УДК 519.72

АВТОМАТИЗАЦИЯ ФОРМИРОВАНИЯ МНОГОМЕРНЫХ ПРЕДСТАВЛЕНИЙ ДАННЫХ

С. В. Зыкин

Омский филиал Института математики имени С. Л. Соболева СО РАН

Поступила в редакцию 12.11.2010 г.

Аннотация. В статье рассматривается проблема автоматизации построения многомерных представлений данных (гиперкубов) с целью последующей аналитической обработки. В качестве исходного представления данных предполагается использование базы данных реляционного типа. Для управления содержимым гиперкубов, в том числе реализации ограничений на данные, применяются различные контексты. Автоматизация построения гиперкубов достигается за счет использования свойств схемы исходной базы данных.

Ключевые слова: Реляционная база данных, гиперкуб, контекст.

Abstract. In paper the automation problem of construction of multidimensional data representations (hypercubes) is considered with the purpose of the subsequent analytical processing. The relational database is used as source data representation. For management of contents of hypercubes, including the constraint realizations, are applied various contexts. The automation of hypercubes construction by use of the scheme properties of source database is achieved.

Keywords: Relational database, hypercube, context.

1. ВВЕДЕНИЕ

Анализ накопленной информации является актуальной для многих исследовательских и прикладных задач. Наиболее популярным способом ее решения в настоящее время является технология оперативной аналитической обработки данных OLAP (online analytical processing). Основой OLAP-технологии является построение гиперкубического (многомерного) представления данных. Проблема автоматизации анализа данных актуальна как для пользователей больших корпоративных информационных систем, так и для пользователей сравнительно небольших баз данных (БД), поскольку одни и те же данные приходится многократно реорганизовывать вручную, чтобы обработать их тем или иным алгоритмом поиска закономерностей.

В работах, посвященных OLAP, значительное внимание уделяется исследованию свойств моделей гиперкубов [1–3] и операциям их преобразования [2, 4] с целью получения представления, необходимого для анализа данных. Особое внимание уделяется построению иерархий в измерениях [2, 3, 5–7], что позволяет гарантировать корректность операций агрегации данных. В работах [3, 5, 7] рассматриваются нормальные формы для многомерных моделей данных, которые позволяют контролировать значения NULL в иерархиях измерений. В большинстве работ предполагается постоянное хранение гиперкубических представлений для обеспечения наибольшей скорости обработки данных. При этом считается, что схемы гиперкубов являются неизменными и заранее хорошо продуманы.

В данной работе предполагается, что основой аналитической работы пользователя является необходимость формирования новых гиперкубов из исходного реляционного представления данных. Это необходимо при выявлении скрытых закономерностей в данных и для проведения анализа данных, не предусмотренного при проектировании складов данных. Следовательно, основное внимание необходимо акцентировать на сокращении времени формирования схемы нового гиперкуба, а формирование представления гиперкуба должно быть выполнено автоматически алгоритмами, соответствующими выбранному классу схем.

В работе [8] для автоматизации формирования гиперкуба предлагается использовать последовательность преобразований:

$$RRD \Rightarrow TJ \Rightarrow GC,$$

где RRD – реляционное представление данных, TJ – таблица соединений, GC – гиперкуб (далее соответствующие модели будут формально оп-

Работа выполнена по гранту РФФИ № 09-07-00059-а.

ределены). В данном случае *RRD* – представление исходной модели данных, *GC* – целевой. Наличие промежуточного представления *TJ* позволяет динамически управлять содержимым гиперкуба. В работе [8] при формировании *TJ* используются частичные соединения отношений, что позволяет автоматически формировать содержимое измерений и мер гиперкуба, не перекладывая эту проблему на пользователя. От него требуется только определить имена атрибутов на схеме БД, которые будут являться координатами и мерами гиперкуба, и логические условие общего вида, накладываемое на представление *GC*. Однако, такой способ формирования ограничивает допустимые реализации *GC*, прежде всего за счет ограниченного количества реализаций измерений гиперкуба.

В данной работе предлагается изменить способ формирования таблицы *TJ* за счет отдельного формирования представлений координат гиперкуба. Это сделает более трудоемким процесс создания схемы гиперкуба, но позволит увеличить количество реализаций измерений гиперкуба. Такой подход более близок к традиционным технологиям, например: Microsoft Analysis Services и ORACLE Analytic Workspace Manager. Однако, использование промежуточного представления *TJ* позволяет более гибко управлять содержимым гиперкуба. Упрощение процесса формирования гиперкуба предлагается достигнуть за счет использования свойств схемы исходной реляционной БД.

