

АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ СИСТЕМ УПРАВЛЕНИЯ ДЕДУКТИВНЫМИ БАЗАМИ ДАННЫХ

Дацун Наталья Николаевна, Пушкаренко Сергей Анатольевич
Кафедра прикладной математики и информатики ДонНТУ
Кафедра компьютерной инженерии ГФ ОМУРЧ “Украина”

В настоящее время в базах данных (БД) накоплен большой объем информации. Традиционно управление БД выполняется с помощью программной реализации в виде систем управления базами данных (СУБД). Однако разрабатываются и аппаратно-программные средства вычислительной техники для реализации хранения, обработки, ввода и вывода информации, например, в виде машин баз данных. Но использование информации, хранящейся в современных БД, не достаточно эффективно. Наиболее распространённый в настоящее время язык запросов SQL не позволяет в полной мере осуществить

извлечение всей необходимой информации из БД. Многие запросы не могут быть представлены в рамках одного SQL-оператора. При этом SQL-запросы теряют свою декларативность, становятся процедурными, сложными для написания и восприятия, что часто приводит к ошибкам и, соответственно к неверным результатам запросов. Поэтому возникает необходимость в альтернативных средствах извлечения информации, наиболее перспективным из которых является использование языков логического программирования (ЛП) в качестве языков запросов к реляционным БД (РБД). Такие системы называют дедуктивными базами данных.

Структура запроса на языке логического программирования (ЯЛП) и SQL-запроса во многом сходна. Запрос на ЯЛП при этом более компактен и более прост для восприятия.

Между РБД и логическим программированием есть много общего [1]:

- а) таблица БД соответствует предикату ЛП;
- б) поле БД соответствует аргументу предиката ЛП;
- в) запись БД соответствует факту языка ЛП;
- г) представление БД соответствует правилу языка ЛП;
- д) запрос БД соответствует цели языка ЛП.

Такое сходство упрощает реализацию СУБД, основанных на ЯЛП.

Традиционно в качестве языка ЛП используется язык Пролог, его диалекты или модификации, например, Дейталоги. Любую таблицу или множество таблиц можно описать на прологе или Дейталоге. Такие программы обычно называют базами данных Пролога (Дейталога соответственно).

Язык логического программирования Пролог может работать как мощный язык запросов БД. Однако использование Пролога имеет ряд недостатков. К ним можно отнести следующие [1]:

- а) покортежная обработка. Пролог возвращает отдельные кортежи по одному, а не множество кортежей (Пролог в традиционном использовании ориентирован на домены);
- б) процедурность и чувствительность к порядку.

Вычисление в Прологе (согласование цели в БД) зависит от порядка правил и фактов в БД и от порядка предикатов в теле правила. В действительности программист на Прологе использует чувствительность к порядку для построения эффективных программ, жертвуя, таким образом, декларативной природой логического программирования в пользу процедурности. Напротив, языки БД (такие, как SQL или реляционная алгебра) являются непроцедурными: выполнение запроса к БД не зависит от порядка поисковых предикатов или кортежей БД;

в) специальные предикаты. Программисты на Прологе управляют выполнением программ с помощью специальных предикатов (используемых, например, для ввода-вывода, или воздействия на бэгрекинг). В этом заключается другая важная причина потери декларативной природы языка, не имеющей аналогов в других языках программирования;

г) функциональные символы. Пролог содержит функциональные символы, используемые обычно для построения рекурсивных функций и сложных структур данных; эти функции не требуются для манипулирования плоской реляционной БД, хотя они могли бы оказаться полезными для оперирования сложными объектами в БД.

Альтернативой Прологу является язык Дейталога, который представляет собой один из вариантов интеграции возможностей РБД и логического языка запросов к БД. Он который сконструирован для использования в качестве языка БД. Он является непроцедурным, множественным, не чувствительным к порядку, не имеющим специальных предикатов и функциональных символов. Таким образом, в Дейталоге устранены четыре указанных выше недостатка Пролога

Синтаксически Дейталога напоминает чистый Пролог. Все прологовские правила для представления запросов и ограничений, являются также правильными запросами Дейталога. Их выполнение приводит к образованию множества кортежей.

Преимущества Дейталога по сравнению с Прологом [1]:

а) программы на Дейталоге - чисто декларативные (а

прологовские имеют операционную семантику);

б) базовая схема интерпретации Дейталога – восходящая. В ее основе лежат методы неподвижной точки программы и реляционная алгебра, в отличие от Пролога, в котором используется нисходящая схема, основанная на резолюционной стратегии, поиске “в глубину” и бэктрегинге для конструирования деревьев доказательств;

в) Пролог позволяет создавать системы, в которых средства поддержки логических программ и БД рассматриваются изолированно, и требуется выделять предикаты экстенциональной БД для интерпретации доступа к системе БД из логической программы. Дейталога же ориентирован на создание интегрированных дедуктивных систем.

В настоящее время на территории СНГ разработкой компиляторов языка Дейталога занимаются Московский государственный университет, Харьковский национальный университет радиоэлектроники и другие учебные и научные заведения.

Московским государственным университетом разработан проект «Deviser» LCPR-системы языка Дейталога [2]. Реализация проекта выполнена на Bigloo – функциональном языке программирования Scheme, в свою очередь реализованном на C++. Эта Дейталога-система работает с СУБД “SQLite”.

Институтом прикладной математики им. М.В. Келдыша Российской Академии наук разработан экспериментальный прототип дедуктивной системы – дедуктивный язык для работы с автоматически структурируемыми данными (авторы Е.Л. Китаев, М.И. Слепенков). Система обладает языком класса Datalog. Система предназначена для работы с полуструктурированными данными, ее можно использовать как для выполнения запросов к БД, так и для трансформации и реструктуризации данных. При построении языка были использованы подходы, развитые в семействе языков объектных логик.

Рабочей группой Datalab НИИ системных технологий кафедры искусственного интеллекта Харьковского

национального университета радиоэлектроники [3] разработан квантифицированный Дейталога (“DLQ”) и объектно-дедуктивный Дейталога, работающие с СУБД MS Access.

В основе квантифицированного Дейталога лежит понятие кванторного выражения (атрибутив), имеющего смысл придаточного предложения в правиле программы или подзапроса SQL. Содержательно атрибутив интерпретируется словом “существует” и позволяет непосредственно выражать логические конструкции существования в программе (В традиционном Дейталога как и в Прологе нет возможности для записи кванторов и все переменные неявно квантифицированы квантором всеобщности [4]). Объектно-дедуктивный Дейталога поддерживает иерархию наследования и полиморфизм.

Таким образом, в настоящее время актуальной является проблема разработки аппаратно-программных средств реализации систем управления дедуктивными базами данных.

СПИСОК ЛИТЕРАТУРЫ

1. Чери С., Готлоб Г., Танка Л. Логическое программирование и базы данных. - М.: Мир, 1992.
2. Официальный сайт МГУ. – <http://sp.cmc.msu.ru/datalog/>.
3. Московская Секция АСМ. SIGMOD. – <http://synthesis.ipi.ac.ru/sigmod/seminar>
4. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. – М.: Финансы и статистика, 1990. – 320 с.