

¹ ассистент кафедры алгебры, ГВУЗ «ДГПУ»

e-mail: rubannn@gmail.com

ГРАФОВЫЕ БАЗЫ ДАННЫХ

Рассмотрены отличительные особенности использования баз данных на основе графов, их преимущества относительно реляционных баз при решении некоторых задач.

Ключові слова: *графовые базы данных, граф, NoSQL*

Введение

Графовая база данных – это база данных, предназначенная для хранения данных в виде графа. Т.е. это хранения данных по вершинам и ребрам. По определению, графовая база данных – это любая структура для хранения, где взаимосвязанные элементы соединены без применения индекса. Смежные по структуре элементы доступны при разыменовании физического показателя. Существует несколько типов графов, которые могут хранить: от «однотипного» ненаправленного графа до гиперграфа, включая свойственные подграфы.

Таким образом, база данных в виде графа должна соответствовать следующим критериям:

- хранение оптимизировано для данных, представленных в виде графа, с обеспечением хранения данных по вершинам и ребрам;
- хранение оптимизировано для просмотра графа без использования индекса. Графовая база данных оптимизирована для запросов используя близость данных, начиная от одного или нескольких корневых узлов, а не использование глобальных запросов;
- гибкая модель данных для некоторых решений: нет необходимости в определении типов данных для вершин и ребер в отличие от более ограниченной табличной модели реляционной базы данных;
- встроенная API с точкой входа для классических алгоритмов теории графов (кратчайший путь, алгоритм Дейкстры, A^* и др.)

Основная часть

Движение *NoSQL* (англ. not only SQL, не только SQL) стало популярным в последние несколько лет, в особенности благодаря тому, что *NoSQL* может решить ряд вопросов, по которым реляционные базы данных значительно уступают:

- доступность, для обработки больших наборов данных, и их разделение (реляционные базы данных уделяют внимание корректности, доступности и устойчивости к сбоям сети (теорема CAP));
- гибкость схемы;
- трудности в моделировании и обработке сложных структур, таких как деревья, графы, или слишком большого количества связей.

Графовые базы данных чаще всего работают используя такие пункты:

- обработка взаимосвязанных данных;
- легкое управление сложными и гибкими моделями данных;
- предложение исключительного функционирования для локального чтения путем обхода графа.

База данных на основе графа подходит для изучения данных, структурированных как граф (или производных, таких как дерево), особенно когда взаимосвязь между элементами значительна. Идеальный вариант использования запроса – начать с одной или нескольких вершин и обойти граф. Всегда есть возможность дать более широкие запросы, например, *.findAll* (найти все элементы данного типа), но в данном случае должен использоваться механизм индексации. Такой механизм может быть разработан специально для графа (супервершины — часть индексированных данных) или же использовать другое известное решение (например Apache Lucene).

В отличие от этого, реляционные базы данных хорошо подходят для схожих с *.findAll* запросов благодаря внутренней структуре таблиц. Группировки набора данных обычно вполне достаточно.

Несмотря на название, реляционные базы данных менее подходят для изучения взаимосвязей. Обычно применяются индексы, либо внешние ключи. В базах данных на основе графа обход данных происходит, используя физические указатели, в то время как внешние ключи являются логическими указателями.

Рассмотрим эти различия на примере. В базе будем хранить данные о сотруднике ВУЗ, наименовании кафедры, и педагогический стаж. Например, необходимо найти людей, которые работают на факультете математики.

При помощи реляционной модели мы можем выполнить следующий запрос, который вероятно потребует поиска 3 индексов, соответствующих внешним ключам в модели.

```
Select Person.Name from Person, Faculty, WorksIn
where Faculty.Name='математики'
and WorksIn.FacultyId = Faculty.Id
and WorksIn.PersonId = Person.Id
```

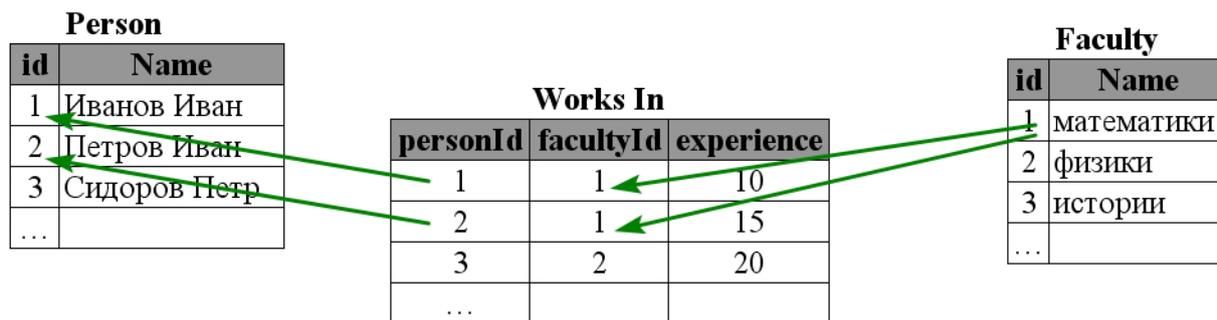


Рис. 1: Реляционная схема.

