УДК 519.72

ФОРМИРОВАНИЕ МНОГОМЕРНЫХ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ПРОМЕЖУТОЧНЫХ ПРЕДСТАВЛЕНИЙ[1]

С.В. Зыкин, А.Н. Полуянов

Рассмотрена проблема автоматизации формирования многомерных таблиц данных из исходной реляционной базы данных с произвольной схемой. Для формирования размерностей таких таблиц предложено использовать ранее сформированные системой представления данных. Возможность повторного использования представлений определяется путем вычисления областей истинности логических выражений.

Ключевые слова: многомерная таблица, реляционная база данных, домен.

ВВЕДЕНИЕ

Технологии аналитической обработки данных (OLAP) сформировались как направление исследований в середине 1990-х г. [1] и в настоящее время служат основой информационной поддержки принятия решений. За прошедшие годы в этой области были получены значительные фундаментальные [2—4] и прикладные [5—8] результаты.

Основное внимание в данной работе уделяется проблеме динамического формирования многомерных таблиц из реляционных баз данных (РБД) с помощью зарезервированных представлений данных на компьютере пользователя, оснащенном графическим процессором (GPU). Эта проблема связана с детально исследованной и получившей широкое применение на практике области оптимизации запросов, поскольку нацелена на сокращение объема передаваемых данных с сервера БД. Зарезервированные данные достаточно активно используются на практике в системах управления базами данных (СУБД). Однако в большинстве случаев это относится к повторному использованию блоков данных, предварительно записанных в буферный пул, без какого-либо предварительного анализа содержимого блоков на предмет возможности частичного или комбинированного использования сохраненных в них данных. Работа СУБД ограничивается тем, что при выполнении очередного запроса блоки данных не запрашиваются с внешних устройств, если они есть в буферном пуле, т. е. анализируются номера блоков, а не их содержимое. Наиболее продвинутое решение существует в СУБД Oracle благодаря применению повторно используемых SQL-операторов. Сервер Oracle предварительно просматривает разделяемый пул в поисках готового результата. К недостаткам такого подхода следует отнести необходимость создания вручную запросов с повторяющимися SQL-операторами и отсутствие возможности частичного и комбинированного использования сохраненных результатов в автоматическом режиме.

В области теоретических исследований ситуация складывается следующим образом. Огромное число публикаций посвящено проблеме построения оптимального плана запроса на основе формальных правил, в которых не используются области определения предикатов в SQL-операторах (логическая оптимизация) либо эти области учитываются при вычислении статистических оценок для оптимизации физического доступа к БД. Близкими по методам решения являются задачи выполнения запросов на потоках данных [9, 10], однако различные, по сравнению с настоящей работой, цели приводят к различным результатам. Наиболее близка к рассматриваемой проблеме работа [11]. В ней рассматриваются конъюнктивные запросы над доменами данных с предикатами в виде арифметических сравнений и представлены алгоритмы вычисления запросов с использованием существующих представлений. В настоящей работе рассматривается специальный вид универсального реляционного запроса над отношениями базы данных, а не над отдельными доменами. Хотя цели

[1] Работа выполнена при финансовой поддержке РФФИ (проект № 12-07-00066-а).

в обеих работах совпадают, результаты различны по указанной причине. В частности, в настоящей работе нет необходимости разрабатывать алгоритмы, так как их замещает реляционная алгебра.

1. ПОСТАНОВКА ЗАДАЧИ

В работе [12] для автоматизации формирования многомерной таблицы (гиперкуба) предлагается использовать последовательность преобразований

$$RDB \Rightarrow TJ \Rightarrow GC,$$

где RDB — реляционное представление данных, TJ — таблица соединений, GC — гиперкуб (далее соответствующие модели будут формально определены). В данном случае RDB — представление исходной модели данных, GC — целевой. Наличие промежуточного представления TJ позволяет динамически управлять содержимым гиперкуба благодаря определению контекстных ограничений и логическим ограничениям на данные.

