

Обзор возможностей СУБД Neo4j

Пальмов Сергей Вадимович, кандидат технических наук, доцент
Поволжский государственный университет телекоммуникаций и информатики;
Самарский государственный технический университет (г. Самара)
Мячина Анна Сергеевна, студент 2 курса,
факультет «Информационные системы и технологии»
Поволжский государственный университет телекоммуникаций и информатики (г. Самара)

На сегодняшний день большая часть данных существует в форме отношений между различными объектами. Эти отношения более ценны, чем сами данные. В данной статье описывается один из типов нереляционных моделей: модель данных на основе графов. К последней относятся такие инструменты, как ArangoDB, HyperGraphDB, OrientDB и Neo4j. Система управления базами данных Neo4j, имеющая интегрированный язык запросов Cypher, рассматривается более подробно. Описаны преимущества и недостатки графической базы данных Neo4j. Приведен пример создания простой модели данных.

Ключевые слова: база данных, СУБД, NoSQL, граф.

DBMS Neo4j Review

Palmov Sergey Vadimovich, Candidate of Science Engineering, Associate Professor
Povolzhskiy State University of Telecommunications and Informatics,
Samara State Technical University (Samara)
Myachina Anna Sergeevna, 2nd year student, faculty "Information Systems and Technologies"
Povolzhskiy State University of Telecommunications and Informatics (Samara)

Nowadays, data mostly exists in the form of relationships between different objects. These relationships are more valuable than the data itself. The paper describes one of the types of non-relational models: a graph-based data model. It contains such instruments as ArangoDB, HyperGraphDB, OrientDB and Neo4j. The Neo4j database management system, which has the integrated query language Cypher, is considered in more detail. The advantages and disadvantages of the Neo4j graphic database are described. An example of creating a simple data model is given.

Keywords: data base, DBMS, NoSQL, graph.

DOI: 10.5281/zenodo.3888099

Для упрощения и организации первых моделей данных использовались системы управления реляционными базами данных (СУБД), занимающие значительное место среди инструментов хранения данных и обеспечивают лучшее сочетание простоты, стабильности, гибкости, производительности, масштабируемости и совместимости. Они используются для хранения структурированных данных одного типа, которые не хранят взаимосвязи между данными и представлены в виде двумерных таблиц (строк и столбцов) [1, с.103].

Необходимость разрабатывать современные приложения и обрабатывать все более крупные объемы данных является одним из факторов, влияющих на внедрение нового класса нереляционных баз данных NoSQL (Not only SQL), которые пытаются решить проблемы масштабируемости и доступности за счет атомарности и согласованности данных. NoSQL оптимизирована для интуитивно понятной разработки, позволяющая расширить возможности в тех областях, где реляционная модель и SQL недостаточно гибки и масштабируемы [2, с.30]. Гибкость моделей обеспечивает более быструю разработку и пошаговую реализацию, что хорошо подходит для частично структурированных и неструктурированных данных. Одним из наиболее популярных и актуальных подвидов нереляционного хранения данных является графовая БД.

К методам реализации нереляционных графовых БД относятся СУБД, которые реализованы на разных языках программирования:

- ArangoDB – масштабируемая и полностью управляемая графическая база данных, хранилище документов и поисковая система. ArangoDB Oasis – это управляемый сервис для ArangoDB. Собственный язык AQL, позволяющий получать доступ к одним и тем же данным с использованием различных шаблонов доступа, таких как обходы, объединения, поиск, ранжирование, геопространственные данные или любые другие комбинации [7];

- HyperGraphDB – это универсальный механизм хранения данных с открытым исходным кодом, основанный на мощном формализме управления знаниями, известном как направленные мультиграфы (способ запросов – Java или P2P) [7];

- OrientDB – является ведущей в мире мультимодельной базой данных NoSQL с открытым исходным кодом, которая обеспечивает работу самых мощных современных компаний и приложений (работа построена на языке SQL) [7];

- Neo4J – это хорошо масштабируемая база данных собственных графов, созданная специально для использования не только данных, но и взаимосвязей между данными (реализована на языке Java) [3]. В отличие от других баз данных, Neo4j объединяет дан-

ные в том виде, в котором они хранятся, что позволяет им проходить через соединения в более быстром порядке.

