УДК 004.655.3/.652.43

РЕАЛИЗАЦИЯ МЕТОДОВ СРАВНЕНИЯ МНОЖЕСТВ СРЕДСТВАМИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Федорченко К.А.
Херсонский государственный университет

В статье рассматриваются вопросы, связанные с использованием современных реляционных баз данных. Описаны различные запросы, которые используют реляционное деление "Great Divide" и рассмотрены варианты применения этих запросов. Приведено описание выражений с использованием реляционной алгебры и примеры на языке SQL.

Ключевые слова: *реляционные базы данных, реляционное деление, сравнение множеств.*

Постановка проблемы

Накопление информации в базах данных информационных систем управления высшим учебным заведениями, заставляет искать средства дополнительного анализа и обработки этих данных, для возможности дальнейшее прогнозировании и принятия адекватных решений в различных ситуациях. Наиболее распространенными базами данных являются реляционные базы данных, которые до сих пор остаются самыми распространенными и популярными в мире. Применение инструментов для работы с OLAP(*online analytical processing* аналитическая обработка в реальном времени) и DM(*Data Mining*, Интеллектуальных анализ данных) обладают широкими возможностями с точки зрения анализа данных, однако эти инструменты не всегда доступны в учебных заведениях. Поэтому разработчикам приходится реализовывать многие алгоритмы стандартными средствами реляционной базы данных.

При анализе данных часто возникает необходимость производить сравнение не только полей, но множества записей (кортежей).

В современных реляционных базах данных практически отсутствуют реализации операторов сравнения многомерных множеств, поэтому для сравнения множеств(отношений) программисту приходиться каждый раз реализовывать механизм сравнения самостоятельно через элементарные операции с использованием временных таблиц, функций, курсоров.

Анализ последних достижений и публикаций

В реляционной алгебре для сравнения множеств существует оператор «деление», определенный Коддом [1]. Причем данный оператор не является элементарным и выражается через другие операторы «проекции», «вычитания» и «произведения». Однако при его использовании мы можем получить только собственное подмножество множества. Причем в классическом определении оператор деления определен для бинарного и унарного отношений, так называемое, «Small Divide» [2]. Для деления «многомерных» отношений используется «Great divide» [3]. Очень подробно оператор деления во всех его проявлениях рассмотрен в диссертации [4]. Операторы, позволяющие получать равные множества или подмножества, были введены позднее К. Дейтом и Х. Дарвеном в «Третьем манифесте» [5] – SUBSET (подмножество) SUBSETEQ (равенство), которые еще не реализованы в современных реляционных СУБД.

Цель статьи

Рассмотрим возможность реализации данных операторов(SUBSET, SUBSETEQ), элементарными операторами реляционной алгебры с возможностью в дальнейшем получить эквивалентное выражение языка SQL.

Основная часть статьи

Определим базовые понятия деления и используемые обозначения.

Определение 1.

«Great Divide»(большое деление) – будем называть выражение вида $r_1 \div r_2 = r_3$,

где отношения r_1, r_2, r_3 определяются соответственно схемами $S_1(X \cup Y)$, $S_2(Y \cup Z)$, $S_2(X \cup Z)$, таких что $X = \{x_1 \dots x_k\}, Y = \{y_1 \dots y_l\}, Z = \{z_1 \dots z_m\}$.

Определение 2.

«Обобщенным делением» [6] будем называть выражение

$$r_1 \div r_2 = (\pi_X(r_1) \times \pi_Z(r_2)) - \pi_{X \cup Z}((\pi_X(r_1) \times r_2) - (r_1 \times \pi_Z(r_2))).$$

Данное выражение является громоздким и общим для преобразования в эквивалентное SQL выражение, поэтому предлагается использовать следующее определение оператора деления, в котором используется реляционная алгебра с расширениями (оператор существования $\exists$).

Определение 3.

«Делением с расширениями» будем называть выражение

$$r_1 \div r_2 = \sigma_{\exists(\pi_Y((\sigma_{Z=t_2}(r_2))) - \pi_Y((\sigma_{X=t_1}(r_1))))}(\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)),$$

$$\forall\ t_1 \in (\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)) \text{ и } t_1 \in (\pi_X(r_1)),\ t_2 \in (\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)) \text{ и } t_2 \in (\pi_Z(r_2))$$

При использовании оператора деления мы можем получить только собственное подмножество, для отношений r_1, r_2 относительно $Y = \{y_1 \dots y_l\}$, удовлетворяющих следующему условию $\pi_Y(\sigma_{X=t_1}(r_1)) \subseteq \pi_Y(\sigma_{Z=t_2}(r_2))$. «Деление с расширениями» легко преобразуется в SQL выражение с двумя вложенными подзапросами с оператором EXISTS, так называемым «SQL Double Double».