Рассмотрим формализацию задачи. Пусть задана схема базы данных RDB: $R = \{R_1, R_2, ..., R_k\}$, полученная в результате нормализации отношений [13, 14]. Отношения R_i определены на множестве атрибутов $U = \{A_1, A_2, ..., A_n\}$. Пусть $\langle R_i \rangle$ — схема отношения, множество атрибутов, на которых определено отношение R_i. Предположим, что схема R является редуцированной [14], т. е. не существует двух отношений таких, что $\langle R_i \rangle \subseteq \langle R_j \rangle$, $i \neq j$. Кортеж $t[X]$ — совокупность значений атрибутов $A_j \in X \subseteq \langle R_i \rangle$, 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. Результат сравнения (арифметического, строкового и т. д.) с неопределенным значением дает значение $UNKNOWN$.

Определим размерности: $\{D_1, D_2, ..., D_d\}$, $d > 0$, где D_l — упорядоченное, в соответствии с иерархией (сначала корневой атрибут, затем его потомок и т. д.), множество расширенных имен атрибутов: $R_i.A_j$, $A_j \in \langle R_i \rangle$. Каждая размерность считается координатой в многомерном пространстве. Реализацией размерности служит некоторое отношение, упорядоченное в соответствии с иерархией (сначала по корневому атрибуту, затем по его потомку и т. д.) множество кортежей, определенное на множестве атрибутов D_l. Каждый кортеж реализации размерности является отсчетом на соответствующей координате. Для каждой размерности задается ограничение в виде логической формулы F_l, переменными в которой являются расширенные имена атрибутов. Атрибуты в выражении F_l могут не принадлежать размерности D_l, а ограничение будет задаваться опосредованно, за счет контекстов. Пусть M — непустое множество мер, заданных в виде расширенных имен атрибутов. Реализацией мер служит некоторое отношение, т. е. множество кортежей, определенное на множестве атрибутов M.

Определение 1. Гиперкубом будем называть модель логического многомерного представления данных, где каждому кортежу реализации M соответствует ровно один кортеж каждой реализации D_l. ♦

Формальные определения гиперкубов в работах [2—4] представлены через определение операций над ними и явное ограничение в виде функциональной зависимости мер от размерностей: $D_1 D_2 ... D_d \rightarrow M$.

В настоящей работе не предполагается наличие этого ограничения, что позволит использовать содержательные, как правило, не ключевые атрибуты в размерностях [15]. В результате в одной ячейке гиперкуба будет несколько значений (список) атрибута $R_i.A_j \in M$. Это не отвергает применения традиционных операций с гиперкубами, например, drill down и roll up, но потребует иной их реализации. Кроме того, все представление становится более компактным по причине меньшего числа ячеек в рабочей области гиперкуба.

Пример 1. Рассмотрим фрагмент схемы базы данных вуза:

Специальности (*Номер специальности*, Специальность),

Предметы (*Номер предмета*, Предмет),

Учебная нагрузка (*Номер специальности*, *Номер предмета*, *Семестр*, *Вид занятия*, Количество часов),

Контроль (*Номер специальности*, *Номер предмета*, *Семестр*, *Контроль успеваемости*),

где закурсивлены ключевые атрибуты отношений. Одно из возможных представлений гиперкуба приведено в таблице.

В таблице атрибуты размерностей представлены жирным шрифтом, атрибуты мер — курсивом, значения атрибутов — обычным шрифтом.

Схема гиперкуба в таблице может быть представлена в виде:

{Специальности.Специальность} × {Учебная_нагрузка.
Семестр {Предметы.Предмет {Учебная_нагрузка.
Вид_занятия (Учебная_нагрузка.Количество_часов)}
(Контроль.Контроль_успеваемости)}},

где D_1 = {Специальности.Специальность} и D_2 = {Учебная_нагрузка.Семестр {Предметы.Предмет {Учебная_нагрузка.Вид_занятия}}} — размерности, M = {Учебная_нагрузка.Количество_часов,Контроль.Контроль_успеваемости} — меры.

Логическое ограничение: F = (Контроль.Семестр = 2 ∧ (Предметы.Предмет = 'Математика' ∨ Предметы.Предмет = 'Физика')).