Целью статьи является изучение графовой модели данных на примере СУБД Neo4j и рассмотрение практического примера ее применения. На сегодняшний день Neo4j является самой популярной графовой БД, в первую очередь благодаря тому, что бесплатная версия предлагает все необходимые инструменты для полноценной работы с графами [6, с.56]. Графовая платформа Neo4j построена на базе собственной графовой базы данных Neo4j. Собственная графовая база данных Neo4j поддерживает транзакционные приложения и графическую аналитику [4, с.63]. Любой запрос Cypher, который изменяет граф данных представляет собой транзакцию и всегда либо выполнится полностью, либо не выполнится вовсе. Сохраняет транзакции ACID для полностью последовательных и надежных данных, идеально подходит для постоянно работающих глобальных корпоративных приложений.

К преимуществам Neo4j относятся:

- гибкость – предоставляет гибкую и простую модель данных, которую можно легко модифицировать в соответствии с приложениями и отраслями;
- масштабируемость – масштабирование с использованием распределенных аппаратных кластеров;
- анализ в реальном времени – обработка графиков обеспечивает нулевую задержку и производительность в режиме реального времени независимо от количества связей;
- доступность – доступен для больших приложений;
- простой поиск – возможность легко представить и извлекать связанные данные;
- язык запросов Cypher – собственный язык графовых запросов, который обеспечивает наиболее эффективный и выразительный способ описания запросов отношений [5, с.28].

Вместе со всеми достоинствами графической БД Neo4j также существуют и недостатки, но их заметно меньше:

- занимает достаточно много места на диске, по сравнению с реляционными СУБД;
- при выполнении простых запросов производительность ниже, чем в реляционных базах;
- имеет специфичную модель представления данных, подходящая не для всех предметных областей.

Графовая БД может хранить любые данные, используя несколько простых концепций:

- вершина (узел) – является основным элементом данных. Может быть соединен с другими узлами через отношения;
- свойство – это пара ключ-значение для описания узлов и отношений графа. Свойство можно индексировать и ограничивать;
- отношение (ребро) – являются важным строительным блоком графической базы данных, соединяющие два узла между собой;
- метка – связывает общее имя с набором узлов или отношений. Узел или отношение могут содержать одну или несколько меток.

Хранилище типа «ключ-значение» – простейшее хранилище данных NoSQL с точки зрения интерфейса прикладного программирования. Клиент может либо получить значение по ключу, либо записать значение по ключу, либо удалить ключ из хранилища данных. Значение представляет собой двоичный объект данных, который записан в хранилище без детализации его внутренней структуры [2, с.103].

Интерфейс Neo4j состоит из трех основных кнопок – Start Learning, Write Code и Monitor (Рис.1).

Start Learning представляет собой интерактивный учебник для начинающих, где можно ознакомиться с основными понятиями.

Write Code демонстрирует общие шаблоны запросов.

Monitor отображена текущая информация о базе данных. Состоит из таких данных, как Store Size, ID Allocation, Page Cache, Transactions, Databases. Для их отображения также можно ввести команду «: sysinfo» в командной строке.

В верхней части окна располагается строка редактора. Начиная набор команд с двоеточия, можно увидеть список всех доступных команд с их кратким описанием (Таблица 1).

Продемонстрируем работу Neo4j на простом примере. Создается БД предметной области «киноиндустрия». Исходными данными будет информация об актерах и фильмах, в которых они снимались. Следовательно, БД будет содержать два типа вершин (узлов): «Person» и «Movie». Чтобы создать узел, нужно ввести команду, состоящую из CREATE (создание; задает свойства узла) и RETURN (аналог SELECT; описывает результат, возвращаемый в результате выполнения команды).

