**

**«Хранилища данных»**

Студент (А. Костромин)

(личная подпись) (И. Фамилия)

**Москва 2021**

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc87121834)

[ХОД РАБОТЫ 5](#_Toc87121835)

[1 Установка Neo4j 5](#_Toc87121836)

[2 Создание социального графа 6](#_Toc87121837)

[РЕЗУЛЬТАТЫ 8](#_Toc87121838)

[ЗАКЛЮЧЕНИЕ 11](#_Toc87121839)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc87121840)

# ВВЕДЕНИЕ

Neo4j является ведущей в мире графической базой данных с открытым исходным кодом, которая разработана с использованием технологии Java. Он легко масштабируется и не содержит схем (NoSQL). По состоянию на 2015 год считается самой распространённой графовой СУБД. Разработчик — американская компания Neo Technology, разработка ведётся с 2003 года.

Данные хранит в собственном формате, специализированно приспособленном для представления графовой информации, такой подход в сравнении с моделированием графовой базы данных средствами реляционной СУБД позволяет применять дополнительную оптимизацию в случае данных с более сложной структурой. Также утверждается о наличии специальных оптимизаций для SSD-накопителей, при этом для обработки графа не требуется его помещение целиком в оперативную память вычислительного узла, таким образом, возможна обработка достаточно больших графов.

Основные транзакционные возможности — поддержка ACID и соответствие спецификациям JTA, JTS и XA. Интерфейс программирования приложений для СУБД реализован для многих языков программирования, включая Java, Python, Clojure, Ruby, PHP, также реализовано API в стиле REST.

Граф представляет собой графическое представление набора объектов, где некоторые пары объектов связаны ссылками. Он состоит из двух элементов — узлов (вершин) и отношений (ребер).

База данных графиков — это база данных, используемая для моделирования данных в форме графиков. Здесь узлы графа изображают сущности, в то время как отношения изображают ассоциацию этих узлов.

В СУБД используется собственный язык запросов — Cypher, но запросы можно делать и другими способами, например, напрямую через Java API и на языке Gremlin, созданном в проекте с открытым исходным кодом TinkerPop. Cypher является не только языком запросов, но и языком манипулирования данными, так как предоставляет функции CRUD для графового хранилища [[1](https://ru.wikipedia.org/wiki/Neo4j)].

# ХОД РАБОТЫ

## Установка Neo4j

Для выполнения лабораторной работы необходимо:

* скачать и установить десктопный клиент Neo4j Desktop с официального сайта [[3](https://neo4j.com/download-center/#desktop)];
* настроить и создать новый проект;
* добавить новую базу данных и запустить ее;
* получить доступ по адресу [[i](http://localhost:7474/browser/)].

Сама программа представлена на Рисунке 1.

