

**С. В. Зыкин**, д-р техн. наук, проф., e-mail: szykin@mail.ru,  
Институт математики им. С. Л. Соболева СО РАН,  
**В. С. Зыкин**, аспирант, e-mail: vszykin@mail.ru,  
Омский государственный технический университет

## Коммутативные преобразования в базе данных при редактировании многотабличных запросов

*Рассматриваются теоретические аспекты формирования приложений для редактирования данных в базах данных реляционного типа. Эта актуальная проблема до сих пор не получила удовлетворительного решения. Приложение должно работать с представлением данных — таблицей, сформированной из множества отношений базы данных с использованием запросов типа "соединение—селекция—проекция". Требованиям к таким приложениям является реализация функций редактирования данных: дополнение, удаление и модификация кортежей в таблице. В соответствии с выполненным редактированием приложение должно преобразовать исходную базу данных так, чтобы преобразование удовлетворяло условию коммутативности. Для указанных преобразований получены формулы реляционной алгебры, доказана корректность преобразований.*

**Ключевые слова:** реляционная база данных, коммутативные преобразования, реляционная алгебра, редактирование представлений

### Введение

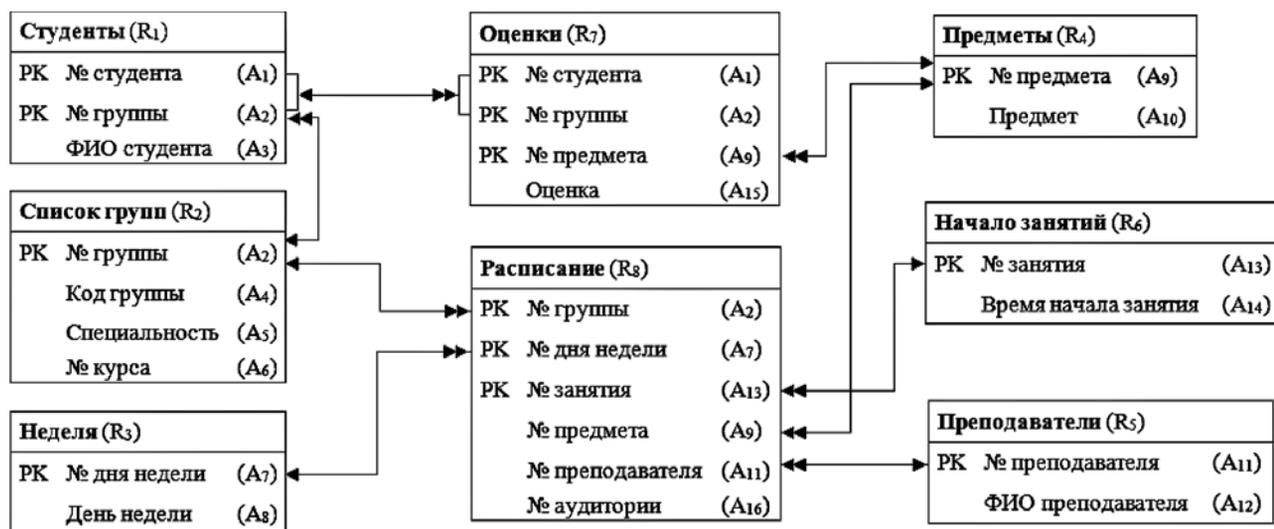
Представления (*view*) в базах данных (БД) играют важную роль как средство структурирования информации для обеспечения потребностей конкретных пользователей. Доступ к чтению через представления легко реализуется средствами языков запросов, тогда как обновление данных в БД с помощью представлений является довольно сложной проблемой. Для разработчиков программного обеспечения — систем управления базами данных (СУБД) — редактирование данных с помощью представлений является актуальной проблемой, поскольку соответствующий инструментарий существенно сократит время разработки типовых приложений.

Ограничения на обновления представлений данных присутствуют, начиная с первого стандарта языка SQL (Structured Query Language). Суть их сводится [1] к необходимости СУБД "для каждой строки представления найти соответствующую строку в исходном отношении, а для каждого обновляемого столбца представления — соответствующий столбец в исходной таблице (отношении)". При этом, "если представление пропускает столбец исходной таблицы (отношения) с ограничением NOT NULL без спецификации DEFAULT, то нельзя выполнить вставку но-

вой строки в таблицу (отношение) посредством данного представления. Причина в том, что в этом случае нет способа указать значение данных для пропущенного столбца".

Большинство коммерческих СУБД лишь частично ослабляют базовые ограничения стандарта SQL [2–6]. Например, выделенные столбцы в представлении могут быть предназначены только для просмотра, попытка редактирования таких столбцов блокируется, тогда как значения в других столбцах можно изменять в пределах заданных ограничений на соответствующие столбцы в исходном отношении.

Наиболее продвинутая технология редактирования многотабличных представлений реализована в Microsoft Access. В режиме конструктора формируется многотабличный запрос (представление), в котором действительно допускаются операции дополнения, удаления и модификации кортежей. Однако имеются существенные недостатки. Например, в запросе требуется присутствие ключевых атрибутов для связи между отношениями. Обычно эти атрибуты — номера объектов, которые лишены семантики, что создает проблемы при вводе новых значений. Имеются очевидные ошибки, например, в представлении "Расписание занятий по специальности" (пример 1) попытка переноса одного занятия с по-



Фрагмент схемы БД "Вуз"

недельника на среду путем редактирования дня недели привела к замене всех значений "Понедельник" на значение "Среда" в справочном отношении  $R_3$  (см. рисунок). При переходе из режима конструктора в режим SQL возможность редактирования запросов блокируется. Основная причина перечисленных и иных проблем: должно быть установлено соответствие кортежей редактируемой таблицы и кортежей отношений БД, что не всегда удается сделать. Для решения проблем редактирования представлений данных СУБД имеют механизм триггеров, но это позволяет решить только частную проблему частным способом.

Далее под представлением будем понимать таблицу данных, сформированную в результате выполнения запроса к БД. Таблицы исходной БД будем называть отношениями. Для демонстрации предлагаемых решений рассмотрим фрагмент схемы БД "Вуз" (см. рисунок). На схеме атрибуты первичного ключа отношения помечены символами РК, связи на схеме соответствуют ссылочным ограничениям целостности.

Именам отношений и атрибутов на схеме БД "Вуз" в скобках поставлены в соответствие символные обозначения.

## 1. Связанные работы

Работа [7] была одной из первых по данной тематике. В ней исследуются свойства отображений состояний базы данных в состояния представлений, основанных на морфизме. Представлена схема коммутативности композиции преобразований, исследования сопровождаются примерами. Однако отсутствуют какие-либо ал-

горитмы, соответствующие преобразованиям, и метод "константного дополнения", предложенный авторами [7], слишком ограничительный, чтобы быть полезным для создания технологии.

В работе [8] рассмотрена проблема автоматического перевода обновлений пользовательских представлений в соответствующие обновления исходной БД. Прежде всего, были исследованы свойства корректности таких преобразований и условия их существования. Условия были получены сначала с точки зрения расширения схемы и представления, а затем синтаксически с использованием функциональных зависимостей и ключей. Для определения операций преобразования БД были определены два графа: граф-трассировка и граф зависимостей представления. Проведенные исследования показали, что точные и уникальные переводы обновлений могут быть возможны только в довольно жестких условиях. Необходимость построения графов и условия корректности в виде исчисления предикатов делают затруднительным использование этого материала для формирования технологии. Кроме того, преобразования, описанные в работе [8], современные СУБД частично делают автоматически.

В работе [9] предпринята попытка решения данной проблемы в общем виде: допускается модификация кортежей во всех отношениях представления и работа с комбинациями представлений. Для решения проблемы неоднозначности интерпретации операций преобразования предлагается ряд эвристических критериев и правил, которые в общем случае не гарантируют корректного результата. В крайнем случае предполагается вмешательство пользователя в процесс модификации БД.

В работе [10] рассматривается обратная задача — обновление представлений данных. В общем случае обновление представлений неоднозначно и требует дополнительного представления, которое содержит всю информацию, опущенную из данного представления, и остается постоянным во время преобразований. Изучаются некоторые вычислительные проблемы, связанные с применением этой общей методологии в контексте реляционных баз данных. Показано, что поиск минимального дополнения к данному представлению является *NP*-полной задачей. Корректность преобразований проверяется тестированием, исследуется вычислительная сложность тестирования.

В работе [11] рассмотрена проблема перевода обновлений представлений в базы данных, где база данных и представление моделируются как абстракции данных. Абстракция данных состоит из набора состояний и набора примитивных операторов обновления, представляющих функции перехода состояния. Показано, как сложные программы обновления могут быть созданы из примитивных операторов обновления, и как программы обновления версий транслируются в программы обновления баз. Исследованы свойства эквивалентности и сериализуемости переходов между состояниями. В работе определено понятие дополнения и показано влияние постоянного дополнения на изменения в представлении.

В работе [12] изучается проблема трансляции обновления представления для реляционной модели данных, в которой базовые отношения могут содержать значения *null*. Предполагается наличие репрезентативного экземпляра, который считается правильным представлением всех данных в базе данных; класс изучаемых представлений состоит из полных проекций репрезентативного экземпляра. Рассматриваются только отдельные схемы баз данных, т. е. схемы, для которых глобальная согласованность подразумевается локальной согласованностью. Условие независимости отношений является сильно ограничительным. Так, на приведенном выше рисунке отношение "Расписание" содержит суперключ почти для всех отношений, следовательно, все они зависимы. Кроме того, для обеспечения корректности преобразования предполагается формирование расширения модифицируемого отношения до размеров универсума (отношения, определенного на всем множестве атрибутов), что само по себе является препятствием для практического использования такого подхода.

В работе [13] продолжены исследования свойств модификации представлений, основанных на "*constant complement approach*" [7]. Получены интересные теоретические результаты, в том числе применительно к множеству внешних схем.

Однако результаты пока далеки от практического использования.

Наиболее близка к нашей тематике работа [14], в которой обсуждается проблема корректности формирования представлений, основанных на реляционных операциях: соединение, селекция и проекция. Автором рассмотрена схема межмодельных преобразований, основанная на условиях коммутативности. Также рассмотрены пять критериев правильности выполненных преобразований, представлены варианты алгоритмов выполнения операций дополнения, удаления и модификации кортежей в алгоритмах. К недостаткам работы [14] можно отнести отсутствие модели для описания представлений, что является препятствием к созданию технологической цепочки преобразований. Рассмотренные примеры с командами SQL не заменяют отсутствующую модель. Кроме того, для обеспечения корректности преобразования требуется наличие ключевых атрибутов в представлениях. В реальных БД ключами чаще всего являются автоинкрементные атрибуты, в которых хранятся номера соответствующих объектов. Значения таких атрибутов недоступны для редактирования и не обладают какой-либо семантикой. Следовательно, работа пользователей с представлениями будет существенно затруднена.

В данной работе учтены перечисленные недостатки. Исследования доведены до уровня технологии.

## 2. Метод коммутативных преобразований данных

Построение межмодельных отображений начинается с выбора либо с формирования некоторой универсальной модели, которая имеет расширенные возможности в части описания и представления данных. В данной работе предполагается, что универсальная модель известна и полностью определена (реляционная). Далее формируется либо выбирается целевая (специализированная) модель данных, с которой и предполагается дальнейшая работа. Чаще всего целевая модель определяется посредством пользовательского описания данных. Конечной целью построения отображения является разработка программного обеспечения, поддерживающего интерфейс между исполняющей средой (программным обеспечением) для универсальной модели и исполняющей средой для целевой модели, в том числе для модели пользовательского представления. Существенной деталью построения является то, что пользователь должен иметь возможность корректно выполнять весь спектр операций над данными, не выходя за рамки своего приложения.

Модель информационной системы (модель данных) запишем в виде алгебраической системы:

$$\Omega = \langle M, D, O, P \rangle,$$

где  $M$  — логическая схема данных;  $D$  — совокупность допустимых состояний БД;  $O$  — набор операций для модели  $\Omega$ ;  $P$  — совокупность предикатов, ограничивающих допустимые состояния  $D$ ;  $M$  и  $D$  в совокупности являются носителем системы.

Модель  $\Omega$  будем считать исходной, а  $\Omega' = \langle M', D', O', P' \rangle$  — целевой (пользовательской). Следовательно, необходимо построение преобразования  $\Omega \Rightarrow \Omega'$ .

В работе [15] рассмотрен метод построения отображения моделей данных, основанный на свойстве коммутативности. В работе [16] получено обобщение метода на случай отсутствия биективности состояний исходной и целевой моделей данных. Такая ситуация является актуальной, когда пользовательское приложение взаимодействует не со всей БД, а только с ее частью. Спецификой данной работы является полностью обновляемое представление данных пользователя при активизации приложения: выполнение многотабличного запроса. Далее рассмотрим последовательность построения преобразования исходной модели  $\Omega$  в целевую  $\Omega'$ .

1. Формирование многотабличного запроса осуществляется в терминах стандартного языка запросов. В настоящее время таковым является язык SQL. Запрос будет записан компактнее, если воспользоваться реляционной алгеброй [17, 18], которая является теоретической основой языка SQL. Переход из одного состояния БД в другое осуществляется за счет выполнения операции редактирования: дополнение, удаление и модификация кортежа в каком-либо отношении.

*Определение 1.* Совокупность всех значений данных в БД, неизменных в течение некоторого промежутка времени, будем называть элементарным состоянием.

При работе с моделью  $\Omega'$  пользователь выполняет команду  $f'_p$ , которая переводит модель  $\Omega'$  из состояния  $d'_i \in D'$  в состояние  $d'_j \in D'$ . В программном обеспечении должны быть реализованы алгоритмы, выполняющие соответствующие преобразования модели  $\Omega$  из состояния  $d_i \in D$  в состояние  $d_j \in D$ .

*Определение 2.* Преобразование  $\Omega \Rightarrow \Omega'$  будем считать корректным, если выполнены условия коммутативности:

$$d_i - (Q) \rightarrow d'_i - (f'_p) \rightarrow d'_j,$$

$$d_i - (Alg_p) \rightarrow d'_i - (Q) \rightarrow d'_j,$$

где  $Q$  — многотабличный запрос;  $Alg_p$  — алгоритм преобразования исходной БД, соответствующий команде редактирования  $f'_p$ .

Другими словами, в состояние  $d'_j$  можно перейти двумя различными способами, но результат должен быть один и тот же. В определении 2 предполагается, что при выполнении преобразований состояние БД не изменяется другими приложениями.

3. Для исходной модели данных  $\Omega$  может быть задана совокупность ограничений на допустимые состояния. Для реляционной модели данных это, прежде всего, первичные ключи, внешние ключи и ограничения домена для атрибутов. Первичные ключи позволяют реализовать в БД функциональные зависимости [19]. Кроме того, функциональные зависимости могут быть реализованы с помощью уникального индекса и, в крайнем случае, за счет триггера. Теоретической основой для внешних ключей являются зависимости включения [20]. Эти зависимости реализуются за счет связей на схеме БД. Таким образом, ограничения целостности в совокупности определяют допустимые состояния данных и переходы между ними для исходной модели  $\Omega$ . Поскольку запрос  $Q$  определяет соответствующие состояния моделей  $\Omega$  и  $\Omega'$ , то соответствующие ограничения на допустимые состояния и переходы модели  $\Omega'$  также определяются этим запросом. Образы ограничений целостности должны быть достаточными для выполнения исходных ограничений модели  $\Omega$ : для целевой модели  $\Omega'$  могут быть заданы более "жесткие" ограничения, если реализация образов ограничений, являющихся необходимыми и достаточными, для модели  $\Omega'$  является затруднительной либо невозможной. В крайнем случае ограничением целостности может стать запрет на модификацию определенной группы данных.

4. Выполнение операции преобразования  $f'_p$  для модели  $\Omega'$  должно сопровождаться преобразованием состояния модели  $\Omega$ . Для этого в системе должен быть реализован алгоритм  $Alg_p$ , соответствующий операции  $f'_p$  и удовлетворяющий условию коммутативности. Таким образом, для каждой операции  $f'_p$  должен быть реализован свой алгоритм преобразования БД —  $Alg_p$ . Из этого следует, что набор операций  $f'_p$  должен быть строго регламентирован.

При разработке алгоритмов  $Alg_p$  основное внимание должно быть уделено корректности преобразований. Однако выполнение алгоритмов  $Alg_p$  предполагается при работе пользователя с приложением (*online*), следовательно, должно быть обеспечено приемлемое время их выполнения.

### 3. Редактирование многотабличных представлений данных

Рассмотрим формальную постановку задачи. Исходная реляционная база данных (РБД) пред-

ставлена в виде множества отношений  $R_1, R_2, \dots, R_k$ . Целевая модель — реляционная таблица данных, являющаяся результатом выполнения запроса к РБД:

$$Q = \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m])),$$

где  $\pi_{X_0}$  — операция проекции по множеству атрибутов  $X_0$ ;  $\sigma_F$  — операция селекции;  $F$  — логическое выражение на атрибутах отношений  $R'_1, R'_2, \dots, R'_m$ ;  $\bowtie$  — операция естественного соединения отношений РБД;  $R'_i[X_i]$  — краткая запись операции проекции отношения  $R'_i$  по атрибутам  $X_i$ . Атрибут  $A_j$  отношения  $R'_i$  принадлежит  $X_i$ ,  $i = 1, 2, \dots, m$ : если  $A_j \in X_0$ , и/или существует  $R'_l$ ,  $l \neq i$ :  $A_j \in \langle R'_l \rangle$ , и/или  $A_j \in \langle F \rangle$ , других атрибутов в  $X_i$  нет.

Рассмотрим соответствие обозначений. Схема данных исходной модели данных есть множество схем отношений БД:  $M = \{\langle R_1 \rangle, \langle R_2 \rangle, \dots, \langle R_k \rangle\}$ , где  $\langle R_i \rangle$  — все атрибуты отношения  $R_i$ ;  $M' = \bigcup_{i=1}^m \langle R'_i \rangle$  — заголовок таблицы, соответствующий результату выполнения запроса  $Q$ ;  $P$  — первичные и внешние ключи БД;  $P'$  — реализованные зависимости БД (образы первичных и внешних ключей);  $O$  и  $O'$  — базисные для БД операции дополнения, удаления и модификации кортежей в отношениях. Допустимые состояния  $D$  и  $D'$  определяются предикатами  $P$  и  $P'$  соответственно.

Условие коммутативности для запроса интерпретируется следующим образом: если пользователь удалил, добавил или модифицировал кортеж, то после повторной загрузки данных именно этот кортеж должен быть удален, добавлен или модифицирован. Остальные кортежи должны остаться без изменений с точностью до порядка их следования.

Рассмотрим требования к составу и структуре запроса  $Q$ .

1. Отношения  $R'_1, R'_2, \dots, R'_m$  должны иметь упорядочение по внешним ключам: главные отношения в последовательности стоят раньше, подчиненные — позже. Таким образом, должен существовать частичный порядок, в котором есть только одно отношение  $R'_m$ , не имеющее подчиненных отношений. Это отношение будет соответствовать семантике приложения: выполненные операции в приложении будут реализовываться только соответствующими операциями в  $R'_m$ . Далее это отношение будем называть целевым. Фактическое установление частичного порядка над отношениями не является принципиальным, однако алгоритмическая реализация преобразования БД в виде схемы итерирования отношений [21] в большинстве случаев позволяет

существенно сократить промежуточные результаты за счет внешних ключей и, как следствие, добиться сокращения времени работы алгоритма.

Сформулируем условия (ограничения) на атрибуты запроса  $Q$ . Для корректного выполнения операций должны быть выполнены следующие условия.

2.  $\langle R'_m \rangle \rightarrow X_0$ : атрибуты отношения  $R'_m$  функционально определяют атрибуты приложения  $X_0$ .

3.  $\langle R'_m \rangle \subseteq X_0 \cup X_1 \cup \dots \cup X_{m-1}$  — отсутствие свободных атрибутов в целевом отношении.

Представленные условия далее должны быть реализованы в инструментальной исполняющей среде при генерации пользовательских многотабличных приложений. С использованием фрагмента БД (см. рисунок) могут быть сгенерированы представления для различных приложений: "Ведомость на экзамен", "Сводная ведомость по предмету", "Сводная ведомость для группы", "Расписание занятий для группы", "Расписание занятий на день недели" и т. п. Каждый кортеж представления для перечисленных приложений имеет ровно один кортеж прообраз в целевом отношении, что упрощает процесс редактирования. Рассмотрим более сложный пример приложения "Расписание занятий по специальности", в котором одному кортежу представления соответствует несколько кортежей целевого отношения.

**Пример 1.** С использованием обозначений, данных на приведенном выше рисунке, запрос на расписание занятий по специальности может иметь следующий вид:

$$Q_1 = \pi_{X_0}(\sigma_{A_{10}=\text{Предмет}}(R_2[A_2 A_5 A_6] \bowtie R_3 \bowtie R_4 \bowtie R_5 \bowtie R_6 \bowtie R_8)),$$

где  $X_0 = A_5 A_6 A_8 A_{14} A_{12} A_{16}$ . При старте приложения значение параметра 'Предмет' выбирается из списка предметов (таблица  $R_4$ ). С помощью приложения, в основе которого используется представление  $Q_1$ , должна быть возможность дополнения и удаления занятия. Кроме того, должны быть доступны: перенос занятия на другое время и/или день недели, замена преподавателя, замена аудитории.

### 3.1. Операция удаления кортежа

Пусть в приложении кортеж  $u$  удален из таблицы, соответствующей запросу  $Q$ . Далее запрос и результат его выполнения будем обозначать одним и тем же символом  $Q$ .

В соединении отношений

$$R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m]$$

множество кортежей  $T$ , удовлетворяющих логическому выражению  $F$  и принимающих значения

кортежа  $u$  на атрибутах  $X_0$ , можно выразить формулой

$$T = \sigma_{F \wedge (X_0 = u)} (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m]),$$

где выражение  $(X_0 = u)$  означает равенство значений одноименных атрибутов. Тогда операция удаления кортежа  $u$  из таблицы  $Q$  сводится к удалению кортежей

$$R''_m = R'_m \setminus \pi_{\langle R'_m \rangle} (T) \quad (1)$$

в отношении  $R'_m$ , где символом  $\setminus$  помечена реляционная операция вычитания [18]. Покажем, что данное преобразование будет корректным (коммутативным). Это означает, что таблица

$$Q' = \pi_{X_0} (\sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R''_m [X_m]))$$

будет отличаться от таблицы  $Q$  только отсутствием одного кортежа  $u$ . Заметим, что при выполнении операции проекции дублированные кортежи удаляются.

**Теорема 1.** Операция удаления кортежа  $u \in Q$  коммутативна.

*Доказательство.* 1. Покажем, что кортеж  $u$  будет отсутствовать в  $Q'$ . Допустим обратное:  $u \in Q'$ . Следовательно, существует кортеж  $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R''_m [X_m])$  и  $t[X_0] = u$ . Поскольку  $R''_m [X_m] \subseteq R'_m [X_m]$ , то  $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$  и, следовательно, должен быть удален, что противоречит предположению.

2. Покажем, что не произошло удаление лишних кортежей, если  $u' \in Q$  и  $u' \neq u$ , тогда  $u' \in Q'$ . Предположим обратное: кортеж  $u'$  отсутствует в  $Q'$ . Следовательно, любой кортеж  $t' \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$ , для которого  $t'[X_0] = u'$ , должен удовлетворять условию для удаления  $t'[\langle R'_m \rangle] \in T$ . Однако в  $T$  содержатся только те кортежи, для которых существует  $t \in \sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_m [X_m])$  и  $t[X_0] = u$ . Следовательно, для любого кортежа  $t'$  найдется кортеж  $t$ , для которых  $t'[\langle R'_m \rangle] = t[\langle R'_m \rangle]$  и  $t'[X_0] \neq t[X_0]$ , что противоречит ограничению  $\langle R'_m \rangle \rightarrow X_0$ . Полученные противоречия доказывают коммутативность операции удаления кортежа.

Аналитическое выражение (1) является замещением алгоритма  $Alg_1$  для удаления кортежа в многотабличном запросе. В исполняющей среде это выражение достаточно для формирования команды SQL с последующим ее выполнением посредством какой-либо СУБД. Однако эффективное выполнение этой команды не гарантируется, в том числе по причине значительного объема промежуточных данных. В исполняющей среде

предлагается реализовать алгоритм итерирования отношений с динамическим выбором последовательности просматриваемых отношений [21].

### 3.2. Операция дополнения кортежа

По аналогии с предыдущей операцией рассмотрим аналитическое выражение операции дополнения кортежа в терминах реляционной алгебры. Пусть кортеж  $u$  дополняется к таблице  $Q$ , тогда в исходной БД должны быть выполнены следующие преобразования:

$$R''_m = R'_m \cup T, \quad (2)$$

где

$$T = \pi_{X_m} (\sigma_F (T' \bowtie u));$$

$$T' = \pi_Y (\sigma_{Z = u[Z] \& F} (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_{m-1} [X_{m-1}]));$$

$$Z = X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1});$$

$$Y = (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}),$$

если  $Z = \emptyset$ , то будем считать, что формула  $Z = u[Z]$  принимает значение "Истина",  $F$  — проекция формулы  $F$  на подпространство атрибутов  $\langle F \rangle = \langle F \rangle \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1})$  [22]. В случае операции дополнения решение может отсутствовать по следующим причинам: 1) может оказаться пустым множество  $T'$  — отсутствуют кортежи, с которыми может быть выполнено соединение (формула  $Z = u[Z]$  принимает значение "Ложь" для всех кортежей в  $T'$ ); 2) может оказаться пустым множество  $T$  — отсутствуют кортежи, на которых  $F$  принимает значение "Истина".

Покажем, что данное преобразование будет корректным (коммутативным), если  $T' \neq \emptyset$  и  $T \neq \emptyset$ . Это означает, что таблица

$$Q' = \pi_{X_0} (\sigma_F (R'_1 [X_1] \bowtie R'_2 [X_2] \bowtie \dots \bowtie R'_{m-1} [X_{m-1}] \bowtie R''_m [X_m]))$$

будет отличаться от таблицы  $Q$  только присутствием одного кортежа  $u$ .

**Теорема 2.** Если  $T' \neq \emptyset$  и  $T \neq \emptyset$ , то операция дополнения кортежа  $u \in Q$  коммутативна.

*Доказательство.* Все кортежи  $Q$  будут принадлежать  $Q'$ , поскольку в отношениях  $R'_i$ ,  $i = 1..m$ , кортежи не были удалены. Покажем, что кортеж  $u$ , и только он, будет добавлен в  $Q'$ . Предварительно заметим, что  $\langle R'_m \rangle = X_m$ , т. е. все атрибуты целевого отношения  $R'_m$  участвуют в реализации запросов  $Q$  и  $Q'$ . Допустим обратное, существует атрибут  $A_j \in \langle R'_m \rangle$  и  $A_j \notin X_m$ . Тогда по правилам формирования  $X_i$  должно быть выполнено  $A_j \notin X_i$ ,  $i = 1, \dots, m-1$ . Из ограничения 3 (см. разд. 3) следует, что  $A_j \in X_0$ , но такие атрибуты должны быть включены в  $X_m$  по построению. Следова-

тельно,  $\langle R'_m \rangle \subseteq X_m$ . Включение в обратную сторону следует из свойства операции проекции. Таким образом, имеем

$$X_m \subseteq X_0 \cup X_1 \cup X_2 \cup \dots \cup X_{m-1}$$

и

$$X_0 \subseteq X_1 \cup X_2 \cup X_3 \cup \dots \cup X_m.$$

С учетом полученных неравенств все кортежи  $T'$  на атрибутах  $Z$  совпадают с кортежем  $u[Z]$  по построению. При формировании  $T$  кортежи по атрибутам  $X_0 \setminus Z$  будут равны значениям соответствующих атрибутов  $u[X_0 \setminus Z]$ . По свойствам операций реляционной алгебры имеем:

$$\begin{aligned} Q' &= \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie R'_m[X_m])) = \\ &= \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie (R'_m[X_m] \cup T))) = \\ &= Q \cup \pi_{X_0}(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie T)). \end{aligned}$$

Множество кортежей  $\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_{m-1}[X_{m-1}] \bowtie T)$  по предположению не пусто, и все кортежи этого множества по атрибутам  $X_0$  совпадают с кортежем  $u$ . Следовательно, таблица  $Q'$  будет отличаться от  $Q$  только наличием одного и только одного кортежа  $u$ . Что и требовалось доказать.

**Замечание.** Для выполнения условия коммутативности в формуле (2) к  $R'_m$  достаточно добавить только один кортеж из множества  $T$ , однако необходимо дополнение всех кортежей множества  $T$ , что соединит вновь введенную информацию со всей существующей информацией в БД. Это обеспечивается выбором значений свободных атрибутов  $X_1 \cup X_2 \cup X_3 \cup \dots \cup X_m \setminus Z$  при формировании  $T'$ .

Как и в случае с операцией удаления кортежа, полученное аналитическое выражение (2) является замещением алгоритма  $Alg_2$  для дополнения кортежа в многотабличном запросе. В исполняющей среде формируется команда SQL, соответствующая формуле (2).

### 3.3. Операция модификации кортежа

В общем случае операция модификации кортежа сводится к операциям удаления старого кортежа  $u$  и дополнения нового  $u'$ . Алгоритму  $Alg_3$  (операция модификации) соответствует следующее выражение реляционной алгебры:

$$\begin{aligned} R''_m &= R'_m \setminus \pi_{\langle R'_m \rangle}(\sigma_{F \wedge (X_0=u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \\ &\dots \bowtie R'_m[X_m])) \cup \pi_{X_m}(\sigma_F(\pi_Y(\sigma_{Z=u[Z] \& F}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \\ &\dots \bowtie R'_{m-1}[X_{m-1}])) \bowtie u')), \end{aligned}$$

где

$$\begin{aligned} Z &= X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}); \\ Y &= (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}). \end{aligned}$$

Применив эквивалентные преобразования, получим:

$$R''_m = R'_m \setminus \pi_{X_m}(\sigma_F(T \bowtie u)) \cup \pi_{X_m}(\sigma_F(T \bowtie u')), \quad (3)$$

где

$$\begin{aligned} T &= \pi_Y(\sigma_{(Z=u[Z]) \vee (Z=u'[Z]) \& F}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \\ &\dots \bowtie R'_{m-1}[X_{m-1}]))); \end{aligned}$$

$$\begin{aligned} Z &= X_0 \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}); \\ Y &= (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \dots \cup X_{m-1}). \end{aligned}$$

В формуле (3) использованы обозначения формулы (2) и следующие преобразования. Выражение  $\sigma_{F \wedge (X_0=u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie R'_m[X_m])$  было заменено на эквивалентное  $\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \dots \bowtie u)$ . В  $T$  объединены списки удаляемых и дополняемых кортежей. Если какие-либо кортежи в  $T$  принадлежат обоим спискам, то при формировании  $T \bowtie u$  и  $T \bowtie u'$  каждый кортеж участвует только в своей операции и лишние кортежи в  $T$  игнорируются. Заметим, если  $u' = \emptyset$ , то формула (3) эквивалентна операции удаления кортежа  $u$ , если  $u = \emptyset$ , то (3) — дополнение кортежа  $u'$ .

В большинстве приложений каждый кортеж-представление имеет один кортеж-образ, в целевом отношении, также достаточно выполнить одну команду UPDATE.

**Пример 2.** Для демонстрации рассмотрим операцию модификации кортежа в примере 1. Пусть при старте приложения выбрано ограничение на название предмета "Введение в специальность". Схема представления (заголовок) следующая:  $\langle \text{Специальность}(A_5) \rangle$ ,  $\langle \text{№ курса}(A_6) \rangle$ ,  $\langle \text{День недели}(A_8) \rangle$ ,  $\langle \text{Время начала занятия}(A_{14}) \rangle$ ,  $\langle \text{ФИО преподавателя}(A_{12}) \rangle$ ,  $\langle \text{№ аудитории}(A_{16}) \rangle$ . Пусть в представлении кортеж (Физика, 1, Вторник, 11-40, Иванов И. И., Б-314) заменили кортежем (Химия, 1, Среда, 9-45, Петров П. П., А-212). Исходному кортежу в целевом отношении будет соответствовать три кортежа (три группы на специальности 'Физика'):  $(A_2 = 15, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$ ,  $(A_2 = 16, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$ ,  $(A_2 = 17, A_7 = 2, A_{13} = 3, A_9 = 6, A_{11} = 362, A_{16} = \text{'Б-314'})$ . В соответствии с предложенной технологией в целевом отношении перечисленные корте-

жи должны быть заменены четырьмя кортежами (четыре группы на специальности 'Химия'): ( $A_2 = 22, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$ ), ( $A_2 = 23, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$ ), ( $A_2 = 24, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$ ), ( $A_2 = 25, A_7 = 3, A_{13} = 2, A_9 = 6, A_{11} = 412, A_{16} = 'A-212'$ ). Если бы значения атрибута  $A_2$  совпадали для всех новых кортежей, было бы достаточно одной команды UPDATE для выполнения замены. Однако для данного примера самым эффективным способом является выполнение двух команд SQL: удаление (DELETE) старых кортежей и дополнение (INSERT) новых.

## Заключение

Рассмотренная в данной работе теория является основой для создания новой технологии генерации многотабличных приложений. Для этого создается программа-шаблон, в которую в диалоговом режиме вставляются компоненты запроса  $Q$ . В ходе диалога проверяются ограничения на запрос с использованием реализованных зависимостей. Операции реляционной алгебры в виде команд SQL вставляются в программу-шаблон. В результате получается готовая к исполнению программа, реализующая редактирование многотабличных представлений. Аналогичная по структуре технология была предложена в работе [23] и реализована для динамического формирования многомерных данных (плоский вариант) в среде Microsoft Office [24].

## Список литературы

1. **Forta B.** SQL in 10 Minutes, Sams Teach Yourself. 4th Ed. Indiana, USA: Sams Publishing PTG, 2013. 288 p.
2. **Baklarz G.** DB2 9 for Linux, UNIX, and Windows: DBA Guide 6 edition. NY, USA: IBM Press, 2007. 1136 p.
3. **Oracle** Database Online Documentation 11g Release 1 (11.1). Database Administrator's Guide, 2017. 299 p. URL: [https://docs.oracle.com/cd/B28359\\_01/server.111/b28310/toc.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28310/toc.htm) (дата обращения: 31.10.17).
4. **Бондарь А.** Microsoft SQL Server 2014. СПб: БХВ-Петербург, 2015. 592 с.
5. **McLaughlin B.** PHP & MySQL: The Missing Manual, 2nd Edition. Sebastopol, USA: O'Reilly Media, 2012. 546 p.
6. **Worsley J., Drake J.** Practical PostgreSQL. Sebastopol, USA: O'Reilly Media, 2011. 640 p.
7. **Bancilhon F., Spyrtatos N.** Update semantics of relational views // ACM Trans. Database Syst. 1981. Vol. 6, N. 4. P. 557–575.

8. **Dayal U., Bernstein P. A.** On the correct translation of update operations on relational views // ACM Trans. Database Syst. 1982. Vol. 7, N. 3. P. 381–416.
9. **Masunaga Y.** A Relational Database View Update Translation Mechanism // In Proceedings of the 10th International Conference on Very Large Data Bases (VLDB '84), San Francisco, CA, USA. 1984. P. 309–320.
10. **Cosmadakis S., Papadimitriou C.** Updates of Relational Views // J. ACM. 1984. Vol. 31, N. 4. P. 742–760.
11. **Gottlob G., Paolini P., Zicari R.** Properties and update semantics of consistent views // ACM Trans. Database Syst. 1988. Vol. 13, N. 4. P. 386–524.
12. **Langerak R.** View updates in relational databases with an independent scheme // ACM Trans. Database Syst. 1990. Vol. 15, N. 1. P. 40–66.
13. **Lechtenböcker J.** The impact of the constant completion approach towards view updating // In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '03). ACM, New York, NY, USA. 2003. P. 49–55.
14. **Keller A.** Algorithms for translating view updates to database updates for views involving selections, projections, and joins // In Proceedings of the fourth ACM SIGACT-SIGMOD symposium on Principles of database systems (PODS '85). ACM, New York, NY, USA. 1985. P. 154–163.
15. **Калиниченко Л. А.** Методы и средства интеграции неоднородных баз данных. М.: Наука, 1983. 423 с.
16. **Zykin S. V.** Generation of User View for a Relational Database by Mappings // Programming and Computer Software. 1999. Vol. 25, N. 3. P. 173–183.
17. **Maier D.** The theory of relational databases // Computer Science Press, 1983. 637 p.
18. **Ullman J.** Principles of database systems // Computer Science Press, 1980. 379 p.
19. **Hartmann S., Link S.** The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations // ACM Transactions on Database Systems. 2012. Vol. 37, N. 2. P. 1–40.
20. **Köhler H., Link S.** Inclusion Dependencies Reloaded // The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15), Melbourne, Australia, October 18–23, 2015. P. 1361–1370.
21. **Зыкин С. В., Полуянов А. Н.** Итерирование отношений базы данных для запросов специального вида // Материалы X Международной IEEE научно-технической конференции "Динамика систем, механизмов и машин", Омск, 15–17 ноября, 2016. Т. 4, № 1. С. 18–21.
22. **Mosin S. V., Zykin S. V.** Truth Space Method for Caching Database Queries // Моделирование и анализ информационных систем. 2015. Т. 22, № 2. С. 248–258.
23. **Zykin S. V.** Automation of the Interface Formation Between Multidimensional and Relational Representation of the Data // Relational Databases and Open Source Software Developments. NY: Nova Science Publishers, 2011. P. 43–66.
24. **Зыкин С. В.** Автоматизация формирования табличных приложений // Моделирование и анализ информационных систем. 2013. Т. 20, № 4. С. 41–54.