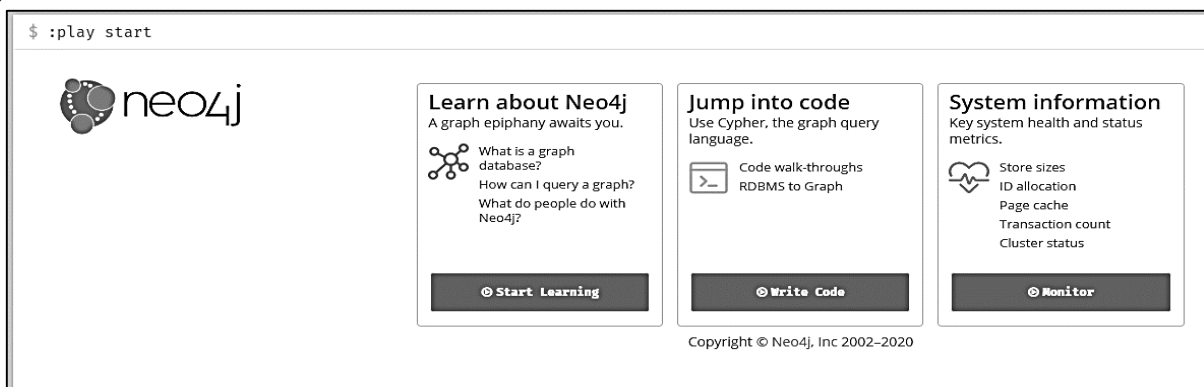


Рис.1. Интерфейс Neo4j

Таблица 1. Доступные команды в Neo4j

Команда	Описание
:get	Send HTTP GET to Neo4j's REST interface (отправить HTTP GET в интерфейс REST Neo4j)
:put	Send HTTP PUT to Neo4j's REST interface (отправить HTTP PUT в интерфейс REST Neo4j)
:help	Learn about other topics (узнать о других темах)
:play	Loads a mini-deck with either guide material or sample (ознакомиться с примерами работы с графами)
:post	Send HTTP POST to Neo4j's REST interface (отправить HTTP POST в интерфейс REST Neo4j)
:head	Send HTTP HEAD to Neo4j's REST interface (отправить HTTP HEAD на интерфейс REST Neo4j)
:param	Define a parametr (определить параметр)
:clear	Remove all frames from the stream (удалить все кадры из потока)
:server	Manage the connection to Neo4j (управление подключением к Neo4j)
:params	Show you a list of all your current parameters (показать список всех текущих параметров)
:schema	Display indexes and constraints (показать индексы и ограничения)
:delete	Send HTTP DELETE to Neo4j's REST interface (отправить HTTP DELETE на интерфейс REST Neo4j)
:queries	List your servers and clusters running queries (список серверов и кластеров, выполняющих запросы)

Таким образом, запрос, например, будет выглядеть так: «CREATE (n: Person {name: `Kevin Bacon`}) RETURN n». Кнопка «Graph» позволяет увидеть узлы в графическом виде. Вывод запроса показан на Рис.2.

Добавляются еще узлы с другими актерами по типу запроса на Рис.2 с изменением только фамилии и имени актера. Далее выводятся в новом запросе все узлы типа «Person»: «MATCH (people:Person) RETURN people» (Рис.3).

Далее добавляются узлы с названиями фильмов типа «Movie» и так же выводятся вместе: «MATCH (films:Movie) RETURN films» (Рис.4). Также можно вывести узлы по отдельности типа «Person» и «Movie»

нажав на кнопку с соответствующим названием по типу, которые находятся в разделе «Node Labels».

Далее соединяются узлы типа «Person» и «Movie» вместе: «MATCH (people:Person), (films:Movie) RETURN people, films». Также можно вывести все узлы при помощи кнопки «*», которая находится на панели «Database Information». Добавляются отношения между различными узлами. Например, запрос на создание отношения между актером «Tom Cruise» и фильмом «A Few Good Men» с указанием роли в фильме будет выглядеть так: «(TomCruise)-[:ACTED_IN {roles: [Lt. Daniel Kaffee]}]->(AFewGoodMen)» (Рис.5).

