

## СИНХРОНИЗАЦИЯ ДАННЫХ МЕЖДУ ТАБЛИЦАМИ СПЕЦИАЛЬНОГО ВИДА И БАЗОЙ ДАННЫХ

© 2025 С.В. Зыкин<sup>1</sup>, В.С. Зыкин<sup>1</sup>, Н.С. Шепелев<sup>2</sup>

<sup>1</sup>Институт математики им. С.Л. Соболева СО РАН  
(630090 Новосибирск, пр. ак. Коптюга, д. 4),

<sup>2</sup>Омский государственный технический университет  
(644050 Омск, пр. Мира, д. 11)

E-mail: [zykin@ofim.oscsbras.ru](mailto:zykin@ofim.oscsbras.ru), [vszykin@mail.ru](mailto:vszykin@mail.ru), [n06k@mail.ru](mailto:n06k@mail.ru)

Поступила в редакцию: 17.11.2025

Корректная автоматизация доступа к информации может быть выполнена за счет создания инструментальных средств, основанных на теории межмодельных отображений и обеспечении коммутативности преобразований данных. Данная статья посвящена технологии передачи данных между реляционной базой данных и табличным представлением данных специального вида. Структура таблицы является удобным средством работы пользователя, поскольку дает возможность не только редактировать данные, синхронизированные с базой данных, но и выполнять различные виды анализа с использованием электронных таблиц. В общем случае размер таблицы может быть огромным. В данной статье предлагается методика сокращения размера таблицы за счет логических ограничений при загрузке данных. При этом появляются две проблемы: фиктивные пустые значения и потеря пустых значений, необходимых для редактирования данных. В работе предложено решение этих проблем за счет использования промежуточного представления данных в виде запроса к базе данных, в котором присутствуют логические ограничения. Специальная форма этих ограничений, согласованная со стандартом SQL, является необходимым условием при решении проблемы пустых значений. Для этой цели формируются подмножества отношений из частичного порядка, который соответствует ссылочной целостности в базе данных. Полученные подграфы используются для формирования размерностей таблицы. В заключение статьи представлен анализ корректности преобразований.

*Ключевые слова:* реляционная модель данных, логические ограничения, коммутативность.

### ОБРАЗЕЦ ЦИТИРОВАНИЯ

Зыкин С.В., Зыкин В.С., Шепелев Н.С. Синхронизация данных между таблицами специального вида и базой данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2025. Т. 14, № 4. С. 25–39. DOI: 10.14529/cmse250402.

### Введение

Для взаимодействия с информационными ресурсами разрабатываются различные приложения, позволяющие делать выборочный просмотр и редактирование данных, представленных в удобном для пользователей формате. При этом существуют два подхода. Первый подход подразумевает, что приложение предоставляет пользователю регламентированный набор функций (например, просмотр расписания занятий на неделю в учебном заведении), что ограничивает возможности пользователей (например, при составлении семестрового расписания консультаций и отчетов) и является рутинной работой для программистов. Вторым подход подразумевает гибкий механизм формирования пользовательских представлений данных на основе некоторых эвристических правил и предположений. Использование эвристик приводит к потере некоторой информации из поля зрения, либо к ошибочному появлению сведений и документов. Следовательно, на первый план выходит проблема корректности таких преобразований.

Проблему корректности интеграции неоднородных данных на основе межмодельных отображений впервые исследовал Л. А. Калиниченко. В работе [1] представлен способ формализации моделей данных, обеспечивающий корректное преобразование в процессе взаимодействия гетерогенных систем с несколькими базами данных (БД). В основе предложенной методики лежат коммутативные преобразования между разнородными моделями и центральной обобщенной моделью. Предложенная методика позволяет формировать корректные преобразования данных для одной модели, если произошли какие-либо изменения в другой. В данной статье методика используется для обоснования корректности преобразований.

В статье [2] рассматривается технология передачи данных из электронных таблиц в реляционную БД и обратно. При этом БД создается динамически с использованием методов анализа содержимого электронных таблиц. По содержимому таблиц определяются функциональные зависимости, строится их минимальное покрытие, что достаточно для построения отношений БД, удовлетворяющих требованиям третьей нормальной формы (3НФ). Загрузка БД осуществляется за счет применения аналога проекции к электронной таблице. В работе не обсуждается редактирования данных, а только массовая загрузка, и логические ограничения при этом не используются. Однако такой подход дает возможность сделать вывод о корректности преобразований.

В статье [3] рассматривается технология управления БД из электронных таблиц без использования макросов либо встроенных языков программирования. Для этой цели используются формулы в электронных таблицах (операторы реляционной алгебры реализуются с помощью функций). В качестве заготовки создается электронная таблица с пустыми рабочими листами для данных и рабочими листами, заполненными формулами для запросов. Когда пользователь вводит, изменяет или удаляет данные в рабочих листах данных, формулы в рабочих листах запросов автоматически вычисляют фактические результаты запросов к БД. Редактирование данных в БД с использованием электронных таблиц не предусмотрено. Логические ограничения на данные используются внутри функций. Корректность преобразований в сфере ответственности программиста.

Для автоматизации составления запросов при подготовке документов на предприятиях [4] была разработана математическая модель для генерации документов путем интеграции приложений Visual Basic for Application (VBA), ActiveX Data Object (ADO) и Extensible Markup Language (XML). На основе анализа типов данных документов разработаны файлы конфигурации, настроены шаблоны документов и использован алгоритм автоматической генерации документов. Данная модель была использована в практических задачах и показала свою эффективность. Технология не предусматривает редактирование данных и их возврат обратно в БД.

После того, как сформировано пользовательское представление данных возникает проблема их редактирования, и возврата новых значений в БД. Для этой цели необходимо воспользоваться методикой, подложенной в работе [1].

