Математические структуры и моделирование
2004, вып. 13, с. 161–169

УДК 519.6

ВЫПОЛНЕНИЕ СВОЙСТВА СОЕДИНЕНИЯ БЕЗ ПОТЕРЬ НА НЕКОТОРЫХ ТИПАХ СХЕМ БАЗ ДАННЫХ

Д.В. Малыгин

Building a correct query is of great importance in query-generating systems, and particularly, in systems that translate natural language queries to SQL-queries. It is sufficient to bound a set of relations, which are addressed in the query, to confront only objects related in the data domain. This problem is solved by execution of nonloss join on the set of relations forming the query. In this article a special case of nonloss join execution is considered - nonloss joins on the "star" and "snowflake" schemas, that are used in OLAP applications.

Введение

Важное значение в системах автоматической генерации запроса и, в частности, систем преобразования естественноязыкового (ЕЯ)-запроса в SQL-запрос, принимает вопрос корректности формируемого запроса. Требуется ограничить множество отношений, участвующих в запросе, для того, чтобы не были сопоставлены друг другу объекты, не связанные в предметной области. Указанную проблему решает выполнение на множестве отношений запроса свойства соединения без потерь.

В статье рассматривается частный случай выполнения свойства соединения без потерь при наложении определенных условий на схему. Как частный случай, рассматривается выполнение свойства соединения без потерь на типах схем «звезда» (star schema) и «снежинка» (snowflake schema), применяемых в базах данных OLAP.

1. Случай с функциональными зависимостями

Свойство соединения без потерь информации применяется для множества отношений в следующих случаях:

При проектировании схемы реляционной базы данных и проведении декомпозиции исходного отношения требуется исполнение данного свойства для обеспечения восстанавливаемости исходного отношения из декомпозиции.

E-mail: malygin_dv@mail.ru

Институт математики им. В.А.Стеклова СО РАН, лаборатория МППИ

При построении запросов, а также автоматизации формирования запроса также необходимо выполнение свойства для множества отношений запроса. Особенно важной проверка свойства становится при автоматизации построения запроса (в частности, при преобразовании ЕЯ-запроса в SQL-запрос).

При построении запросов также необходимо выполнение свойства для множества отношений запроса.

При этом операция естественного соединения не вырождается в операцию прямого произведения, что не дает получить в результате запроса кортежи, отсутствовавшие в исходном отношении.

Таким образом, полезной становится информация о заведомом исполнении данного свойства для некоторых распространенных типов схем баз данных.

Обозначения. Обозначим множество атрибутов отношения R как $\{R\}$. Обозначим также первичный ключ отношения R как K.

Предложение 1. *Выберем произвольную схему базы данных с множеством отношений* $R = \langle R_1, \ldots, R_n \rangle$.

Выберем множество $R^* \subset R$, $R^* = \langle R_1^*, \ldots, R_m^* \rangle$ *таким, что любое отношение из* R^* *не связано с любым другим отношением связью «один-ко-многим», т.е. для каждого* R_i^* *из* R^* *не существует отношения* R_k *из* R *такого, что* $K_i^* \subset \{R_k\}$, *но* K_i^* *не является подмножеством* K_k.

Если

1. *Для любых двух отношений* R_i *и* R_k *можно найти последовательность отношений* $\langle R_{i1}, \ldots, R_{in} \rangle$, $R_{i1} = R_i$, $R_{in} = R_k$ *такую что каждые два соседних элемента последовательности связаны ограничением внешнего ключа, т.е. для соседних отношений* R_{ik} *и* $R_{i(k+1)}$, *где* $K_{ik} \subset \{R_{i(k+1)}\}$ *или* $K_{i(k+1)} \subset \{R_{ik}\}$.

2. *Все отношения из* R^* *связаны между собой отношением* $1:1$, *т.е. для каждого* R_i^* *из* R^* *существует* R_k^* *из* R^* *такое, что* $K_i^* = K_k^*$,

то множество отношений схемы обладает свойством соединения без потерь.

Доказательство.

Выберем любое отношение R' из R^*.

Для каждого отношения R_i из R можно найти последовательность отношений $\{R_{i1}, \ldots, R_{in}\}$, $R' = R_{i1}$, $R_{in} = R_i$ такую, что каждый предыдущий элемент последовательности связан с последующим связью «много-к-одному» или «один-к-одному», т.е. для отношений R_{ik} и $R_{i(k+1)}$ ключ K_{ik} лежит в $\{R_{i(k+1)}\}$.