В случае графа, запрос потребует поиска 1 индекса, затем поиск взаимосвязи путем прямого разыменования физических указателей.

Находим узел у которого Faculty.Name='математики' (Faculty1) и отбираем все узлы которые связаны отношением Works.

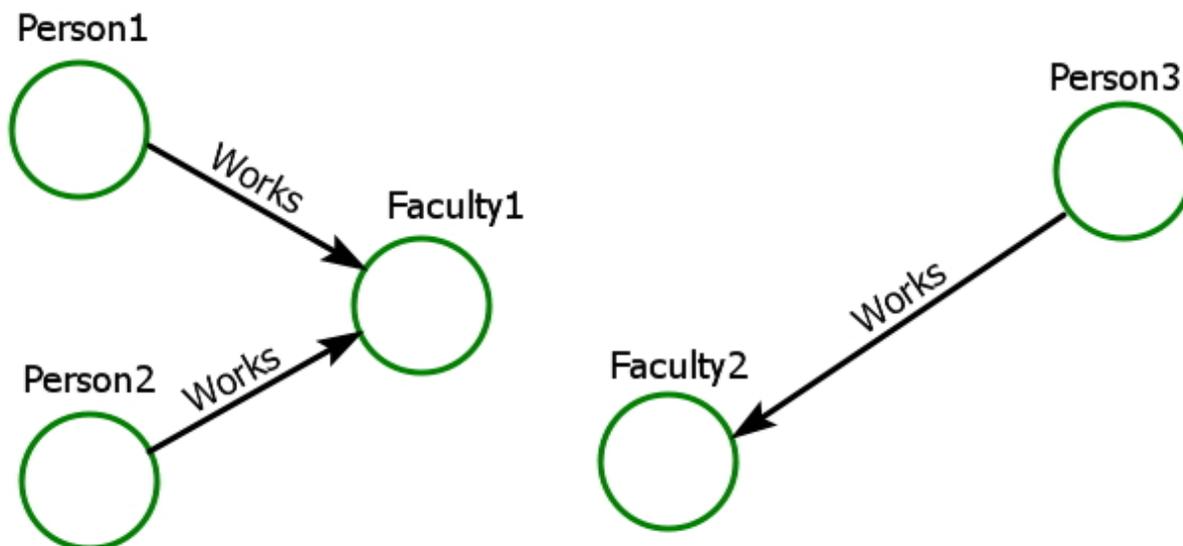


Рис. 2: Графовая схема.

На этом примере можно хорошо рассмотреть ситуацию, когда производительность графовой базы данных будет выше, чем реляционной базы данных.

Разница в производительности будет еще более значительна при увеличении объема данных, потому что кроме первого индекса поиска, для обхода нужно найти начальный узел(ы), запрос будет фактически выполняться за постоянное время. А в реляционной базе данных, при разименовании каждого индекса которое будет происходить за время $O(\log_2 N)$, если для индексов использовать B -деревья (N — число записей в таблице).

Сравнение работы графовых баз данных и других типов баз данных — сложная задача из-за их разницы по структуре и предназначению. Для запросов, графовые базы данных будут более эффективны в отличие от реляционных базы данных, если модель доступа к данным была специально разработана для обхода графа. Различие будет еще более значительным, если имеет значение глубина обхода, или если она заранее не известна (так как задачу с использованием эквивалентного SQL-запроса сложно оптимизировать).

Выводы

Таким образом при разработке структуры базы данных необходимо рассматривать варианты при которых данная структура будет опираться на свойства графа и если данные предполагают использование графовых схем, то их использование может существенно сократить доступ к данным, а на больших объемах значительно уменьшить время обработки информации. Особенно большой интерес использования графовых баз данных возникает при разработки всевозможных социальных сетей.

Литература

- [1] *Ian Robinson* Graph Databases / Ian Robinson, Jim Webber, Emil Eifrem. — O'Reilly Media, 2013. — 178 p.
- [2] *Jonas Partner* Neo4j in Action / Jonas Partner, Aleksa Vukotic, Nicki Watt. — Manning Publications, 2012. — 350 p.
- [3] *Shashank Tiwari* Professional NoSQL / Tiwari Shashank. — Wrox, 2011. — 408 p.
- [4] *Эрик Редмонд* Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Эрик Редмонд, Джим Р. Уилсон; [Под ред. Жаклин Картер]; [пер. с англ. А.А. Слинкин]. — М.: ДМК Премм, 2013. — 384 с.: ил. алгоритмы / Пер. с англ. — Москва: Вильямс, 2007. — 400 с.