БАЗЫ ДАННЫХ
часть II

Технологии базы данных для WWW

Даниэла Флореску, Алон Леви, Альберто Мендельсон **Технологии баз данных для World-Wide Web: обзор** // Системы управления базами данных - Издательство «Открытые Системы», #04-05, 1998 г.

БД для WWW

Популярность World-Wide Web (WWW) превратила его в главное средство распространения информации.

Основной возникающий здесь вопрос – как управлять большими объемами данных?

Новый контекст WWW вынуждает значительно расширить ранее используемые технологии.

Основная цель лекции состоит в том, чтобы классифицировать различные задачи, к решению которых применялись концепции баз данных.

БД для WWW

Сосредоточимся на трех классах задач, связанных с управлением информацией в среде WWW:

1. Моделирование и запросы в WWW.
2. Выборка и интеграция информации.
3. Разработка и реструктуризация Web-сайтов.

БД для WWW

Моделирование и запросы в WWW: Предположим, что мы рассматриваем Web как ориентированный граф, узлы которого являются страницами Web, а дуги – связями между страницами. Рассмотрим задачу формулировки запросов для поиска определенных страниц Web. При этом запросы могут быть основаны на содержании нужных страниц и на структуре связей, соединяющих эти страницы.

БД для WWW

Выборка и интеграция информации: Некоторые Web-сайты могут рассматриваться на более тонком уровне гранулярности, чем страницы, как контейнеры структурированных данных. В связи с ростом числа таких сайтов становятся актуальными две следующие задачи. **Первая** - осуществлять выборку данных, представленных в структурированном виде (например, множество кортежей) из HTML-страниц, их содержащих. Если мы рассматриваем сайты такого рода как автономные неоднородные базы данных, возникает **вторая** задача – формулировка запросов, которые требуют интеграции данных.

БД для WWW

Разработка и реструктуризация Web-сайтов: Другой аспект применения концепций и технологий баз данных – разработка и реструктуризация Web-сайтов, а также управление ими. Конструирование Web-сайтов может начинаться либо с некоторых исходных данных (хранимых в базах данных или в структурированных файлах), либо путем реструктуризации уже существующих Web-сайтов. Выполнение этой задачи требует использования каких-либо методов моделирования структуры Web-сайта и языков для реструктуризации данных таким образом, чтобы они соответствовали желаемой структуре.

БД для WWW

Представление данных для задач Web/DB

Создание систем для решения любой из указанных выше задач требует выбора какого-либо метода для моделирования предметной области. В частности, нам необходимо в этих задачах моделировать сам Web, структуру Web-сайтов, внутреннюю структуру страниц Web, и, наконец, содержание сайтов с более тонкой степенью гранулярности. Далее необходимо обсудить главные факторы, характеризующие модели данных, используемые в приложениях Web.

БД для WWW

Представление данных для задач Web/DB

Графовые модели данных: Нам необходимо моделировать множество страниц Web, а также связи между ними. Эти страницы могут полностью находится на нескольких сайтах либо на единственном сайте. Следовательно, естественный способ моделирования этих данных основан на модели данных ***помеченных графов***. В этой модели узлы представляют страницы Web (или внутренние компоненты страниц), а дуги – связи между страницами. Метки на дугах могут рассматриваться как имена атрибутов.

БД для WWW

Представление данных для задач Web/DB

Модели слабоструктурированных данных: Второй аспект моделирования данных для приложений Web заключается в том, что во многих случаях структура этих данных не является постоянной.

Слабоструктурированными называются данные, обладающие какими-либо из следующих характеристик:

1. Схема не задана заранее и может неявно содержаться в данных.
2. Схема сравнительно велика (в смысле объема данных) и может часто изменяться.
3. Схема является описательной, а не предписывающей.
4. Данные не являются строго типизированными, т.е., для различных объектов значения одного и того же атрибута могут иметь различные типы.

БД для WWW

Представление данных для задач Web/DB

Модели слабоструктурированных данных были основаны на помеченных ориентированных графах. В модели слабоструктурированных данных не налагается какого-либо ограничения на множество дуг, которые исходят от данного узла в графе, или на типы значений атрибутов.

В связи с упомянутыми выше характеристиками слабоструктурированных данных становится важной в этом контексте дополнительная возможность – запрашивать схему (т.е., метки дуг в графе).