Действительно, если R_i принадлежит R^*, то такая последовательность примет вид $\{R', R_i\}$ (по п.1 условия).

Если же R_i принадлежит $R \setminus R^*$, тогда возможность построения такой последовательности обеспечивается п.1 условия, а тип связи между двумя соседними элементами последовательности обеспечивается из выбора множества R^* (оно не содержит отношений, связанных с другими связью «один-ко-многим»).

Выберем последовательно все отношения из множества $R \setminus R'$. Как показано ранее, для каждого такого отношения R_i существует последовательность

$\langle R_{i1},\ \ldots,\ R_{in}\rangle$, $R_{i1} = R_i$, $R_{in} = R'$, такая что каждый предыдущий элемент последовательности связан с последующим связью «один-ко-многим» или «один-к-одному».

Согласно алгоритма проверки свойства соединения без потерь [2] выпишем таблицу проверки свойства:

	A_1	$\ldots$	A_n
R_1			
$\ldots$			
R'			
$\ldots$			
R'_n			

На пересечении строки, соответствующей отношению R_i и столбца, соответствующего атрибуту A_k, стоит «1», если $A_k \in \{R_i\}$, иначе «0».

Последовательно пройдем по всем элементам данной последовательности. Текущий элемент обозначим как R_{ik}, последующий – как $R_{i(k+1)}$. Из свойств данной последовательности вытекает, что $K_{ik} \subset \{R_{i(k+1)}\}$. Таким образом, существует функциональная зависимость $K_{i(k+1)} \to K_{ik}$, а значит и $K_{i(k+1)} \to \{R_{ik}\}$. Учитывая предыдущие отношения в данной последовательности, получим, что $K_{i(k+1)} \to \{R_{i1},\ \ldots,\ R_{ik}\}$.

Согласно [2], получим, что т.к. на пересечении строк R_{ik} и $R_{i(k+1)}$ и столбцов, относящихся к атрибутам $K_{i(k+1)}$, стоят «1», то ячейки строки $R_{i(k+1)}$, соответствующие ячейкам, имеющим значение «1» в строке R_{ik} (это ячейки, соответствующие атрибутам $\{R_{i1},\ \ldots,\ R_{ik}\}$), также примут значение «1».

Таким образом, последовательно выбрав все отношения из $R \setminus R'$, построив последовательности от данных отношений до R', получаем, что все ячейки строки R' имеют значение «1». Значит, согласно алгоритма [2], множество отношений схемы обладает свойством соединения без потерь. ■

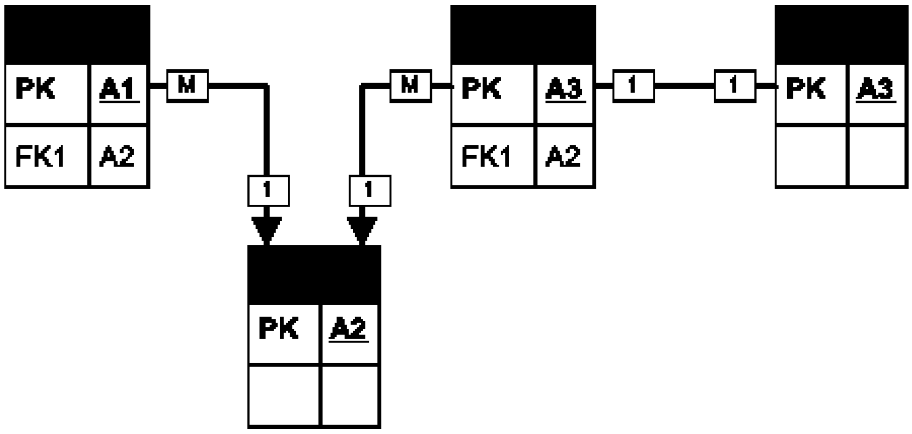
Замечание 1. Можно рассмотреть схему базы данных как граф. Вершинами графа можно принять отношения. Если два отношения связаны между собой ограничением внешнего ключа, то можно рассматривать эту связь как дугу между вершинами. Причем, если два отношения связаны связью «один-к-одному», то дугу можно считать ненаправленной, а если «один-ко-многим» или «многие-к-одному», то дугу можно считать направленной.