Пошаговое нахождение равных множеств описано Дейтом [7], но данный подход требует создания большого количества промежуточных временных отношений (таблиц), что часто очень неудобно. Поэтому необходимо получить общий вид выражения, позволяющего получить равные подмножества.

Определение 4.

«Обобщенным эквивалентным делением» будем называть выражение

$$r_1 \div_= r_2 = (\pi_X(r_1) \times \pi_Z(r_2)) - \pi_{X \cup Z}((\pi_X(r_1) \times r_2) - (r_1 \times \pi_Z(r_2)))$$

$$-\pi_{X \cup Z}((r_1 \times \pi_Z(r_2)) - (\pi_X(r_1) \times r_2))$$

Данное выражение получается из «обобщенного деления».

Определение 5.

«Эквивалентным делением с расширениями» будем называть выражение

$$r_1 \div_= r_2 = \sigma_{\exists(\pi_Y(\sigma_{X=t_1}(r_1)) \cup \pi_Y(\sigma_{Z=t_2}(r_2)) - \pi_Y(\sigma_{X=t_1}(r_1)) \cap \pi_Y(\sigma_{Z=t_2}(r_2)))}(\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)),$$

$$\forall\ t_1 \in (\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)) \text{ и } t_1 \in (\pi_X(r_1)),\ t_2 \in (\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2)) \text{ и } t_2 \in (\pi_Z(r_2))$$

Условие равных подмножеств также можно представить тождественным выражением

$$r_1 \div_= r_2 = \sigma_{\exists(\pi_Y(\sigma_{X=t_1}(r_1)) - \pi_Y(\sigma_{Z=t_2}(r_2))) \vee \exists(\pi_Y(\sigma_{Z=t_2}(r_2)) - \pi_Y(\sigma_{X=t_1}(r_1)))}(\pi_{X \cup Z}(r_1 \triangleright\triangleleft r_2))$$

Используя определения 4 или 5, получим равные подмножества для отношений r_1, r_2 относительно $Y = \{y_1 \dots y_l\}$, удовлетворяющих следующему условию

$$\pi_Y(\sigma_{X=t_1}(r_1)) = \pi_Y(\sigma_{Z=t_2}(r_2)).$$

Определение 6.

«Обобщенным делением для собственных подмножеств» будем называть выражение

$$r_1 \div_\subset r_2 = (\pi_X(r_1) \times \pi_Z(r_2)) - \pi_{X \cup Z}((\pi_X(r_1) \times r_2) - (r_1 \times \pi_Z(r_2)))$$

$$-\pi_{X \cup Z}((r_1 \times \pi_Z(r_2)) - ((r_1 \times \pi_Z(r_2)) - (\pi_X(r_1) \times r_2)))$$

Определение 7.

«Делением для собственных подмножеств с расширениями» будем называть выражение

$$r_1 \div_{\subset} r_2 = \sigma_{\exists\left(\pi_Y\left(\sigma_{Z=t_2}(r_2)\right)-\pi_Y\left(\sigma_{X=t_1}(r_1)\right)\right)\vee\exists\left(\pi_Y\left(\sigma_{X=t_1}(r_1)\right)-\pi_Y\left(\sigma_{Z=t_2}(r_2)\right)\right)}\left(\pi_{X\cup Z}(r_1 \triangleright\triangleleft r_2)\right),$$

$$\forall\ t_1 \in \left(\pi_{X\cup Z}(r_1 \triangleright\triangleleft r_2)\right) \text{ и } t_1 \in \left(\pi_X(r_1)\right),\ t_2 \in \left(\pi_{X\cup Z}(r_1 \triangleright\triangleleft r_2)\right) \text{ и } t_2 \in \left(\pi_Z(r_2)\right)$$

Используя определения 6 или 7, получим собственные подмножества для отношений r_1, r_2 относительно $Y = \{y_1 \ldots y_l\}$, удовлетворяющих следующему условию $\pi_Y\left(\sigma_{X=t_1}(r_1)\right) \subset \pi_Y\left(\sigma_{Z=t_2}(r_2)\right)$.