Фрагмент учебного плана в вузе

Семестр	2							
Предмет	Математика				Физика			
Вид занятия	Лекции	Практика	Лабораторные работы	Контроль успеваемости	Лекции	Практика	Лабораторные работы	Контроль успеваемости
Специальность	Кол-во часов				Кол-во часов			
История	36	36	18	Зачет, экзамен	18	18	18	Зачет
Филология	18	18	18	Зачет	36	36	18	Зачет, экзамен.
Правоведение	48	48	24	Зачет, экзамен	48	48	24	Зачет, экзамен

В таблице размерность D_2 имеет два терминальных уровня {Предметы.Предмет} и {Учебная_нагрузка.Вид_занятия}, так как каждый из них имеет собственную сопоставленную меру, хотя между ними установлено иерархическое подчинение. ♦

При формировании представления в примере 1 использовались так называемые контексты и таблица соединения, определение и способ формирования которых рассматриваются далее. Предлагается не ограничивать пользователя в выборе структуры гиперкуба, а только подсказывать ему корректные решения по формированию гиперкуба при заданных мерах и размерностях.

2. ФОРМИРОВАНИЕ КОНТЕКСТОВ

Кратко рассмотрим понятие контекста, правила и причины его формирования [15]. Пусть *DEP* — множество зависимостей (функциональных, многозначных, соединения и включения) на отношениях из R, $C = \{R_1, R_2, ..., R_m\}$ — произвольное подмножество отношений *RDB*.

Определение 2. Множество C при $m > 0$ будем называть контекстом, если оно удовлетворяет свойству соединения без потерь информации (СБПИ) [13, *с. 165*, 14, *с. 125*] на зависимостях *DEP*.

Замечание 1. 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 употребления термина «контекст» для совокупности таких значений.

Первоначальный выбор размерностей и мер гиперкуба выполняется 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 $R^0 = \{R_1^0, R_2^0, ..., R_q^0\}$, участвующее в обязательном порядке сначала в формировании таблицы соединения, а потом — гиперкуба. Совокупность отношений, по которым строится гиперкуб, должна удовлетворять свойству СБПИ [16], поскольку лишние кортежи в промежуточном представлении данных дают ошибочные значения в рабочей области гиперкуба. Следовательно, дальнейшая задача состоит в дополнении множества R^0 отношениями из R, чтобы результирующее множество удовлетворяло свойству СБПИ на множестве зависимостей *DEP*, т. е. 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 работе [14] предложен алгоритм последовательной генерации контекстов по сформулированным критериям, которые увеличивают вероятность более быстрого достижения результата.

Для решения проблемы дублирования значений в списках мер в работе [15] введено и обосновано понятие ключа KM_j атрибута меры $R_i.A_j \in M$.

В существующих OLAP-системах, в том числе в «Microsoft Analysis Services» и «ORACLE Analytic Workspace Manager», свойство СБПИ не анализируется, что служит основной причиной ошибок

либо ограничений при формировании представлений гиперкубов.

В терминах реляционной алгебры выражение для контекста имеет вид:

$$TJ(C) = \pi_X(\sigma_F(R_1 \bowtie R_2 \bowtie ... \bowtie R_m)), \qquad (1)$$

где $C = \{R_1, R_2, ..., R_m\}$, π_X — проекция по атрибутам X (вырезка по столбцам), σ_F — селекция (вырезка по строкам), $\bowtie$ — операция естественного соединения. Выражение (1) может быть преобразовано к оптимальному виду с учетом свойств операций реляционной алгебры [13, 14]. В данной работе предлагается использование единичных представлений $TJ(C)$ для формирования размерностей, а не их комбинаций, поэтому преобразование выражения (1) будет дублировать функции СУБД по оптимизации запроса.

В работе [15] предложен алгоритм формирования гиперкуба GC из совокупности представлений $TJ(C_s)$ с раздельным формированием размерностей. Это позволяет эффективно управлять содержимым GC, при этом гарантируется его корректность: в рабочей области присутствуют все значения, соответствующие текущему состоянию RDB, и отсутствуют избыточные значения.