Проблема редактирования данных в БД с использованием пользовательских представлений данных исследовалась в работе [5]. Для обеспечения коммутативности преобразований использовалась промежуточная модель «Таблица связанных соединений», в которой дополнительный «вектор вхождений» позволял однозначно идентифицировать отношение и кортеж в БД, к которым относятся сделанные модификации. Основная идея формирования «Таблицы связанных соединений» заключается в переборе различных комбинаций

отношений БД и формировании промежуточной таблицы, если совокупность отношений в комбинации удовлетворяет условию соединения без потери информации (СБПИ). Далее промежуточные таблицы объединяются в единую таблицу с удалением подчиненных кортежей. В итоге получается NP-трудная задача. Следовательно, значительные интервалы времени формирования и преобразования «Таблицы связанных соединений» не позволяют выполнять работу в оперативном режиме со значительным объемом данных. В данной работе промежуточное представление данных состоит из одного запроса, что делает задачу полиномиальной, но усложняет задачу определения пустых значений (в «Таблице связанных соединений» эта задача решается сразу с использованием вектора вхождений). Кроме того, в данной статье анализ свойства СБПИ (квадратичный алгоритм по памяти и по времени) заменен проверкой выводимости зависимости с использованием линейного по памяти и по времени алгоритма.

Редактирование многотабличного запроса к БД рассмотрено в работе [6]. Поскольку результирующая таблица (запрос) является реляционной, то алгоритмы формирования этой таблицы и возврата отредактированных значений в БД удалось сформулировать в терминах реляционной алгебры без использования промежуточных представлений данных. Реализация технологии на PostgreSQL показала удовлетворительные результаты на тестовых БД. Недостатком является то, что реляционная таблица (запрос) не является удобным средством для работы пользователя.

В данной работе получила развитие технология межмодельных коммутативных преобразований [1]. Отказ от биективности состояний моделей данных [7] позволяет работать только с частью данных, накладывать логические ограничения на формируемые таблицы, что существенно улучшает восприятие их пользователем.

Статья оформлена следующим образом. В разделе 1 представлено описание задачи и обсуждение возникающих проблем. В разделе 2 рассмотрена структура и состав логических ограничений на данные при их загрузке из БД. Раздел 3 содержит описание способа формирования размерностей табличного представления данных и анализ размерностей. В разделе 4 рассмотрены алгоритмы формирования таблицы, синхронизация ее содержимого с БД и определены условия коммутативности преобразований. Обоснование корректности преобразований представлено в разделе 5. В заключении приводится краткая сводка результатов, полученных в работе, и указаны направления дальнейших исследований.

## 1. Обсуждение проблемы

В работе [8] рассматривается модель данных «Трансформация», частный случай которой впервые был представлен в работе [9] под названием «Семантическая трансформация». Для формального определения «Трансформации» рассмотрим следующие обозначения:  $\mathcal{R} = \{R_1, R_2, \dots, R_k\}$  — реляционная БД, где  $R_i$  — отношения (таблицы), удовлетворяющие требованиям ЗНФ;  $[R_i]$  — схема отношения  $R_i$  (заголовок таблицы);  $R_i[S]$  — проекция  $R_i$ , по атрибутам  $S$  (вырезка по столбцам).  $U$  — конечное множество атрибутов, на которых задана БД. Связи между отношениями БД устанавливают ограничения ссылочной целостности и удовлетворяют типизированным ациклическим зависимостям включения [10].

Определим трансформацию  $Tr$  как табличное представление данных, где значения множества атрибутов  $X = \{X_1, X_2, \dots, X_m\}$  задают наименования строк, а значения множества атрибутов  $Y = \{Y_1, Y_2, \dots, Y_n\}$  задают наименования столбцов. Непустые множества атрибутов  $X \subset U$  и  $Y \subset U$  далее будем называть размерностями.

Особенность трансформации в том, что первые  $m$  столбцов таблицы будут озаглавлены именами атрибутов  $X_i$ . Последующие столбцы будут иметь составные имена из  $n$  значений  $y_{j,l_j}$  атрибутов  $Y_j$ ,  $j = \overline{1, n}$ . Содержимое  $i$ -й строки таблицы состоит из значений:

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

где  $p$  — количество столбцов в таблице, озаглавленных значениями атрибутов  $Y$ . В  $i$ -й строке и  $j$ -м столбце,  $j = \overline{1, m}$ , содержится значение атрибута  $X_j$ . В последующих столбцах  $i$ -й строки содержатся значения  $z_{i,l}$  атрибута  $Z \in U$ , который далее будем называть мерой. В рассматриваемой технологии значения меры подлежат редактированию с передачей новых значений в БД.

С практической точки зрения не целесообразно наличие общих атрибутов в размерностях. Кроме того, мера не должна принадлежать размерностям. Пусть  $X \cap Y = \emptyset$ ,  $Z \notin X \cup Y$ . Структура трансформации представлена в табл. 1.

Таблица 1. Пример трансформации

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

В табл. 1  $x_{ip}$  — значения атрибута  $X_i$ ,  $y_{js}$  — значения атрибута  $Y_j$ ,  $z_{ij}$  — значения атрибута  $Z$ .

Для каждой размерности строится подграф из частичного порядка, содержащий атрибуты этой размерности. При этом, множества атрибутов  $X$  и  $Y$  упорядочиваются в таблице  $Tr$  так, чтобы на верхних уровнях находились атрибуты с меньшим количеством значений, что сокращает количество дублирования на нижних уровнях. В работе [11] предложен алгоритм для построения таких иерархий в размерностях, в котором используются функциональные и многозначные зависимости исходной реляционной БД и мощности соответствующих доменов. Для удобства восприятия таблицы значения атрибутов  $X$  и  $Y$  упорядочиваются в каком-либо лексикографическом порядке.

**Определение 1.** Рассмотрим произвольное отношение  $R_i \in \mathcal{R}$ , пусть  $W$  и  $S$  — некоторые подмножества атрибутов  $[R_i]$ . Будем говорить, что в  $R_i$  реализована *функциональная зависимость*  $W \rightarrow S$ , если для любой реализации  $R_i$  не могут присутствовать два кортежа  $t_1, t_2 \in R_i$ , такие что  $t_1[W] = t_2[W]$  и  $t_1[S] \neq t_2[S]$ .