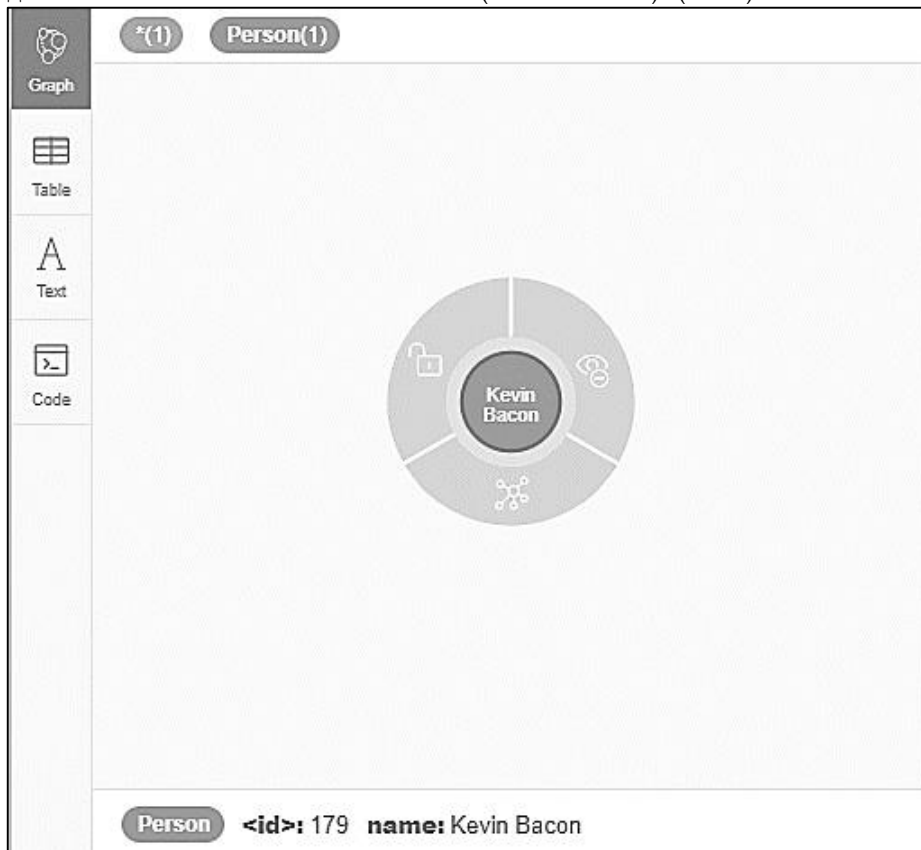


Рис.2. Вывод запроса на создание узла

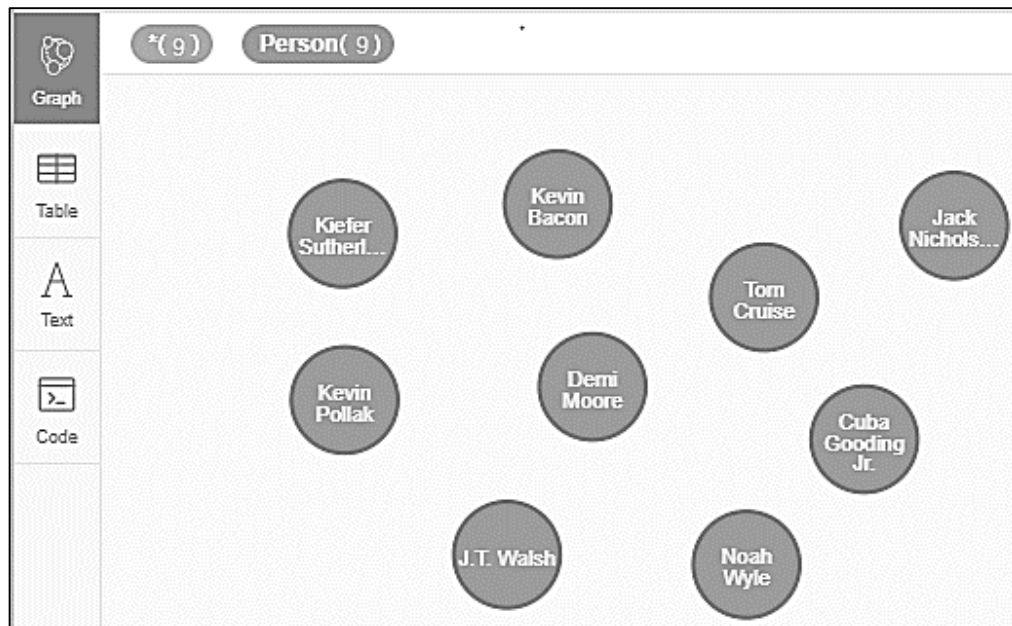


Рис.3. Вывод всех узлов «Person»