БД для WWW

Представление данных для задач Web/DB

Другая отличительная черта моделей, используемых в приложениях Web/DB – присутствие специфических для Web конструкций в представлении данных. Например, в некоторых моделях различаются унарное отношение, идентифицирующее страницы, и бинарное отношение для связей между страницами. Кроме того, мы можем различать связи внутри Web-сайта и внешние связи.

БД для WWW

Представление данных для задач Web/DB

К свойствам второго порядка, которыми различаются обсуждаемые здесь модели данных, можно отнести: (1) способность моделирования некоторого порядка на множестве элементов в базе данных, (2) моделирование вложенных структур данных и (3) поддержка типов коллекций (множества, мультимножества, массивы).

БД для WWW

Представление данных для задач Web/DB

Важный аспект языков запросов данных в приложениях Web – необходимость генерировать сложные структуры в результате обработки запроса. Например, результат некоторого запроса в системе управления Web-сайтом может представлять собой граф, моделирующий этот Web-сайт. Следовательно, фундаментальная характеристика многих из языков запросов состоит в том, что их выражения запросов содержат компонент структурирования наряду с традиционным компонентом фильтрации данных.

БД для WWW

Моделирование Web и запросы

Первыми инструментальными средствами для обработки запросов в Web, были известные поисковые машины, которые теперь широко развернуты и активно используются. Они основаны на поисковых индексах слов и фраз, встречающихся в документах, обнаруженных «роботами» (crawler) Web. Совсем недавно были предприняты усилия, направленные на преодоление ограничений этой парадигмы за счет использования в запросах структуры связей. Прототип поисковой машины Web следующего поколения Google интенсивно использует структуру Web для повышения производительности функционирования «робота» и индексирования.

БД для WWW

Гипертекстовые/документальные языки запросов: Ряд моделей и языков запросов для структурированных документов и гипертекстов был предложен еще в период, предшествующий появлению Web. Новый аспект этого подхода заключается в возможности производить запросы относительно структуры с помощью переменных-путей.

БД для WWW

Графовые языки запросов: Исследования, связанные с использованием графов для моделирования баз данных, которые были стимулированы такими приложениями, как разработка программного обеспечения и управление компьютерными сетями, привели к созданию языков целого ряда языков, например, G, G+ и GraphLog, основанных на графах. Они поддерживают использование в запросах правильных выражений путей и графовых конструкций.

БД для WWW

Языки запросов для слабоструктурированных данных: Такие языки запросов для слабоструктурированных данных, как Lorel, UnQL и STRUDQL, также используют помеченные графы как гибкую модель данных. В отличие от графовых языков запросов, они делают акцент на возможности запрашивать схему и приспосабливаться к нерегулярностям в данных, таких, например, как опущенные или повторяющиеся поля, неоднородные записи.

БД для WWW

Язык WebSQL: В языке WebSQL предлагается моделировать Web как реляционную базу данных, состоящую из двух (виртуальных) отношений: *Документ* и *Якорь*. Отношение Документ содержит по одному кортежу для каждого документа из Web, а отношение Якорь – по одному кортежу для каждого якоря в каждом документе из Web. Такая реляционная абстракция Web позволяет использовать для формулировки запросов язык, подобный SQL.

Если бы *Документ* и *Якорь* были фактическими отношениями, можно было бы просто использовать SQL. Но поскольку эти отношения являются полностью виртуальными, нельзя оперировать ими непосредственно.

БД для WWW

Семантика WebSQL зависит от *материализации* частей этих отношений путем спецификации представляющих интерес документов во фразе FROM запроса. Основным способом материализации является навигация из известных URL. Для описания такой навигации используются правильные выражения путей.

Атом такого правильного выражения может иметь форму d1 = > d2, означающую, что документ d1 указывает на d2, и d2 хранится на ином сервере, чем d1.

Он может иметь также форму d1 – > d2, которая, в свою очередь, означает, что d1 указывает на d2, и d2 хранится на том же самом сервере, что и d1.

БД для WWW

Предположим, например, что мы хотим найти список триплетов вида (d1, d2, метка), где d1 – документ, хранимый на нашем локальном сайте, d2 – документ, хранимый где-либо еще, и d1 указывает на d2 с помощью связи, помеченной меткой. Допустим также, что все наши локальные документы достижимы из www.mysite.start. Тогда указанную задачу можно решить с помощью запроса:

```
SELECT d.url, e.url, a.label
FROM Document d SUCH THAT «www.mysite.start» -> d,
        Document e SUCH THAT  d => e,
        Anchor a SUCH THAT a.base = d.url
WHERE a.href = e.url
```

БД для WWW

Предложение FROM порождает экземпляры двух переменных, определенных на отношении *Документ* (d и e), и одной переменной a на отношении *Якорь*. Области определения переменной d принадлежит каждый локальный документ, а e принимает значения на множестве всех не локальных документов, достижимых непосредственно из d.