**Определение 2.** Множество атрибутов  $V$  отношения  $R_i$  является *потенциальным ключом*, если для любого кортежа  $t \in R_i$  совокупность значений  $t[V]$  может присутствовать только в кортеже  $t$ .

**Замечание 1.** Один из потенциальных ключей отношения  $R_i$ , не содержащий избыточных атрибутов, назначается первичным ключом  $R_i$ .

Очевидным условием существования таблицы  $Tr$  является невозможность присутствия в одной ячейке таблицы более одного значения атрибута меры  $Z$ . Это условие реализуется за счет использования функциональных зависимостей в исходной БД. Для множества отношений  $R = \{R'_1, R'_2, \dots, R'_p\} \subseteq \mathcal{R}$ , участвующих в формировании  $Tr$ , выделяются соответствующие им индексные файлы, обладающие свойством уникальности. По выделенным индексным файлам определяется множество реализованных (выполнимых) в БД функциональных зависимостей  $F = \{F_1, F_2, \dots, F_q\}$ . Заметим, что зависимость  $W \rightarrow S$  будет выводима, если атрибуты  $S$  принадлежит замыканию  $W_F^+$  на множестве зависимостей  $F$  [12, 13]. В силу полноты и непротиворечивости системы аксиом функциональных зависимостей [12] выводимая зависимость является выполнимой в БД, а выполнимая зависимость выводимой. Следовательно, далее достаточно проверять только выводимость зависимостей.

В качестве промежуточного представления данных будем использовать запрос «проекция–селекция–соединение» (1), представленный в терминах реляционной алгебры. Для управления размером таблицы  $Tr$  будем использовать логические ограничения на значения атрибутов.

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

где  $\bowtie$  — операция естественного соединения;  $\pi$  — операция проекции,  $\sigma$  — операция селекции,  $L$  — логическая формула на атрибутах отношений  $R'_i$ . Кортежи в соединении отношений:

$$Q_1 = R'_1 \bowtie R'_2 \bowtie \dots \bowtie R'_p,$$

подстановка которых в формулу  $L$  дает значение FALSE или UNKNOWN, будут отсутствовать в  $Q$ . Заметим, что  $Q_1$  является универсальной реляционной таблицей, удовлетворяющей требованиям первой нормальной формы (1НФ). Далее будем считать, что  $Q_1$  соответствует фактическому содержанию БД на отношениях  $R$ . Результат выполнения запроса  $Q$  является источником для формирования трансформации  $Tr$ .

Отсутствие значения атрибута  $Z$  в ячейке таблицы  $Tr$  означает, что в  $R$  значениям атрибутов  $X$  и  $Y$  не сопоставлено ни одного значения атрибута  $Z$ , то есть, в  $Q_1$  отсутствует соответствующий кортеж. Ситуация меняется при наложении ограничения  $L$ . Логическое ограничение может удалить соответствующий кортеж в  $Q_1$ . В этом случае в  $R$  векторам значений  $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$  и  $\vec{y}_j = (y_{1,j}, y_{2,j}, \dots, y_{n,j})$  соответствует значение  $z_{ij}$ , а в таблице  $Tr$  соответствующая ячейка оказывается пустой. Такая ситуация будет вводить в заблуждение пользователя, и он может попытаться заполнить эту ячейку новым значением  $\hat{z}_{ij}$ , что является ошибкой. Возможны ситуации, когда  $z_{ij}$  в таблице  $Tr$  не пусто, а в БД этому значению не сопоставлены вектора  $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$  и  $\vec{y}_j = (y_{1,j}, y_{2,j}, \dots, y_{n,j})$ , или этим векторам в БД сопоставлено другое значение  $\hat{z}_{ij}$ .

Самый простой вариант ошибочного содержимого формулы  $L$  — использование ограничения на значения меры  $Z$ . В общем случае причина появления ошибочно пустого значения в таблице  $Tr$  определена в утверждении 1.

**Утверждение 1.** Пусть кортеж  $t$  принадлежит  $Q_1$ , а в  $Q$  он удален ( $L(t)=\text{FALSE}$ ), но остался кортеж  $t_1$  совпадающий с  $t$  по  $Y$  ( $L(t_1)=\text{TRUE}$ ), но отличающийся по  $X$ , и остался кортеж  $t_2$  совпадающий с  $t$  по  $X$ , но отличающийся по  $Y$  ( $L(t_2)=\text{TRUE}$ ). Тогда в  $Tr$  на

пересечении строки  $\vec{x}_i$  и столбца  $\vec{y}_j$  будет находиться фиктивное пустое значение, тогда как в БД ему соответствует значение  $t[Z]$ .

Причиной появления значения  $z_{ij}$ , отличного от фактического в БД, может быть разрушение зависимости  $XY \rightarrow Z$ .

## 2. Структура и состав логических ограничений

Формулу  $L$  будем задавать в виде конъюнкции логических условий, определенных на различных атрибутах множества  $U$ :

$$L = L_1 \wedge L_2 \wedge \dots \wedge L_s. \quad (2)$$

В формуле (2)  $L_i, i = \overline{1, s}$ , дизъюнкция атомарных логических выражений:

$$L_i = l_{i1}(A_i) \vee l_{i2}(A_i) \vee \dots \vee l_{iv}(A_i), \quad (3)$$

где логические выражения  $l_{ij}, j = \overline{1, v}$ , определены для **одного** атрибута  $A_i \in X \cup Y$ .

