

УДК 681.3.016



ОРГАНИЗАЦИЯ ЗАПРОСОВ К РАСПРЕДЕЛЕННЫМ ДАННЫМ СРЕДСТВАМИ ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ

С.С. Тянянский¹, Ю.А. Мальков²¹ ХНУРЭ, г. Харьков, Украина, tanyansky_ss@yahoo.com;² ХНУРЭ, г. Харьков, Украина, malkov@smtp.ru

Рассматривается представление реляционных баз данных в виде множества логических фактов и правил. Описаны возможные варианты отличий в физической структуре отношений со схожим информационным наполнением. Рассмотрен метод создания совместного отображения данных с различной физической структурой, основанный на медиаторах и логическом программировании.

БАЗЫ ДАННЫХ, ФЕДЕРАЛИЗАЦИЯ, ДЕДУКТИВНЫЕ БАЗЫ ДАННЫХ, ИНТЕГРАЦИЯ БАЗ ДАННЫХ, ДЕЙТАЛОГ

Введение

В процессе развития информационных систем (ИС) крупных предприятий или организаций внедряется большое множество различных информационных подсистем. Как правило, эти подсистемы имеют свои собственные базы данных (БД) для накопления информации и последующего ее использования. В связи с информационной и логической связанностью бизнес-процессов предприятия, автоматизируемых данными подсистемами, часть информации неизбежно дублируется. Для исключения накопления одинаковой информации в различных подсистемах предприятия создают системы интеграции данных. Это позволяет использовать данные одной подсистемы другими. Для интеграции данных используют методы консолидации и федерализации. При консолидации данные переносятся в одну общую ИС, с общей структурой, управляемую реляционной системой управления базами данных (СУБД) промышленного уровня. После переноса данных все системы используют общую БД. Среди недостатков данной технологии необходимо отметить потерю данных при интеграции, а также значительные ресурсы памяти для их хранения. Другим подходом решения задачи интеграции данных является федерализация. Он позволяет не изменяя существующих ИС, извлекать данные из первичных источников и интегрировать их таким образом, чтобы они отвечали требованиям поступившего запроса.

На данный момент основным средством управления данными являются реляционные системы БД, использующие в качестве операционной спецификации алгебру множеств. Языки запросов, построенные на таких операциях, имеют ряд ограничений, таких как отсутствие рекурсивных функций и строгая привязка к структуре данных.

В настоящее время активно развиваются декларативные языковые средства в системах обработки данных.

Вопросами применения логического программирования в реляционных базах данных начали

рассматриваться еще в конце 70-х годов. Среди исследователей данной тематики можно выделить работы Дж. Малпаса [1], С. Чери и других [2], Стерлинга и Шапиро [3], Девяткова В.В. [4], в которых описана методика применения логических языков, и в частности Дейталога, в качестве языка определения и манипулирования данными.

Вопрос о применении средств логического программирования для реализации системы интеграции данных в литературе ранее не рассматривался. В связи с этим целью настоящей статьи является исследование и разработка метода приведения БД с неоднородной структурой к единому виду, используя интенциональные свойства таблиц. Также предложен подход к организации поиска данных средствами логического программирования.

Ниже рассматривается методика совместного использования данных при помощи средств логического языка Дейталога, поддерживаемого дедуктивными базами данных (ДБД).

1. Применение логических языков для манипулирования объектами БД

Рассмотрим механизм ДБД для реализации системы интеграции данных, основанной на федерализации. Это позволит оперировать объектами базы данных при помощи логических выражений, а сами объекты базы данных описывать логическими фактами и правилами. В качестве языка используем язык логического программирования Дейталога, разработанный специально для БД [2, 4].

Отношение БД будем именовать предикатом и представлять в виде набора базовых дизъюнктов (фактов) вида (1):

$$t(a_1, a_2, \dots, a_n), \quad (1)$$

где, t – имя предиката, а a_1, a_2, \dots, a_n – аргументы предиката, определяющие его аридность.

Факты являются входными данными Дейталога-программы и все их множество представляют экстенциональной базой данных (ЭБД).

Кроме ЭБД существует интенциональная база данных (ИБД) – результат выполнения Дейталога-

программы. ИБД строится из ЭБД с помощью правил, определяющих ее содержимое.

Правило – это конструкция вида

$$q :- p_1, \dots, p_n, \quad (2)$$

разделенная символом условной дизъюнкции “:-”. Слева от символа “:-” находится голова правила, а справа – тело. Приведенное правило означает, что цель q будет истинной, если будут истинны факты p_1, \dots, p_n , стоящие в теле правила.

В Дейталоге при помощи правил осуществляют запросы данных из множества фактов, а также создают новые логические конструкции, с которыми в дальнейшем можно работать, как с фактами.

