

MEANS OF INTEGRATION OF HETEROGENEOUS DATA CORPORATE INFORMATION AND TELECOMMUNICATION SYSTEMS

Filatov V. A., Rudenko D. A., Grinyova E. E.
Kharkov National University of Radio Electronics
14, Lenin Ave, Kharkov, 61166, Ukraine
Ph.: (057) 7021214, e-mail: filatov_val@ukr.net

Abstract — The key features and characteristics of a declarative database for corporate information technology systems are identified. It is shown that the use of syntactic structures of deductive databases fairly confirms with logical rules for constructing intension database. It is shown that the use of rules for describing the properties of the database allows recursively computing the values of the query.

СРЕДСТВА ИНТЕГРАЦИИ НЕОДНОРОДНЫХ ДАННЫХ В КОРПОРАТИВНЫХ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Филатов В. А., Руденко Д. А., Гринева Е. Е.
Харьковский национальный университет радиоэлектроники
пр. Ленина, 14, Харьков, 61166, Украина
тел.: (057) 7021214, e-mail: filatov_val@ukr.net

Аннотация — Определены основные свойства и характеристики декларативного описания данных для корпоративных информационно-телекоммуникационных систем. Показано, что использование синтаксических конструкций дедуктивных баз данных хорошо сопрягаются с логическими правилами для построения интенционала базы данных. Показано, что использование правил для описания свойств баз данных позволяет рекурсивно вычислять значения запроса.

I. Введение

Развитие информационных систем невозможно без создания необходимой телекоммуникационной среды, обеспечивающей требуемый уровень сервиса при передаче данных.

К настоящему времени разработано и внедрено большое количество технологий передачи данных, обеспечивающих необходимые скорости передачи, защиту и управление. Данные технологии поддерживаются производителями аппаратных и программных средств и обеспечивают разработчиков и потребителей телекоммуникационных систем необходимыми возможностями для решения задач формирования интегрированной информационной системы.

Современные системы управления базами данных (СУБД), обладая мощным арсеналом средств решения типичных задач, таких, как загрузки и обновления базы данных (БД), поиск по условиям, задачи прямого счета, имеют в то же время ограниченные возможности по решению сложных задач (в частности, в неоднородных структурах). Для реализации последних в рамках традиционных технологий пользователь вынужден, во-первых, осуществлять глубокую проработку задачи, связанную не только с описанием структуры и правил функционирования системы, но и с явным описанием процедур поиска решения. Во-вторых, эти процедуры, описанные на каком-либо формальном языке, необходимо "перевести" на входной язык СУБД, что само по себе является трудоемкой задачей.

Таким образом, применение технологий обработки неоднородной информации, основанных на использовании традиционных языков программирования баз данных (например, языка SQL), оказывается неэффективным.

II. Основная часть

На основании проведенного анализа интегрированные БД можно определить как комплекс алгебраиче-

ских и логических средств, ориентированный на разработку прикладных программ, независимых от СУБД при одновременном взаимодействии с несколькими, возможно, неоднородными БД. Следует отметить, что интеграция должна строиться таким образом, чтобы обеспечивалась возможность эволюционного развития информационно-телекоммуникационных систем.

Определим основные характеристики декларативного языка и проведем параллель между исчислением предикатов первого порядка (в частности, реляционным исчислением) и правилами логического программирования, которые коррелируют с *L*-правилами, описывающими существование и свойства информационных объектов предметной области [1].

Логический язык, рассматриваемый далее, является прототипом языка DataLog и состоит из импликативных правил типа «если..., то...». В этих правилах выражаются следующие факты: из определенной комбинации кортежей в определенном отношении можно вывести, что некоторый другой кортеж входит в другое отношение, или получить ответ на запрос.

Прикладным инструментом для реализации задач интеграции является компилятор дедуктивной БД (ДБД) как средство трансляции логического языка обработки данных. Модель ДБД представляет собой тройку

$$M^{DDB} = \langle EDB, IDB, I \rangle, \quad (1)$$

где *EDB* - экстенциональная база данных (ЭБД): соответствующие отношения хранятся в БД, *IDB* - интенциональная (вычисляемая) база данных (ИБД): отношения вычисляемые посредством применения одного или нескольких правил, *I* - множество ограничений целостности.

Отношения БД в контексте DataLog представляется с помощью предикатов. Каждый предикат обладает фиксированным количеством аргументов. Предикат с аргументом будем называть атомом. Синтак-

сис атома представляет собой n -арную функцию типа $P(x_1, \dots, x_n)$, возвращающую значение булевого типа. Если R – это таблица с n атрибутами, перечисленными в некотором порядке, то R можно трактовать как предикат, соответствующий таблице, при этом атом $R(I_{o_1}, \dots, I_{o_n})$ будет иметь значение *True*, если список $(I_{o_1}, \dots, I_{o_n})$ является строкой таблицы R , в противном случае атому соответствует значение *False*.

Операторы, подобные операторам традиционной реляционной алгебры, в DataLog описываются с помощью правил. Согласование реляционных терминов и терминов логического программирования можно представить следующими пунктами:

- реляционный атом соответствует заголовку правила;
- символ “ \leftarrow ” можно трактовать как условие “если”;
- тело правила, которое может состоять из одного или нескольких атомов, трактуется как подцель правила, которые могут быть как реляционными, так и арифметическими атомами;
- в случае нескольких подцелей, они объединяются оператором *AND* (“ \wedge ”), и при необходимости каждой подцели может предшествовать оператор логического отрицания *NOT* (“ \neg ”).

