

УДК 004.652

© С. В. Зыкин, А. Н. Полуянов, В. С. Зыкин

РЕДАКТИРОВАНИЕ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ «ТРАНСФОРМАЦИЯ»

Проблема согласования семантики данных, представленных в рамках различных моделей, остается актуальной значительный промежуток времени. Прежде всего, это связано с удобством работы пользователей, которые привыкли к определенным инструментальным средствам, например, к электронным таблицам. Подготовленные в этих средах данные нуждаются в загрузке в централизованную базу данных, что позволяет избавиться от дублирования и противоречивости данных. Препятствием на этом пути является проблема согласования данных. Редактирование данных непосредственно в базе данных является сложной задачей для пользователей непрограммистов. Традиционным способом решения этой проблемы является разработка специальных приложений, которые имеют ограниченный функционал. В данной работе предлагается технология, которая позволяет редактировать данные в базе данных с использованием электронных таблиц, делая доступным их богатый функционал. Основным отличием от аналогичных подходов является использование модели «Трансформация», которая делает представление данных удобным для восприятия человеком. Поскольку модель данных «Трансформация» существенно отличается от реляционной модели, появляется необходимость согласования данных между базой данных и электронными таблицами. Для решения аналогичных проблем Л. А. Калиниченко предложил методику коммутативных преобразований в базах данных. В данной работе эта методика, с некоторыми изменениями, используется в алгоритмах передачи данных из базы данных в «Трансформацию» и обратно. В статье представлен обзор работ по проблеме согласования данных в различных источниках, представлено описание модели данных «Трансформация», в том числе: описание схемы таблицы, условий существования экземпляра таблицы и операций редактирования данных. В работе приводится описание алгоритма загрузки данных в таблицу из базы данных и алгоритма преобразования данных в базе данных в соответствии с изменениями в таблице, определены условия коммутативности преобразований, представлено доказательство корректности преобразований.

Ключевые слова: модель данных, трансформация, коммутативность.

DOI: [10.35634/vm240409](https://doi.org/10.35634/vm240409)

Введение

В процессе развития реляционная модель благодаря своим свойствам получила наибольшую популярность. Однако растущее количество информации с различным описанием и структурой хранения возрождает проблему согласования и интеграции информации. Реляционная модель не всегда удовлетворяет предъявляемым требованиям. Прежде всего это относится к способу хранения данных и к представлению данных в приложениях: нормализованные таблицы требуют сложных механизмов выполнения запросов на физическом уровне и сложных алгоритмов формирования пользовательских представлений данных.

Теоретические проблемы интеграции и согласования неоднородных по структуре данных впервые начал исследовать Л. А. Калиниченко. В работе [1] рассматривается методика формализации моделей данных, средства интеграции неоднородных систем с неоднородными базами данных (БД). В основе методики согласования используются коммутативные преобразования данных периферийных моделей с центральной обобщенной моделью. Использование ядра обобщенной модели в качестве основы позволяет разрабатывать и верифицировать алгоритмы реализации запросов в различных средах.

В статье [2] представлены классические методы формирования реляционных баз данных с использованием электронных таблиц. Посредством анализа содержимого электронных таблиц определяются функциональные зависимости в данных, строится минимальное покрытие зависимостей и на этой основе определяется структура реляционной БД. Содержимое БД формируется за счет применения операции проекции к электронной таблице. С использованием запроса SQL имеется возможность восстановить содержимое электронной таблицы. В статье [2] не обсуждается редактирование отдельных данных, а только массовая загрузка данных. Кроме того, исходные данные в электронных таблицах не содержат каких-либо трансформаций, и фактически являются аналогом универсального реляционного отношения.

В статье [3] электронная таблица используется в качестве движка базы данных за счет использования формул в электронных таблицах. Все операторы реляционной алгебры реализуются с помощью функций электронных таблиц. Все функции задаются программистом и обычный пользователь не может их модифицировать. Пользователь вводит, изменяет или удаляет данные в рабочих листах данных, формулы в рабочих листах запросов автоматически вычисляют фактические результаты запросов. Таким образом, электронная таблица служит хранилищем данных и выполняет запросы SQL. Редактирование данных в БД с использованием электронных таблиц не предусмотрено.

Для решения проблемы низкой эффективности при составлении документов по сбору данных на предприятиях и необходимости ручной корректировки ошибок в работе [4] была построена математическая модель для автоматического создания документов, запрограммированная в среде Visual Basic for Application (VBA). В модели реализованы классификация и анализ основных типов данных документов, разработаны правила хранения файлов конфигурации и настройки шаблонов документов. Разработанная модель была использована в системе автоматического формирования документов задач спутниковой платформы, и практикой применения доказано, что модель осуществима и эффективна. Редактирование данных в БД средствами VBA не предусмотрено.

В работе [5] представлен язык подстановок для генерации электронных документов на основе содержимого БД и файлов. Проектируемый язык предполагает возможность работы как с одной БД, так и с большим числом однотипных баз. Для сложных элементов документа разработана система шаблонов, что позволяет генерировать таблицы со сложной структурой и вложенными подтаблицами и расширенными ячейками. Технология позволяет группировать в удобном для чтения виде большие объемы разнотипных данных и использовать произвольные форматы входных и выходных данных. Редактирование данных в БД в предложенной технологии не предусмотрено.

Перечисленные работы акцентируют внимание на пользе представления данных в виде структурированных таблиц. Однако, это лишь малая толика публикаций, в которых в том или ином виде таблицы являются предметом и/или инструментом исследования. Особенно часто это можно наблюдать в публикациях по экономической тематике. На практике, работая с табличным представлением данных, у пользователя (заказчика) практически сразу возникает вопрос: «Могу ли я редактировать в таблице данные, которые получены из БД?» Сама по себе проблема формирования табличного представления данных из БД не является сложной. Но возврат отредактированных в таблице данных обратно в БД в общем случае не имеет однозначного решения. Для этого требуется формализовать модели данных и на их основе разработать алгоритмы преобразования данных в БД.

В данной работе используются межмодельные коммутативные преобразования [1,6] для интеграции ресурсов с различными представлениями данных. Построение преобразований позволяет значительно автоматизировать рутинные операции по обработке информации: сбору, согласованию, структуризации и т. д. Для достижения этой цели представлено фор-

мальное описание используемых моделей.

§ 1. Модель данных «Трансформация»

Пользовательское табличное представление данных далее будем называть трансформацией. Впервые термин “Семантическая трансформация” был использован в работе [7] для частного случая таблицы. Учитывая популярность такого представления данных, разработчики Microsoft Access дополнили язык запросов SQL инструкцией TRANSFORM, которая формирует представление аналогичное [7], и допускает агрегирование данных. Редактирование данных в таком представлении не допускается.

На основе анализа рассмотренных представлений данных построим модель данных. Введем обозначения для реляционной БД: $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$, полученная в результате нормализации отношений R_i ; $[R_i]$ — множество атрибутов, на которых определено отношение R_i (заголовок таблицы, схема отношения); $R_i[S]$ — проекция отношения R_i , по атрибутам S . U — конечное множество атрибутов, на которых заданы отношения \mathcal{R} . В соответствии с требованием нормализации каждое из R_i , $i = \overline{1, k}$, как минимум, должно удовлетворять требованиям третьей нормальной формы (ЗНФ). Кроме того, отношения \mathcal{R} должны быть связаны ограничениями ссылочной целостности, которые соответствуют типизированным ациклическим зависимостям включения [8]:

Определение 1. Зависимость включения $R_i[S] \subseteq R_j[W]$, определенную на множестве кортежей отношений R_i и R_j , будем называть *типизированной*, если и только если $S = W$.

Замечание 1. Множество атрибутов S в отношении R_i называется внешним ключом, отношение R_j называется главным, а отношение R_i называется внешним.

Определение 2. Множество зависимостей включения будем называть *ациклическим*, если ациклическим является направленный граф, в котором отношения R_i являются узлами, а зависимости $R_i[W] \subseteq R_j[W]$ являются ребрами.

Минимальное покрытие типизированных зависимостей включения [9, 10] реализовано в БД ссылочной целостностью.

Замечание 2. Реализованные циклические зависимости включения блокируют работу БД: в отношениях, входящих в цикл, невозможно ни вставить, ни удалить записи.

Обозначим Tr — табличное представление данных (трансформация). Схему (заголовок) Tr зададим в следующем виде:

$$Sch(Tr) = \{X, Dom(Y)\},$$

где $X = \{X_1, X_2, \dots, X_m\}$ и $Y = \{Y_1, Y_2, \dots, Y_n\}$ — непустые множества атрибутов: $X \subset U$ и $Y \subset U$; X_i и Y_j — отдельные атрибуты; $Dom(Y) = Dom(Y_1) \times Dom(Y_2) \times \dots \times Dom(Y_n)$, где $Dom(Y_j)$ — значения атрибута Y_j (множество значений в БД). Таким образом, первые m столбцов таблицы будут озаглавлены именами атрибутов X_i . Последующие столбцы будут иметь составные имена из n значений: $y_{1,l_1}, y_{2,l_2}, \dots, y_{n,l_n}$, где y_{j,l_j} значение атрибута Y_j , $j = \overline{1, n}$.

Каждая i -я строка (кортеж) таблицы содержит значения:

$$(x_{i,1}, x_{i,2}, \dots, x_{i,m}, z_{i,(m+1)}, z_{i,(m+2)}, \dots, z_{i,(m+p)}),$$

где p — мощность $Dom(Y)$. В i -й строке и j -м столбце, $j = \overline{1, m}$, содержится значение атрибута X_j . В последующих столбцах содержатся значения $z_{i,l}$ атрибута $Z \in U$, соответствующие значениям атрибутов X и Y . Правила установления такого соответствия рассматриваются далее. Множества атрибутов X и Y упорядочиваются в представлении таблицы Tr

с целью достижения компактности [11]. При этом используются функциональные и многозначные зависимости в исходной реляционной БД и мощности соответствующих доменов. Для удобства восприятия таблицы значения атрибутов X и Y упорядочиваются, например, по возрастанию.

С практической точки зрения целесообразно использовать следующие ограничения: $X \cap Y = \emptyset$, $Z \notin X \cup Y$. Кроме того, мощность доменов X и Y не должна быть слишком большой, чтобы таблица Tr была обозримой. Пример трансформации представлен в таблице 1.

Таблица 1. Пример трансформации

X_1	X_2	X_3	y_{11}				y_{12}			
			y_{21}		y_{22}		y_{21}		y_{22}	
			y_{31}	y_{32}	y_{31}	y_{32}	y_{31}	y_{32}	y_{31}	y_{32}
x_{11}	x_{21}	x_{31}	z_{11}		z_{13}	z_{14}		z_{16}		z_{18}
		x_{32}					z_{25}			
	x_{22}	x_{31}		z_{32}		z_{34}			z_{37}	
		x_{32}	z_{41}		z_{43}			z_{46}		
x_{12}	x_{21}	x_{31}	z_{51}			z_{54}			z_{57}	
		x_{32}		z_{62}			z_{65}			z_{68}
	x_{22}	x_{31}	z_{71}		z_{73}			z_{76}		
		x_{32}	z_{81}			z_{84}			z_{87}	

Редактирование данных в БД с использованием табличных представлений исследуется не впервые. Подход к решению этой проблемы был предложен в работе [12]. Для согласования данных в таблице и в БД использовалась промежуточная модель «Таблица связанных соединений», в которой были реализованы зависимости на данных, соответствующие зависимостям БД. Дополнительный «вектор вхождений» позволял однозначно идентифицировать отношение и кортеж, в которых произошли изменения. К недостаткам такого подхода можно отнести длительное время формирования «Таблицы связанных соединений», что позволяет работать только со слабо изменчивыми БД.

Подход к редактированию многотабличных представлений данных был предложен в работе [13]. С использованием коммутативных преобразований данных был разработан алгоритм преобразования БД. При этом не использовались какие-либо промежуточные модели данных. Опытная реализация такого подхода показала удовлетворительные результаты на тестовых БД. Однако этот подход имеет ограниченную сферу применения, поскольку итоговая реляционная таблица не является удобным представлением для работы пользователя. В данной статье поставлена цель избавления от перечисленных недостатков.

На содержимое и функциональные возможности трансформации оказывают влияние ограничения целостности, реализованные в БД. Условием существования таблицы Tr является отсутствие наложения различных значений атрибута Z в одной ячейке таблицы. Этому должны способствовать реализованные ограничения целостности в БД. Из условия существования таблицы Tr следует, что таким ограничением является функциональная зависимость $XY \rightarrow Z$, где XY обычное объединение двух множеств атрибутов ($XY = X \cup Y$). Основным средством реализации функциональных зависимостей в БД являются индексные файлы, обладающие свойством уникальности (Primary key, Unique).

Формирование структуры Tr начинается с определения расширенных имен атрибутов X , Y и Z . Расширенное имя атрибута состоит из имени отношения R_i и имени атрибута A_j , разделенных между собой точкой ($R_i.A_j$). Таким образом определяется начальное множество отношений $R = \{R'_1, R'_2, \dots, R'_p\} \subseteq \mathcal{R}$, участвующее в формировании Tr , и множество реализованных функциональных зависимостей $F = \{F_1, F_2, \dots, F_q\}$ (определяется по индексным файлам со свойством уникальности в отношениях R).

Реализация зависимости $XY \rightarrow Z$ проверяется за счет построения замыкания $\{XY\}_F^+$ на множестве зависимостей F : если $Z \in \{XY\}_F^+$, то зависимость $XY \rightarrow Z$ является выводимой из F [14, 15]. Поскольку система аксиом функциональных зависимостей полна [14], то все выводимые зависимости являются выполнимыми (логическим следствием). Следовательно, ограничения целостности в виде зависимостей F , реализованные в отношениях БД, не дадут ввести данные, которые будут противоречить $XY \rightarrow Z$.

В качестве промежуточного представления данных будем использовать упрощенную форму запроса «проекция–селекция–соединение», которая может быть записана в терминах реляционной алгебры:

$$Q = \pi_{XYZ}(R'_1 \bowtie R'_2 \bowtie \dots \bowtie R'_p), \quad (1)$$

где \bowtie — операция естественного соединения; π — операция проекции.

Для корректного редактирования данных в БД должны быть выполнены следующие условия.

- Каждому непустому пересечению $[R'_i] \cap [R'_j]$ отношений R'_1, R'_2, \dots, R'_p должно соответствовать ограничение ссылочной целостности. Кроме того, ограничения должны упорядочивать указанные отношения по внешним ключам: главные отношения в последовательности стоят раньше соответствующих им внешних отношений. Таким образом, на отношениях устанавливается частичный порядок, в котором должно быть только одно максимальное отношение $R'_Z: Z \in [R_Z]$. Это отношение будет соответствовать семантике приложения, поскольку в нем будут реализованы операции редактирования БД.
- Существует некоторое множество атрибутов V , которое удовлетворяет условиям: $V \subseteq XY$, $V \subseteq [R_Z]$, $Z \notin V$ и $V \rightarrow XY$.
- Во множестве R должно быть одно отношение R_Z , содержащее атрибут Z , что необходимо для однозначной интерпретации значений Z при формировании Tr .

Замечание 3. Частичный порядок над отношениями R позволит вводить только те данные, относительно которых остальная информация уже определена. Перечисленные ограничения являются естественными при использовании синтетического или декомпозиционного подходов при проектировании схемы БД, и не накладывают значительных ограничений на предлагаемую технологию.

Если условие реализации зависимости $XY \rightarrow Z$ не выполнено, либо отсутствует частичный порядок над отношениями, то множество отношений R должно быть дополнено новыми отношениями из \mathcal{R} . Это не может быть сделано автоматически, поскольку всякий раз будет меняться семантика запроса (1). Программное обеспечение может только подсказать пользователю допустимые варианты, из которых он должен выбрать вариант, соответствующий семантике решаемой задачи. Далее будем считать, что зависимость $XY \rightarrow Z$ для множества отношений R реализована и над отношениями установлен частичный порядок.

§ 2. Формирование представления таблицы Tr

Рассмотрим алгоритм Alg_1 , который предназначен для формирования Tr .

- Шаг 1. Для каждого атрибута множеств X и Y вычисляются области определения $Dom(X_i)$ и $Dom(Y_j)$. Для этого используется содержимое таблицы БД, из которой соответствующий атрибут выбран (определяется по расширенному имени атрибута). Значения в каждой области определения упорядочиваются в каком-либо лексикографическом порядке.
- Шаг 2. Для получения оптимальной иерархии размерностей атрибуты множеств X и Y упорядочиваются в соответствии с алгоритмом [11].
- Шаг 3. Полученные вектора значений $(x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и $(y_{1,l_1}, y_{2,l_2}, \dots, y_{n,l_n})$ используются для формирования строк и столбцов таблицы Tr .
- Шаг 4. Последовательно просматриваются кортежи Q , и значение атрибута Z помещаются в ячейку Tr , соответствующую строке $(x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и столбцу $(y_{1,l_1}, y_{2,l_2}, \dots, y_{n,l_n})$. Значения атрибутов X и Y определяются текущим кортежем Q .

В теореме 1 формулируется важный факт для построения Tr , однако, доказательство теоремы является очевидным.

Теорема 1. *Представление Tr существует тогда и только тогда, когда Q удовлетворяет функциональной зависимости $XY \rightarrow Z$.*

Замечание 4. Формулировка теоремы 1 предполагает отсутствие неопределенных значений для атрибутов X и Y в представлении Q . Тогда как атрибут Z может принимать такое значение, что соответствует пустой ячейке в Tr .

Теорема 2. *Если функциональная зависимость $XY \rightarrow Z$ реализована на множестве отношений R , тогда Q удовлетворяет $XY \rightarrow Z$.*

Доказательство. Предположим обратное: существуют два кортежа $t_1, t_2 \in Q$, значения в которых совпадают по атрибутам XY , то есть $t_1[XY] = t_2[XY]$, и $t_1[Z] \neq t_2[Z]$. Пусть V_1, V_2, \dots, V_p — последовательность множеств атрибутов, упорядоченная в соответствии с очередностью их присоединения к замыканию $(XY)_F^+$. На начальном этапе $(XY)_F^+ = XY$ по аксиоме рефлексивности [14]. Зависимость $XY \rightarrow V_1$ принадлежит множеству F , следовательно, существует отношение $R'_i \in R$, в котором эта зависимость реализована. По свойству операции \bowtie атрибуты V_1 принадлежат Q в единственном экземпляре (значения одноименных атрибутов должны совпадать, поэтому дублированные столбцы удаляются из соединения). Следовательно, $t_1[V_1] = t_2[V_1]$. Последующее присоединение множества V_2 к замыканию $(XY)_F^+$ осуществляется за счет зависимости $W \rightarrow V_2$, которая принадлежит множеству F , и множество атрибутов W уже содержится в замыкании. По построению множества F существует отношение $R'_j \in R$, в котором зависимость $W \rightarrow V_2$ реализована. Следовательно, $t_1[V_2] = t_2[V_2]$. Продолжая дальше аналогичные рассуждения, получим, что для всех множеств V_s , $s = \overline{1, p}$, выполнено $t_1[V_s] = t_2[V_s]$. Поскольку зависимость $XY \rightarrow Z$ выводима, то атрибут Z принадлежит замыканию, и существует множество V_s , которому принадлежит атрибут Z . Следовательно, $t_1[Z] = t_2[Z]$, что противоречит сделанному предположению. \square

Замечание 5. В отличие от [13] не требуется выполнение свойства соединения без потерь информации [14] на множестве отношений R , поскольку «лишние», с точки зрения этого свойства, кортежи будут удовлетворять зависимости $XY \rightarrow Z$, следовательно, не будут препятствовать формированию Tr .

§ 3. Преобразование представления таблицы Tr

Рассмотрим единичную операцию Op для преобразования значения атрибута Z в представлении таблицы Tr : замена одного значения z_{ij} на другое значение \hat{z}_{ij} . При этом оба значения z_{ij} и \hat{z}_{ij} могут быть непустыми, либо одно из них может быть пустым (значение отсутствует). Равенство значений, либо одновременное отсутствие значений z_{ij} и \hat{z}_{ij} означает, что операция не выполнялась.

Для выполнения преобразований в БД, соответствующих операции Op , будем использовать алгоритм Alg_2 :

- Шаг 1. Пусть \hat{z}_{ij} — измененное значение атрибута Z в таблице Tr , ему соответствуют векторы значений $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и $\vec{y}_j = (y_{1,j}, y_{2,j}, \dots, y_{n,j})$ атрибутов X и Y соответственно.
- Шаг 2. Поиск кортежа $t \in Q$, для которого $t[XY] = (\vec{x}_i, \vec{y}_j)$. Если кортеж найден, то выполняется Шаг 3, иначе Шаг 4.
- Шаг 3. В отношении R_Z выполняется замена в кортеже u : $u[Z] = \hat{z}_{ij}$, для которого $u[V] = t[V]$. Конец алгоритма.
- Шаг 4. В отношении R_Z дополняется новый кортеж u :

$$u[V \cap X] = \vec{x}_i[V \cap X], \quad u[V \cap Y] = \vec{y}_j[V \cap Y] \quad \text{и} \quad u[Z] = \hat{z}_{ij}.$$

Если возникла ошибка в связи с нарушением ссылочной целостности в БД, то переход на Шаг 5, иначе конец алгоритма.

- Шаг 5. Информирование пользователя об отсутствии связанных данных для векторов \vec{x}_i и \vec{y}_j , возврат значения z_{ij} на свое место в таблице Tr . Конец алгоритма.

Замечание 6. Алгоритм Alg_2 можно выполнять одновременно для всех измененных значений атрибута Z в таблице Tr , для сравнения используя, например, скрытую копию исходной таблицы Tr . Однако, это не даст существенного ускорения выполнения операций, поскольку для каждого измененного значения требуется отдельный поиск в представлении запроса Q . Тогда как единичные изменения могут выполняться асинхронно с работой пользователя. Алгоритм Alg_2 является реализацией операции Op в реляционной модели. При отображении реляционной модели в $Q + Tr$ сохраняется семантика этой операции (обновления атрибутов из Z).

Условие коммутативности преобразований (1), Alg_1 , Alg_2 и Op можно представить в следующем виде:

$$DB_1 \xrightarrow{(1)} Q_1 \xrightarrow{Alg_1} Tr_1 \xrightarrow{Op} Tr_2, \quad (2)$$

$$DB_1 \xrightarrow{Alg_2} DB_2 \xrightarrow{(1)} Q_2 \xrightarrow{Alg_1} Tr_2, \quad (3)$$

где DB_1 и DB_2 — начальное и конечное состояния БД, Tr_1 и Tr_2 — начальное и конечное состояния трансформации. Другими словами, из DB_1 в Tr_2 можно перейти двумя различными путями, но результат должен быть один и тот же. На практике это гарантирует корректность выполненных преобразований в БД.

Теорема 3. Преобразования (2) и (3) удовлетворяют условию коммутативности.

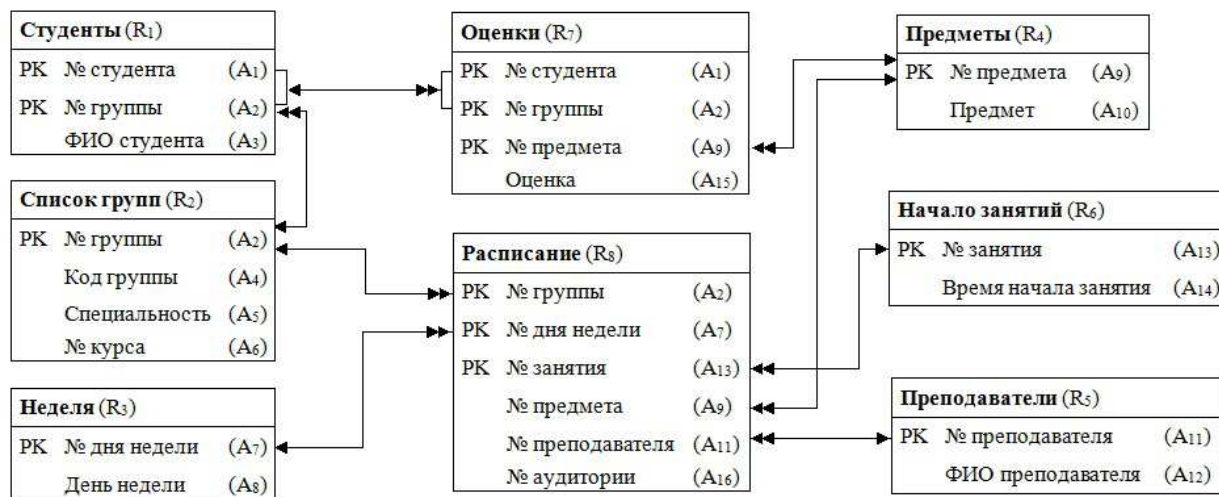


Рис. 1. Схема БД

Доказательство. Для доказательства коммутативности преобразований достаточно показать, что образ состояния DB_2 в трансформации существует и совпадает с Tr_2 . Из теорем 1 и 2 следует, что для существования Tr_2 достаточна реализация зависимости $XY \rightarrow Z$ в представлении DB_2 .

Предположим, что Q_2 содержит, по крайней мере, две строки (t_1 и t_2), которые совпадают по атрибутам XY и различаются по атрибуту Z . Эти кортежи должны быть сформированы с участием кортежей u_1 и u_2 , принадлежащих R_Z , для которых $u_1[Z] \neq u_2[Z]$. Поскольку $V \rightarrow XY$ и $V \subseteq [R_Z]$, то $V \rightarrow Z$ и $u_1[V] \neq u_2[V]$. Однако, $V \subseteq XY$, следовательно, $u_1[V] = u_2[V]$. Полученное противоречие опровергает предположение, препятствующее существованию Tr_2 .

Допустим, что выполнен Шаг 3 алгоритма Alg_2 . Поскольку атрибут Z принадлежит только одному отношению множества R , то его значение не влияет на результат операции естественного соединения в (1), следовательно, состав кортежей в Q_1 и Q_2 будет совпадать, поскольку операция естественного соединения выполняется по одним и тем же атрибутам, значения которых не изменились. Исключением является новое значение \hat{z}_{ij} , загруженное из R_Z вместо старого значения z_{ij} . Поскольку в Q_2 реализована зависимость $XY \rightarrow Z$, то во всех кортежах, совпадающих по атрибутам XY , значения атрибута Z будут совпадать. Это гарантирует, что в результате работы алгоритма Alg_1 из Q_2 будет получена таблица Tr_2 .

Допустим, что на Шаге 4 алгоритма Alg_2 в отношении R_Z был успешно дополнен новый кортеж u . Это означает, что во всех отношениях R были найдены кортежи, совпадающие по одноименным атрибутам, что гарантируется отсутствием нарушения ссылочной целостности. Вновь дополненные кортежи получены за счет соединения с кортежем u , следовательно, совпадают по атрибутам $[R_Z]$ и по атрибутам V . Поскольку $V \rightarrow XY$, то эти кортежи совпадают по атрибутам XY . В результате по алгоритму Alg_1 будет получена таблица Tr_2 . \square

§ 4. Пример формирования таблицы Tr

Рассмотрим фрагмент схемы БД «Вуз» (рис. 1), на котором продемонстрируем варианты формирования таблицы «Трансформация».

На схеме атрибуты первичного ключа отношений помечены символами РК, связи на схеме соответствуют ссылочным ограничениям целостности: одиночные стрелки указывают

на связанные атрибуты главных отношений, двоянные стрелки соответствуют внешним ключам. Отношения и атрибуты на схеме поименованы (в скобках).

Сформируем приложение «Сводная ведомость». Пусть $X = \{R_2.A_2, R_2.A_4, R_1.A_3\}$, $Y = \{R_4.A_9, R_4.A_{10}\}$, $Z = \{R_7.A_{15}\}$. Множество атрибутов $V = \{R_7.A_1, R_7.A_2, R_7.A_9\}$ удовлетворяет ограничениям, отношения R удовлетворяют условию частичного порядка с одним максимальным элементом R_7 . Внешний вид приложения представлен в таблице 2.

Таблица 2. Приложение «Сводная ведомость»

№ группы	Код группы	№ студента	ФИО студента	1	2	3	4	...
				Физика	Химия	История	Биология	...
1	Б-211	1	Иванов	3		5		
		2	Петров		4			
		3	Сидоров	4			3	
1	Х-212	1	Ковалев	3		4		
		2	Попов		4		5	

В реализации таблицы 2 без потери информативности можно сделать скрытыми строки и столбцы с номерами, что сократит объем заголовков. Однако, остающиеся строки и столбцы, в общем случае, не будут содержать ключевых атрибутов, что приведет к неоднозначной интерпретации результатов.

Отношения множества R , занимающие минимальные места в частичном порядке, можно ограничить логическими выражениями. Это не изменит функциональные зависимости и частичный порядок на отношениях, но сократит объем таблицы. Для введения более сложных ограничений требуются дополнительные исследования.

Кроме приложения «Сводная ведомость» с использованием схемы БД (рис. 1) могут быть сформированы другие приложения. Например, приложение «Расписание занятий», в котором $X = \{R_8.A_7, R_3.A_8, R_8.A_{13}, R_6.A_{14}\}$, $Y = \{R_8.A_2, R_2.A_4\}$, $Z = \{R_8.A_{16}\}$. Множество атрибутов $V = \{R_8.A_2, R_8.A_7, R_8.A_{13}\}$, частичный порядок на отношениях R имеет один максимальный элемент R_8 . Для редактирования атрибутов $R_8.A_9$ и $R_8.A_{11}$ необходимо использовать «Поле со списком», что в современных электронных таблицах является стандартным компонентом. Кроме того, редактирование атрибутов $R_8.A_{16}$, $R_8.A_9$ и $R_8.A_{11}$ может быть выполнено в одном приложении. В этом случае в таблице Tr должны присутствовать столбцы для каждого редактируемого атрибута.

Заключение

Рассмотренный в данной статье материал относится к проблеме построения отображения реляционной БД в пользовательское представление данных. На примере «Трансформации» показаны этапы построения такого отображения, где целевым свойством является коммутативность. Многообразие моделей данных делает данную тематику практически неисчерпаемой. Причем, преобразование представления данных актуально не только для редактирования, но и для подготовки массивов данных (датасетов) при проведении различных видов анализа, в том числе для целей машинного обучения. Самое очевидное преобразование связано с разделением массива данных по какому-либо признаку. Например, преобразование $R_i^* = R_i.A_j$, в котором отношение R_i разделяется на множество отношений, в каждом из которых на атрибут A_j накладывается ограничение типа равенство.

В ближайших планах находится разработка экспериментального программного обеспечения для межмодельных преобразований в среде Microsoft Office. Выбор среды разработки

обусловлен наличием функционально насыщенных электронных таблиц и возможностью обмена данными между различными компонентами среды. Выбор иных инструментальных средств значительно усложнит разработку за счет организации интерфейса с электронными таблицами, например, потребуется привлечения языка Python со специальными библиотеками.

Финансирование. Работа выполнена в рамках государственного задания ИМ СО РАН, проект FWNF-2022-0016.