С целью последующего создания программного обеспечения (ПО) атомарные выражения  $l_{ij}$  должны быть согласованы со стандартом языка SQL:

1. Арифметические операции сравнения  $A_i \Theta const$ , где операция  $\Theta$  принимает одно из значений классической шестерки ( $\Theta \in \{=, \neq, >, <, \geq, \leq\}$ ), константа  $const$ , должна быть согласована по типу с атрибутом  $A_i$  ( $\Theta$ -сравнимы), в SQL сравнимыми являются числа в различных форматах, даты, время ...
2. Операция  $A_i$  BETWEEN  $const_1$  AND  $const_2$  — значение атрибута  $A_i$  должно удовлетворять интервальному ограничению.  $const_1 \leq A_i \leq const_2$ . Операция  $A_i$  NOT BETWEEN  $const_1$  AND  $const_2$  требует, чтобы значение атрибута  $A_i$  находилось за пределами указанного интервала.
3. Операция  $A_i$  IN  $List$ , где  $List$  список значений, получит значение TRUE, если в текущем кортеже значение атрибута  $A_i$  будет совпадать с каким-либо значением в списке  $List$ . Операция  $A_i$  NOT IN  $List$  получит значение TRUE, если значение атрибута  $A_i$  будет отсутствовать в списке  $List$ .
4. Ограничение на символьные строки задаются в операции  $A_i$  LIKE  $Str$ . Операция получит значение TRUE, если строка, значение атрибута  $A_i$ , содержит в себе строку  $Str$ , заданную шаблоном. Обратная операция имеет вид  $A_i$  NOT LIKE  $Str$ .

Перечисленные варианты операций используют только часть операций языка SQL [14]. Например, не используется предикат EXISTS, поскольку в нем не заданы имена атрибутов, что исключает возможность явного управления размерностями трансформации  $Tr$ .

Естественным требованием к размерностям  $X$  и  $Y$  является отсутствие неопределенных значений. Для этого достаточно расширить логическую формулу  $L$  в запросе (1). С учетом структуры логической формулы (2) в нее дополняются условия определенности: конъюнкция операторов  $A_i \neq emp$ , где  $emp$  обозначает пустое значение, (в команде SQL:  $A_i$  IS NOT NULL) для атрибутов  $A_i$ , которые не входят в логическое выражение  $L$ , но принадлежат одной из размерностей. Если атрибут  $A_i$  входит в  $L$ , то на неопределенном значении атрибута формула примет значение UNKNOWN, что по умолчанию приведет к удалению соответствующего кортежа в  $Q_1$  вместе с неопределенным значением.

Далее необходимо определиться со способом формирования размерностей  $Tr$  и их взаимодействием с формулой  $L$ .

### 3. Анализ размерностей в трансформации $Tr$

Рассмотрим фрагмент схемы БД учебного заведения (см. рис. 1).

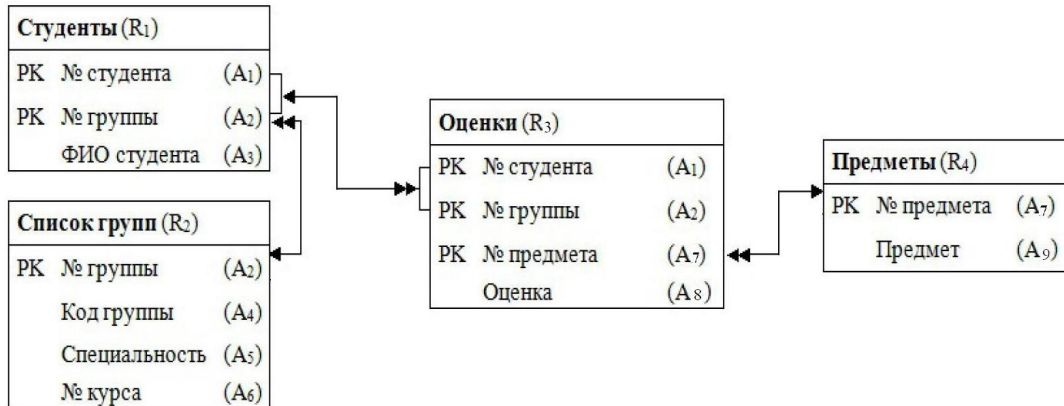


Рис. 1. Фрагмент схемы БД

Обозначение PK соответствует компонентам первичного ключа, стрелками показана ссылочная целостность, где сдвоенные стрелки указывают на внешние ключи. В скобках на схеме приведены символические имена атрибутов и отношений.

Самый простой способ формирования размерностей  $X$  и  $Y$  — получение проекций по этим множествам атрибутов:  $\pi_X(Q)$  и  $\pi_Y(Q)$ , однако, при этом возможны потери некоторых строк и столбцов в таблице  $Tr$  даже если для них логическая формула не принимает значение FALSE или UNKNOWN. Для демонстрации этого результата сформируем приложение «Сводная ведомость» (см. табл. 2), где  $X = \{R_2.A_2, R_2.A_4, R_1.A_3\}$ ,  $Y = \{R_4.A_7, R_4.A_9\}$ ,  $Z = \{R_3.A_8\}$ . Если вычислить  $X$  и  $Y$  с использованием проекций  $\pi_X(Q)$  и  $\pi_Y(Q)$ ,

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	
2	Х-212	1	Ковалев	3		4		
		2	Попов		4		5	

то строки для групп Б-211 и Х-212 будут отсутствовать в табл. 2, если экзамены по всем предметам не проводились. Либо будут отсутствовать столбцы, если экзамены по соответствующим предметам не проводились. Такой результат неприемлем, указанные строки и столбцы должны присутствовать в таблице с пустыми ячейками для проставления оценок. В этом случае формирование размерности  $X$  выполняется по формуле:

$$Dim(X) = \pi_X(\sigma_{L_X}(R_1 \bowtie R_2)),$$

где  $L_X$  логическое ограничение на атрибут «Код группы» и/или на атрибут «ФИО студента». Формирование размерности  $Y$  выполняется по формуле:

$$Dim(Y) = \pi_Y(\sigma_{L_Y}(R_4)),$$

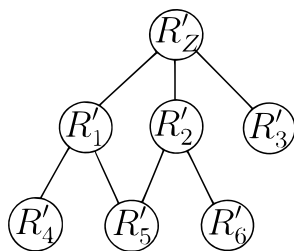
где  $L_Y$  логическое ограничение на атрибут «Предмет». Способ формирования  $L_X$  и  $L_Y$  из формулы  $L$  обсуждается далее.