Параллель между реляционными и логическими языками запросов можно показать на следующем примере. Пусть дана схема БД $R(I_{o_1}, I_{o_2}, I_{o_3}, I_{o_4}, I_{o_5})$ и пусть необходимо найти множество значений из I_{o_1} и I_{o_2} , для которых $I_{o_3} \leq N$ (где $N = \text{const}$), тогда правило будет иметь вид

$$P(x_1, x_2) \leftarrow R(x_1, x_2, x_3, x_4, x_5) \wedge x_3 \leq N, \quad (2)$$

где $(x_1, x_2, x_3, x_4, x_5)$ – предикатные переменные, соответствующие информационным объектам $(I_{o_1}, I_{o_2}, I_{o_3}, I_{o_4}, I_{o_5})$.

Аналогичный запрос, выраженный в терминах реляционной алгебры, будет иметь вид

$$R \equiv \pi_{I_{o_1}, I_{o_2}} (\sigma_{I_{o_3} \leq N} (R(I_{o_1}, I_{o_2}, I_{o_3}, I_{o_4}, I_{o_5}))), \quad (3)$$

Покажем, что такой язык может быть транслирован в правила логического программирования для работы с табличным представлением данных. С одной стороны, в нерекурсивной форме правила DataLog обладают теми же выразительными свойствами, что и языки реляционной модели, с другой стороны, аппарат рекурсии DataLog позволяет представить запросы, которые не могут быть описаны непосредственно средствами SQL.

Проведем редукцию между основными операциями реляционной алгебры и логическими правилами, тем самым подтвердив эквивалентность выразительной мощности нерекурсивного языка DataLog и реляционной алгебры. Поведение каждого оператора алгебры можно выразить с помощью одного или нескольких логических правил [2, 3].

Рассмотрим некоторые операции. Пусть даны два отношения $R(A_1, A_2, A_3, A_4)$ и $S(A_1, A_2, A_3, A_4)$, для построения пересечения $R \cap S$, тогда правило DataLog будет иметь вид

$$P(x_1, x_2, x_3, x_4) \leftarrow R(x_1, x_2, x_3, x_4) \text{ AND } S(x_1, x_2, x_3, x_4), \quad (4)$$

где P – предикат из интенционала БД, которому соответствует отношение $R \cap S$. Чтобы кортеж $\langle x_1, x_2, x_3, x_4 \rangle$ был добавлен в итоговое отношение P , он должен в обеих подцелях получить значение *True*, то есть принадлежать отношениям R и S одновременно.

Для объединения отношений $R \cup S$ необходимо воспользоваться двумя правилами

$$\begin{aligned} P(x_1, x_2, x_3, x_4) &\leftarrow R(x_1, x_2, x_3, x_4); \\ P(x_1, x_2, x_3, x_4) &\leftarrow S(x_1, x_2, x_3, x_4). \end{aligned} \quad (5)$$

Первое правило определяет, что каждый кортеж из R должен быть включен в P . Аналогично, второе правило предполагает, что отношение P должно содержать все кортежи из S . Таким образом, совместно используемые правила предусматривают включения в P каждого кортежа отношения из $R \cup S$.

Если другие правила, в заголовках которых используется предикат P , не определены, способов пополнения P какими-либо другими кортежами не существует, следовательно, содержимое P точно совпадает с набором данных $R \cup S$.

Из рассмотренных методов приведения операций реляционной алгебры к логическим правилам можно сделать вывод, что если условие выбора использует только конъюнкции, то такая формула может быть выражена одним правилом, если в условии присутствует хотя бы одна дизъюнкция, то такая формула описывается несколькими правилами в соответствии с количеством литералов дизъюнкции.

Необходимо отметить, что ситуация, когда сложное выражение реляционной алгебры может быть описано с помощью одного правила DataLog, не является всегда допустимой и зависит от алгебраического выражения.

III. Заключение

1. Предложен метод использования правил языка БД для формирования запросов к таблицам, поддерживаемым реляционной СУБД. Трансляция реляционного языка в язык логики предикатов реализована посредством редукции реляционных операций в правила DataLog. Такой подход дает возможность обрабатывать данные в информационно-телекоммуникационных системах с неоднородной структурой

2. Усовершенствован метод декомпозиции сложных выражений реляционной алгебры, что значительно сокращает время вывода конечного результата.

3. Получила дальнейшее развитие технология приведения табличных документов к реляционному виду, которая в отличие от существующих, позволяет организовать доступ к интегрированной информации с помощью динамических запросов.

IV. References

- [1] Tanyanskiy S.S. Semantika reljacionnyh baz dannyh s neopredeljonnyimi znachenijami. Radioelektronika i informatika. 1998, № 1(02), S. 51-53.
- [2] Garsia-Molina G. Sistemy baz dannyh. Polnyj kurs. M.: Vil'jams, 2003. 1088 s.
- [3] Ul'man Dzh. Vvedenie v sistemy baz dannyh. M.: Lori, 2006, 379 s.