**УДК 004.658.3**

**А.Б. Кунгурцев,** канд. техн. наук, доц.,
**С.Л. Зиноватная,** инженер,
Одес. нац. политехн. ун-т

# МОДЕЛЬ РЕСТРУКТУРИЗАЦИИ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ ПУТЕМ ДЕНОРМАЛИЗАЦИИ СХЕМЫ ОТНОШЕНИЙ

*О.Б. Кунгурцев, С.Л .Зіноватна.* **Модель реструктурізації реляційної бази даних шляхом денормалізації схеми відносин.** Розглядаються варіанти денормалізації бази даних, математичне подання схеми бази даних у результаті застосування конкретного варіанту, умови застосування, переваги використання й вимоги до додаткової обробки даних.

*A.B. Kungurtsev, S.L. Zinovatnaya.* **The model of restructurization the relational data base by using scheme denormalization of relations**. Variants of data base denormalization, mathematical representation of data base scheme as a result of concrete variant application, application conditions, advantages of usage and requirements to additional data processing are considered.

При создании структуры реляционной базы данных (БД) отношения нормализуются для устранения избыточности данных, которая может вызвать проблемы при модификации этих отношений. В нормализованной БД количество отношений, иерархически связанных между собой, может оказаться значительным. Для выполнения запросов потребуется находить соответствующие отношения и связывать данные, чтобы извлечь нужную информацию или обработать ее. При этом более интенсивно, чем в ненормализованной БД, используется центральный процессор, требуется больший объем памяти и большее число операций ввода-вывода.

Если предъявляются особые требования к производительности приложения, то одним из способов увеличения скорости выполнения запросов является денормализация БД — внесение в реляционную схему изменений, при которых уменьшается уровень нормализованности хотя бы одного отношения [1].

Реструктуризация базы данных различными методами денормализации потребует ввода в базу данных или в приложение, работающее с ней, кода, обеспечивающего поддержание целостности данных, которая может быть нарушена благодаря избыточности.

Не существует четких определений различных вариантов денормализации, а также формальных подходов к выполнению реструктуризации базы данных и модификации запросов информационной системы, работающей с БД. Как правило, дается описание денормализации на основе некоторых примеров, и при этом набор различных вариантов денормализации тоже различается [2…4].

Представляется целесообразным исследовать и систематизировать варианты реструктуризации реляционной базы данных с целью повышения производительности информационной системы.

Предлагается классифицировать варианты денормализации по количеству задействованных в реструктуризации отношений и по инкрементации (декрементации) итогового количества отношений. Классификацию можно представить в виде дерева (рис. 1).

Для каждого основного варианта денормализации приведена схема БД, реструктурированная в результате применения конкретного варианта, описаны условия применения, определены преимущества и требования к дополнительной обработке.

Информационную систему, в основе которой лежит нормализованная БД, предлагается рассматривать как кортеж

$$I=<M, Q>,$$

где $M=<R, A, P, B, F, D, V>$ — кортеж, описывающий схему БД [5];

$R$ — множество отношений;

$A$ — множество атрибутов отношений;

$P, B, V$ — множество ключевых, неключевых и внешних ключевых атрибутов отношений, соответственно;

$F$ — множество функциональных зависимостей между атрибутами;

$D$ — множество иерархических связей между отношениями;

$Q = \{Q_i\}, i = \overline{1,n}$ — множество запросов к базе данных;

$Q_i =< q_i, s_{q_i}, T_i >$ — тройка, описывающая $i$-й запрос к базе данных;

$q_i$ — текст $i$-го запроса;

$s_{q_i}$ — количество выполнений $i$-го запроса;