Пусть задана схема БД $RD = \{R_1, R_2, \ldots, R_k\}$, полученная в результате нормализации отношений [9, 10]. Отношения R_i определенны на множестве атрибутов $U = \{A_1, A_2, \ldots, A_n\}$. Пусть $[R_i]$ – схема отношения, множество атрибутов, на которых определено отношение R_i. Предположим, что схема *RD* редуцирована [10], то есть не существует двух отношений таких, что $[R_i] \subseteq [R_j]$, при $i \neq j$. Кортеж $t[X]$ – совокупность значений атрибутов $A_j \notin X \subseteq [R_i]$, заданных в кортеже $t \in R_i$. Неопределенное значение NULL атрибута A_j в кортеже t: $t[A_j]$ = NULL, не равно любому другому значению, в том числе другому неопределенному значению.

Для демонстрации предлагаемых технологических решений рассмотрим примеры.

Пример 1. Пусть задана упрощенная схема БД учебного заведения. Множество атрибутов *U* задано следующим набором: A_1 – ФИО студента, A_2 – наименование предмета, A_3 – код группы, A_4 – наименование специальности, A_5 – количество часов по предмету, A_6 – оценка по предмету {отл., хор., удовл., неудовл., зачет, не зачтено}, A_7 – вид контроля успеваемости {экзамен, зачет}, A_8 – номер студента в группе, A_9 – порядковый номер группы, A_{10} – порядковый номер предмета, A_{11} – номер специальности, A_{12} – вид занятия {лекция, практика, лаб.раб.}. В фигурных скобках уточняется область определения некоторых атрибутов. Для выделенных атрибутов существуют следующие зависимости:

$$DEP = \{A_8A_9 \to A_1, A_{10} \to A_2,$$
$$A_9 \to A_3A_{11}, A_{11} \to A_4,$$
$$A_{10}A_{11}A_{12} \to A_5, A_8A_9A_{10} \to A_6,$$
$$A_{10}A_{11} \to A_7,$$
$$A_{10} \twoheadrightarrow A_8A_9(A_{12}), A_{11} \twoheadrightarrow A_9(A_{10})\},$$

где одиночной стрелкой обозначены функциональные зависимости, а сдвоенной – многозначные. По правилам построения нормальных форм [9, 10] схема БД будет содержать следующие отношения:

Студенты = $R_1(\underline{A_8}, \underline{A_9}, A_1)$,
Предметы = $R_2(\underline{A_{10}}, A_2)$,
Список групп = $R_3(\underline{A_9}, A_3, A_{11})$,
Специальности = $R_4(\underline{A_{11}}, A_4)$,
Учебная нагрузка = $R_5(\underline{A_{10}}, \underline{A_{11}}, \underline{A_{12}}, A_5)$,
Оценки = $R_6(\underline{A_8}, \underline{A_9}, \underline{A_{10}}, A_6)$,
Контроль успеваемости = $R_7(\underline{A_{10}}, \underline{A_{11}}, A_7)$,

где подчеркнуты ключевые атрибуты отношений. После формирования отношений множество *DEP* должно быть дополнено зависимостями включения [11, 12, 13]:

$$\{R_1[A_9] \subseteq R_3[A_9], R_3[A_{11}] \subseteq R_4[A_{11}],$$
$$R_5[A_{10}] \subseteq R_2[A_{10}], R_5[A_{11}] \subseteq R_4[A_{11}],$$
$$R_6[A_8A_9] \subseteq R_1[A_8A_9], R_6[A_{10}] \subseteq R_2[A_{10}],$$
$$R_7[A_{10}] \subseteq R_2[A_{10}], R_7[A_{11}] \subseteq R_4[A_{11}]\}.$$

Зависимости включения на схеме БД реализуются связями (внешними ключами).

Используя БД с указанной схемой, можно выполнить следующие виды анализа: 1) успеваемость студентов по группам или по предметам; 2) зависимость успеваемости от преподавателя или специальности, 3) нагрузку студентов по предметам и т.д. Каждый из перечисленных видов анализа требует собственного представления данных.

Рассмотрим два примера формирования многомерного представления данных, для наглядности ограничимся двумя измерениями.

Пример 2. Учебный план в Вузе.