3. ИССЛЕДОВАНИЕ СВОЙСТВ ПРОМЕЖУТОЧНЫХ ПРЕДСТАВЛЕНИЙ ДАННЫХ

Промежуточные представления данных обозначим как $P = \{P_1, P_2, ..., P_m\}$, где $P_v = \pi_{Xv}(\sigma_{Fv}(R_1^v \bowtie R_2^v \bowtie ... \bowtie R_{s(v)}^v))$, $s(v)$ — количество отношений в базе данных, использованных при формировании контекста $R^v = \{R_1^v, R_2^v, ..., R_{s(v)}^v\}$ и затем использовавшихся для формирования представления данных P_v. В терминах отображений это есть инъекция пар индексов в конечное множество натуральных чисел: $\{(v, 1), (v, 2), ..., (v, s(v))\} \to \{1, 2, ..., k\}$, где k — общее количество отношений в базе данных, $v = 1...m$. Целевое выражение, которое надо будет получить из представлений P, запишем в виде:

$$P^* = \pi_{X^*}(\sigma_{F^*}(R_1^* \bowtie R_2^* \bowtie ... \bowtie R_l^*)), \qquad (2)$$

что соответствует выражению в формуле (1). Пусть $Dom(F_v)$ — область определения логического выражения F_v, т. е. область значений переменных величин в формуле, для которых она принимает значение $TRUE$. Формула F_v состоит из логических операций, в которых явным образом специфицированы расширенные имена атрибутов $R_i.A_j$ (атрибут A_j в отношении R_i):

- операция сравнения $Expr_1\ \theta\ Expr_2$, θ — операция сравнения ($\theta \in \{=, \neq, >, <, \geq, \leq\}$), $Expr_i$ — согласованные по типам допустимые выражения, определенные на множестве расширенных имен атрибутов и констант;
- операция $Expr_1$ [NOT] BETWEEN $Expr_2$ AND $Expr_3$ (содержимое в прямоугольных скобках [·] для предиката не является обязательным при написании);
- операция $Expr$ [NOT] IN S, где S — список значений либо подзапрос, результатом которого является столбец атрибута A_j в отношении R_i;
- операция Str_1 [NOT] LIKE Str_2, где Str_i — строки;
- операция $Expr\ \theta$ ALL/ANY S.

Перечисленные варианты операций используют не все возможности языка SQL. Например, предикат EXISTS не используется, поскольку в нем явно не специфицированы расширенные имена атрибутов, предикат NULL используется в данной работе для другой цели. Сложность компонентов предикатов $Expr$, S и Str определяется возможностями программного обеспечения по вычислению областей определения $Dom(F_v)$, что необходимо для вычисления $TJ(C_s)$ из промежуточных представлений P_v.

При вычислении логического выражения F_v может быть получено значение $UNKNOWN$, если на текущем кортеже t атрибут принимает значение $NULL$, поскольку результаты вычисления логических выражений в SQL-запросах соответствуют трехзначной логике. Это приводит к неоднозначной интерпретации результата не только обычными пользователями, но и опытными программистами. Для решения этой проблемы предлагается ограничение: каждому атрибуту, входящему в F^*, явно присваивается признак «Использование неопределенного значения» с двумя взаимоисключающими значениями «Да» или «Нет». Семантика этого признака такова, что если ему присвоено значение «Да», то появление значения $NULL$ для указанного атрибута в текущем кортеже t не служит основанием удаления этого кортежа из дальнейшего рассмотрения. В противном случае значение признака «Нет» гарантирует, что появление значения $NULL$ для указанного атрибута в текущем кортеже t приведет к удалению этого кортежа из дальнейшего рассмотрения. Далее будем полагать, что все атрибуты формулы F^* принадлежат множеству X^*.