Рассмотрим примеры SQL выражений.

В качестве примера рассмотрим две таблицы

```
CREATE TABLE [R1] (
        [A] [varchar] (50) NOT NULL,
        [B] [varchar] (50) NOT NULL
)
CREATE TABLE [R2] (
        [B] [varchar] (50) NOT NULL,
        [C] [varchar] (50) NOT NULL
)
```

Для деления, как отмечено выше, используется «SQL Double Double», который описывается следующим выражением

```
SELECT DISTINCT r1.A, r2.C FROM r1, r2
        WHERE NOT EXISTS(SELECT 1 FROM r2 T2
                                WHERE T2.C = R2.C
                                AND NOT EXISTS(SELECT 1 FROM r1 T1
                                        WHERE T1.A=r1.A and T2.B=T1.B));
```

Согласно определению 5 для получения равных подмножеств получаем следующее выражение

```
SELECT DISTINCT r1.A, r2.C FROM r1, r2
        WHERE NOT EXISTS(SELECT 1 FROM r2 T2
                                WHERE T2.C = R2.C
                                AND NOT EXISTS(SELECT 1 FROM r1 T1
                                        WHERE T1.A=r1.A and T2.B=T1.B))
        AND NOT EXISTS(SELECT 1 FROM r1 T1
                                WHERE T1.A = R1.A
                                AND NOT EXISTS(SELECT 1 FROM r2 T2
                                        WHERE T2.C=r2.C and T2.B=T1.B));
```

Согласно определению 7 для получения собственных подмножеств получаем следующее выражение, причем выражение будет отличаться от предыдущего только вторым условием, которое преобразуется из NOT EXISTS в EXISTS

```
SELECT DISTINCT r1.A, r2.C FROM r1, r2
        WHERE NOT EXISTS(SELECT 1 FROM r2 T2
                                WHERE T2.C = R2.C
                                AND NOT EXISTS(SELECT 1 FROM r1 T1
                                        WHERE T1.A=r1.A and T2.B=T1.B))
        AND EXISTS(SELECT 1 FROM r1 T1
                                WHERE T1.A = R1.A
                                AND NOT EXISTS(SELECT 1 FROM r2 T2
                                        WHERE T2.C=r2.C and T2.B=T1.B));
```

Примеры приведены для таблиц, содержащих по одному атрибуту; если при сравнении используется несколько атрибутов, то во все запросы добавляются необходимые атрибуты

T1.A= T2.A (A_1= T2.A_1 and A_2= T2.A_2 and … and T2.A_k=T2.A_k), T1.B= T2.B (B_1= T2.B_1 and B_2= T2.B_2 and … and T2.B_n=T2.B_n), T1.C= T2.C (C_1= T2.C_1 and C_2= T2.C_2 and … and T2.C=T2.C_n)

Выводы

Применение реляционного деления может быть полезным при сложном анализе данных в реляционных СУБД, как разработчикам, так и администраторам. Раскрытие механизмов построения запроса с использованием реляционного деления позволяет упростить и ускорить написание SQL запросов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. CODD, E. F. 1972. Relational completeness of data base sublanguages. In Courant Computer Science Symposia No. 6: Data Base Systems. Prentice-Hall, New York, pp. 67-101.
2. Мейер Д. Теория реляционных баз данных. – М.: «Мир». – 1987. – 608 с.
3. Hugh Darwen and Chris Date. Into the Great Divide. In Chris Date and Hugh Darwen, editors, Relational Database: Writings 1989–1991, pages 155–168. Addison-Wesley, Reading, Massachusetts, USA, 1992.
4. Rantzau Ralf. Query Processing Concepts and Techniques for Set Containment Tests Dissertation 2003[Электронный ресурс], Режим доступа к источнику – (http://elib.uni-stuttgart.de/opus/volltexte/2004/1619/)
5. C. J. Date, Hugh Darwen. “Foundation for Object/Relational Databases: The Third Manifesto”, Addison-Wesley Pub Co; (June 1998)
6. Robert Demolombe. Generalized Division for Relational Algebraic Language. Information Processing Letters, 14(4): pages 174–178, 1982.
7. К. Дейт. Введение в системы баз данных. 7-е изд. СПб.: Вильямс, 2001. 1072 с.: ил.