В свою очередь, значением переменной a может являться каждая связь, которая исходит из документа d. Дополнительное условие, предписывающее, чтобы целевым документом связи a был документ e, задается предложением WHERE.

БД для WWW

Язык W3QL подобен, по существу, WebSQL, с некоторыми значительными различиями: он использует внешние программы (аналогично определяемым пользователем функциям в объектно-реляционных языках) для спецификации условий, налагаемых на содержание файлов, а не формирование условий в синтаксисе языка, и это обеспечивает механизмы для обработки форм, встречающихся в процессе навигации.

БД для WWW

WQL, язык запросов проекта WebDB подобен WebSQL, но он в большей мере поддерживает функциональные возможности SQL, допуская, например, агрегацию и группирование, и, кроме того, обеспечивает ограниченную поддержку запросов внутридокументной структуры.

БД для WWW

Рассмотренные выше языки интерпретируют страницы Web как атомарные объекты с двумя свойствами: они могут содержать или не содержать некоторые текстовые образцы и они могут указывать на другие объекты. Опыт использования таких языков показывает, что имеется две основные области приложений, для которых они могут быть полезны: (1) создание оболочек (wrapping) для данных, трансформация и реструктуризация данных, (2) создание и реструктуризация Web-сайтов. В обеих этих областях приложений часто оказывается существенной возможность иметь доступ к внутренней структуре страниц Web из языка запросов.

БД для WWW

Язык WebOQL: Основная структура данных в WebOQL – гипердерево. *Гипердеревья* – это упорядоченные деревья с помеченными дугами, причем имеется два типа дуг – внутренние и внешние. Внутренние дуги используются для представления структурированных объектов, а внешние – для представления связей (обычно гиперссылок) между объектами. Дуги снабжаются метками, в качестве которых используются записи. Язык WebOQL позволяет манипулировать как отдельными гипердеревьями, так и Web в целом, и они (гипердеревья и Web) могут создаваться в результате обработки запроса.

БД для WWW

На рисунке показано гипердерево, содержащее описания публикаций нескольких исследовательских групп.