СПИСОК ЛИТЕРАТУРЫ

1. Kalinichenko L. A. Methods and tools for equivalent data model mapping construction // *Advances in Database Technology – EDBT'90*. Berlin–Heidelberg: Springer, 1990. P. 92–119.
<https://doi.org/10.1007/BFb0022166>
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back // *Proceedings of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation*. ACM, 2009. P. 179–188. <https://doi.org/10.1145/1480945.1480972>
3. Tyszkiewicz J. Spreadsheet as a relational database engine // *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010. P. 195–206.
<https://doi.org/10.1145/1807167.1807191>
4. Mi Lin, Li Chuanrong, Du Peng, Zhu Jiajia, Yuan Xinfang, Li Ziyang. Construction and application of an automatic document generation model // *2018 26th International Conference on Geoinformatics*. IEEE, 2018. P. 1–6. <https://doi.org/10.1109/GEOINFORMATICS.2018.8557127>
5. Белых Е. А., Гольчевский Ю. В. Подход к проектированию языка подстановок для генерации электронных документов, содержащих сложные таблицы // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*. 2019. Т. 29. Вып. 3. С. 422–437.
<https://doi.org/10.20537/vm190311>
6. Zykin S. V., Zykin V. S. Commutative transformations in multi-model databases // *2023 Dynamics of Systems, Mechanisms and Machines (Dynamics)*. IEEE, 2023. P. 1–4.
<https://doi.org/10.1109/Dynamics60586.2023.10349549>
7. Цаленко М. Ш. Моделирование семантики в базах данных. М.: Наука, 1989.
8. Koehler H., Link S. Inclusion dependencies and their interaction with functional dependencies in SQL // *Journal of Computer and System Sciences*. 2017. Vol. 85. P. 104–131.
<https://doi.org/10.1016/j.jcss.2016.11.004>
9. Missaoui R., Godin R. The implication problem for inclusion dependencies: a graph approach // *ACM SIGMOD Record*. 1990. Vol. 19. Issue 1. P. 36–40. <https://doi.org/10.1145/382274.382402>
10. Zykin V. S., Zykin S. V. Analysis of typed inclusion dependences with null values // *Automatic Control and Computer Sciences*. 2018. Vol. 52. Issue 7. P. 638–646.
<https://doi.org/10.3103/S0146411618070258>
11. Редреев П. Г. Построение иерархий в многомерных моделях данных // *Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика*. 2009. Т. 9. Вып. 4. Ч. 1. С. 84–87. <https://doi.org/10.18500/1816-9791-2009-9-4-1-84-87>
12. Редреев П. Г. Построение табличных приложений со списочными компонентами // *Информационные технологии*. 2009. № 5. С. 7–12. <https://elibrary.ru/item.asp?id=12137302>
13. Зыкин С. В., Зыкин В. С. Коммутативные преобразования в базе данных при редактировании многотабличных запросов // *Информационные технологии*. 2018. Т. 24. № 5. С. 330–338.
<https://doi.org/10.17587/it.24.330-338>
14. Ульман Дж. Основы систем баз данных. М.: Финансы и статистика, 1983.
15. Maier D. *The theory of relational databases*. Rockville: Computer Science Press, 1983.
<https://web.cecs.pdx.edu/~maier/TheoryBook/TRD.html>

Поступила в редакцию 22.07.2024

Принята к публикации 10.11.2024

Зыкин Сергей Владимирович, д. т. н., профессор, лаборатория методов преобразования и представления информации, Институт математики им. С. Л. Соболева СО РАН, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4.

ORCID: <https://orcid.org/0000-0002-0576-2149>E-mail: szykin@mail.ru

Полуянов Андрей Николаевич, к. т. н., старший научный сотрудник, лаборатория методов преобразования и представления информации, Институт математики им. С. Л. Соболева СО РАН, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4.

ORCID: <https://orcid.org/0000-0002-8304-0803>E-mail: pan@ofim.oscsbras.ru

Зыкин Владимир Сергеевич, к. ф.-м. н., научный сотрудник, лаборатория методов преобразования и представления информации, Институт математики им. С. Л. Соболева СО РАН, 630090, Россия, г. Новосибирск, пр. Академика Коптюга, 4.

ORCID: <https://orcid.org/0000-0002-6492-2464>E-mail: vszykin@mail.ru

Цитирование: С. В. Зыкин, А. Н. Полуянов, В. С. Зыкин. Редактирование данных с использованием модели «Трансформация» // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2024. Т. 34. Вып. 4. С. 613–625.

S. V. Zykin, A. N. Poluyanov, V. S. Zykin

Editing data using the “Transformation” model

Keywords: data model, transformation, commutativity

MSC2020: 68P15

DOI: [10.35634/vm240409](https://doi.org/10.35634/vm240409)

The problem of coordinating the semantics of data presented within different models has remained relevant for a significant period of time. First of all, this is related to the convenience of work for users who are accustomed to certain tools, for example, spreadsheets. The data prepared in these environments needs to be loaded into a centralized database, which makes it possible to get rid of duplication and inconsistency of data. An obstacle to this path is the problem of data reconciliation. Editing data directly in the database is a difficult task for non-programmer users. The traditional way to solve this problem is to develop special applications that have limited functionality. This paper proposes a technology that allows editing data in a database using spreadsheets, making their rich functionality available. The main difference from similar approaches is the use of the “Transformation” model, which makes the presentation of data convenient for human perception. Since the “Transformation” data model differs significantly from the relational model, there is a need to reconcile data between the database and spreadsheets. To solve similar problems, L. A. Kalinichenko proposed a method of commutative transformations in databases. In this paper, this technique, with some modifications, is used in algorithms for transferring data from a database to “Transformation” and back. The article presents an overview of works on the problem of data matching in various sources, a description of the data model “Transformation”, including: a description of the table schema, conditions for the existence of a table instance and data editing operations. The paper describes an algorithm for loading data into a table from a database and the algorithm for transforming data in a database in accordance with changes in the table, defines the conditions for the commutativity of the transformations, and presents a proof of the correctness of the transformations.