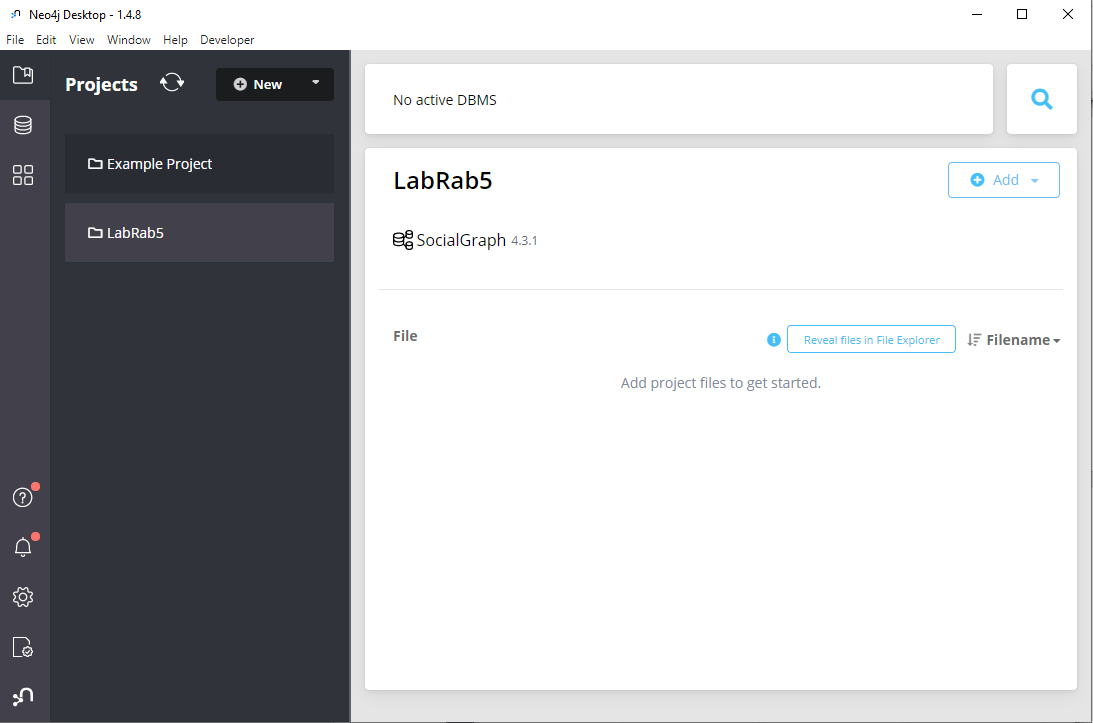


Рисунок 1 – Neo4j Desktop

Пример работы с интерфейсом в браузере представлен на Рисунке 2. Сверху присутствует консоль для команд, которая будет использоваться в дальнейшем для создания графа.



Рисунок 2 – Neo4j в браузере

## Создание социального графа

Выполнение лабораторной работы происходило пошагово. Первым шагом было создание нодов с типом «Person». Данные ноды будут использоваться для обозначения людей и их атрибутов. Было создано 13 нодов данного типа. Для их создания использовалась следующая команда:

CREATE (n: Person {name:’Артем Иванов’}) RETURN (n)

Следующим шагом стало создание нодов типа «Messenger». Данные ноды будут использоваться для обозначения и хранения информации о мессенджерах. В результате было создано 4 нода с таким типом. Для их создания использовалась следующая команда:

CREATE (n: Messenger {name:’Viber’})

RETURN (n)

После этого были созданы связи между нодами с типом «Person». Для этого были использованы связи с названиями: «FRIEND», «PARENT». Первая связь означает, что два человека являются друг другу друзьями, вторая означает то, что один из нодов является родителем другому. Всего таких связей было создано 9 и 2 соответственно. Для создания связей были использованы следующие команды:

MATCH (a:Person{name:’Олеся Куценко’}),

(b:Person{name:’Евгений Топов’})

MERGE (a)-[r:FRIEND]->(b)

Следующим шагом стало добавление атрибутов для людей. В данном случае атрибутом является возраст. Все атрибуты были добавлены при помощи следующей команды:

MATCH (n:Person {name:’Артем Иванов’})

SET n.age = 35

RETURN n

И наконец, были созданы связи среди нодов с типом «Person» и «Messenger». Название связи: «USE». То есть, связь показывает то, что человек пользуется тем или иным мессенджером. Всего таких связей было создано 20. Для создания связей были использованы следующие команды:

MATCH (a:Person{name: ’Петр Абрамов’}),

(b:Messenger{name:’Viber’})

MERGE (a)-[r:USE]->(b)

Подсчет количества нодов и связей может быть реализован с помощью специальных команд, а также с помощью визуальной оценки в самой системе. На Рисунке 3 отображено меню, где отображается количество нодов и рёбер.



Рисунок 3 – Количество нодов и рёбер

# РЕЗУЛЬТАТЫ

В результате получился граф, содержащий:

* с 17 нодами (13 человек и 4 мессенджера);
* 31 рёбрами (20 для связи людей и мессенджеров, 9 для обозначения дружеской связи и 2 для обозначения родительской связи).

Итоговый граф изображен на Рисунке 4.