Пусть формула F^* в выражении (2) имеет следующий вид: $F^*(..., Q_l, ...)$, где Q_l — элемен-

тарные операции. Тогда после предложенного преобразования она примет следующий вид: $F^*(..., Q_l', ...) \wedge_{i,j} (R_i.A_j \neq NULL)$, где $\wedge_{i,j}(R_i.A_j \neq NULL)$ — конъюнкция по всем атрибутам формулы F^*, для которых не допустимо значение *NULL*. Операция $Q_l' = (Q_l \vee_{i,j} (R_i.A_j = NULL))$, где $\vee_{i,j}(R_i.A_j \neq NULL)$ — дизъюнкция по всем атрибутам предиката Q_l, для которых допустимо значение *NULL*. Внешние скобки для предиката Q_l' определяют приоритет выполнения операций. Несложно убедиться, что в рамках трехзначной логики преобразованная формула принимает только значения *TRUE* и *FALSE*. Кроме того, несложно убедиться, что в рамках двузначной логики, когда в кортежах отсутствуют неопределенные значения, исходная формула F^* будет эквивалентна преобразованной формуле, поэтому семантика представления P^* практически не искажается. Для раскрытия термина «практически» рассмотрим наиболее неудобный пример: пусть $F^* = R_1.A_2 > 3 \vee R_3.A_4 < 4$ и на кортеже t атрибут $R_1.A_2$ принимает допустимое значение *NULL*, а предикат $R_3.A_4 < 4$ принимает значение *FALSE*. Тогда преобразование формулы F^* на кортеже t будет иметь значение *TRUE*, что не совсем очевидно. Далее будем предполагать, что все формулы F^* и F_v являются преобразованными.

Рассмотрим проблему вычисления представления данных P^* из существующих промежуточных представлений P_v.

Теорема 1. $P^* \subseteq \pi_{X^*}(\sigma_{F^*}(P_v))$, *если:*

а) $X^* \subseteq X_v$,

б) $\{R_1^v, R_2^v, ..., R_{s(v)}^v\} \subseteq \{R_1^*, R_2^*, ..., R_l^*\}$,

в) $Dom(F^*) \subseteq Dom(F_v)$.

Доказательство. Пусть существует кортеж $t^* \in P^*$. По условию теоремы надо показать, что $t^* \in \pi_{X^*}(\sigma_{F^*}(P_v))$. Из условия $t^* \in P^*$ следует, что существует кортеж $t' \in R_1^* \bowtie R_2^* \bowtie ... \bowtie R_l^*$ и $t^*[X^*] = t'[X^*]$. Следовательно, существуют кортежи $t^*_i \in R^*_i$:

$$t_i^*[\langle R_i^* \rangle] = t'[\langle R_i^* \rangle], \quad i = 1, ..., l, \qquad (3)$$

и для любых пар i и j, для которых $\langle R_i^* \rangle \cap \langle R_j^* \rangle \neq \varnothing$, выполнено:

$$t_i^*[\langle R_i^* \rangle \cap \langle R_j^* \rangle] = t_i'[\langle R_i^* \rangle \cap \langle R_j^* \rangle], \; i, j = 1, ..., l. \quad (4)$$

Поскольку условия (3) и (4) выполнены для всего множества отношений $\{R_1^*, R_2^*, ..., R_l^*\}$, то они выполнены для любого его подмножества, в том числе для $\{R_1^v, R_2^v, ..., R_{s(v)}^v\}$. Следовательно, кортежи t_i^* из отношений R_j^v при выполнении операции естественного соединения образуют кортеж t'', для которого выполнено: $t'[X_v] = t''[X_v]$. Поэтому $t''[X^*] = t'[X^*] = t^*[X^*]$.

Из условия $t^* \in P^*$ следует, что $F^*(t') = TRUE$. Поскольку $Dom(F^*) \subseteq Dom(F_v)$, то формулы F^* и F_v определены на атрибутах, общих для кортежей t' и t'', следовательно, $F_v(t'') = TRUE$. Это значит, что $t'' \in R_1^v \bowtie R_2^v \bowtie ... \bowtie R_{s(v)}^v$ и $t^* = t''[X^*] \in \pi_{X^*}(\sigma_{F^*}(P_v))$. ♦

Замечание 2. Предложенные в теореме 1 условия гарантируют, что данные, необходимые для формирования представления P^*, содержатся в промежуточном представлении P_v. Однако, в нем могут быть лишние кортежи, которые дают значение *TRUE* при подстановке в формулу F^*. Дело в том, что эти кортежи будут удалены при выполнении операции естественного соединения с отношениями, которых не хватает в множестве $\{R_1^v, R_2^v, ..., R_{s(v)}^v\}$ для совпадения с множеством $\{R_1^*, R_2^*, ..., R_l^*\}$. Используя области определения атрибутов в P_v, по которым выполняется соединение, у СУБД можно запросить минимально необходимый набор данных для определения лишних кортежей.