**Определение 3.** Для обеспечения корректности преобразования данных в БД определим следующие три ограничения:

1. Отношения  $R = (R'_1, R'_2, \dots, R'_p)$  образуют частично упорядоченное множество относительно ссылочной целостности:  $R'_i \preceq R'_j$  если  $R'_i$  является главным отношением, а  $R'_j$  — внешним. Частичный порядок содержит один максимальный элемент  $R'_Z \in R$  ( $Z \in [R'_Z]$ ), и изолированный элемент принадлежит частичному порядку, если  $p = 1$ . Отношение  $R'_Z \in R$  соответствует семантике приложения, поскольку в нем будут реализованы операции редактирования БД.
2. В отношении  $R_Z$  множество атрибутов  $V$  является потенциальным ключом, для которого  $V \subseteq XY$ ,  $Z \notin V$  и  $V \rightarrow XY$ .
3. Только одно отношение  $R'_Z$  в  $R$  содержит атрибут  $Z$ .

**Замечание 2.** В рассмотренном приложении  $V = \{R_3.A_1, R_3.A_2, R_3.A_7\}$ ,  $R_4 \preceq R_3$ ,  $R_2 \preceq R_1 \preceq R_3$ , в силу транзитивности  $R_2 \preceq R_3$ .

Частичный порядок на отношениях БД может иметь вид, представленный на рис. 2. Связи задают ссылочные ограничения целостности, соответствующие типизированным за-



**Рис. 2.** Частичный порядок для ограничений целостности

висимостям включения [10]. Максимальный элемент частичного порядка  $R_Z$  на рис. 2 является внешним отношением для всех остальных отношений  $R_1 \dots R_6$  (все они являются главными для  $R_Z$ ).  $R_1$  является внешним для отношений  $R_4$  и  $R_5$ ,  $R_2$  является внешним для отношений  $R_5$  и  $R_6$ . Частичный порядок позволит вводить во внешние отношения только те данные, которые имеют совпадающие значения связанных атрибутов в главных отношениях. В противном случае ввод данных будет заблокирован системой управления базой данных (СУБД).

#### 4. Формирование содержимого трансформации $Tr$

Ранее было отмечено, что в трансформации  $Tr$  не должно быть фиктивных пустых ячеек для атрибута  $Z$  с одной стороны. С другой стороны должны присутствовать реальные пустые ячейки для отсутствующих в БД значений атрибута  $Z$ . Для решения этой задачи предлагается правило формирования размерностей  $Rule_1$ . Рассмотрим это правило для размерности  $X$ .

1. Формируется множество отношений  $R_X$  из множества отношений  $R$ . Каждое из отношений  $R_X$  должно содержать атрибут множества  $X$ .
2. Поиск ближайших общих предков в частичном порядке для множества отношений  $R_X$  (см. замечание 3). Выбор одного из предков для построения размерности по  $X$ .

3. Формируется подграф из всех потомков выбранного предка. Все вместе они образуют множество отношений  $\{R_{X_1}, R_{X_2}, \dots, R_{X_d}\}$ . В этом случае формирование размерности  $X$  выполняется по формуле:

$$Dim(X) = \pi_X(\sigma_{L_X}(R_{X_1} \bowtie R_{X_2} \bowtie \dots \bowtie R_{X_d})),$$

где  $L_X$  логическое ограничение, сформированное по правилу  $Rule_2$  (далее).

**Замечание 3.** Для поиска ближайших общих предков в частичном порядке целесообразно воспользоваться универсальным алгоритмом топологической сортировки Кана [15], либо какой-либо модификацией алгоритма Тарьяна [16], который более эффективен при поиске, но требует предварительной обработки графа. Если направленный граф является деревом, то общий ближайший предок один, но в частичном порядке их может оказаться несколько. Множества потомков для каждого из предков будут отличаться друг от друга не только по составу, но и по семантике. Следовательно, выбор общего предка должен сделать пользователь из предложенного списка, полученного в алгоритме.

При формировании размерностей используются логические ограничения, сформированные по правилу  $Rule_2$ . Рассмотрим это правило для атрибутов размерности  $X$ .

- В формуле (2) конъюнктивный компонент  $L_i$ , определенный для атрибута  $A_i$ , заменяем значением TRUE если  $A_i \in Y \setminus X$ , в противном случае компонент оставляем без изменения. Полученную формулу обозначим  $L_X$ .

**Замечание 4.** Применив правило  $Rule_1$  к множеству атрибутов  $Y$  получим размерность

$$Dim(Y) = \pi_Y(\sigma_{L_Y}(R_{Y_1} \bowtie R_{Y_2} \bowtie \dots \bowtie R_{Y_g})),$$

где логическая формула  $L_Y$  сформирована по правилу  $Rule_2$ , в котором атрибуты  $X$  и  $Y$  меняются местами,  $g$  — количество отношений в подграфе, выделенном в частичном порядке.

Рассмотрим схему алгоритма  $Alg_1$  для формирования  $Tr$ :

- Шаг 1. Упорядочивается множество атрибутов  $X$  и отдельно упорядочивается множество атрибутов  $Y$  для построения оптимальной иерархии размерностей в соответствии с алгоритмом [11].
- Шаг 2. Заголовками строк в  $Tr$  становится таблица  $Dim(X)$ , атрибуты в которой упорядочены в соответствии с шагом 1, а строки отсортированы по значениям атрибутов с приоритетом, соответствующим порядковому номеру атрибута в  $X$ , полученному на предыдущем шаге. Заголовками первых  $m$  столбцов становятся атрибуты  $X$ . Остальные столбцы озаглавлены транспонированной таблицей  $Dim(Y)$ , прошедшей преобразования, аналогичные  $Dim(X)$ .
- Шаг 3. Последовательно просматриваются кортежи  $Q$ . Очередное значение атрибута  $Z$  помещаются в ячейку  $Tr$ , соответствующую заголовку строки  $(x_{i,1}, x_{i,2}, \dots, x_{i,m})$  и заголовку столбца  $(y_{1,l_1}, y_{2,l_2}, \dots, y_{n,l_n})$ , значения которых получены из текущего кортежа  $Q$ .

Определим операцию по преобразованию атрибута  $Z$  в таблице  $Tr$ .