В таблице 1 атрибуты измерений гиперкуба представлены жирным шрифтом, атрибуты мер – курсивом, значения атрибутов – обычным шрифтом.

Пример 3. Сводная ведомость.

При формировании представлений в примерах 2 и 3 использовались контексты и таблица соединения, определение и способ формирования которых будут рассмотрены далее.

2. ФОРМИРОВАНИЕ СХЕМЫ ГИПЕРКУБА

Гиперкуб обычно определяется [1, 2, 4, 7] в виде совокупности измерений: $\{D_1, D_2, \ldots, D_d\}$, где D_l – множество имен атрибутов, и множества имен атрибутов M, которые называются мерами. Значения D_l являются значениями координат гиперкуба, значения M будут располагаться в рабочей области гиперкуба. В примере 3 $d = 2$, D_1 = {<номер студента в группе>, <ФИО студента>}, D_2 = {<наименование предмета>}, M = {<*оценка по предмету*>}. Для того, чтобы в одной ячейке гиперкуба было гарантированно не более одного значения атрибута меры, предполагается выполнение функциональной зависимости:

$$D_1, D_2, \ldots, D_d \to M. \qquad (2.1)$$

Далее для каждого измерения устанавливаются иерархии: в примере 3 для измерения D_1 атрибут <номер студента в группе> является корневым, а атрибут <ФИО студента> является его потомком. Для корректного выполнения операций агрегации над гиперкубом и управления неопределенными значениями в измерениях определены и исследованы нормальные формы гиперкубов [3, 5, 7]. При установлении иерархий в гиперкубе [14] предлагается использовать функциональные зависимости предков от потомков. Такой подход гарантирует отсутствие дублированных значений в потомках, однако, не всегда является удобным при визуализации гиперкуба. Так, в примере 3 в размерности D_1 имеется обратная зависимость потомка от предка, а в примере 2 такие зависи-

Таблица 1

Фрагмент учебного плана в Вузе

	наименование предмета	Математика				Физика			
	вид занятия	Лекции	Практика	лаб.раб.	вид контроля успеваемости	Лекции	практика	лаб.раб.	вид контроля успеваемости
Наименование специальности		количество часов по предмету	количество часов по предмету	количество часов по предмету		количество часов по предмету	количество часов по предмету	количество часов по предмету	
История		36	36	18	зачет.	18	18	18	зачет
Филология		18	18	18	зачет	36	36	18	экзамен
Правоведение		48	48	24	экзамен	48	48	24	зачет

Таблица 2

Фрагмент сводной ведомости

Ограничение: код группы=Ф-220					
наименование предмета		Философия	Математика	Информатика	История
номер студента в группе	ФИО студента	оценка по предмету	оценка по предмету	оценка по предмету	оценка по предмету
1	Иванов И.И.	хор.			хор.
2	Петров П.П.	отл.	хор.		
3	Сидоров С.С.		хор.		удовл.
4	Ковалев К.К.				отл.

мости вообще отсутствуют. В работах [3, 5, 7] предполагается наличие одного терминального уровня (листа) для каждой иерархии измерения. Это снимает проблему определения соответствия мер атрибутам измерений: один терминальный уровень и все меры ему соответствуют. Однако, в примере 2 атрибут меры <*количество часов по предмету*> соответствует атрибуту измерения <вид занятия>, а атрибут меры <*вид контроля успеваемости*> соответствует атрибуту <наименование предмета> того же самого измерения. Этот пример показывает, что в иерархиях могут быть несколько терминальных атрибутов.

Для динамического формирования реализации гиперкуба потребуется иной способ формирования схемы с более широкими возможностями по представлению данных. Исходные множества измерений D_l и мер M будем задавать в виде расширенных имен атрибутов: $R_i.A_j$, $A_j \in [R_i]$, где R_i – наименование i-го отношения и $[R_i]$ – схема i-го отношения.

Рассмотрим новый способ формирования схемы гиперкуба, учитывающий не только иерархии в измерениях, но и наличие в измерениях нескольких терминальных уровней и ограничений на данные.

Схема гиперкуба GC в таблице 1 может быть представлена в следующем виде:

$$\{R_4.A_4\} \times \{R_2.A_2\{R_5.A_{12}(R_5.A_5)\}(R_7.A_7)\}, \quad (2.2)$$

где $D_1 = \{R_4.A_4\}$ и $D_2 = \{R_2.A_2\{R_5.A_{12}\}\}$ – измерения, $M = \{R_5.A_5, R_7.A_7\}$ – меры. В таблице 1 измерение D_2 имеет два терминальных уровня $R_5.A_{12}$ и $R_5.A_5$, так как каждый из них имеет собственную сопоставленную меру, хотя между ними установлено иерархическое подчинение. Если бы меры не были сопоставлены измерениям, то возникает неоднозначность в расположении значений мер. Таким образом, схема (2.2) задает иерархию для измерения D_2, в которой терминальными уровнями фактически является меры. Кроме того, меры должны быть сопоставлены только атрибутам одного измерения, поскольку в противном случае в одной ячейке таблицы надо было бы сопоставить разнородные значения мер, либо дополнительно дробить ячейки в рабочей области гиперкуба.

Схема гиперкуба в таблице 2 имеет вид:

$$[R_2.A_4 = ?]\{R_1.A_8\{R_1.A_1\}\} \times \{R_2.A_2(R_6.A_6)\}, \quad (2.3)$$

где $D_1 = \{R_1.A_8\{R_1.A_1\}\}$ и $D_2 = \{R_2.A_2\}$ – измерения, $M = \{R_6.A_6\}$ – мера. Логическое ограничение: $F = [R_2.A_4 = ?]$ является шаблоном, неизвестные значения в котором запрашиваются у пользователя в начале формирования представления гиперкуба со схемой (2.3).

После того как сформирована схема GC (с установлением иерархий в измерениях и присоединением мер к атрибутам измерений), простейшим решением задачи формирования гиперкуба по множеству отношений $\{R_1, R_2, ..., R_k\}$ при $F = \varnothing$ является выполнение операции естественного соединения отношений для формирования промежуточного представления TJ:

$$TJ = R_1[V_1] \bowtie R_2[V_2] \bowtie ... \bowtie R_k[V_k], \quad (2.4)$$

где $\bowtie$ – операция естественного соединения [9, 10], V_i – множество атрибутов $A_j \in [R_i]$, для которых выполнено: либо существует измерение D_l такое, что $R_i.A_j \in D_l$, либо $R_i.A_j \in M$, либо $R_i.A_j$ атрибут, который участвует в операции естественного соединения с другими отношениями контекста (определяется зависимостями соединения [10]). Некоторые множества V_i будут пустыми, тогда соответствующие отношения R_i будут исключены из выражения (2.4). Далее значения координат формируются в виде проекций по соответствующим атрибутам: $TJ[D_l]$, с необходимой сортировкой кортежей каждого измерения в соответствии с иерархией: ведущим атрибутом для сортировки является корень, затем его потомок и т.д. Завершается построение GC присваиванием значений мер M в рабочей области GC: для каждого кортежа $t \in TJ$ на пересечении значений координат $t[D_l]$, $l = 1,...,d$, ставится значение $t[A_j]$, $R_i.A_j \in M$. В данном случае R_i будет произвольным, так как по свойству операции естественного соединения значения одноименных атрибутов для различных отношений будут совпадать. Во всех остальных ячейках GC ставится значение NULL.

Предложенная процедура формирования GC не решает следующие проблемы:

Если какому-либо набору значений измерения не соответствует ни одного значения меры, то эти значения измерений не появятся в реализации гиперкуба (по свойству операции естественного соединения). Однако отсутствие значений мер также является предметом анализа данных. Так в таблице 2 будет отсутствовать столбец для предмета «Информатика», хотя этот предмет содержится в учебном плане для данной группы.

В таблице 2 задано ограничение на атрибут “Код группы”. Следовательно, наименования

столбцов таблицы не должны содержать наименования предметов, которые в этой группе не изучаются.

Ограничения на значения измерений могут быть заданы опосредованно – через значения на связанные кортежи в отношениях, которые не входят в исходный набор отношений. В таблице 2 для формирования *TJ* достаточно отношений R_1 и R_6, для учета ограничения необходимо дополнить отношение R_3. Однако, если вычислить выражение (2.4) для выделенных отношений, то в нем будут отсутствовать студенты, которые пока не получили ни одной оценки. А такого ограничения в формуле *F* нет.

Для некоторых измерений необходимо иметь декартово произведение исходных отношений (все возможные комбинации атрибутов одного измерения), например, при составлении расписания занятий в соответствующем измерении необходимо иметь все возможные комбинации дней недели и времени начала занятий.

Для решения сформулированных проблем используются контексты. В данной статье предлагается не ограничивать пользователя в выборе структуры гиперкуба, а только подсказывать ему корректные решения по формированию гиперкуба при заданных мерах и измерениях.

3. КОНТЕКСТЫ

Рассмотрим правило формирования таблицы *TJ* для подмножеств отношений R_1, R_2, ..., R_k со схемой $\{A_1,A_2,...,A_n\}$. Правило формирования кортежей $t \in TJ$ следующее: для каждого кортежа u текущего соединения $R_{s(1)} \bowtie R_{s(2)} \bowtie \ldots \bowtie R_{s(p)}$ формируется кортеж $t[A_j] = u[A_j]$, если атрибут A_j принадлежит объединению: $[R_{s(1)}] \cup [R_{s(2)}] \cup \ldots \cup [R_{s(p)}]$, и $t[A_j] = emp$ в противном случае. Рассмотрим отношение частичного порядка над кортежами $t \in TJ$.

Определение 3.1. Кортеж $t \in TJ$ является менее определенным или равным кортежу $t' \in TJ$, когда для любого атрибута A_i выполнено: если $t[A_i] \neq t'[A_i]$, то $t[A_i] = emp$. В этом случае будем писать: $t \leq t'$ и назовем кортеж t подчиненным кортежу t'.

Рассмотренное определение частичного порядка означает то, что кортеж t' содержит в себе все менее определенные либо равные кортежи.

Определение 3.2. Представление данных *TJ* со схемой $\{A_1,A_2,...,A_n\}$, в котором удалены все подчиненные кортежи, будем называть таблицей соединения.

Определение 3.3. Множество кортежей соединения $R_{s(1)} \bowtie R_{s(2)} \bowtie \ldots \bowtie R_{s(p)}$, оставшихся в *TJ* после удаления подчиненных кортежей, будем называть остатком этого соединения.

Далее рассмотрим правила, по которым формируются множества отношений. Пусть $\{R_{s(1)},R_{s(2)},...,R_{s(p)}\}$ – произвольное множество отношений реляционной БД. *DEP* – множество зависимостей на отношениях из *R*: ФЗ – функциональные зависимости, МЗ – многозначные зависимости, ЗС – зависимости соединения [9, 10]. Используя свойство соединения без потери информации (СБПИ) [9], определим контекст.

Определение 3.4. Множество $C_i = \{R_{s(1)},R_{s(2)},..., R_{s(p)}\}$ будем называть контекстом, если оно удовлетворяет свойству СБПИ на зависимостях *DEP*. В противном случае совокупность C_i будем называть псевдоконтекстом.

Не вдаваясь глубоко в семантические проблемы интерпретации результатов реляционных операций, отметим, что в основе контекста лежит операция естественного соединения, которая собирает из различных отношений БД связанные друг с другом по значению данные. Затем эти данные (кортежи) участвуют в формировании новых структур, естественным образом дополняя и ограничивая друг друга, что делает уместным использования термина "контекст" для совокупности исходных отношений.

1. Контекст приложения. Первоначальный выбор измерений и мер гиперкуба предлагается сделать в расширенном виде: $R_i.A_j$, где R_i – наименование отношения из исходной реляционной БД, и A_j – наименование атрибута в этом отношении. Таким образом будет задано начальное множество отношений $R^0 = \{R^0_1, R^0_2, ..., R^0_q\}$, участвующее в обязательном порядке сначала в формировании таблицы соединения, а потом – гиперкуба.

Совокупность отношений, по которым строится гиперкуб должна удовлетворять свойству СБПИ [15], поскольку лишние кортежи в промежуточном представлении данных дают лишние значения в рабочей области гиперкуба. Дальнейшая задача состоит в дополнении множества R^0 отношениями из *RD*, чтобы результирующее множество удовлетворяло свойству СБПИ на множестве зависимостей *DEP*, то есть являлось контекстом. В общем случае таких вариантов дополнения существует несколько.

Каждый из вариантов (контекстов) имеет свою смысловую нагрузку, поэтому окончательный выбор контекста может выполнить только пользователь.

В примере 2 контекстом приложения является множество отношений $\{R_2, R_4, R_5, R_7\}$, в примере 3 – $\{R_1, R_6\}$.

2. Контексты ограничений. На формируемые кортежи в *TJ* накладывается ограничение в виде логической формулы, определенной на совокупности атрибутов. Логическую формулу удобнее всего задавать в виде дизъюнктивной нормальной формы (ДНФ): $F = F_1 \vee F_2 \vee \ldots \vee F_s$, где $F_i = F_{i1} \wedge F_{i2} \wedge \ldots \wedge F_{ip}$. Терм F_{ij} задается в следующем виде: $R_i.A_j.\theta$ <const>, где θ – оператор сравнения и <const> – некоторая константа, или $R_i.A_j.\theta\, R_i.A_j.$, или $R_i.A_j.\theta$ <?>, значение <?> запрашивается у пользователя при активизации приложения.

Совокупность отношений, на которые накладывается условие F_i, соответствуют одному контексту, который выступает как единое целое при формировании представления (отношения из такого контекста не должны давать альтернативные кортежи, а значит и не должны участвовать в формировании представления отдельно от своего контекста). Условия F_i и F_j задают альтернативные ограничения и, следовательно, в представлении должны присутствовать альтернативные кортежи из различных контекстов. Способ формирования контекстов для указанных ограничений совпадает с формированием контекста приложения. Начальное множество отношений задается в самой формуле F_i за счет расширенных имен атрибутов. Заметим, что в формировании контекста, соответствующего F_i, могут участвовать отношения, не принадлежащие контексту приложения. Если для каких-либо формул F_i и F_j будут получены одинаковые контексты, то их целесообразно объединить в один контекст и сопоставить ему формулу $F_i \vee F_j$. Это не повлияет на конечный результат, но сократит число промежуточных вычислений. В примере 3 контекстом ограничения является множество отношений $\{R_1, R_3\}$.

3. Контексты измерений. Для формирования связанных между собой значений измерений в качестве исходных данных достаточно указать совокупность отношений, которые надо дополнить другими отношениями, чтобы они совместно удовлетворяли свойству СБПИ. В примере 3 сводная ведомость должна содержать только те предметы, которые изучаются на данной специальности, выбрав множество отношений $\{R_2, R_4, R_5\}$ получим контекст измерения D_2. Контекст измерения D_1 в этом примере совпадает с контекстом ограничения: $\{R_1, R_3\}$.

4. Псевдоконтексты измерений. Псевдоконтекстом будем называть множество отношений, для которых не обязательно выполнено свойство СБПИ. Использование псевдоконтекста измерения $\{R_2, R_7\}$ в каком-либо представлении гиперкуба позволит иметь комбинации всех предметов со всеми возможными видами контроля успеваемости.

5. Свободные контексты. Произвольное подмножество отношений контекста приложения, удовлетворяющее свойству СБПИ, будем называть свободным контекстом. Остатки таких контекстов позволят выявить значения мер, которым не сопоставлено ни одного значения измерения. Свободные контексты не должны использоваться совместно с контекстами ограничений, так как в противном случае, возможно, что значения мер будут отфильтрованы за счет логического ограничения.

Множество всех контекстов обозначим $C = \{C_0, C_1, \ldots, C_p\}$.

4. ФОРМИРОВАНИЕ КОНТЕКСТОВ

Сформулируем критерии, которые позволят сделать перебор дополняемых отношений направленным.

Замыкание первичного ключа нового отношения R_i совпадает со всем множеством атрибутов в выбранных отношениях. Такое отношение гарантирует выполнение свойства СБПИ и получает приоритет 3.

Отношение R_i содержит внешние ключи уже выбранных отношений R_j. Такое отношение позволяет делать дополнительные подстановки в алгоритме проверки свойства СБПИ и получает приоритет 2.

Дополняемое отношение R_i должно иметь непустое пересечение с уже выбранными отношениями. В противном случае в алгоритме проверки свойства СБПИ невозможно будет сделать ни одной постановки. Такое отношение получает приоритет 1. Остальные отношения получают приоритет 0.

Формируемый контекст не должен содержать лишних отношений, наличие которых обусловлено только порядком присоединения отношений к контексту в алгоритме.

В работе [16] рассмотрен алгоритм формирования контекста приложения. Используем этот алгоритм для формирования всех перечисленных контекстов. Пусть $R^1 = \{R^1_1, R^1_2, ..., R^1_p\}$ – множество отношений не входящих в число выбранных отношений: $R^0 = RD \backslash R^1$. Рассмотрим схему алгоритма, удовлетворяющего сформулированным критериям.

1. Подсчет весов для отношений R^1, и их упорядочение по убыванию весов.