**S. V. Zykin**, D. Sc., Professor, Head of Laboratory, e-mail: szykin@mail.ru,  
Sobolev Institute of Mathematics SB RAS, Novosibirsk, Russia,  
**V. S. Zykin**, Postgraduate, e-mail: vszykin@mail.ru,  
Omsk State Technical University, Omsk, Russia

## Commutative Conversion in the Database when Editing a Multitable Query

*Theoretical aspects of the formation of applications for editing data in relational databases are considered. This important problem has not yet been satisfactorily resolved. That problem has been solved by "hard" restrictions on the views and it requires the highly qualified user when using these components. In this paper, the theoretical aspects of the applications formation for work with a relational database are discussed. The application works with a data view such as table formed by a set of database relations using "join-select-projection" queries. The implementation of data editing functions: inserting, deleting and updating of tuples in the table are the requirement for such applications. The application must convert the source database according to the user's view editing providing that the conversion satisfy the commutativity condition. Formulas of relational algebra are obtained for these transformations. The correctness of the transformations is proved. The theory in this paper is the basis for creating a new technology for generating of multitable applications. The technology involves creating a template program, where a query and its restrictions are implemented. The operations of relational algebra are transformed into SQL commands. As a result, the user obtains a ready-to-run program that implements editing of multitable views.*

**Keywords:** relational database, commutative transformations, relational algebra, views editing

### References

1. Forta B. *SQL in 10 Minutes, Sams Teach Yourself*, 4th Edition. Indiana, USA: Sams Publishing PTG, 2013. 288 p.
2. Baklarz G. *DB2 9 for Linux, UNIX, and Windows: DBA Guide* 6 edition. NY, USA, IBM Press, 2007, 1136 p.
3. Oracle Database Online Documentation 11g Release 1 (11.1). Database Administrator's Guide, 2017. 299 p. URL: [https://docs.oracle.com/cd/B28359\\_01/server.111/b28310/toc.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28310/toc.htm) (date of access: 31.10.17).
4. Bondar' A. *Microsoft SQL Server 2014*. SPb: BHV-Peterburg, 2015. 592 p. (in Russian)
5. Maklaffin B. *PHP & MySQL: The Missing Manual*, 2nd Edition. Sebastopol, USA, O'Reilly Media, 2012, 546 p.
6. Worsley J., Drake J. *Practical PostgreSQL*, Sebastopol, USA, O'Reilly Media, 2011, 640 p.
7. Bancilhon F., Spyratos N. Update semantics of relational views, *ACM Trans. Database Syst.*, 1981, vol. 6, no. 4, pp. 557–575.
8. Dayal U., Bernstein P. A. On the correct translation of update operations on relational views, *ACM Trans. Database Syst.*, 1982, vol. 7, no. 3, pp. 381–416.
9. Masunaga Y. A Relational Database View Update Translation Mechanism, *In Proceedings of the 10th International Conference on Very Large Data Bases (VLDB '84)*, San Francisco, CA, USA, 1984, pp. 309–320.
10. Cosmadakis S., Papadimitriou C. Updates of Relational Views, *J. ACM*, 1984, vol. 31, no. 4, pp. 742–760.
11. Gottlob G., Paolini P., Zicari R. Properties and update semantics of consistent views, *ACM Trans. Database Syst.*, 1988, vol. 13, no. 4, pp. 386–524.
12. Langerak R. View updates in relational databases with an independent scheme, *ACM Trans. Database Syst.*, 1990, vol. 15, no. 1, pp. 40–66.
13. Lechtenbörger J. A The impact of the constant complement approach towards view updating, *In Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS '03)*, ACM, New York, NY, USA, 2003. P. 49–55.
14. Keller A. Algorithms for translating view updates to database updates for views involving selections, projections, and joins, *In Proceedings of the fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS '85)*, ACM, New York, NY, USA, 1985. P. 154–163.
15. Kalinichenko L. A. *Metody i sredstva integracii neodnorodnyh baz dannyh* (Methods and facilities for integrating nonhomogeneous databases), Moscow, Nauka, 1983. 423 p. (in Russian).
16. Zykin S. V. Generation of User View for a Relational Database by Mappings, *Programming and Computer Software*, 1999, vol. 25, no. 3, pp. 173–183.
17. Maier D. *The theory of relational databases*, Computer Science Press, 1983, 637 p.
18. Ullman J. *Principles of database systems*, Computer Science Press, 1980, 379 p.
19. Hartmann S., Link S. The implication problem of data dependencies over SQL table definitions: axiomatic, algorithmic and logical characterizations, *ACM Transactions on Database Systems*, 2012, vol. 37, no. 2, pp. 1–40.
20. Köhler H., Link S. Inclusion Dependencies Reloaded, *The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*, Melbourne, Australia, October 18–23, 2015, pp. 1361–1370.
21. Zykin S. V., Polujanov A. N. Iterirovanie otnoshenij bazy dannyh dlja zaprosov special'nogo vida (Iteration of database relations for special queries), *Materialy X Mezhdunarodnoj IEEE nauchno-tehnicheskoy konferencii "Dinamika sistem, mehanizmov i mashin"*, Omsk, 15–17 November, 2016, vol. 4, no. 1, pp. 18–21 (in Russian).
22. Mosin S. V., Zykin S. V. Truth Space Method for Caching Database Queries, *Modelirovanie i analiz informacionnyh sistem*, 2015, vol. 22, no. 2, pp. 248–258.
23. Zykin S. V. Automation of the Interface Formation Between Multidimensional and Relational Representation of the Data. *Relational Databases and Open Source Software Developments*. NY, Nova Science Publishers, 2011, pp. 43–66.
24. Zykin S. V. Avtomatizacija formirovanija tablichnyh prilozhenij (Automating the formation of tabular applications), *Modelirovanie i analiz informacionnyh sistem*, 2013, vol. 20, no. 4, pp. 41–54 (in Russian).