**Определение 4.** Единичной операцией  $Op$  будем считать замену одного значения  $z_{ij}$  на новое значение  $\hat{z}_{ij}$  атрибута  $Z$  в таблице  $Tr$ . Значения  $z_{ij}$  и  $\hat{z}_{ij}$  могут быть пустыми, либо непустыми. Для дальнейших операций сохраняется старое значение  $z_{ij}$ .

Алгоритм  $Alg_2$  содержимое БД преобразует в соответствии с операцией  $Op$ . Рассмотрим схему этого алгоритма:

- Шаг 1. В таблице  $Tr$  определяются векторы для строки  $\vec{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$  и для столбца  $\vec{y}_j = (y_{1,j}, y_{2,j}, \dots, y_{n,j})$ , соответствующие измененному значению  $\hat{z}_{ij}$ .
- Шаг 2. Поиск кортежа  $t \in Q$ , для которого  $t[XY] = (\vec{x}_i, \vec{y}_j)$ . Если кортеж найден, то переход на шаг 3, в противном случае переход на шаг 4.
- Шаг 3. В отношении  $R_Z$  поиск кортежа  $u$ , для которого  $u[V] = t[V]$ . Выполняется замена  $u[Z] = \hat{z}_{ij}$ . Конец алгоритма.
- Шаг 4. Пусть  $S = [R_Z] \cap (X \cup Y)$  В отношение  $R_Z$  дополняется новый кортеж  $u$ :  $u[S \cap X] = \vec{x}_i[S \cap X]$ ,  $u[S \cap Y] = \vec{y}_j[S \cap Y]$  и  $u[Z] = \hat{z}_{ij}$ . Атрибуты, не вошедшие в множество  $S \cup Z$ , получают значение *emp*. Если СУБД прервала операцию в связи с нарушением ограничений целостности в БД, то переход на шаг 5, иначе конец алгоритма.
- Шаг 5. Информирование пользователя о возникшей ошибке, возврат значения  $z_{ij}$  на прежнее место в  $Tr$ . Конец алгоритма.

**Замечание 5.** Поскольку кортеж  $u$  является частью кортежа  $t$ ,  $V \subseteq [R_Z]$  и  $V \subseteq XY$ , то на шаге 3  $u$  существует. Благодаря транзитивной зависимости  $V \rightarrow Z$  и ЗНФ кортеж  $u$  единственный в  $R_Z$ .

**Замечание 6.** Алгоритм  $Alg_2$  целесообразно выполнять асинхронно с редактированием  $Tr$  сразу после операции  $Op$ .

После определения всех необходимых компонентов коммутативность преобразований с использованием (1),  $Alg_1$ ,  $Alg_2$  и  $Op$  можно представить в следующем виде:

$$DB' \xrightarrow{(1)} Q' \xrightarrow{Alg_1} Tr' \xrightarrow{Op} Tr'', \quad (4)$$

$$DB' \xrightarrow{Alg_2} DB'' \xrightarrow{(1)} Q'' \xrightarrow{Alg_1} Tr'', \quad (5)$$

где  $DB'$  и  $DB''$  — начальное и конечное состояния БД,  $Tr'$  и  $Tr''$  — начальное и конечное состояния трансформации,  $Q'$  и  $Q''$  — результаты выполнения запроса  $Q$  до и после операции  $Op$ . Следовательно, из  $DB'$  в  $Tr''$  можно перейти двумя различными путями, но результат должен быть один и тот же, что гарантирует корректность выполненных преобразований в БД.

## 5. Анализ корректности преобразований

Не сложно убедиться в истинности следующего утверждения, в котором устанавливается связь между трансформацией и функциональными зависимостями.

**Утверждение 2.** В таблице  $Tr$  отсутствует наложение различных значений атрибута  $Z$  в одной ячейке тогда и только тогда, когда  $Q$  удовлетворяет функциональной зависимости  $XY \rightarrow Z$ .

Действительно, достаточность условия следует из определения 1, необходимость следует из условия существования  $Tr$ .

Следующая лемма дает достаточное условие для реализации зависимости в промежуточном представлении данных  $Q$ , следовательно, и в  $Tr$ .

**Лемма 1.**  $Q$  удовлетворяет функциональной зависимости  $XY \rightarrow Z$ , если она реализована в БД на выделенном множестве отношений  $R$ .

*Доказательство.* Пусть  $F$  реализованное множество зависимостей на  $R$ . Рассмотрим произвольную зависимость  $F_i \in F$ . По построению  $F$  существует отношение  $R'_j$ , в котором зависимость  $F_i$  реализована. Произвольный кортеж  $t$  в  $R'_j$  становится частью кортежа в  $Q_1$  в неизменном виде, поскольку операция естественного соединения выполняется по совпадению значений одноименных атрибутов в кортежах других отношений. Следовательно, все кортежи в  $Q_1$  удовлетворяют зависимостям  $F$ . По условию леммы зависимость  $XY \rightarrow Z$  реализована в  $R$  (выводима из  $F$ ), следовательно, она реализована в  $Q_1$ . Поскольку логическая формула  $L$  и последующая проекция только удаляют кортежи из  $Q_1$ , то они не могут нарушить зависимость  $XY \rightarrow Z$  на множестве кортежей  $Q$ .  $\square$

**Замечание 7.** Утверждение 2 и лемма 1 гарантируют, что в таблице  $Tr$  не появятся непустое значение  $z_{ij}$ , отличное от фактического значения в БД.

**Теорема 1.** Результаты преобразований (4) и (5) совпадают и не содержат фиктивных пустых значений.

*Доказательство.* После выполнения преобразований (4) будут определены векторы  $\vec{x}_i$ ,  $\vec{y}_i$  и значения атрибута  $Z$ : старое  $z_{ij}$  и новое  $\hat{z}_{ij}$ . Доказательство построим на сравнении компонентов преобразований (4) и (5).