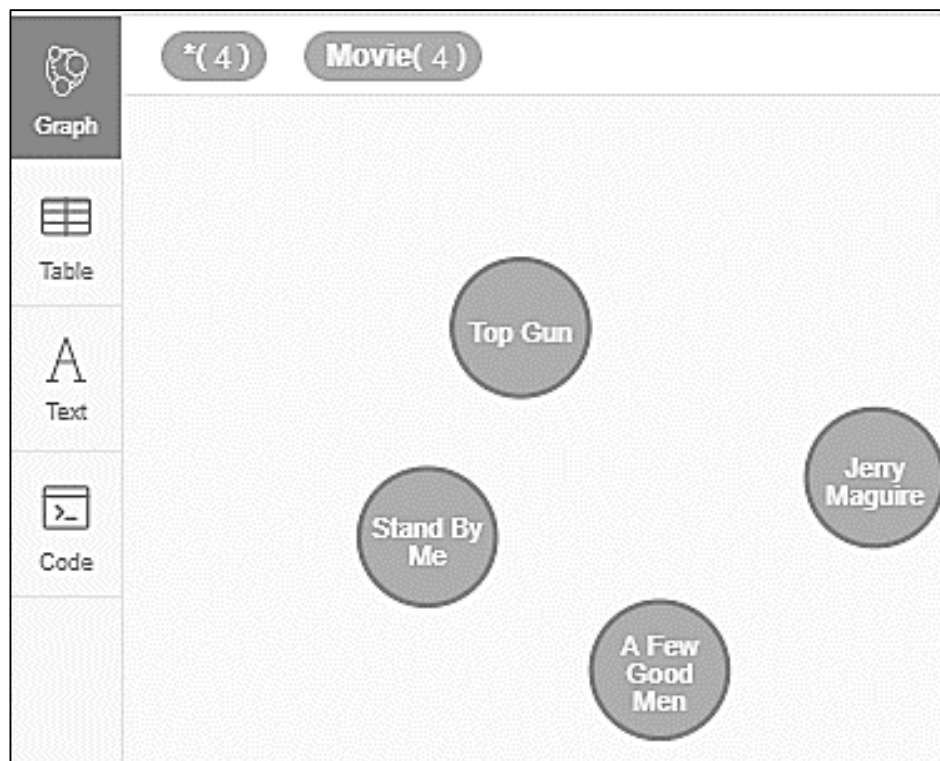


Рис.4. Вывод всех узлов «Movie»

Каждый актер снялся в каком-то фильме, а некоторые актеры даже в нескольких фильмах (несколько отношений). Имея данные о актерах и взаимосвязях между ними, можно узнать, кто является коллегой по фильму конкретного актера.

Для отображения результата запроса можно экспортировать его в файл в четырех различных форматах: CSV, JSON, PNG и SVG.

CSV – текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми (Рис.6).

JSON – текстовый формат обмена данными, основанный на JavaScript.

PNG – растровый формат хранения графической информации.

SVG – графический формат файлов, применяемый в веб-дизайне для векторных изображений.

Интерес к графическим базам данных растет с каждым днем в связи с тем, что такая система представления данных оказалась естественной и востребованной в современном мире. После изучения и применения на практике графовой БД Neo4j можно считать, что последняя эффективно справляется с возложенными на нее задачами. Она позволяет получать более быстрые и эффективные решения, чем при использовании традиционных реляционных СУБД.

