

**Цель работы**

Получить и отработать навыки работы с СУБД Neo4j.

**Теоретическое введение**

Neo4j — графовая система управления базами данных с открытым исходным кодом, реализованная на Java. 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. Расширить программный интерфейс можно как с помощью серверных плагинов, так и с помощью неуправляемых расширений (unmanaged extensions); плагины могут добавлять новые ресурсы к REST-интерфейсу для конечных пользователей, а расширения позволяют получить полный контроль над программным интерфейсом, и могут содержать произвольный код, поэтому их следует использовать с осторожностью.

В СУБД используется собственный язык запросов — Cypher.

Язык запросов Cypher — самый распространенный язык запросов к графовым базам данных, что обусловлено его использованием в СУБД Neo4j. Cypher является декларативным языком и позволяет создавать, обновлять и удалять вершины, ребра, метки и свойства, а также управлять индексами и ограничениями. Для извлечения данных из хранилища используется запрос, содержащий шаблон фильтрации, позволяющий получать:

* (n)-->(m) — все направленные ребра из вершины n в вершину m;
* (n:Person) — все вершины с меткой Person;
* (n:Person:Russian) — все вершины, имеющие обе метки Person и Russian;
* (n:Person {name:{value}}) — все вершины с меткой Person и отфильтрованные по дополнительному свойству;
* (n:Person)-->(m) — ребра между вершинами n с меткой Person и m;
* (n)--(m) — все ненаправленные ребра между вершинами n и m [1].

**Установка Neo4j на Ubuntu.**

Для того, чтобы установить Neo4j, необходимо выполнить следующие команды:

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add –

sudo add-apt-repository "deb https://debian.neo4j.com stable 4.1"

sudo apt install neo4j

sudo systemctl enable neo4j.service

sudo systemctl status neo4j.service

cypher-shell

Далее необходимо скачать и установить файл по с официального сайта разработчика <https://neo4j.com/>.

**Запуск сервера Neo4j:**



Рисунок 1 – Запущенный сервер Neo4j.

Для работы используем встроенное приложение для работы с Neo4j.  получить доступ к Neo4j можно используя URL-адрес http: //localhost: 7474/.

**Добавление вершин:**

Создадим двух работников и место их работы, для этого следует воспользоваться ключевым словом CREATE:

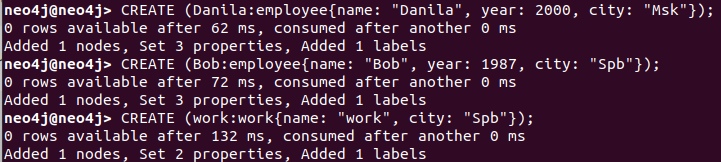


Рисунок 2 – Создание вершин графа.

После этого граф будет иметь следующий вид:

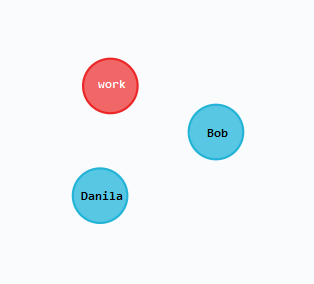


Рисунок 3 – Состояние графа.

**Создание связей:**

Добавим между узлами этого графа зависимости. Пусть работники будут иметь зависимость «worksIn» к месту работы, показывая, что они работают именно там. Введем следующую команду:

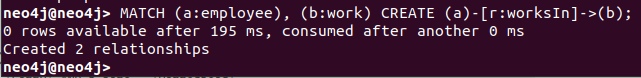


Рисунок 4 – Создание связей между вершинами.

После этого граф будет иметь следующий вид:

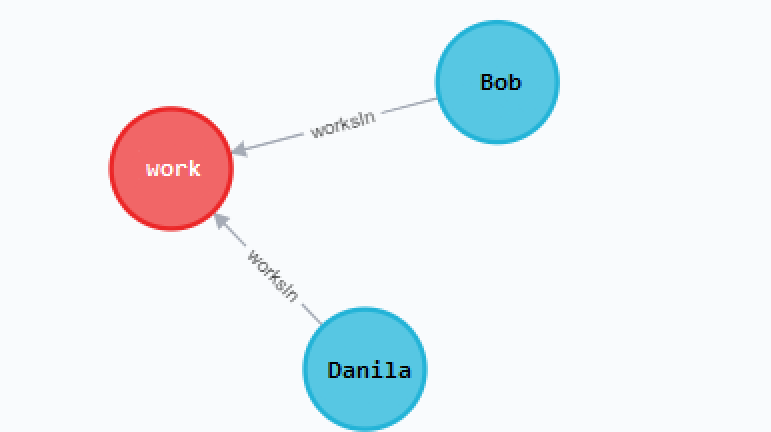


Рисунок 5 – Граф со связями между вершинами.

**Удаление вершины:**

Для удаления следует воспользоваться ключевым словом DELETE. Но поскольку наши работники имеют зависимости, необходимо добавить ключевое слово DETACH для их удаления.

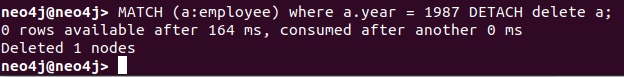


Рисунок 6 – Удаление вершины по заданному условию.

После этого граф будет иметь следующий вид:

****

Рисунок 7 – Граф после удаления одной из вершин.

**Обновление данных:**

Для обновления данных вершины следует воспользоваться ключевым словом SET.

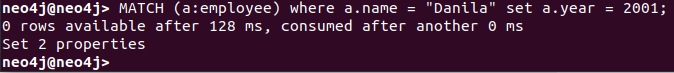


Рисунок 8 – Изменения данных работника.

**Выводы**

В ходе данной работы я получил навыки работы с СУБД Neo4j и успешно отработал типовые задачи.

**Библиографический список**

1. How To Install and Configure Neo4j on Ubuntu 20.04 [Электронный ресурс]. URL: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04>
2. The Neo4j Cypher Manual v4.1 [Электронный ресурс]. URL: https://neo4j.com/docs/cypher-manual/current/