

ОТОБРАЖЕНИЕ РЕЛЯЦИОННЫХ ОТНОШЕНИЙ В СВЯЗИ В СЕМАНТИЧЕСКОЙ МОДЕЛИ

Sajin Iuliana, mrd. TI-151M; Ciorbă Dumitru, dr. conf.univ. inter.;
Poștaru Andrei, drd. lector superior

Universitatea Tehnică a Moldovei

Резюме: Графовые модели обладают значительными достоинствами в представлении связей между данными. Однако их понимание в контексте анализа информации при неполных данных и поиска новых соединений требует более абстрактный семантический подход.

Ключевые слова: графовая база данных, семантическая модель, реляционная база данных, онтология, классы, связи, предикаты.

1. Реляционные и не реляционные хранилища данных

В настоящее время решение различных задач во многих областях человеческой деятельности требует обработки большого количества поступающих данных.

С 1980-х годов реляционные системы управления базами данных (СУБД) стали занимать доминирующее положение среди средств хранения данных.

Сегодня ситуация несколько иная. Появившиеся в последние годы так называемые NoSQL (Not only SQL, не только SQL) хранилища реализуют модели данных, имеющие существенные отличия от традиционной реляционной модели. Основная их цель — расширить возможности баз данных (БД) в тех областях, где реляционная модель и SQL недостаточно гибки, и не вытеснить их там, где они справляются со своими задачами. [1]

Реляционные базы данных отлично справляются с хранением данных, но мало пригодны в области анализа данных. Несмотря на то что реляционные базы данных могут реализовывать связи с помощью внешних ключей, операции соединения требуют навигации, которая может оказаться затратной. Следовательно, в моделях данных с большим количеством связей производительность упадет. [2]

Одной из технологий призванных решить данную проблему стали OLAP-кубы. Благодаря детальному структурированию информации OLAP-кубы позволяют оперативно осуществлять анализ данных и формировать отчеты в различных разрезах и с произвольной глубиной детализации. OLAP кубы обеспечивают высокую скорость обработки запросов, однако они не предоставляют никакой информации о связях между данными, оперируя лишь частью данных, они позволяют видеть данные лишь в разобранном виде, так же кубы достаточно громоздки, занимают много места и трудно модифицируемы.

OLAP кубы предоставляют лишь некоторую суммарную информацию о данных и не подходят для более глубокого и детального анализа данных с учетом их взаимосвязей. В данном случае лучше использовать графовые базы данных.

Графовые базы данных идеально подходят для хранения любых данных, связанных со сложными отношениями, например, социальных сетей, товарных предпочтений или правил приема на работу.

Фундаментальная модель данных графовых баз очень простая: узлы, соединенные ребрами (которые называют также дугами). В таких хранилищах обход узлов требует очень небольших затрат. В основном это объясняется тем, что графовые базы данных переносят большую часть работы, связанной с навигацией по связям, с момента запроса на момент вставки. Это естественно оправдывает себя в ситуациях, когда производительность запроса важнее скорости вставки. [2]

Преимуществами графовой базы данных можно воспользоваться, даже если все данные хранятся в реляционной схеме. В данной статье будет рассмотрен один из шагов алгоритма, обеспечивающий конвертирование реляционной схемы в графовую модель.

2. Семантическая модель данных

Для записи семантических моделей существует несколько языков, основными из которых являются RDF и OWL. RDF позволяет записывать простейшие факты об объектах, классах и свойствах. OWL - язык написания онтологий для семантических моделей и выражает логические аксиомы, описывающих сложные взаимоотношения классов и свойств. [3]

Обычно под онтологией понимается структура в виде графов или сети, состоящая из:

- 1) набора концептов (понятий, классов);
- 2) набора бинарных связей (отношений) между концептами;
- 3) набора экземпляров классов – записей данных, соответствующих классу или отношению.

В данном случае онтология разрабатывается на основе информации о структуре реляционной базы данных. Первым шагом является извлечение классов. Вторым - установление связей между ними.

Классами в онтологии становятся сами таблицы, а атрибутами соответственно поля таблиц с сохранением информации об их типе. Отношения между классами онтологии устанавливаются на основе анализа связей между таблицами базы данных, при этом учитываются не только внешние ключи, но так же уникальные ключи, а в некоторых случаях анализируется и структура таблицы.