Следующая теорема характеризует частный случай, однако проблема лишних кортежей не возникает.

Теорема 2. $P^* = \pi_{X^*}(\sigma_{F^*}(P_v))$ *тогда и только тогда, когда:*

а) $X^* \subseteq X_v$,

б) $\{R_1^v, R_2^v, ..., R_{s(v)}^v\} = \{R_1^*, R_2^*, ..., R_l^*\}$,

в) $Dom(F^*) \subseteq Dom(F_v)$.

Доказательство. Поскольку условия теоремы являются частным случаем теоремы 1, то включение $P^* \subseteq \pi_{X^*}(\sigma_{F^*}(P_v))$ доказано. Необходимо показать, что $\pi_{X^*}(\sigma_{F^*}(P_v)) \subseteq P^*$. Пусть кортеж $t_v \in \pi_{X^*}(\sigma_{F^*}(P_v))$. Покажем, что $t_v \in P^*$. Из условия $t_v \in \pi_{X^*}(\sigma_{F^*}(P_v))$ следует, что существует кортеж $t' \in R_1^v \bowtie R_2^v \bowtie ... \bowtie R_{s(v)}^v$ и $t_v = t'[X^*]$. По свойству операции естественного соединения $R_1^v \bowtie R_2^v \bowtie ... \bowtie R_{s(v)}^v = R_1^* \bowtie R_2^* \bowtie ... \bowtie R_l^*$, следовательно, $t' \in R_1^* \bowtie R_2^* \bowtie ... \bowtie R_l^*$. Из условий $t_v \in \pi_{X^*}(\sigma_{F^*}(P_v))$ и $Dom(F^*) \subseteq Dom(F_v)$ следует, что $F^*(t') = TRUE$, а значит $t_v \in P^*$. ♦

Замечание 3. Выполнение условий теорем 1 и 2 не представляет собой исключительный случай, поскольку набор размерностей при формировании гиперкубов изменяется редко, что позволит формировать эти размерности из промежуточных представлений без обращения к СУБД. Кроме того, используя области определения атрибутов в P_v, у СУБД можно запросить минимально необходимый набор данных для определения значений мер в рабочей области гиперкуба.

ЗАКЛЮЧЕНИЕ

Основной результат данной работы состоит в обосновании условий использования зарезервированных представлений данных для формирования новых гиперкубов. Эти представления данных могут храниться на компьютере пользователя-аналитика и существенно сократить время на формирование данных, необходимых для принятия решений.

Традиционная проблема в системах, работающих не под управлением СУБД, заключается в актуализации данных, полученных из БД. В технологии MOLAP она решается путем периодической актуализации содержимого гиперкуба службой сопровождения. Аналогичный способ может быть применен для актуализации представлений P_v. Для сокращения времени актуализации возможно получить доступ к журналу изменений, который сопровождает СУБД, и актуализировать только те представления, для которых изменились исходные данные в БД. Однако еще раз отметим, что данные, используемые в размерностях гиперкуба, обычно редко модифицируются в БД.

Предлагаемый подход подразумевает в дальнейшем применение графических процессоров для выполнения промежуточных операций фильтрации и соединения различных представлений данных. Исследования в этой области [17, 18] показали, что операции на графическом процессоре выполняются существенно быстрее, чем на центральном процессоре, для небольших объемов данных. При увеличении объемов производительность падает. Причина очевидна: пока данные помещаются в быструю память графического процессора, операции с ними выполняются быстро. При увеличении объемов данных появляется необходимость их многократного перемещения между различными видами памяти, что снижает производительность. С одной стороны, производители графических процессоров постоянно наращивают быструю память, с другой — повторное использование данных наиболее актуально для размерностей многомерных таблиц, для которых требуется сравнительно небольшой объем памяти, что в совокупности делает целесообразным применение графических процессоров при решении указанной проблемы.