В данной терминологии утверждение можно переформулировать следующим более очевидным способом:

Предложение 2. *Если*

1. *Схема базы данных представляет собой связный граф,*
2. *Множество таких отношений схемы, которые не связаны с любым другим отношением связью «один-ко-многим», связно,*

то множество отношений схемы обладает свойством соединения без потерь.



Следствие 1. *Если граф схемы базы данных представляет собой дерево, в котором дуга «предок-потомок» представляет собой связь типа* $M : 1$ *или* $1 : 1$*, то множество отношений схемы обладает свойством соединения без потерь.*

Доказательство. Действительно, из условия следствия вытекает первое условие утверждения 2, а также и второе условие (т.к. множество, упомянутое во втором условии утверждения 2, представляет собой корень данного дерева, а множество из одного элемента всегда связно).

■

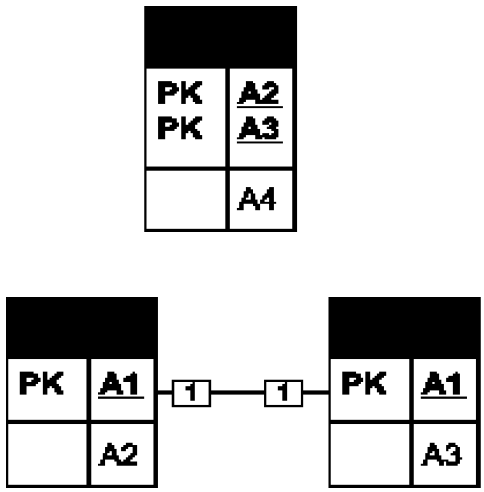
2. Примеры

Пример 1. Покажем на контрпримере необходимость второго условия утверждения 1. Действительно, рассмотрим схему базы данных, приведенную на рисунке.

В данном случае не выполняется второе условие утверждения, т.е. множество отношений $\{R_1, R_3, R_4\}$, не связанных с любым другим отношением связью «один-ко-многим», не связно.

Согласно алгоритму проверки свойства соединения без потерь [2] выпишем таблицу проверки свойства:

	A_1	A_2	A_3
R_1	1	1	
R_2		1	
R_3		1	1
R_4			1



После преобразований по алгоритму таблица примет вид:

	A_1	A_2	A_3
R_1	1	1	
R_2		1	
R_3		1	1
R_4		1	1

В данной таблице нет строк, состоящих только из «1», следовательно, множество отношений схемы не обладает свойством соединения без потерь.

Пример 2. Покажем на контрпримере, что утверждение, обратное утверждению 1, неверно, т.е. из свойства соединения без потерь не следует связность графа отношений.

Действительно, рассмотрим схему базы данных, приведенную на рисунке.

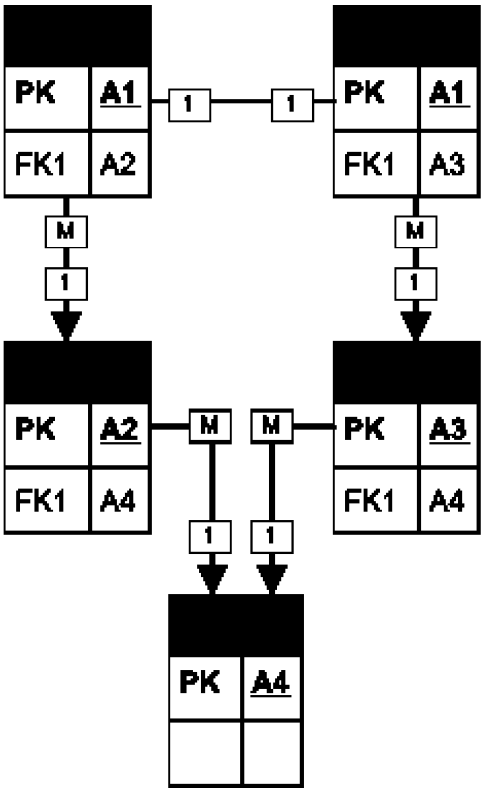
Согласно алгоритму проверки свойства соединения без потерь [2] выпишем таблицу проверки свойства:

	A_1	A_2	A_3	A_4
R_1	1	1		
R_2			1	
R_3		1	1	1

После преобразований по алгоритму таблица примет вид:

	A_1	A_2	A_3	A_4
R_1	1	1	1	1
R_2	1	1	1	1
R_3		1	1	1

Т.к. в таблице есть строки, состоящие только из «1», то множество отношений схемы обладает свойством соединения без потерь, хотя граф связного множества отношений не связен.