Далее рассматриваются основные типы связей в реляционных базах данных.

Связь 1:1

Первый тип отношений это 1:1, означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице. Связь 1:1 может отображать два аспекта отношений между сущностями: композицию и наследование.

Композиция используется, чтобы преодолеть ограничения РСУБД или с целью увеличения производительности. Данный тип связи выявляется на основе информации об уникальном и внешнем ключах. Для отображения композиции в графовой модели используется предикат has a.

При выявлении отношения наследования могут возникнуть некоторые трудности. Как парсеру понять на основе информации о ключах и структуре таблиц, что между данными двумя таблицами существует отношение наследования, а не композиции? Простой анализ данных в связанных таблицах не поможет решить данную проблему. Для определения отношения типа наследования необходимо изменить структуру родительской таблицы, добавив поле Дискриминатор, в которую будет заноситься имя таблицы-"потомка". Таким образом с каждым первичным ключом из родительской таблицы будет ассоциирована одна из таблиц потомков. В графовой модели данный тип отношений представляется в виде иерархии классов.[5]

Связь 1:n

Этот тип отношений наиболее часто встречающийся. В типе связей один ко многим одной записи первой таблицы соответствует несколько записей в другой таблице. Данный тип отношения выявляется на основе информации о внешних ключах. В графовой модели для указания данного типа отношений используется предикат "has some" .

Связь n:n

Связь многие-ко-многим создается с помощью трех таблиц. Две таблицы – “источника” и одна соединительная таблица. Кроме первичного ключа соединительная таблица состоит из двух полей, двух внешних ключей, которые ссылаются на первичные ключи таблиц. В графовой модели промежуточная таблица удаляется, если она не содержит никакой дополнительной информации кроме информации о связи таблиц, с помощью предиката "has some" связь устанавливается напрямую между двумя сущностями.

Если таблица не удаляется, то отношения устанавливаются как в случае 1:n.

На рисунке 1 представлена реляционная база данных, которая будет сконвертирована в графовую в соответствии с принципами, изложенными выше.

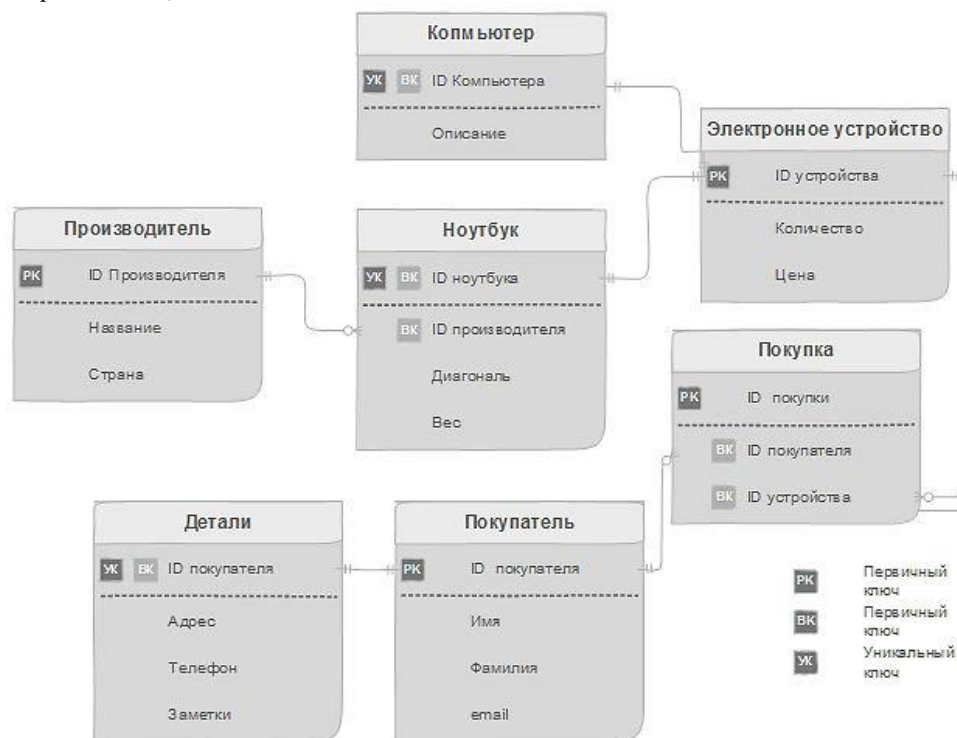


Рисунок 1 - Схема реляционной базы данных

На рисунке 2 представлена полученная в результате преобразования графовая модель.