Пусть алгоритм  $Alg_2$  завершился выполнением шага 3. Состояния  $DB'$  и  $DB''$  различаются одним значением атрибута  $Z$  в единственном ( $V \subseteq XY$ ) кортеже  $u$  отношения  $R_Z$ :  $u[Z] = z_{ij}$  и  $u[Z] = \hat{z}_{ij}$  соответственно. Множество зависимостей  $F$  при этом не изменится. Поскольку атрибут  $Z$  принадлежит только  $R_Z$ , то результаты выполнения запросов  $Q'$  и  $Q''$  будут отличаться значением этого атрибута в кортежах  $t[XY] = (\vec{x}_i, \vec{y}_i)$ . Остальные значения в кортежах  $Q'$  и  $Q''$  будут совпадать. Поскольку размерности  $Dim(X)$  и  $Dim(Y)$  остаются без изменений, то в результате преобразований (5) будет получено представление данных  $Tr''$ .

Пусть  $Alg_2$  успешно завершил работу на шаге 4. При дополнении нового кортежа  $u \in R_Z$  множество зависимостей  $F$  не изменится, поскольку  $V \subseteq XY$  и  $V$  является потенциальным ключом в  $R_Z$ . За счет нового кортежа  $u$  в  $Q''$  появится множество новых кортежей  $T$ , которых не было в  $Q'$ . Среди кортежей  $T$  найдется хотя бы один кортеж  $t$  с определенными значениями  $XY$ , поскольку соответствующие размерности есть в  $Tr'$ . Остальные будут отсеяны расширенной логической формулой  $L$ . Для кортежа  $t$  выполнено  $t[V] = u[V]$  по свойству операции естественного соединения, следовательно,  $t[XY] = (\vec{x}_i, \vec{y}_i)$ , поскольку  $V$  потенциальный ключ и зависимость  $V \rightarrow XY$  выводима из  $F$ . Из леммы 1 следует, что  $t[Z] = u[Z]$ . Следовательно, в результате преобразований (5) получим трансформацию  $Tr''$ .

Причина появления фиктивного пустого значения определена в условии 1. Способ построения размерностей на основе подграфа и структура логического ограничения  $L$  (конъюнкция ограничений) гарантируют, что в  $Tr$  не появятся соответствующие строки и столбцы, следовательно, не появится фиктивное пустое значение.  $\square$

**Замечание 8.** Раздельное формирование размерностей не препятствует появлению всех допустимых строк и столбцов, а использование только иерархий в размерностях позволяет увеличить количество пустых ячеек, доступных для редактирования. В конкретных ситуациях можно подобрать ограничения и размерности, которые дают большее количество не фиктивных пустых значений, например, использовать дизъюнкцию для различных атрибутов. Однако в общем случае это может привести к ошибкам. К фиктивным пустым ячейкам в  $Tt$  приводит использование в логической формуле  $L$  не принадлежащих размерностям атрибутов, в том числе, когда они находятся в отношениях без атрибутов размерностей. Тогда выполнение условия 3 будет недостаточно, потребуются дополнительные ограничения в виде зависимостей функциональных и/или соединения.

## Заключение

В данной статье получила развитие технология автоматического формирования пользовательского представления данных из реляционной БД с использованием таблицы «Трансформация». Логические ограничения при формировании «Трансформации» позволяют сделать ее обозримой при работе с произвольными объемами данных. Основной проблемой является возможное присутствие в таблице фиктивных пустых значений и возможное отсутствие реальных пустых значений. Для решения этой проблемы использован специальный вид логических ограничений, и способ формирования размерностей таблицы, основанный на ссылочной целостности в БД.

Структура и содержимое «Трансформации» позволяют использовать ее для анализа данных. В статье предложены алгоритмы синхронизации измененных данных в «Трансформации» с БД. Корректность достигается за счет коммутативности соответствующих преобразований.

В работе [8] представлено экспериментальное программное обеспечение, разработанное в среде Microsoft Office, которое позволило выявить проблемные места в рассматриваемой технологии и найти для них решение. В ближайших планах находится продолжение исследований «Трансформации» с целью ослабления ограничений на ее структуру и состав: увеличение количества одновременно редактируемых атрибутов БД, использование при редактировании внешних таблиц БД. Разработка экспериментального программного обеспечения планируется в среде PostgreSQL и Microsoft Office.

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

## Литература

1. Kalinichenko L.A. Methods and tools for equivalent data model mapping construction // Advances in Database Technology - EDBT'90. Vol. 416. Springer, 1990. P. 99–119. Lecture Notes in Computer Science. DOI: 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, Savannah, GA, USA, 2009. ACM, 2009. P. 79–188. DOI: 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 (SIGMOD '10). ACM, New York,

- NY, USA, 2010. P. 195–206. DOI: 10.1145/1807167.1807191.
4. Mi L., Li C., Du P., *et al.* Construction and application of an automatic document generation model // 26th International Conference on Geoinformatics, Kunming, China, 2018. P. 1–6. DOI: 10.1109/GEOINFORMATICS.2018.8557127.
  5. Редреев П.Г. Построение табличных приложений со списочными компонентами // Информационные технологии. 2009. № 5. С. 7–12.
  6. Зыкин В.С., Цымблер М.Л. Обновление многотабличных представлений на основе коммутативных преобразований базы данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 92–106. DOI: 10.14529/cmse190206.
  7. Zykin S.V., Zykin V.S. Commutative Transformations in Multi-Model Databases // Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russian Federation, 2023. P. 1–4. DOI: 10.1109/Dynamics60586.2023.10349549.
  8. Zykin S.V., Poluyanov A.N., Zykin V.S. Tool Environment for Editing Spreadsheet Applications // Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russian Federation, 2024. P. 1–5. DOI: 10.1109/Dynamics64718.2024.10838693.
  9. Цаленко М.Ш. Моделирование семантики в базах данных. М.: Наука, 1989. 288 с.
  10. Koehler H., Link S. Inclusion dependencies and their interaction with functional dependencies in SQL // J. Comput. Syst. Sci. 2017. Vol. 85. P. 104–131. DOI: 10.1016/j.jcss.2016.11.004.
  11. Редреев П.Г. Построение иерархий в многомерных моделях данных // Известия Саратовского университета. Серия Математика. Механика. Информатика. 2009. Т. 9, № 4, ч. 1. С. 84–87. DOI: 10.18500/1816-9791-2009-9-4-1-84-87.
  12. Ульман Дж. Основы систем баз данных. М.: Финансы и статистика, 1983. 334 с.
  13. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987. 608 с.
  14. Мосин С.В. Сравнение областей истинности запросов к реляционной базе данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2016. Т. 5, № 1. С. 85–99. DOI: 10.14529/cmse160108.
  15. Kahn A.B. Topological sorting of large networks // Communications of the ACM. 1962. Vol. 5, no. 11. P. 558–562. DOI: 10.1145/368996.369025.
  16. Eswaran K.P., Tarjan R.E. Augmentation problems // SIAM J. on Computing. 1976. Vol. 5, no. 4. P. 653–665.