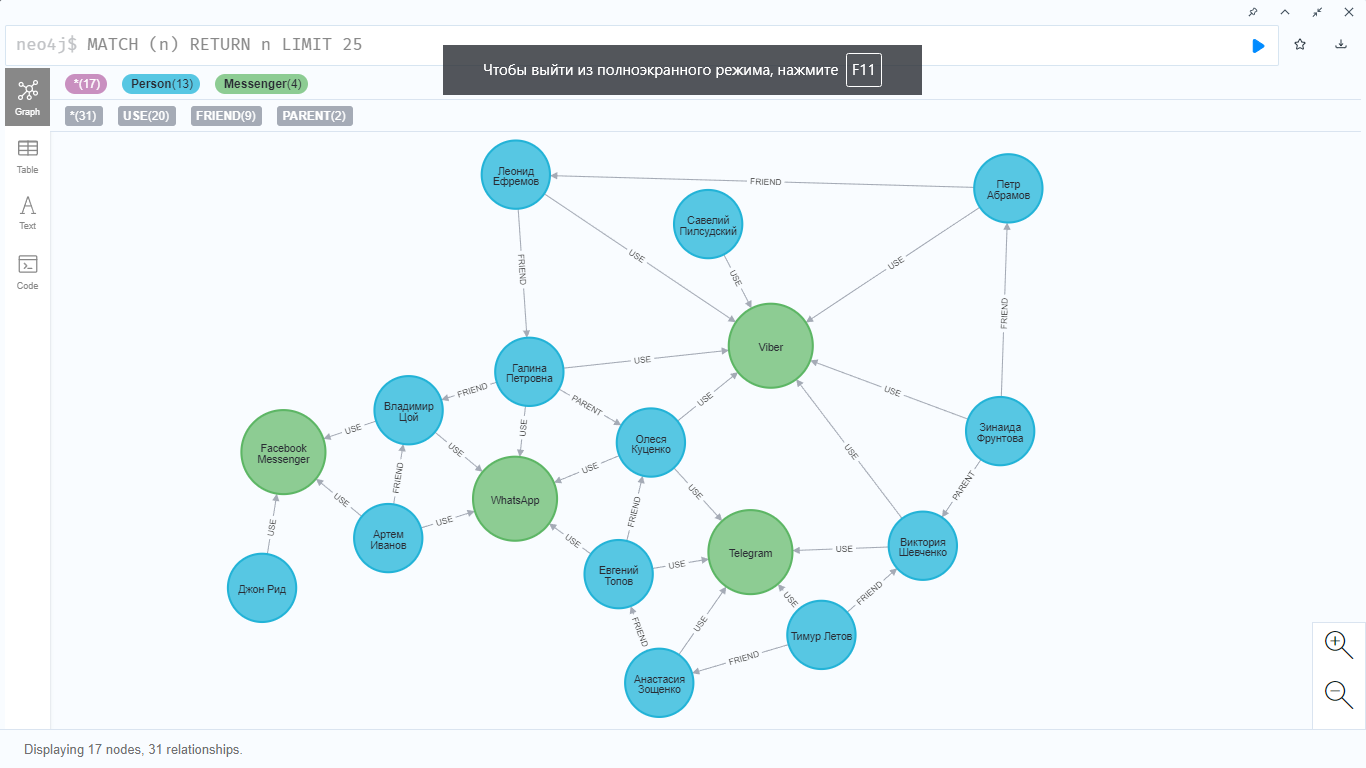


Рисунок 4 – Итоговый граф

Первичный визуальный осмотр графа показывает разделение людей на кластеры. К примеру, есть кластер людей, которые пользуются Viber, кластер людей, которые пользуются Facebook Messenger, кластер людей, которые пользуются Telegram. Стоит отметить, что все пользователи WhatsApp так или иначе пользуются иными мессенджерами.

Оценим также граф с характеристикой возраста (Рисунок 5).