2. Формирование сочетаний без повторений из отношений R^1, сначала по одному, затем по два и так далее. Сочетания начинаются с наименьших значений, и далее последовательно увеличиваются, например сочетания по два элемента: (1,2), (1,3),..., (2,3).... Текущее сочетание отношений совместно с R^0 проверяется на выполнение свойства СБПИ. Если свойство выполнено, то полученное сочетание дополняется к множеству контекстов.

3. В процессе выполнения алгоритма пользователю предлагается выбрать нужный контекст приложения.

Замечание. Такой алгоритм на ближайших итерациях находит дополнительные отношения с наибольшей вероятностью образующие наименьший контекст с исходным множеством отношений R^0. Для формирования псевдоконтекстов не требуется использование данного алгоритма.

5. ТАБЛИЦА СОЕДИНЕНИЯ

По формуле (2.4) формируется начальное представление *TJ*, соответствующее контексту приложения C_0. Каждый кортеж представления $t \in C_0$ подставляется в логическую функцию $F(t)$, если $F(t)$ = FALSE, то кортеж t удаляется. Функция $F(t)$ = TRUE, если какой-либо ее терм F_{ij} не определен на кортеже t (отсутствует атрибут или не определен результат операции при сравнении со значением NULL).

Для формирования представлений, соответствующих контекстам C_i, $i = 1,...,p$, воспользуемся формулой

$$TJ_i = (\delta_F (R_{i1} \bowtie R_{i2} \bowtie ... \bowtie R_{il}))[X], \quad (5.1)$$

где δ – операция селекции [9], $R_{i1}, R_{i2}, ..., R_{il}$ – отношения контекста C_i, X – множество атрибутов $A_j \in [TJ_i]$, для которых существует контекст C_l такой, что $A_j \in [C_l]$, $l \neq i$, $C_l \in C$.

Далее к представлению *TJ* дополняются кортежи промежуточных соединений контекстов по следующему правилу: в массиве $m(p)$ последовательно формируем сочетания без повторений из p элементов по l, $l = p, p-1, ..., 1$. Если множество контекстов $\{C_{m(1)}, C_{m(2)}, ..., C_{m(l)}\}$ обладает свойством СБПИ, то формируется отношение (при $l = 1$ свойство СБПИ не проверяется):

$$TJ_i = (\delta_F (C_{m(1)} \bowtie C_{m(2)} \bowtie ... \bowtie C_{m(l)})). \quad (5.2)$$

Селекция δ_F присутствует в выражении, поскольку неопределенные значения на кортежах в (5.1) могут оказаться определенными в (5.2).

Для каждого кортежа $t \in TJ_i$ формируется кортеж t': $t'[A_j] = t[A_j]$ если $A_j \in [TJ]$, и $t'[A_j]$ = NULL в противном случае. Кортеж t' дополняется к *TJ*, если в *TJ* отсутствует кортеж t'': если $t'[A_j] \neq t''[A_j]$, то $t'[A_j]$ = NULL, для всех атрибутов $A_j \in [TJ]$. Множество дополненных кортежей в *TJ* за счет реализации TJ_i будем называть остатком соединения TJ_i.

Рассмотренная схема формирования *TJ* может быть реализована различными способами. В том числе с использованием операций LEFT JOIN, RIGHT JOIN и TOTAL JOIN. Однако, рассмотренная схема формирования *TJ* существенно проще: если пользователю требуется наличие остатка какого-либо соединения отношений, в том числе отдельного отношения, ему достаточно объявить это соединение контекстом или псевдоконтекстом и не задумываться о правильном порядке выполнения операций, поскольку операция естественного соединения коммутативна.

Заключительным этапом является формирование представления *GC*. Способ формирования *GC* из *TJ* аналогичен преобразованию выражения (2.4).

6. ЗАКЛЮЧЕНИЕ

Рассмотренная технология является детализацией и развитием технологии, предложенной в статье [17]. Сформулированные в статье требования к системам OLAP часто сводятся только к быстрой (за доли секунд) реакции системы на запросы пользователя-аналитика. Не во всех прикладных областях требуется такая быстрая реакция аналитика на быстро изменяющуюся ситуацию. В большинстве случаев аналитик должен вдумчиво выполнить различные виды анализа над различными представлениями данных с участием ИТ-специалиста. Рассмотренная в данной статье технология предназначена, прежде всего, для ИТ-специалиста, чтобы он мог за

приемлемое время подготовить для обработки необходимое представление данных. Хотя, после некоторого обучения, аналитик самостоятельно может освоить данную технологию.