Зыкин Сергей Владимирович, д.т.н., профессор, ведущий научный сотрудник, заведующий лабораторией МППИ, Институт математики им. С.Л. Соболева СО РАН (Омск, Российская Федерация)

Зыкин Владимир Сергеевич, к.ф.-м.н., научный сотрудник, Институт математики им. С.Л. Соболева СО РАН (Омск, Российская Федерация)

Шепелев Никита Сергеевич, аспирант Омского государственного технического университета (Омск, Российская Федерация)

## SYNCHRONIZATION OF DATA BETWEEN SPECIAL-TYPE TABLES AND THE DATABASE

© 2025 S.V. Zykin<sup>1</sup>, V.S. Zykin<sup>1</sup>, N.S. Shepelev<sup>2</sup><sup>1</sup>*Sobolev Institute of Mathematics SB RAS**(ac. Koptuyug avenue 4, Novosibirsk, 630090 Russia),*<sup>2</sup> *Omsk State Technical University (Mira avenue 11, Omsk, 644050 Russia)**E-mail: zykin@ofim.oscsbras.ru, vszykin@mail.ru, n06k@mail.ru*

Received: 17.11.2025

Correct automation of access to information can be achieved by creating tools based on the theory of inter-model mappings and ensuring the commutativity of data transformations. This paper is devoted to the technology of data transfer between a relational database and a special type of tabular representation of data. The table structure is a convenient tool for the user, since it allows not only to edit data synchronized with the database, but also to perform various types of analysis using spreadsheets. In general case, the table size can be huge. This paper proposes a technique for reducing table size by applying logical constraints when loading data. This introduces two problems: fictitious empty values and the loss of empty values needed to edit the data. The work proposes a solution to these problems by using an intermediate representation of data in the form of a database query that contains logical constraints. A special form of these constraints, consistent with the SQL standard, is necessary to deal with the null value problem. For this purpose, subsets of relations are formed from a partial order that corresponds to referential integrity in the database. The resulting hierarchies are used to form the table dimensions. The paper concludes with an analysis of the correctness of the transformations.

*Keywords: relational data model, logical constraints, commutativity.*

## FOR CITATION

Zykin S.V., Zykin V.S., Shepelev N.S. Synchronization of Data between Special-Type Tables and the Database. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2025. Vol. 14, no. 4. P. 25–39. (in Russian) DOI: 10.14529/cmse250402.

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 4.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Kalinichenko L.A. Methods and tools for equivalent data model mapping construction. Advances in Database Technology - EDBT'90. Vol. 416. Springer, 1990. P. 99–119. Lecture Notes in Computer Science. DOI: 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, Savannah, GA, USA, 2009. ACM, 2009. P. 79–188. DOI: 10.1145/1480945.1480972.
3. Tyszkiewicz J. Spreadsheet as a relational database engine. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10). ACM, New York, NY, USA, 2010. P. 195–206. DOI: 10.1145/1807167.1807191.
4. Mi L., Li C., Du P., *et al.* Construction and application of an automatic document generation

- model. 26th International Conference on Geoinformatics, Kunming, China, 2018. P. 1–6. DOI: 10.1109/GEOINFORMATICS.2018.8557127.
5. Redreev P.G. Construction of Applications with the List Components Information Technology. 2009. No. 5. P. 7–12.
6. Zykin V.S., Zymbler M.L. Updating Multi-table Views Based on Commutative Database Transformations. Bulletin of the South Ural State University. Computational Mathematics and Software Engineering. 2019. Vol. 8, no. 2. P. 92–106. (in Russian) DOI: 10.14529/cmse190206.
7. Zykin S.V., Zykin V.S. Commutative Transformations in Multi-Model Databases. Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russian Federation, 2023. P. 1–4. DOI: 10.1109/Dynamics60586.2023.10349549.
8. Zykin S.V., Poluyanov A.N., Zykin V.S. Tool Environment for Editing Spreadsheet Applications. Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russian Federation, 2024. P. 1–5. DOI: 10.1109/Dynamics64718.2024.10838693.
9. Tsalenko M.Sh. Modeling semantics in databases. Moscow: Nauka, 1989. 288 p. (in Russian).
10. Koehler H., Link S. Inclusion dependencies and their interaction with functional dependencies in SQL. J. Comput. Syst. Sci. 2017. Vol. 85. P. 104–131. DOI: 10.1016/j.jcss.2016.11.004.
11. Redreev P.G. Construction of hierarchies in multidimensional data models. Bulletin of the Saratov University. Series Mathematics. Mechanics. Computer Science. 2009. Vol. 9, no. 4(1). P. 84–87. DOI: 10.18500/1816-9791-2009-9-4-1-84-87.
12. Ullman J. Principles of Database Systems. Moscow: Finance and Statistics, 1983. 334 p. (in Russian).
13. Meyer D. The Theory of Relational Databases. Moscow: Mir, 1987. 608 p. (in Russian).
14. Mosin S.V. Truth Space Comparison of Relational Database Queries. Bulletin of the South Ural State University. Computational Mathematics and Software Engineering. 2016. Vol. 5, no. 1. P. 85–99. DOI: 10.14529/cmse160108.
15. Kahn A.B. Topological sorting of large networks. Communications of the ACM. 1962. Vol. 5, no. 11. P. 558–562. DOI: 10.1145/368996.369025.
16. Eswaran K.P., Tarjan R.E. Augmentation problems. SIAM J. on Computing. 1976. Vol. 5, no. 4. P. 653–665.