При построении сложных правил часто используются переменные – строки буквенно-цифровых символов конечной длины, начинающиеся с прописной буквы. Например X , Query_2 и так далее. Они необходимы для подстановки атрибутов предиката из тела правила в его голову, а также задания условий в теле правила. Необходимо отметить, что переменные, которые появляются в голове правила, обязательно должны упоминаться в теле правила.

2. Система федерализации данных, основанная на медиаторах

Медиаторы – программные компоненты, обеспечивающие поддержку виртуальных баз данных, которые транслирует каждый запрос пользователя в один или несколько запросов, адресованных различным СУБД [5]. По результатам обработки частных запросов медиатор синтезирует ответ на исходный запрос.

Рассмотрим общую схему построения системы интеграции БД с применением медиаторов (рис. 1).

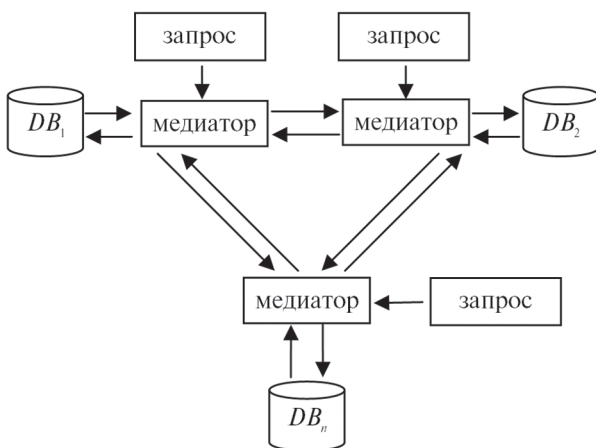


Рис. 1. Схема системы интеграции данных на основе медиаторов

Представим информационную систему предприятия в виде множества баз данных информационных подсистем $DB = \{DB_1, \dots, DB_n\}$. Некоторые отношения в этих базах имеют схожее информационное наполнение, но между ними существуют

различия в физической структуре, обусловленные различными вариантами реализации схемы при проектировании и различной степенью нормализации.

Рассмотрим схему построения медиатора для произвольной БД с применением средств логического программирования (рис. 2).

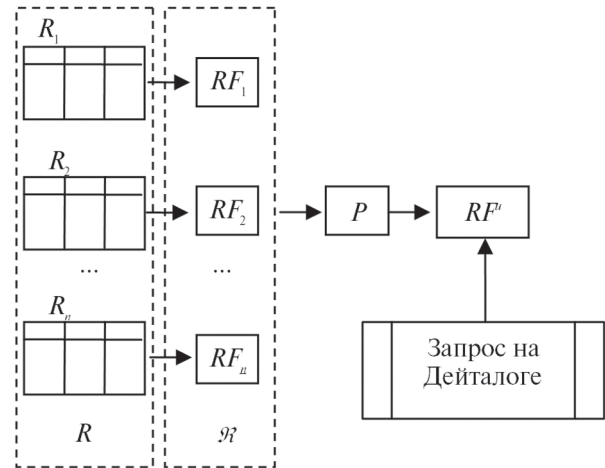


Рис. 2. Схема построения медиатора с применением логического программирования

Представим каждое отношение базы данных $R = \{R_1, R_2, \dots, R_n\}$ в виде множества $\mathfrak{R} = \{RF_1, RF_2, \dots, RF_n\}$, где RF_i – множество фактов вида (1), и определим логические правила $P = \{P_1, P_2, \dots, P_m\}$, где P_i – выражения вида (2), которые позволяют выделить интенционал $RF^m = \{rfu_1, rfu_2, \dots, rfu_k\}$ для отношений R , где rfu_i – выражение вида (1). Представленная схема построения медиатора позволяет формировать запросы к БД различной структуры с использованием языковых средств Дейталога.

3. Построение интенционала интегрируемых БД

Рассмотрим варианты приведения различных отношений R_i к общему интенционалу RF^m на примере интеграции отношений баз данных учета успеваемости студентов. Первым и возможно самым распространенным вариантом различий между отношениями являются отличия в физической структуре схемы БД, то есть различная степень нормализации.

Рассмотрим процедуру создания интенционала RF^m для БД DB_1 и DB_2 , имеющих различные схемы. Пусть R_1 – некоторое отношение из DB_1 (рис. 3), а R_2 и R_3 – отношения из DB_2 (рис. 4, 5), связанные по полю id.