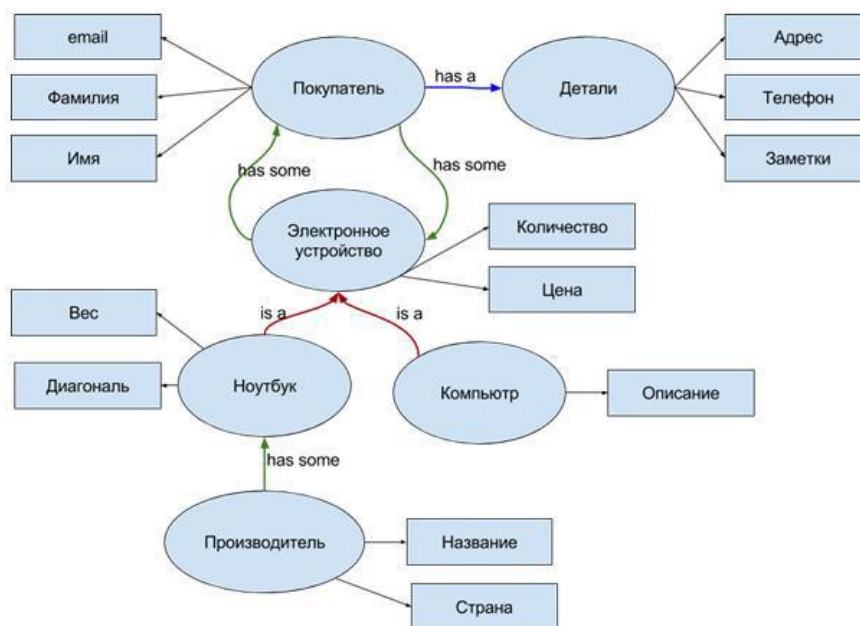


Рисунок 2 - Графовая модель

Для того чтобы обеспечить совместимость между графовой и реляционной моделью, во время преобразования создается файл с метаданными о произведенных преобразованиях. Если какие-либо структурные элементы были переименованы в графовой модели, то данная информация обязательно сохраняется, так же как информация об удаленных таблицах и о внешних ключах.

3. Заключение

Таким образом, были рассмотрены принципы, которые позволяют преобразовать реляционные связи в отношения между сущностями графовой модели. В заключении можно отметить, что несмотря на преобладание систем управления и серверов БД, построенных на реляционных принципах, графовые БД начинают постепенно занимать свое место в платформах для аналитической интеллектуальной обработки информации. Хотя первичный импорт данных и может занять определенное время, также следует учитывать и то, что это не частый процесс и данное время будет компенсировано за счет ускорения обработки информации в графовой структуре. Такие хранилища идеально подходят для задач, требующих учета и выявления большого количества взаимосвязей. Практически все задачи, в которых количество взаимосвязей между сущностями превышает количество сущностей или основной целью которых является анализ взаимосвязей, могут рассматриваться как кандидаты на решение средствами графовых систем.

Список использованных источников

1. Dominguez-Sal D., Urbon-Bayes P., Gimenez-Vano A., Gomez-Villamor S., Martinez-Bazan N., Larriba-Pey J.L. Survey of graph database performance on the HPC scalable graph analysis benchmark. Proceedings of the 2010 int. conf. on Web-age information management (WAIM'10). Berlin, Heidelberg, Springer-Verlag, 2010, pp. 37–48.
2. Ю.Р. Мухина, к.п.н., доцент кафедры «Информационные технологии и системы» ЧОУ ВПО «Южно-Уральский институт управления и экономики», Обзор NoSQL решений управления данными. Управление в информационных и кибернетических системах, 2013
3. Владислав Головков, Андрей Портнов, Виктор Чернов, RDF — инструмент для неструктурированных данных. Открытые системы, №9, 2012
4. Lina Lubyte, Sergio Tessaris, Extracting ontologies from relational databases, 2007 — In Proc. of the 20th Int. Workshop on Description Logics
5. Hibernate, Community Documentation, 2004. [Электронный ресурс]. - <https://docs.jboss.org/hibernate/orm/3.5/reference/en/html/inheritance.html>