Пример 3. Приведенная на рисунке схема базы данных обладает свойством соединения без потерь, поскольку множество $\{R_1, R_2\}$, связно.

Согласно алгоритму проверки свойства соединения без потерь выпишем таблицу проверки свойства:

	A_1	A_2	A_3	A_4
R_1	1	1		
R_2	1		1	
R_3		1		1
R_4			1	1
R_5				1

После преобразований по алгоритму таблица примет вид:

	A_1	A_2	A_3	A_4
R_1	1	1	1	1
R_2	1	1	1	1
R_3		1		1
R_4			1	1
R_5				1

Как видно из таблицы, множество отношений схемы базы данных удовлетворяет свойству соединения без потерь.

Замечание 1. Полученное утверждение может быть применено для определения выполнения свойства соединения без потерь. Для некоторых распространенных типов схем баз данных можно заранее определить выполнение данного свойства.

В качестве примера рассмотрим типы схем «звезда» (star schema) и «снежинка» (snowflake schema), применяемые в базах данных OLAP [3].

Тип схемы данных «звезда» представляет собой схему отношений, в которой одно отношение является выделенным и называется «фактографическим» (fact table), данное отношение связано со всеми остальными отношениями схемы («отношениями измерений», dimension table) связью «многие-к-одному»; остальные отношения схемы связаны ограничением внешнего ключа только с фактографическим отношением.

Тип схемы данных «снежинка» представляет собой тип «звезда» со следующим расширением: отношения измерений в свою очередь могут быть связаны ограничением внешнего ключа с другими отношениями измерений, однако от каждого отношения измерений до фактографического отношения существует только одна последовательность связей между отношениями, причем каждое предыдущее отношение в данной последовательности связано с последующим связью «один-ко-многим».

Из следствия 1 легко можно заключить, что схемы базы данных, относящиеся к типу «звезда» или «снежинка», заведомо обладают свойством соединения без потерь.

3. Случай с многозначными зависимостями

Предложение 3. *Выберем произвольную схему базы данных с множеством отношений* $R = \langle R_1, \ldots, R_n \rangle$.

Выберем множество $R^* \subset R$, $R^* = \langle R_1^*, \ldots, R_m^* \rangle$ *таким, что любое отношение из* R^* *не связано с любым другим отношением связью «один-ко-многим», т.е. для каждого* R_i^* *из* R^* *не существует отношения* R_k *из* R *такого, что* $K_i^* \subset \{R_k\}$, *но* K_i^* *не лежит в* K_k.

Если

1. *Для любых двух отношений* R_i *и* R_k *можно найти последовательность отношений* $\langle R_{i1}, \ldots, R_{in} \rangle$, $R_{i1} = R_i$, $R_{in} = R_k$ *такую что каждые два соседних элемента последовательности связаны ограничением внешнего ключа, т.е. для соседних отношений* R_{ik} *и* $R_{i(k+1)}$ $K_{ik} \subset \{R_{i(k+1)}\}$ *или* $K_{i(k+1)} \subset \{R_{ik}\}$.

2. *Существует отношение* R' *из* R^* *такое, что для любого отношения* R_i *из* R^* *существует многозначная зависимость* $\oslash \to\to K'(K_i)$,

то множество отношений схемы обладает свойством соединения без потерь.

Доказательство. Доказательство утверждения аналогично доказательству утверждения 1.

Для каждого отношения R_i из $R \backslash R^*$ существует отношение R_i^* из R^* и можно найти последовательность отношений $\{R_{i1}, \ldots, R_{in}\}$, $R_i^* = R_{i1}$, $R_{in} = R_i$ такую,

что каждый предыдущий элемент последовательности связан с последующим связью «много-к-одному» или «один-к-одному», т.е. для отношений R_{ik} и $R_{i(k+1)}$ $K_{ik} \subset \{R_{i(k+1)}\}$.