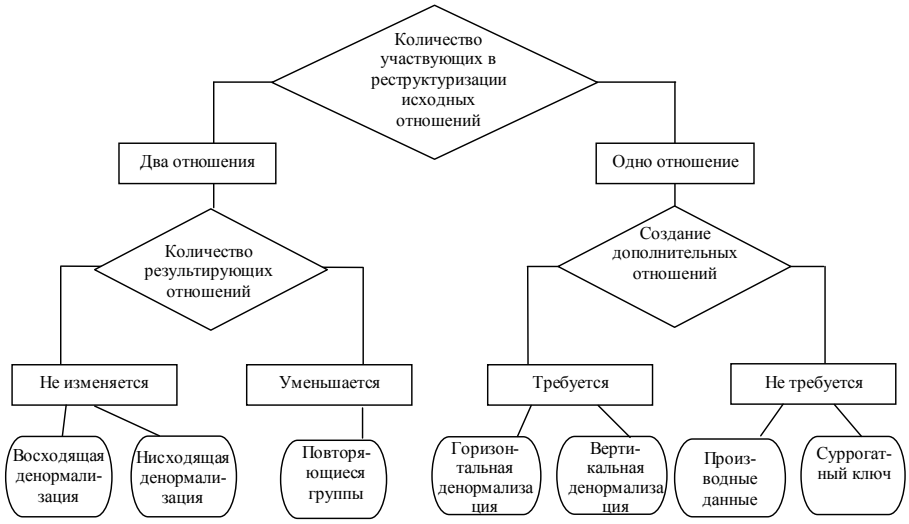


*Рис. 1. Классификация основных вариантов денормализации*

$T_i \in \{Sel, Ins, Del, Upd\}$ — тип $i$-го запроса (выборка, вставка, удаление, обновление, соответственно).

Структуру денормализованной БД можно представить в виде кортежа

$$M' =< R', A', P', B', F', D', V' >,$$

в котором после реструктуризации каждый компонент может быть модифицирован или остаться без изменений. Далее для каждого варианта денормализации представлено описание только модифицированных компонентов кортежа $M'$.

Если реструктуризации подлежат два отношения, находящиеся между собой в иерархической взаимосвязи “один - ко многим”, то в результате количество отношений остается неизменным, и в одно из исходных отношений добавлен дополнительный атрибут (атрибуты), или в результате остается единственное отношение.

*Восходящая денормализация* состоит во включении в родительское отношение дополнительного атрибута, значение которого определяется на основе агрегации данных из дочернего отношения, подчиненное отношение остается неизменным.

Компоненты $M'$ БД после применения восходящей денормализации:

$$R' = R \setminus R_i \cup R_i';$$

$$A'_{R_i'} = A_{R_i} \cup b'_{R_i};$$

$$B'_{R_i} = B_{R_i} \cup b'_{R_i},\ b'_{R_i'} = f(R_{ij}),\ R_{ij} = \sigma_{v_{R_iR_j} = p_{R_i}}(\pi_{R_j}),\ (R_i, R_j) \in D;$$

$$F' = F \cup f',\ f' : p_{R_i} \to b'_{R_i'},$$

где $R_i$ и $R_j$ исходные родительское и дочернее отношения, соответственно;

$R_i'$ — отношение $R_i$ после реструктуризации;

$b_{R_i'}'$ — добавленное в исходное отношение неключевое поле;

$p_{R_i}$ — первичный ключ отношения $R_i$;

$v_{R_iR_j}$ — внешний ключ, определяющий взаимосвязь $(R_i, R_j) \in D$;

$\sigma$ — операции выбора из отношения по указанному условию;

$\pi_{R_j}$ — операция проекции для отношения $R_j$;

$f'$ — функциональная зависимость между первичным ключом $p_{R_i}$ отношения $R_i'$ и добавленным неключевым полем $b_{R_i}'$.

Этот вариант денормализации следует применять, если часто требуется доступ к информации об агрегированном значении для кортежа родительского отношения, при этом агрегированное значение вычисляется на основании кортежей дочернего отношения.

Преимущества данного варианта — исключение операции соединения в запросах и сокращение времени на расчет агрегированного значения.

Дополнительная обработка заключается в пересчете значения введенного атрибута при каждом изменении кортежей дочернего отношения для поддержания актуального агрегированного значения.

*Нисходящая денормализация* состоит во включении в дочернее отношение неключевого атрибута из родительского отношения, родительское отношение остается неизменным.

Этот вариант можно применять, если часто применяются запросы для вывода неключевого атрибута из родительского отношения и атрибутов из дочернего отношения.

Компоненты $M'$ денормализованной схемы базы данных:

$$R' = R \setminus R_j \cup R_j';$$

$$A_{R_j'}' = A_{R_j} \cup b_{R_j'}';$$

$$B_{R_i}' = B_{R_j} \cup b_{R_j'}',\ b_{R_j'}'' \subseteq b_{R_i}'',\ (R_i, R_j) \in D;$$

$$F' = F \cup f',\ f' : v_{R_iR_j} \to b_{R_j'}',$$

где $R_j'$ — отношение $R_j$ после реструктуризации;

$b_{R_j}$ — атрибут, добавленный в отношение $R_j$;

$b_{R_j'}''$ — множество значений атрибута $b_{R_j}$.

Преимуществом рассматриваемого варианта является исключение операции соединения при выполнении запросов.

Однако для поддержания непротиворечивого состояния продублированного неключевого атрибута при каждом изменении в родительском отношении потребуется выполнение дополнительного кода.

При реструктуризации путем введения *повторяющихся групп* два исходных отношений в БД заменяются единственным отношением, т.е. строчноориентированное отношение преобразуется в ориентированное по столбцам (введение повторяющихся групп атрибутов), или, др. словами, создается отдельный атрибут (группа атрибутов) для каждого экземпляра дочернего отношения.

Компоненты $M'$ реструктурированной базы данных:

$$R' = R \setminus (R_i \cup R_j) \cup R'_{ij};$$

$$A''_{R'_{ij}} = A_{R_i} \cup A_{R'_j},\ A_{R'_j} = \bigcup_{k=1}^{K} (A_{R_{jk}} \setminus v_{R_i R_j}),\ K = \max\left(\left|\sigma_{v_{R_i R_j} = p_{R_i}}(R_j)\right|\right);$$

$$B'_{R'_{ij}} = B_{R_i} \cup A_{R'_j};$$

$$D' = D \setminus (R_i, R_j);$$

$$V' = V \setminus V_{R_i R_j},$$

где $R'_{ij}$ — отношение, полученное в результате преобразования исходных отношений $R_i$ и $R_j$, $(R_i, R_j) \in D$;

$K$ — коэффициент повторения группы.

$K$ представляет собой максимальное количество экземпляров дочернего отношения, связанных с одним экземпляром родительского отношения.

Денормализация введением повторяющихся групп может быть применена, если значение $K$ относительно невелико.

Преимуществом такой денормализации является сокращение размера отношения и, следовательно, снижение времени его сканирования.

Дополнительная обработка состоит в том, что потребуется использовать более громоздкие запросы.

При горизонтальной и вертикальной денормализации реструктуризация затрагивает единственное отношение в исходной базе данных, в результате реструктуризации создаются дополнительные отношения, общее количество отношений увеличивается.

При *горизонтальной денормализации* вместо исходного отношения $R_1$ создаются $n$ отношений $R_1, \ldots, R_k, \ldots, R_n$ с одинаковой структурой, кортеж помещают в $R_k$, если для него выполняется условие $C_k$, для каждого кортежа истинным является только одно условие $C_k$.

Компоненты $M'$ БД для горизонтальной денормализации:

$$R' = R \setminus R_l \cup \{R'_{l_k}\},\ k = \overline{1,n},\ \bigcup_{k=1}^{n} R'_{l_k} = R_l,\ R'_{l_k} = \sigma_{C_k}(R_l),\ C_k = f(a);$$

$$A'_{R'_{l_k}} = A_{R_l} \quad \text{для} \quad \forall k;$$

$$D' = D \setminus (D_{R_i R_l} \cup D_{R_l R_j}) \cup D_{R_i R'_{l_k}} \cup D_{R'_{l_k} R_j},$$

где $R_l$ — реструктурируемое отношение;

$a$ — кортеж отношения $R_l$;

$C_k$ — предикат, определяющий, будет ли включен кортеж $a$ в отношение $R'_{l_k}$.

Применять такой вариант следует: для отношений, содержащих много данных; если необходим доступ к кортежам по логическим подмножествам (отдел, филиал организации и т.д.); если существует подмножество большого набора данных, используемое активнее других.

Преимущества: сокращение времени на сканирование отношения и сокращение времени ожидания разблокирования таблицы при одновременном доступе нескольких пользователей.

При этом необходима дополнительная обработка, которая состоит в определении, в какое отношение вносить новый кортеж, и в объединении отношений, если требуется получить общее множество кортежей.

При *вертикальной денормализации* исходное отношение $R_1$ разбивается на $n$ отношений $R_1, \ldots, R_k, \ldots, R_n$; в отношение $R_k$ помещаются ключевые атрибуты отношения $R_1$ и некоторое подмножество неключевых атрибутов этого отношения; подмножества неключевых атрибутов отношений $R_1, \ldots, R_k, \ldots, R_n$ не пересекаются.

Компоненты $M'$ после реструктуризации:

$$R' = R \setminus R_l \cup \{R'_l\}, \mid R'_l \mid \leq \mid R_l \mid;$$
$$A' = A \setminus A_{R_l} \cup \{A_{R'_l}\},\ A_{R'_l} = p_{R_l} \cup B'_{R'_l}, \quad B'_{R'_l} \subseteq B_{R_l},\ \bigcap B'_{R'_l} = \varnothing \ \text{ для } \ \forall R'_l;$$
$$P' = P \setminus P_{R_l} \cup \{P_{R'_l}\},\ P_{R'_l} = P_{R'_l} \quad \text{для } \ \forall R'_l;$$
$$B' = B \setminus B_{R_l} \cup \{B'_{R'_l}\},$$

где $|R_l|$ — мощность отношения $R_l$;

Вариант применяется, если отдельные атрибуты отношения используются значительно реже других атрибутов или отдельные атрибуты отношения заполнены неплотно.

Преимущества применения: сокращение времени сканирования отношения за счет уменьшения длины кортежа и сокращение количества кортежей в отношении.

Дополнительная обработка состоит в соединении результирующих отношений при необходимости получения полного множества атрибутов.

При денормализации с использованием *производных данных* или *суррогатного ключа* рассматривается единственное исходное отношение, в результате реструктуризации количество отношений в БД не изменяется, но к исходному отношению добавляется один или несколько атрибутов.

При использовании *производных данных* в отношение помещается дополнительный атрибут, содержащий данные, которые могут быть вычислены на основе значений других атрибутов этого же кортежа, такие атрибуты называются производными (или вычисляемыми).

Компоненты $M'$ денормализованной БД:

$$R' = R \setminus R_l \cup R'_l;$$
$$A'_{R'_l} = A_{R_l} \cup b'_{R_l}, \qquad b'_{R'_l} = f(a'),\ a' = \pi(a_{R_l})$$
$$B'_{R'_l} = B_{R_l} \cup b'_{R'_l};$$
$$F' = F \cup f', \quad f' : p_{R_l} \to b'_{R'_l};$$

где $b_{R'_l}$ — дополнительный неключевой атрибут отношения $R_l$;

$a'$ — значения подмножества атрибутов кортежа отношения $R_l$;

$f'$ — функциональная зависимость между первичным ключом $R_l$ и производным атрибутом.

Использовать производные данные следует, если требуется выполнять поиск по производному значению; требуется сложное вычисление или в системе существуют запросы, содержащие агрегатные расчеты на основе производного значения.

Преимущества этого варианта денормализации: данные предоставляются немедленно по запросу, без задержки на вычисление; существует возможность создания индекса по вычисленному полю, что приводит к ускорению поиска.

Дополнительная обработка состоит в пересчете производного значения при каждом изменении исходных значений.

При создании *суррогатного ключа* составной первичный ключ, используемый в подчиненных отношениях в качестве внешнего ключа, заменяется на производный ключ из одного атрибута, т.е. в исходное отношение помещается дополнительный атрибут, значение которого определяется на основании значений компонентов составного ключа; этот атрибут становится ключевым.

Компоненты $M'$ для БД после реструктуризации:

$$R' = R \setminus R_l \cup R'_l;$$
$$A'_{R'_l} = A_{R_l} \cup p_{R'_l}, \qquad p_{R'_l} = f(P_{R_l});$$
$$B'_{R'_l} = B_{R_l} \cup P_{R_l};$$

$$F' = F \cup \{f'\}, \quad f' : p_{R_l'} \to b'_{R_l} \text{ для } \forall b'_{R_l} \subseteq B'_{R_l'};$$

$$D' = D \setminus \{(R_l, R_j)\} \cup \{(R_l', R_j)\};$$

$$V'_{R_j} = V_{R_j} \setminus V_{R_l R_j} \cup V_{R_l' R_j}, \ (R_l, R_j) \in D,$$

где $f'$ — функциональная зависимость между вновь созданным первичным ключом и каждым неключевым атрибутом отношения $R_l$ ;

$R_j$ — дочернее отношение для $R_l$ ;

$V_{R_l R_j}$ — внешний ключ отношения $R_j$ , обеспечивающий взаимосвязь ( $R_l$ , $R_j$ ).

Из определения следует, что выполнять такую реструктуризацию имеет смысл, если ключ состоит из нескольких полей, и на него неоднократно выполняются ссылки в других отношениях.

Преимущество варианта — при соединении отношений выполняется меньшее количество операций сравнения, а сам кортеж дочернего отношения имеет меньшую длину.

Дополнительная обработка состоит в вычислении значения нового ключа и в пересчете значения нового ключа при каждом изменении значений компонентов составного ключа.

Время выполнения запросов к исходной БД

$$t = \sum_{i=1}^{n} t_{q_i} s_{q_i} ,$$

где $t_{q_i}$ — время выполнения запроса $q_i$.

Время выполнения запросов к реструктурированной базе данных $t'$ :

$$t' = \sum_{i=1}^{n} t_{q_i'} \ s_{q_i} + \sum t_m s_m,$$

где $q_i'$ — модифицированный запрос к реструктурированной базе данных;

$t_m$ — время выполнения дополнительной обработки для поддержания целостности данных при модификации отношений;

$s_m$ — количество выполнений дополнительного кода за рассматриваемое время.

Решение о применении конкретного варианта реструктуризации определяется соотношением

$$\frac{t'}{t} << 1 .$$

Вычисление значений $t$ и $t'$ предусматривает статистическое исследование конкретной информационной системы.

Предложенная модель реструктуризации позволяет определить направления исследования существующей базы данных для выявления резервов увеличения производительности информационной системы в целом и разработать правила выполнения реструктуризации базы данных для приведения к выбранному варианту модели.

**Литература**

1. Коннолли Т. Базы данных: Проектирование, реализация, сопровождение. Теория и практика / Коннолли Т., Бегг К., Страчан А. — М.: Издат. дом “Вильямс”, 2001. — 1120 с.
2. Интернет и базы данных. Денормализация базы данных: Портал для веб-мастеров и веб-программистов. — http://www.wwwmaster.ru/ article.php?nart=22. — 15.07.2006.
3. Хансен Г. Базы данных: разработка и управление / Хансен Г., Хансен Д. — М.: ЗАО “Изд-во БИНОМ”, 1999. — 704 с.

4. Kris V. T. Using Dynamic View Generation to offset Effects of Performance based Denormalisation / Учеб.-консультац. центр “ФОРС” — http://www.fors.com.ru/eoug97/papers/0039.htm. — 15.07.2006.
5. Кунгурцев А.Б. Анализ целесообразности реструктуризации базы данных методом введения нисходящей денормализации / Кунгурцев А.Б., Зиноватная С.Л. — Тр. Одес. политехн. ун-та. — Одесса, 2006. — Вып. 1(25). — С. 104 — 108.

Поступила в редакцию 10 июля 2006 г.