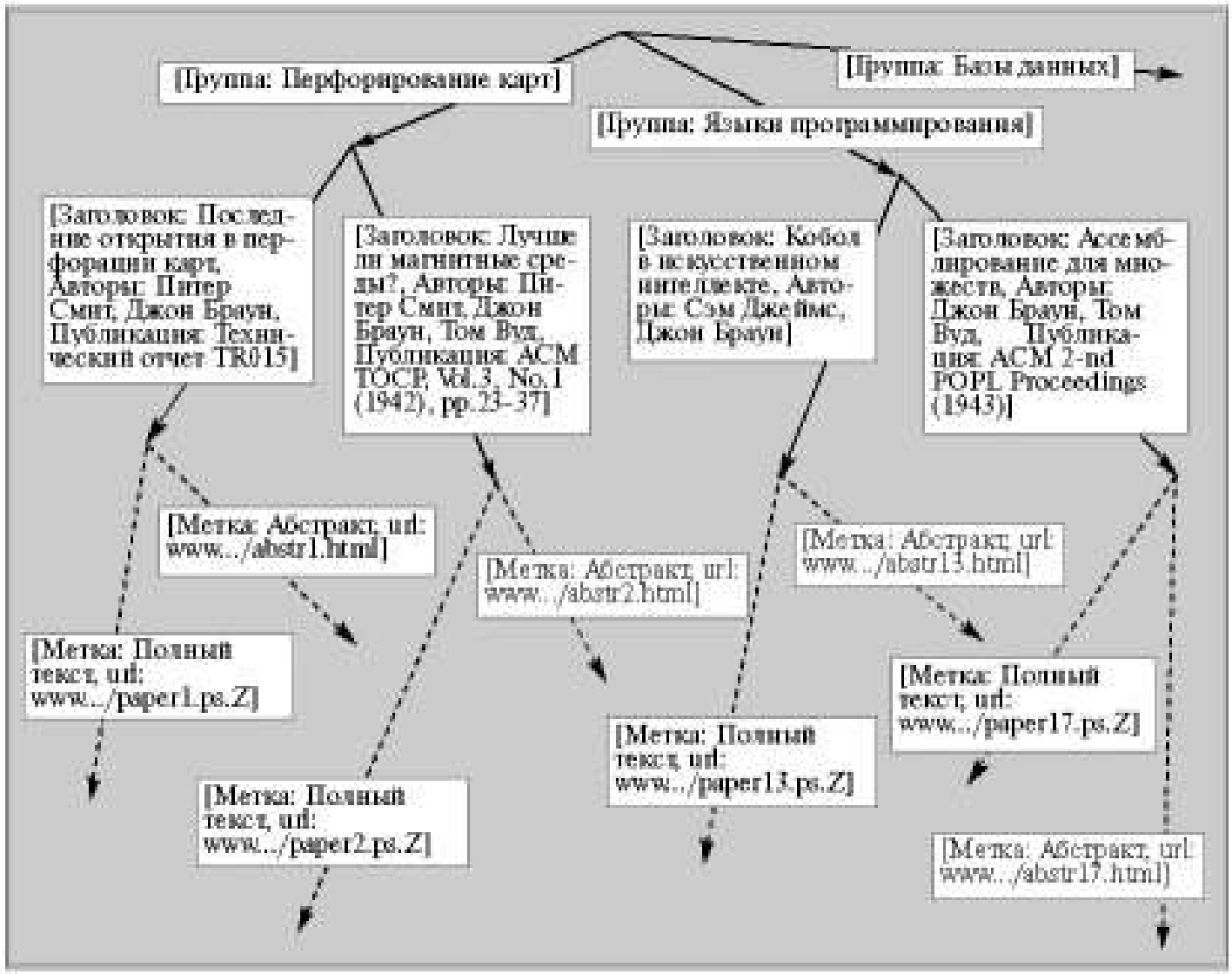

Рис. 1: Пример гипердерева

БД для WWW

Предположим, например, что база данных документов на рисунке выше имеет имя СтатьиПоИнформатике и что мы хотим осуществить из нее выборку названия и URL полных текстов статей Смита. Тогда нужно использовать следующий запрос:

```
select [y.Название, y'.Url]
from x in СтатьиПоИнформатике, y in x'
where y.Авторы ~ «Смит»
```

БД для WWW

В этом запросе переменная x определена на множестве простых деревьев базы данных СтатьиПоИнформатике, а при заданном значении x переменная y, в свою очередь, принимает значения на множестве простых деревьев x'. Переменная x' обозначает результат применения к дереву x оператора ('), который возвращает первое поддерево его параметра. Тот же самый оператор используется для извлечения из дерева y его первого поддерева в y'.Url. Квадратные скобки обозначают оператор, который строит дугу, помеченную записью, образуемой аргументом (в приведенном примере предполагается, что запись включает поля с указанными именами). Наконец, тильда (~) представляет собой предикат сопоставления со строковым образцом: его левый аргумент – строка, а правый – образец.

БД для WWW

Рассмотренные выше запросы отображают гипердерево в другое гипердерево, или, если говорить в более общих терминах, запрос – это функция, которая отображает один Web в другой. Например, следующий запрос создает новую страницу для каждой исследовательской группы (использующей имя группы как URL). Каждая страница содержит публикации соответствующей группы.

```
select x' as x.Группа
from x in СтатьиПоИнформатике
```

БД для WWW

В общем случае фраза select имеет вид:

select q1 as s1, q2 as s2, ..., qm as sm

где каждое qi – это запрос, а каждое из si – или запрос строки или схема. Фразы «as» создают URL s1, s2,... , sm, которые присваиваются новым страницам, полученным в результате выполнения каждого из запросов qi.

БД для WWW

Шаблоны навигации: Шаблоны навигации – это правильные выражения в алфавите предикатов, определенных над записями. Они позволяют специфицировать структуру путей, по которым необходимо следовать для того, чтобы найти значения переменных.

Предположим, что мы имеем некоторый программный продукт, документация к которому представлена в формате HTML, и мы хотим сформировать полнотекстовый индекс для нее. Такие документы образуют сложный гипертекст, но можно просматривать их и последовательно, следуя по связям, помеченным меткой «Следующий». Мы можем получить эту информацию, используя следующий запрос:

```
select [ x.Url, x.Текст ]

from x in browse(«root.html»)

        via (^*[Текст ~ «Следующий»]>)*
```

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

БД для WWW

ВОПРОСЫ ?

СОВЕТ ДНЯ: