

УДК 62.501.72:57

Р. Т. ВОЛКОЛУПОВА

ПОСТРОЕНИЕ БАЗЫ ДАННЫХ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

Использование высокого уровня организации структурных и функциональных отношений в организационных системах, обеспечивающих жизнеспособность биологическим системам, рассмотрено в [1]. Для исследования такого рода систем требуется специальное построение баз данных. Концепция этих баз данных, обусловленная возможностями централизации информации, позволяет ввести принципиально новую технологию обработки данных и существенно повысить эффективность исследования функционирования такой организационной системы.

Одним из главных требований к базе данных является минимизация избыточности данных, используемых в процессе функционирования. Использование в ходе решения исследовательских задач одних и тех же данных по несколько раз в различных программах позволяет в определенных условиях снизить затраты на вводе исходных данных. Важным требованием является также независимость данных, которая трактуется как свобода пользователя от структуры обрабатываемых данных. Стремление разработчиков оградить программы пользователей от необходимости их переработки при каждой модификации структуры данных или свести эту переработку к минимуму объясняется существенной зависимостью стоимости системы от стоимости программных средств. Обеспечение этих требований может быть осуществлено решением ряда задач по рациональному использованию базы данных. Сформулируем одну из них и изложим предлагаемый метод её решения.

Задача. Имеется соответствующая база данных иерархической структуры. Требуется организовать наиболее экономичный формализованный процесс выбора необходимых частей базы данных.

Решение данной задачи рассмотрим на примере. Организационную систему представим в виде системы, в которой четко определены как отдельные элементы (реквизиты или информационные массивы), их совокупности, подсистемы, так и взаимосвязи.

Тогда наиболее подходящей и распространенной моделью структуры такой системы будет линейный конечный граф, соединяющий геометрическую наглядность структуры с её содержательностью.

Если каждому элементу сетевой системы идентифицировать некоторую вершину графа x_i , геометрический образ которой на графе будет задан точкой, то совокупность их определит множество $x_i \in X$. Взаимосвязи между элементами, определенные на графе с помощью отрезков кривых, идентифицируем u_j . Они образуют множество связей $u_j \in U$. Тогда совокупность множества элементов X и множества связей U представляют собой граф $G(X, U)$ с вершинами $x_i \in X$ и ребрами $u_j \in U$, моделирующий структуру рассматриваемой системы [2]. Вершины таких графов моделируют простые элементы системы, часто называемые двухполосниками.

Однако, в рассматриваемой сложной системе при определенной степени детализации, кроме простых элементов, двухполосников, рассматриваются и некоторые подсистемы, состоящие, в свою очередь, из простых элементов и даже подсистем меньших размеров. В графе такие подсистемы задаются многополосниками.

Многополосник – это подграф, моделирующий сетевую подсистему, внутренняя структура которой не раскрывается при рассмотрении данного графа. Он связан с другими элементами графа с помощью вершинполосов, не имеющих в графе связи друг с другом. При этом вершинполосы являются двухполосниками.

Таким образом, модель структуры сложной системы может быть задана моделирующим графом, который определен следующим образом. Под моделирующим графом $G(X, U)$ будем понимать математическое представление структуры сложной сетевой системы, состоящей из простых элемен-

тов, двухполосников, составляющих множество простых вершин графа x_k , и подсистем, многополосников, заданных в графе полюсамивершинами $x_l, x_k, \cup x_k = X$, а также совокупностью взаимосвязей U вершин и вершинполосов, определяющих ребра моделирующего графа.

Тогда математической моделью структуры системы базы данных, на первом уровне абстракции будет моделирующий граф.

Графы, содержащие многополосники, изучены недостаточно. Предложенное представление многополосников позволяет рассматривать их как совокупность невзаимосвязанных вершинполосов, равноправных с вершинами графов. Тогда описания, преобразования, топологические исследования моделирующих графов, содержащих многополосники, ведутся также как с графами, состоящими из двухполосников. Связь вершин графа с вершинами подграфов осуществляется через их вершиныполосы. Все топологические понятия, действительные для графов с двухполосниками при данном представлении графа, содержащего многополосники, остаются в силе и для последнего. Это обстоятельство позволяет обобщить два вида моделирующих графов в один общий и в дальнейшем вести все рассуждения для общего моделирующего графа, допускающего в качестве своих элементов модели двухполосников и многополосников.

Представление структуры системы её моделью, графом, позволяет абстрагироваться от специфики системы, рассматривать её в обобщенном виде и упрощать процесс последующего исследования посредством использования имеющихся моделей. Тогда структурные исследования сетевых систем могут рассматриваться как практическое толкование теории графов.

Исследования структуры базы данных на её модели, общем моделирующем графе позволяет, образно говоря, увидеть "лес" (информационное содержание данных), а не "отдельные деревья" (конкретные значения данных).

Назовем многополосники рассматриваемой модели структуры сетевой системы, базы данных, подструктурами, а общий моделирующий граф для краткости – просто графом.

Рассмотрение структуры базы данных на логикоматематическом уровне абстракции позволяет построить логическую схему для отдельных подструктур многополосников в виде подсхем, а затем интегрировать их для определения общей логической схемы структуры [3]. Тогда общая логическая схема будет представлять собой таблицу используемых типов данных, в качестве которых выступают имена реквизитов. Структуру базы данных на концептуальном уровне будем называть информационной структурой. Она является проблемноориентированной и системнонезависимой, т.е. независимой от конкретной системы базой данных и операционной системы. Воспользуемся обоими рассмотренными уровнями абстракции для решения задачи идентификации подструктур структуры базы данных.

Итак, модель структуры базы данных, общий моделирующий граф, представлен совокупностью логических подсхем, что позволяет конкретному пользователю использовать только отдельные части интегрированной информации, рассматриваемые как элементы структуры базы данных.

Для одной и той же схемы структуры базы данных можно специфицировать несколько подсхем, которые пересекаются. В отдельных случаях подсхема может равняться всей схеме структуры базы данных либо специфицировать всего один тип записей, под которыми будем понимать модель данных "плоский файл", в которой категории называются типами записей, свойства категории – атрибутами. Однако, в одной подсхеме нельзя специфицировать элементы, принадлежащие разным многополосникам: каждая подсхема соответствует только одному из них. Подсхема составляется в соответствии с нуждами конкретной прикладной программы, её можно использовать и для других программ, если она удовлетворяет их требованиям.

Представление многополосников подсхемами позволяет лучшим образом идентифицировать подструктуру баз данных, что обеспечит эффективность функционирования по времени и стоимости процесса выбора данных, а также позволит сократить объем используемой памяти. Рассмотрим изложенное выше утверждение на тестовом примере.

Геометрически математическую модель базы данных можно представить древовидным графом.

Для каждой вершины графа определяется наибольшее из расстояний $d(x_i, x_j)$ ко всем остальным его вершинам. Наибольшее из таких чисел будет диаметром графа, наименьшее – его радиусом. Вершины дерева, для которых максимальное расстояние до других вершин равно радиусу графа, называются корневыми. Для дерева нашей математической модели, диаметр которого равен 5, а радиус – 3, корневым вершинам дадим названия x_1, x_3 .

Пронумеруем сверху вниз и слева направо числами натурального ряда понятия нижнего уровня иерархии базы данных, т.е. присвоим им числовые идентификаторы. Представим эти числа в виде векторов. Пусть $H(1,2,3,4)$ номер зоны, где зона – это логическая или физическая группировка последовательности данных. $M_1(1,7,13,19)$ – минимальное значение каждой зоны, $P=1$ шаг, $M_2(6,12,18,20)$ – максимальное значение каждой зоны.

Для свертывания, которое состоит в ассоциации некоторых чисел к одному целому числу, используем так называемую формулу мультипликативных оснований.

$$N = n_1 + B_1 * n_2 + B_1 * B_2 * n_3 + \dots + B_1 * B_2 * \dots * B_h * n_{h+1} + \dots + B_1 * B_2 * \dots * B_{p-1} * n_p, \quad (1)$$

где B_1, B_2, \dots, B_{p-1} – основания словаря (под основанием словаря следует понимать совокупность закодированных слов, которые помещаются в одной зоне); n_1, n_2, \dots, n_p – совокупность чисел, которые должны храниться в зонах; * – умножение. Обратное выражение

$$N = Q_1 * B_1 + N_1, \quad (2)$$

где N_1 – остаток от деления N на B_1 ; Q_1 – результат деления N на B_1 . Основание словаря определяется по формуле

$$B = (M - m) / p + 1, \quad (3)$$

где M, m – максимальное и минимальное значение зоны соответственно, шаг P .

Найдем основание словаря для каждой зоны с помощью векторов: $B_1 = (6 - 1) / 1 + 1 = 6$, $B_2 = (12 - 7) / 1 + 1 = 6$, $B_3 = (18 - 13) / 1 = 6$, $B_4 = (20 - 19) / 1 + 1 = 2$.

Из формулы кодирования

$$Z = (N - m) / P, \quad (4)$$

где N – число, которое хранится в записи, находим N, n_1, n_2, \dots, n_p для каждого кортежа $\langle 01, 07, 13 \rangle, \langle 02, 08, 14, 19 \rangle, \langle 03, 09, 15, 20 \rangle, \langle 04, 10, 16 \rangle, \langle 05, 11, 17 \rangle, \langle 06, 12, 18 \rangle$.

Из (4) для кортежа $\langle 01, 07, 13 \rangle$ $n_1 = (1 - 1) / 1 = 0$; $n_2 = (7 - 7) / 1 = 0$; $n_3 = (13 - 13) / 1 = 0$; $N = 0 + 6 * 0 + (6 * 6) * 0 = 0$.