Основным методологическим принципом в данной статье является то, что операционная база данных должна удовлетворять принципам независимости, неизбыточности, непротиворечивости и т.д. Эта база данных является ядром приложений для множества пользователей, а не только отдельно взятого аналитика.

Автору данной статьи неоднократно приходилось принимать участие в обработке медицинской информации. Опыт показывает, что основное время тратится на реорганизацию одних и тех же данных для их последующей обработки тем или иным алгоритмом анализа. Повысить производительность работы аналитика можно за счет выполнения следующих принципов:

1. операционная база данных нормализована и не зависит от конкретных приложений;

2. имеется эффективный инструмент для формирования различных пользовательских представлений данных;

3. имеется функционально развитый пакет программ для аналитической обработки данных, интерфейс которого согласован с инструментом пункта 2.

СПИСОК ЛИТЕРАТУРЫ

1. *Vassiliadis P., Sellis T.* A survey of logical models for OLAP databases // SIGMOD Rec., Vol. 28, № 4, 1999. P. 64–69.

2. *Pedersen T. B., Jensen C. S., Dyreson C. E.* A foundation for capturing and querying complex multidimensional data // Inf. Syst., Vol. 26, № 5, 2001. – P. 383–423.

3. *Lechtenborger J., Vossen G.* Multidimensional normal forms for data warehouse design // Inf. Syst., Vol. 28, № 5, 2003. P. 415–434.

4. *Li H.–G., Yu H., Agrawal D., Abbadi A. E.* Progressive ranking of range aggregates // Data & Knowledge Engineering, Vol. 63, № 1, 2007. P. 4–25.

5. *Lehner W., Albrecht J., Wedekind H.* Normal forms for multidimensional databases// Proceedings of the Tenth International Conference on Scientific and Statistical Database Management, 1998, P. 63–72.

6. *Giorgini P., Rizzi S., Garzetti M.* Goal-oriented requirement analysis for data warehouse design // In Proceedings of the 8th ACM international Workshop on Data Warehousing and OLAP: DOLAP '05, 2005. P. 47–56.

7. *Mazon J., Trujillo J., Lechtenborger J.* Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. Data Knowl. Eng. Vol. 63, № 3. 2007. – P. 725–751.

8. *Zykin S. V.* Formation of Hypercube Representation of Relational Database // Programming and Computer Software, 2006, Vol. 32, № 6. – P. 348–354.

9. *Ullman J.* Principles of database systems // Computer Science Press, 1980. – 379 p.

10. *Maier D.* The theory of relational databases // Computer Science Press, 1983. – 637 p.

11. *Casanova M., Fagin R., Papadimitriou C.* Inclusion Dependencies and Their Interaction with Functional Dependencies // Journal of Computer and System Sciences. – 1984. – № 28(1). – P. 29 – 59.

12. *Missaoui R., Godin R.* The Implication Problem for Inclusion Dependencies: A Graph Approach // SIGMOD Record. – 1990. – Vol. 19, – № 1. – P. 36 – 40.

13. *Levene M., Vincent M. W.* Justification for Inclusion Dependency Normal Form // IEEE Transactions on Knowledge and Data Engineering. – 2000. – Vol. 12, – № 2. – P. 281 – 291.

14. *Nguyen T. B., Tjoa A. M.* Conceptual Multidimensional Data Model Based on MetaCube. In Proceedings of the First International Conference on Advances in Information Systems // Springer – 2000. – P. 24–33.

15. *Miller L., Nila S.* Data Warehouse Modeler: A CASE Tool for Warehouse Design // Thirty-First Annual Hawaii International Conference on System Sciences. 1998. Vol. 6. P. 42–48.

16. *Зыкин С. В., Полуянов А. Н.* Автоматизация формирования представлений данных для их аналитической обработки // Вестник компьютерных и ин-формационных технологий, № 4, 2010. – С. 3–9.

17. *Codd E. F., Codd S. B., Salley C. T.* Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate // Codd & Date, Inc. 1993. – 31 p.

Зыкин Сергей Владимирович – заведующий лабораторией методов преобразования и представления информации Омского филиала Института математики СО РАН, доктор технических наук, профессор. Телефон: 8-3812-236739. E-mail: zykin@ofim.oscsbras.ru.

Zykin Sergey Vladimirovich – head of laboratory of methods of transformation and presentations information of Omsk Branch of Sobolev Institute of Mathematics SB RAS, technical scientific doctor, professor. Phone: 8-3812-236739. E-mail: zykin@ofim.oscsbras.ru.