Возможность построения такой последовательности обеспечивается п.1 условия, а тип связи между двумя соседними элементами последовательности обеспечивается из выбора множества R^* (оно не содержит отношений, связанных с другими связью «один-ко-многим»).

Выберем последовательно все отношения из множества $R \setminus R^*$. Как показано ранее, для каждого такого отношения R_i существует последовательность $\langle R_{i1}, \ldots, R_{in}\rangle$, $R_{i1} = R_i$, $R_{in} = R_i^*$ (R_i^* принадлежит R^*), такая что каждый предыдущий элемент последовательности связан с последующим связью «один-ко-многим» или «один-к-одному».

Согласно алгоритма проверки свойства соединения без потерь [2] выпишем таблицу проверки свойства:

	A_1	$\ldots$	A_n
R_1			
$\ldots$			
R'			
$\ldots$			
R'_n			

На пересечении строки, соответствующей отношению R_i и столбца, соответствующего атрибуту A_k, стоит «1», если $A_k \in \{R_i\}$, иначе «0».

Последовательно пройдем по всем элементам данной последовательности. Текущий элемент обозначим как R_{ik}, последующий — как $R_{i(k+1)}$. Из свойств данной последовательности вытекает, что $Kik \subset \{R_{i(k+1)}\}$. Таким образом, существует функциональная зависимость $K_{i(k+1)} \to K_{ik}$, а значит и $K_{i(k+1)} \to \{R_{ik}\}$. Учитывая предыдущие отношения в данной последовательности, получим, что $K_{i(k+1)} \to \{R_{i1}, \ldots, R_{ik}\}$.

Согласно [2], получим, что т.к. на пересечении строк R_{ik} и $R_{i(k+1)}$ и столбцов, относящихся к атрибутам $K_{i(k+1)}$, стоят «1», то ячейки строки $R_{i(k+1)}$, соответствующие ячейкам, имеющим значение «1» в строке R_{ik} (это ячейки, соответствующие атрибутам $\{R_{i1}, \ldots, R_{ik}\}$), также примут значение «1».

Таким образом, перебрав все отношения из $R \setminus R^*$, получим, что каждому отношению R_i^* из R^* соответствует множество отношений $< R_{i1}, \ldots, R_{in} >$, при этом $K_i^* \to \{R_{i1}\} \cup \cdots \cup \{R_{in}\}$ и строка, соответствующая отношению R_i^*, содержит «1» в ячейках, соответствующих столбцам $\{R_{i1}\} \cup \cdots \cup \{R_{in}\}$.

Теперь возьмем в качестве текущей строки таблицы проверки свойства соединения без потерь строку, соответствующую отношению R' и выберем все отношения из $R^* \setminus R'$. Для каждого R_i из $R^* \setminus R'$ из [2] и п.2 условия следует, что в таблицу будет добавлена строка, содержащая «1» в столбцах, соответствующих множествам атрибутов K' и K_i. Далее, т.к. имеются соответствующие функциональные зависимости, получаем значение «1» в колонках, значения которых равны «1» в текущей строке таблицы и в колонках, соответствующих атрибутам $\{R_{i1}\} \cup \cdots \cup \{R_{in}\}$.

Таким образом, последовательно пройдя все отношения из $R^* \setminus R'$, получим что все ячейки текущей строки таблицы имеют значение «1». Значит, согласно алгоритма [2], множество отношений схемы обладает свойством соединения без потерь.

■

4. Заключение

В данной работе рассмотрено выполнение свойства соединения без потерь для схем баз данных заранее заданного типа, показано заведомое выполнение свойства для некоторых типов схем, часто используемых в практической деятельности.

Схема базы данных рассматривается как граф и свойство соединения без потерь ставится в зависимость от связности на графе множества отношений, не связанных с другими отношениями связью «один-ко-многим». Также рассматривается выполнение свойства при выполнении более слабых посылок, с использованием многозначных зависимостей.

Дальнейшее развитие работа может получить в нахождении других типов схем, обладающих указанным свойством.

Литература

1. Мейер Д., *Теория реляционных баз данных.* М.: Мир, 1987.
2. Ульман Дж., *Основы систем баз данных.* М.: Финансы и статистика, 1983
3. Чаудхури С., Дайал У., Ганти В., *Технология баз данных в системах поддержки принятия решений.* Открытые системы, №01/2002 (www.citforum.ru \ consulting \ BI \ bd_decision \ )

!