Для кортежа $\langle 02, 08, 14, 19 \rangle$ $n_1 = (2 - 1) / 1 = 1$; $n_2 = (8 - 7) / 1 = 1$; $n_3 = (14 - 13) / 1 = 1$; $n_4 = (19 - 19) / 1 = 0$; $N = 1 + 6 * 1 + (6 * 6) * 1 + (6 * 6 * 6) * 0 = 43$.

Аналогично для остальных кортежей $\langle 03, 09, 15, 20 \rangle, \langle 04, 10, 16 \rangle, \langle 05, 11, 17 \rangle, \langle 06, 12, 18 \rangle$.

Соответственно $N = 302$, $N = 129$, $N = 172$, $N = 215$.

Из геометрической модели можно выделить одиннадцать многополосников, моделирующих подструктуры: $\langle X_1, X_2, X_3 \rangle, \langle X_1, X_2 \rangle, \langle X_1, X_2, X_4 \rangle, \langle X_2, X_4 \rangle, \langle X_1, X_3, X_5, X_6 \rangle,$

$\langle X_1, X_3 \rangle, \langle X_1, X_3, X_5 \rangle, \langle X_3, X_5 \rangle, \langle X_3, X_5, X_6 \rangle, \langle X_5, X_6 \rangle, \langle X_1, X_2, X_3, X_4, X_5 \rangle$.

Воспользуемся тем, что всякой числовой последовательности $A = (a_i^j)$ составленной из m элементов $a_i^j \in \overline{0, n-1}$, отвечающих ограничению $a_i^j \neq a_\ell^j, i \neq \ell, \ell = \overline{1, m}$ можно поставить в соответствие кодномер $0 \leq N_j \leq |A_m| - 1$ с помощью выражения

$$N_j = \sum_{i=1}^m (a_i^j - S(a_i^j)) \rho_i, \quad (5)$$

где $S(a_i^j) = \sum_{r=1}^{i-1} \text{SIGN}(\text{SIGN}(a_i^j - a_r^j) + 1)$; $\rho_i = \prod_{S=i+1}^m (n+1-S) = (n-i)/(n-m)!$; A_m – количество

разных последовательностей ($|A_m| = n!/(n-m)! = \prod_{i=1}^m (n+1-i)$; $a_0^j = n$)

Найдем N_j для кортежа $\langle X_1, X_2, X_3 \rangle$, т.е. $\langle 0, 43, 129 \rangle, n = 6, m = 3$

$$N_j = (0-0)5!/3! + (43-1)4!/3! + (129-1)3!/3! = 296.$$

Для кортежа $\langle X_1, X_2 \rangle$, т.е. $\langle 0, 43 \rangle, n = 6, m = 3$

$$N_j = (0-0)5!/4! + (43-1)4!/4! = 42.$$

Для кортежа $\langle X_2, X_4 \rangle = \langle 43, 302 \rangle, n = 6, m = 3$

$$N_j = (43-1)5!/4! + (302-1)4!/4! = 511.$$

Аналогично для остальных кортежей $\langle X_1, X_2, X_4 \rangle = \langle 0, 43, 302 \rangle, \langle X_1, X_3, X_5, X_6 \rangle = \langle 0, 129, 172, 215 \rangle, \langle X_1, X_3 \rangle = \langle 0, 129 \rangle, \langle X_1, X_3, X_5 \rangle = \langle 0, 129, 172 \rangle, \langle X_3, X_5 \rangle = \langle 129, 172 \rangle, \langle X_3, X_5, X_6 \rangle = \langle 129, 172, 215 \rangle, \langle X_5, X_6 \rangle = \langle 172, 215 \rangle, \langle X_1, X_2, X_3, X_4, X_5 \rangle = \langle 0, 43, 129, 302, 172 \rangle.$

Соответственно $N_j = 469, N_j = 2263, N_j = 128, N_j = 683, N_j = 811, N_j = 3458, N_j = 1069, N_j = 2550.$

Рассмотренный метод для хранения схемы базы данных требует 31 число для идентификации реквизитов, тогда как традиционный – 251 литералов. После свертывания, мы будем манипулировать только с 15 числами, которые займут 8 байт памяти.

Таким образом, мы получаем экономию памяти на 50% и соответственно сокращаем время поиска данных. Предлагаемый метод позволяет также избавиться от операций деления и умножения при кодировании и декодировании, поскольку шаг $P = 1$, и уменьшить их стоимость.

Список литературы: 1. Волколупова Р.Т. Построение базы алгоритмов программ специального назначения // Проблемы бионики. Республиканский межведомственный сборник. 1988. Вып. 40. 2. Волколупова Р.Т. Выбор дерева экстремальной длины на графе, содержащем многополюсники // Автоматизированные системы управления и приборы автоматики. 1975. Вып. 35. С.107-112. 3. Мартин Дж. Организация баз данных в вычислительных системах. М.: Мир, 1980. 662 с.

Поступила в редколлегию 2.08.2000