Funding. The research was carried out within the framework of the state assignment of the Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, project FWNF-2022-0016.

REFERENCES

1. Kalinichenko L. A. Methods and tools for equivalent data model mapping construction, *Advances in Database Technology – EDBT’90*, Berlin–Heidelberg: Springer, 1990, pp. 92–119.
<https://doi.org/10.1007/BFb0022166>
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back, *Proceedings of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation*, ACM, 2009, pp. 179–188. <https://doi.org/10.1145/1480945.1480972>
3. Tyszkiewicz J. Spreadsheet as a relational database engine, *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, ACM, 2010, pp. 195–206.
<https://doi.org/10.1145/1807167.1807191>
4. Mi Lin, Li Chuanrong, Du Peng, Zhu Jijia, Yuan Xinfang, Li Ziyang. Construction and application of an automatic document generation model, *2018 26th International Conference on Geoinformatics*, IEEE, 2018, pp. 1–6. <https://doi.org/10.1109/GEOINFORMATICS.2018.8557127>
5. Belykh E. A., Golchevskiy Yu. V. An approach to designing a substitution language for generating electronic documents containing complex tables, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2019, vol. 29, issue 3, pp. 422–437 (in Russian).
<https://doi.org/10.20537/vm190311>
6. Zykin S. V., Zykin V. S. Commutative transformations in multi-model databases, *2023 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, IEEE, 2023, pp. 1–4.
<https://doi.org/10.1109/Dynamics60586.2023.10349549>

7. Tsalenko M. Sh. *Modelirovaniye semantiki v bazakh dannykh* (Modeling semantics in databases), Moscow: Nauka, 1989.
8. Koehler H., Link S. Inclusion dependencies and their interaction with functional dependencies in SQL, *Journal of Computer and System Sciences*, 2017, vol. 85, pp. 104–131.
<https://doi.org/10.1016/j.jcss.2016.11.004>
9. Missaoui R., Godin R. The implication problem for inclusion dependencies: a graph approach, *ACM SIGMOD Record*, 1990, vol. 19, issue 1, pp. 36–40. <https://doi.org/10.1145/382274.382402>
10. Zykin V. S., Zykin S. V. Analysis of typed inclusion dependences with null values, *Automatic Control and Computer Sciences*, 2018, vol. 52, issue 7, pp. 638–646.
<https://doi.org/10.3103/S0146411618070258>
11. Redreev P. G. Construction of hierarchies in multidimensional data models, *Izvestiya of Saratov University. New Series. Series: Mathematics. Mechanics. Informatics*, 2009, vol. 9, issue 4 (1), pp. 84–87 (in Russian). <https://doi.org/10.18500/1816-9791-2009-9-4-1-84-87>
12. Redreev P. G. Construction of applications with the list components, *Informatsionnye Tekhnologii*, 2009, no. 5, pp. 7–12 (in Russian). <https://elibrary.ru/item.asp?id=12137302>
13. Zykin S. V., Zykin V. S. Commutative conversion in the database when editing a multitable query, *Informacionnye Tehnologii*, 2018, vol. 24, no. 5, pp. 330–338 (in Russian).
<https://doi.org/10.17587/it.24.330-338>
14. Ullman J. D. *Principles of database systems*, Rockville: Computer Science Press, 1980. Translated under the title *Osnovy sistem baz dannykh*, Moscow: Finansy i Statistika, 1983.
15. Maier D. *The theory of relational databases*, Rockville: Computer Science Press, 1983.
<https://web.cecs.pdx.edu/~maier/TheoryBook/TRD.html>

Received 22.07.2024

Accepted 10.11.2024

Sergey Vladimirovich Zykin, Doctor of Engineering, Professor, Leading Researcher, Laboratory of Converting Methods and Presenting Information, Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, pr. Akademika Koptyuga, 4, Novosibirsk, 630090, Russia.

ORCID: <https://orcid.org/0000-0002-0576-2149>

E-mail: szykin@mail.ru

Andrey Nikolaevich Poluyanov, Candidate of Engineering, Senior Researcher, Laboratory of Converting Methods and Presenting Information, Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, pr. Akademika Koptyuga, 4, Novosibirsk, 630090, Russia.

ORCID: <https://orcid.org/0000-0002-8304-0803>

E-mail: pan@ofim.oscsbras.ru

Vladimir Sergeevich Zykin, Candidate of Physics and Mathematics, Researcher, Laboratory of Converting Methods and Presenting Information, Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, pr. Akademika Koptyuga, 4, Novosibirsk, 630090, Russia.

ORCID: <https://orcid.org/0000-0002-6492-2464>

E-mail: vszykin@mail.ru

Citation: S. V. Zykin, A. N. Poluyanov, V. S. Zykin. Editing data using the “Transformation” model, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2024, vol. 34, issue 4, pp. 613–625.