фамилия	предмет	дата	группа	оценка
Иванов	математика	2008-11-01	ИС-01	4
Петров	английский	2008-12-04	ЗИ-02	5
Сидоров	математика	2008-12-21	ИС-03	3
Иванов	английский	2008-12-28	ИС-01	3
Сидоров	английский	2008-12-23	ПМ-01	2

Рис. 3. Отношение R_1

фамилия	курс	группа	id
Пушкин	4	пм-2	2
Лермонтов	2	иб-3	3
Есенин	1	пм-3	4

Рис. 4. Отношения R_2

id	предмет	оценка
2	математика	2
3	английский	5
4	английский	3
3	украинский	4

Рис. 5. Отношения R_3

Представим отношения R_1, R_2, R_3 в виде множеств логических фактов $r1, r2$ и $r3$ вида (1). Для этого каждый кортеж исходного отношения запишем в виде базового дизъюнкта, аргументами которого будут выступать значения атрибутов кортежа.

- $r1$ (“Иванов”, “математика”, 2008-11-01, “ИС-01”, 4).
-
- $r1$ (“Сидоров”, “английский”, 2008-12-23, “ПМ-01”, 2).
- $r2$ (“Пушкин”, 4, “пм-2”, 2).
- ...
- $r2$ (“Есенин”, 1, “пм-3”, 4).
- $r3$ (2, “математика”, 2).
- ...
- $r3$ (3, “украинский”, 4).

Определим отношение R_1 как интенционал RF^u для отношений R_1, R_2, R_3 . Чтобы привести отношения R_2 и R_3 к виду RF^u , определим следующее правило:

- $rfu(X, Y, G, Z, K):-r2(X, _, G, I),$
- $r3(I, Y, K), Z=$ ”нет_данных”.

В данном правиле переменная I связывает отношения R_2 и R_3 по полю id. Переменные X, Y, G, K переносят в структуру результирующего множества (цели) RF^u атрибуты “фамилия”, “предмет”, “группа”, “оценка” отношений R_2 и R_3 ; при этом переменная Z добавляет атрибут “дата” для установления эквивалентности схем $RF^u \equiv R_1 \equiv R_2 \cup R_3$.

Кроме проблемы различий в физической структуре таблиц существуют также различия в уровнях абстракции при проектировании БД. Так, типы элементов данных, имеющих несколько отличающихся подтипов, можно представить как отдельными таблицами для каждого подтипа, так и одной таблицей для всех подтипов, введя при этом столбец-селектор, определяющий значение подтипа. Для примера рассмотрим отношения R_4 (рис. 6) и R_5 (рис. 7) с данными об успеваемости двух групп “ПМ-04” и “ИС-01”. Для данной структуры характерным признаком является вынесенный элемент данных “группа” в заголовок отношения, что не соответствует структурам других отношений и усложняет процесс обработки данных.

ПМ-04			
фамилия	дата	предмет	оценка
Шевченко	2008-12-01	физика	3
Петренко	2008-12-21	биология	5
Ищенко	2008-12-26	химия	3

Рис. 6. Отношение R_4

ИС-01			
фамилия	дата	предмет	оценка
Тищенко	2008-11-07	алгебра	4
Зозуля	2008-12-01	геометрия	2
Яценко	2008-12-06	физика	5

Рис. 7. Отношение R_5

Чтобы привести отношения R_4 и R_5 к интенционалу RF^u , запишем их в виде множества логических фактов вида (1) и определим следующие правила:

- $rfu(X, Y, Z, G, K):-r4(X, Z, Y, K), G=$ ”пм-04”,
- $rfu(X, Y, Z, G, K):-r5(X, Z, Y, K), G=$ ”ис-01”.

В правилах (3) переменные X, Y, Z, K переносят атрибуты “фамилия”, “дата”, “предмет”, “оценка” отношений R_4 и R_5 в результирующий набор данных RF^u , а с помощью переменной G вводится отсутствующий в отношениях R_4 и R_5 атрибут “группа”.

Рассмотрим вариант приведения к интенционалу RF^u , кросс-таблицы R_6 (рис. 8).

группа	фамилия	дата	математика	физика	химия	биология
иб-03	Петренко	2008-10-11	3	5	3	4
иб-03	Евтушенко	2008-11-03	4	5	5	2
иб-02	Симоненко	2008-11-04	4	4	3	4

Рис. 8. Отношение R_6

Представим отношение R_6 как множество логических фактов и определим для него правила приведения R_6 к виду RF^u .

- $rfu(X, Y, Z, G, K):-r6(G, X, Z, K, _, _, _), Y=$ ”математика”,
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, K, _, _), Y=$ ”физика”,
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, _, K, _), Y=$ ”химия”,
- $rfu(X, Y, Z, G, K):-r6(G, X, Z, _, _, _, K), Y=$ ”биология”.

В приведенных правилах переменные G, X, Z переносят атрибуты “группа”, “фамилия”, “дата” отношения R_6 в RF^u . С помощью переменной K выбирается значение “оценки”, а Y передает название столбца, соответствующего этой оценке. Символ подчеркивания на месте атрибута обозначает, что этот атрибут не будет присутствовать в конечном множестве.

4. Реализация запросов средствами Дейталог

Для применения аппарата логического программирования к задаче интеграции данных при помощи медиаторов представим каждую БД ис-

ходного множества ИС в виде множества логических фактов вида (1). Полученные множества фактов образуют экстенционал этих БД. Для реализации запросов ко всему множеству БД с различной структурой сформируем интенционал этих БД с помощью логических правил вида (2). Данные правила приведут структуры исходных экстенционалов к единому интенционалу RF^u . Это позволит формировать запросы ко всему распределенному множеству БД.

Запрос данных из множества логических фактов в Дейталоге формируют при помощи логических правил вида (2). Рассмотрим процедуру составления запроса (цели) к множеству логических фактов и правил. В качестве источника данных будем использовать интенционал отношений RF^u , который имеет следующую структуру:

$$\begin{aligned} &rfu(x_{11}, x_{12}, x_{13}, \dots, x_{1k}), \\ &rfu(x_{21}, x_{22}, x_{23}, \dots, x_{2k}), \\ &\dots \\ &rfu(x_{m1}, x_{m2}, x_{m3}, \dots, x_{mk}). \end{aligned} \quad (4)$$

Используя интенционал вида (4), можно формировать запросы вида (5), которые будут использовать множество распределенных БД как единый консолидированный источник данных.

$$q(X1, X2, X3, \dots, XK) :- rfu(X1, X2, X3, \dots, XK), X3=1123. \quad (5)$$

Здесь rfu – исходное множество данных, $X1, X2, X3, \dots, XK$ – набор переменных, переносящих атрибуты rfu в результирующий набор q , а $X3=1123$ условие выборки.

Выводы

В силу ряда причин между схожими по информационному наполнению БД часто существуют отличия в их физической структуре, обусловленные различной степенью нормализации и отличными вариантами реализации схемы. Это делает невозможным использовать множество БД как источник информации для запросов, без внесения изменений в исходные отношения.

Описанная технология интеграции информационных систем на основе медиаторов, реализованных средствами логического программирования, позволяет решить задачу интеграции неоднородных БД за счет приведения исходных отношений к общему интенционалу с помощью логических фактов и правил.

Практическая значимость полученных результатов заключается в возможности создания уни-

версального средства формулирования запросов к БД с различной структурой. В рассматриваемом материале не учитывается синтаксическая несогласованность атрибутов и предполагается, что эта задача решена на этапе формирования метаданных, которые используются при согласовании элементов правил в медиаторах. Данный материал выходит за рамки исследований, проведенных в работе и представленных в статье.

Полученные результаты дают основания для дальнейшего изучения свойств логических выражений языка Дейталога для доказательства корректности выполняемых операций над данными. При этом необходимо рассматривать совокупность запросов в виде набора правил, выполняемых как одна транзакция.

Список литературы: 1. *Маллас, Дж.* Реляционный язык Пролог и его применение [Текст] : пер. с англ. – М. : Наука, 1990. – 464 с. 2. *Чери, С.* Логическое программирование и базы данных [Текст] : пер. с англ. / С. Чери, Г. Готлоб, Л. Танка. – М. : Мир, 1992. – 352 с. 3. *Стерлинг, Л.* Искусство программирования на языке Пролог [Текст] : пер. с англ. / Л. Стерлинг, Э. Шапиро. – М. : Мир, 1990. – 333 с. 4. *Девятков, В.В.* Системы искусственного интеллекта [Текст] / В.В. Девятков. – М. : Издательство МГТУ им. Н.Э. Баумана, 2001. – 352 с. 5. *Корнеев, В.В.* Базы данных: интеллектуальная обработка информации [Текст] / В.В. Корнеев, А.Ф. Гореев, С.В. Васютин, В.В. Райх. – М. : Нолидж, 2001. – 496 с.

Поступила в редколлегию 14.04.2010 г.

УДК 681.3.016

Організація запитів до розподілених даних засобами логічного програмування / С.С. Танянський, Ю.А. Мальков // Біоніка інтелекту: наук.-техн. журнал. – 2010. – № 1 (72). – С. 118–121.

У статті розглянуто питання організації доступу до розподілених даних засобами логічного програмування. Даний підхід дозволяє розв'язувати задачі обробки гетерогенних даних. Запропоновано технологію об'єднання інтенціональних та екстенціональних баз даних в одну структуру, для якої формуються запити споживача.

Л. 8. Бібліогр.: 5 найм.

UDK 681.3.016

Building queries for distributed data via methods of logical programming. / S.S. Tanyansky, Y.A. Malkov // Bionics of Intelligence: Sci. Mag. – 2010. – № 1 (72). – P. 118–121.

The aspects of accessing distributed data via methods of logical programming have been viewed in the article. This method allows to complete the tasks of heterogeneous data processing. The technology of uniting the intensional and extentional databases in one structure, for which the client's queries are formed, has been offered.

Fig. 8. Ref.: 5 items.