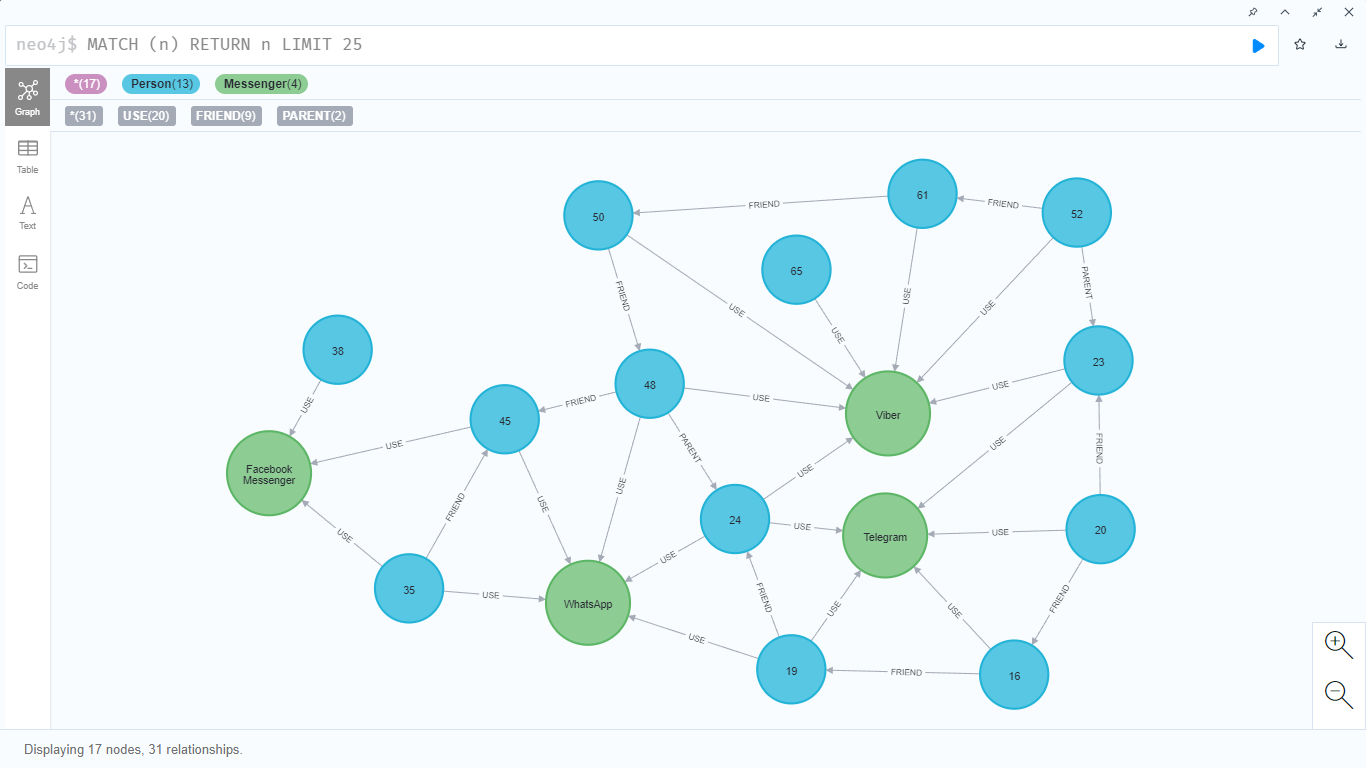


Рисунок 5 – Итоговый граф с атрибутом возраста

Из Рисунка 5 можно сделать следующие выводы:

1. Мессенджером Viber пользуются люди самого старшего возраста.
2. Мессенджером Facebook Messenger пользуются мало людей, т.к. это преимущественно иностранный мессенджер. Помимо этого, можно сказать, что данным мессенджером пользуются люди зрелого возраста (предположительно, деловые люди, которые ведут свой бизнес и которым необходима связь с иностранцами).
3. Мессенджером Telegram пользуются преимущественно молодые люди.
4. Мессенджером WhatsApp пользуются люди разных возрастов.
5. Зачастую дети регистрируются в мессенджерах, которыми пользуются их родители, чтобы поддерживать с ними связь.

Выводы, полученные при помощи визуального анализа, можно проверить с помощью высчитывания среднего возраста по каждому мессенджеру:

MATCH (n)<-[]->(a:Messenger) WHERE a.name = 'Viber'

RETURN avg(n.age)

Результат представлен в Таблице 1.

Таблица 1

Мессенджеры и средний возраст их пользователей

|  |  |  |
| --- | --- | --- |
| № | Мессенджер | Средний возраст |
| 1 | Telegram | 20.4 |
| 2 | Viber | 46 |
| 3 | WhatsApp | 34.2 |
| 4 | Facebook Messenger | 39.3 |

Полученные данные действительно подтверждают ранее выдвинутые выводы. На самом деле, мессенджером Viber пользуются люди более старшего возраста, однако, из-за того, что их дети регистрируются и пользуются данным мессенджером, возраст значительно снижается.

# ЗАКЛЮЧЕНИЕ

Выполнение данной лабораторной работы позволило ознакомится с одной из графовых СУБД – Neo4j. Как и другие представители СУБД Neo4j обладает как преимуществами, так и недостатками. Отметим следующие преимущества Neo4j:

* Гибкая модель данных — Neo4j предоставляет гибкую простую и вместе с тем мощную модель данных, которую можно легко изменять в зависимости от приложений и отраслей.
* Анализ в реальном времени — Neo4j предоставляет результаты на основе данных в реальном времени.
* Высокая доступность — Neo4j отлично доступен для крупных корпоративных приложений реального времени с транзакционными гарантиями.
* Связанные и полуструктурированные данные. Используя Neo4j, вы можете легко представлять связанные и полуструктурированные данные.
* Простой поиск — Используя Neo4j, вы можете не только представлять, но и легко извлекать (перемещаться / перемещаться) связанные данные быстрее по сравнению с другими базами данных.
* Язык запросов Cypher — Neo4j предоставляет декларативный язык запросов для визуального представления графика с использованием синтаксиса ascii-art. Команды этого языка в удобочитаемом формате и очень просты в освоении.
* Нет объединений — используя Neo4j, он НЕ требует сложных объединений для извлечения связанных / связанных данных, так как очень легко получить сведения о соседнем узле или взаимосвязи без объединений или индексов.
* Гибкая модель данных — Neo4j предоставляет гибкую простую и вместе с тем мощную модель данных, которую можно легко изменять в зависимости от приложений и отраслей.
* Анализ в реальном времени — Neo4j предоставляет результаты на основе данных в реальном времени.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Neo4j – Википедия – [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Neo4j
2. Neo4j Краткое руководство – CoderLessons – [Электронный ресурс]. Режим доступа: https://coderlessons.com/tutorials/bazy-dannykh/uznaite-neo4j/neo4j-kratkoe-rukovodstvo
3. Neo4j Desktop Download – Официальный сайт Neo4j – [Электронный ресурс]. Режим доступа: https://neo4j.com/download-center/#desktop