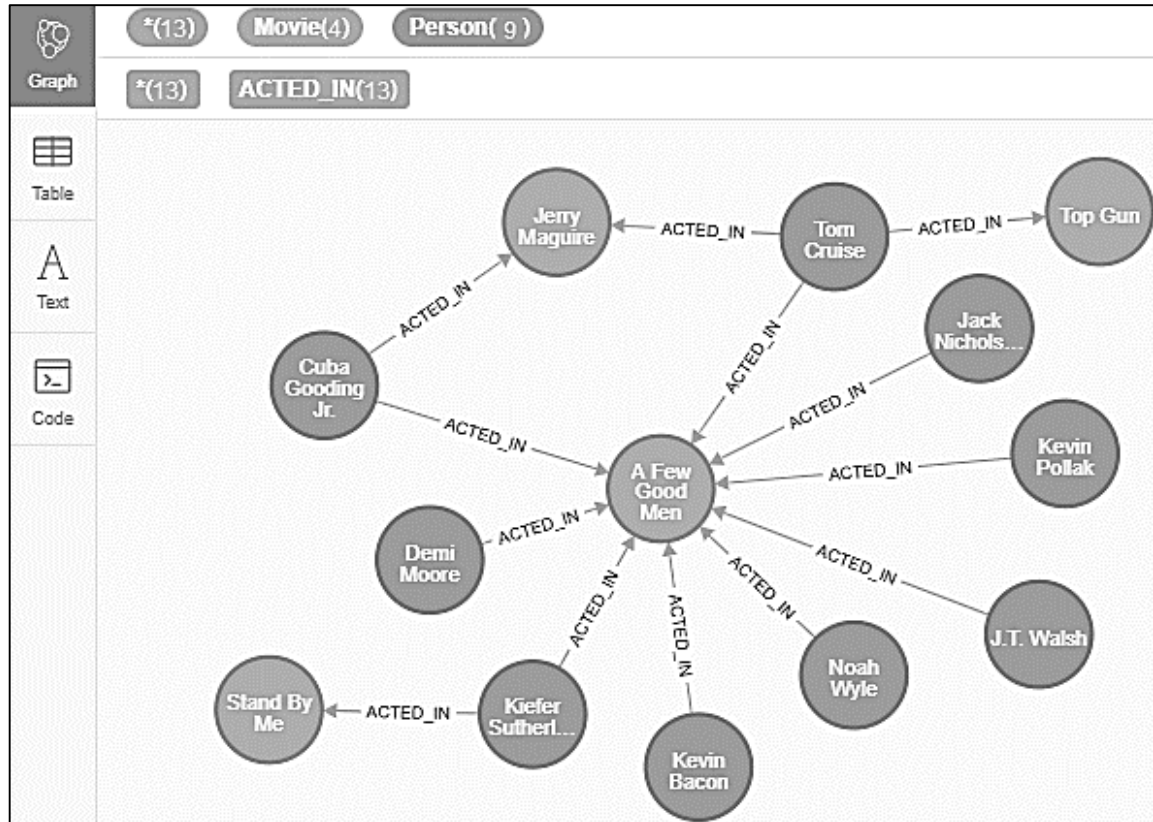


Рис.5. Вывод всех узлов «Person» и «Movie»

	A	B	C	D	E
1	n				
2		{title:Jerry Maguire}			
3		{title:A Few Good Men}			
4		{title:Top Gun}			
5		{title:Stand By Me}			
6		{name:Tom Cruise}			
7		{name:Jack Nicholson}			
8		{name:Demi Moore}			
9		{name:Kevin Bacon}			
10		{name:Kiefer Sutherland}			
11		{name:Noah Wyle}			
12		{name:Cuba Gooding Jr.}			
13		{name:Kevin Pollak}			
14		{name:J.T. Walsh}			

Рис.6. Экспорт БД в формате CSV

Литература:

1. Дейт К. Дж. Введение в системы баз данных. М.: Вильямс, 2005. – 1315 с.
2. М. Фаулер, Дж. Садаладж. NoSQL: новая методология разработки нереляционных баз данных. – М.: «Вильямс», 2013. – 192 с.
3. Neo4j [Электронный ресурс]. URL: <https://neo4j.com/developer/neo4j-browser/> (дата обращения: 10.05.20).
4. Я. Робинсон, Дж. Вебер, Э. Эйфрем. Графовые базы данных. Новые возможности для работы. – М.: «ДМК Пресс», 2016. – 256 с.
5. I. Robinson, J. Webber, E. Eifrem. Graph Databases: New Opportunities for Connected Data. – «O'Reilly», 2015. – 211 с.
6. М. Фаулер, Дж. Садаладж. NoSQL: новая методология разработки нереляционных баз данных. – М.: «Вильямс», 2013. – 148 с.
7. Графовые БД [Электронный ресурс]. URL: <https://oracle-patches.com/db/3919> (дата обращения: 12.05.20).