ЛИТЕРАТУРА

1. *Codd E.F.*, *Codd S.B.*, *Salley C.T.* Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. — Sunnyvale (CA): Codd & Date Inc., 1993. — 31 p.
2. *Lechtenborger J.*, *Vossen G.* Multidimensional normal forms for data warehouse design // Inf. Syst. — 2003. — Vol. 28, N 5. — P. 415—434.
3. *Lehner W.*, *Albrecht J.*, *Wedekind H.* Normal forms for multidimensional databases // Proc. of the Tenth Intern. Conf. on Scientific and Statistical Database Management. — Capri, 1998. — P. 63—72.
4. *Mazon J.-N.*, *Trujillo J.*, *Lechtenborger J.* Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms // Data & Knowledge Engineering. — 2007. — Vol. 63, N 3. — P. 725—751.
5. *Vassiliadis P.*, *Sellis T.* A survey of logical models for OLAP databases // SIGMOD Rec. — 1999. — Vol. 28, N 4. — P. 64—69.
6. *Pedersen T.B.*, *Jensen C.S.*, *Dyreson C.E.* A foundation for capturing and querying complex multidimensional data // Inf. Syst. — 2001. — Vol. 26, N 5. — P. 383—423.
7. *Progressive* ranking of range aggregates / H.-G. Li, et al. // Data & Knowledge Engineering. — 2007. — Vol. 63, N 1. — P. 4—25.
8. *Giorgini P.*, *Rizzi S.*, *Garzetti M.* Goal-oriented requirement analysis for data warehouse design // Proc. of the 8th ACM international Workshop on Data Warehousing and OLAP: DOLAP '05. — Bremen, 2005. — P. 47—56.
9. *Olston C.*, *Jiang J.*, *Widom J.* Adaptive filters for continuous queries over distributed data streams // Proc. of the 2003 ACM SIGMOD Intern. Conf. on Management of Data (SIGMOD '03). — San Diego, 2003. — P. 563—574.
10. *Denny M.*, *Franklin M.J.* Predicate result range caching for continuous queries // Proc. of the 2005 ACM SIGMOD Intern. Conf. on Management of Data (SIGMOD '05). — N.-Y., 2005. — P. 646—657.
11. *Afrati F.*, *Li C.*, *Mitra P.* Rewriting queries using views in the presence of arithmetic comparisons // Theoretical Computer Science. — 2006. — Vol. 368, N 1—2. — P. 88—123.
12. *Зыкин С.В.* Формирование гиперкубического представления реляционной базы данных // Программирование. — 2006. — № 6. — С. 71—80.
13. *Ульман Дж.* Основы систем баз данных. — М.: Финансы и статистика, 1983. — 334 с.
14. *Мейер Д.* Теория реляционных баз данных. — М.: Мир, 1987. — 608 с.
15. *Зыкин С.В.* Формирование гиперкубического представления данных со списочными компонентами // Информационные технологии и вычислительные системы. — 2010. — № 4. — С. 38—46.
16. *Miller L.*, *Nilakanta S.* Data Warehouse Modeler: A CASE Tool for Warehouse Design // Thirty-First Annual Hawaii International Conference on System Sciences. — Kohala Coast, 1998. — Vol. 6. — P. 42—48.
17. *Fast* computation of database operations using graphics processors / N.K. Govindaraju, et al. // Proc. of the 2004 ACM SIGMOD Intern. Conf. on Management of Data (SIGMOD '04). — Paris, 2004. — P. 215—226.
18. *Blas A.D.*, *Kaldewey T.* Why graphics processors will transform database processing // Spectrum, IEEE. — 2009. — Vol. 46, N 9. — P. 46—51.

Статья представлена к публикации руководителем РРС А.К. Погодаевым.

Сергей Владимирович Зыкин — д-р техн. наук, зав. лабораторией, ✉ zykin@ofim.oscsbras.ru,

Андрей Николаевич Полуянов — канд. техн. наук, науч. сотрудник, ✉ Andrey.Poluyanov@gmail.com,

Институт математики им. С.Л. Соболева СО РАН, г. Омск, ☎ (3812) 23-67-39.