УДК 519.711.3, 681.51.015, 681.3.01,658.012.011.56:658.512

Ю.А. Шичкина

ДЕКОМПОЗИЦИЯ РЕЛЯЦИОННОГО ОТНОШЕНИЯ МЕТОДОМ ОГРАНИЧЕННОГО ХАОСА

Изучение и устранение аномалий являются самыми важными в аспекте даталогического этапа проектирования базы данных в целом, поскольку позволяют значительно усовершенствовать ранее созданную реляционную модель. К сожалению, этот этап наименее автоматизирован. Ни одна из современных СУБД не проверяет схему базы данных на наличие аномалий. В данной статье предлагается формализованный метод проверки реляционного отношения на принадлежность к нормальной форме Бойса-Кодда.

Ключевые слова: реляционное отношение, функциональная зависимость, нормальная форма Бойса-Кодда, матрица смежности, ориентированный граф.

Нередко проектирование тех или иных моделей баз данных усложняется появлением аномалий (проблем, мешающих нормальной работе базы данных). Изучение и устранение аномалий являются самыми важными в аспекте даталогического этапа проектирования базы данных в целом, поскольку позволяют значительно усовершенствовать ранее созданную модель.

Реляционные модели данных чаще других подвержены аномалиям [1]. Разновидности аномалий, с которыми приходится иметь дело наиболее часто, перечислены ниже.

1. Аномалия избыточности – одинаковые элементы информации повторяются многократно в нескольких кортежах.

2. Аномалия изменения – один и тот же фрагмент данных изменяется в одном кортеже, но остается нетронутым в другом.

3. Аномалия удаления – удаление какого-то фрагмента приводит к потере целого кортежа.

Полностью устранить аномалию избыточности зачастую не представляется возможным, но можно свести ее к минимуму.

Качество проектируемой схемы отношения позволяет улучшить анализ зависимостей одних атрибутов от других. Анализ схемы на наличие аномалий избыточности проводится с помощью функциональных зависимостей. К сожалению, процесс оптимизации схемы отношения на сегодняшний день является единственным неавтоматизированным этапом в проектировании баз данных. Большинство проектировщиков этот этап просто пропускают, другие тратят достаточно много времени на исследование функциональных зависимостей, больше полагаясь на свой опыт и интуицию, чем на математический аппарат.

В то же время, существует инструментарий в области математики, который позволяет не только формализовать этот этап, но и выстроить алгоритм, с одной стороны, обеспечивающий оптимальное решение задачи по минимизации аномалии избыточности, с другой стороны, позволяющий переложить решение этой задачи «на плечи» компьютера. Таким образом, процесс создания базы данных становится автоматизированным на всех этапах проектирования, от построения ER-модели (например, в ERWin) до создания физической базы данных на SQL-сервере [2].

С этой целью к оптимизируемой реляционной схеме можно применить метод ограниченного хаоса, который заключается, в общем случае, в следующем: собрать все объекты в единую группу, проанализировать ее, найти схожие объекты какого-либо вида и выделить их в отдельную подгруппу. В результате получается некоторое число подгрупп. Если при идентификации на принадлежность нового объекта к той или иной группе возникают затруднения, возможно объединение нескольких групп в одну.

Для разделения объектов на группы необходим классификатор, или онтологическая модель.

Удачно построенная онтологическая модель позволяет:

1. Существенно ускорить поиск нужного объекта в группе объектов;

2. Увеличить достоверность информации (в частности, снизить вероятность появления объектов-близнецов);

3. Сделать заполненность групп этой классификации элементом анализа системы и ее текущего состояния.

Основной характерной чертой онтологического анализа является, в частности, разделение реального мира на классы объектов (at its joints) и определение их онтологий или же совокупности фундаментальных свойств, которые определяют их изменения и поведение.

Итак, на первом этапе оптимизации реляционной схемы объединим все атрибуты схемы в одну группу. Для оптимальной, с точки зрения минимума аномалии избыточности, декомпозиции полученной группы исследуем функциональные зависимости между атрибутами. Наиболее подходящим инструментом при этом является аппарат теории графов.

Современные системы управления объединяют широкое разнообразие физических компонент, или элементов. Наиболее полное представление о

системах различной природы можно получить с помощью графов.

Теория графов достаточно хорошо развита, однако прямое ее применение для представления данных до сих пор встречало затруднения, вызванные следующими обстоятельствами:

1) связи в моделях представления данных относительно просты, матрицы смежности получаются разреженными, что снижает ценность их использования;

2) в графах отражается чаще всего один тип связи (например, 1:1);

3) при постановке задачи представления (моделирования) данных, в отличие от теории управления и математики, в которой широко используются начальные предположения, велик объем неформальной составляющей.

В настоящее время разработано достаточно много алгоритмов, позволяющих при работе с разреженными матрицами экономить не только память, но и расчетное время. Поэтому первое затруднение становится, скорее, преимуществом применения теории графов.

Второе затруднение, применительно к теории графов для отображения функциональных зависимостей, которые лежат в основе данных исследований, отсутствует.

Третье затруднение, к сожалению, остается, т. к. при наличии двух одинаковых с математической точки зрения решений они могут нести совсем разный информационный смысл, и выбор в подобных ситуациях будет всегда оставаться за исследователем.

Для отображения функциональных зависимостей и проведения их анализа теория графов представляет наиболее полный инструментарий.

Одной из важнейших характеристик графа, в применении к анализу функциональных зависимостей, является понятие связности.

Если граф не является связным, то множество его вершин можно единственным образом разделить на непересекающиеся подмножества, каждое из которых содержит все связанные между собой вершины и вместе с инцидентными им ребрами образует связный подграф.

Если существует такая вершина, удаление которой превращает связный граф в несвязный, то она называется точкой сочленения. Граф, имеющий хотя бы одну точку сочленения, является разделимым и называется сепарабельным. Он разбивается на блоки, каждый из которых представляет собой максимальный неразделимый подграф.

Таким образом, несвязный подграф представляет собой совокупность отдельных частей (подграфов), называемых компонентами *k(G)*. Каждая такая компонента представляется своей матрицей смежности $A_{ii}\ (i=1,2,\ldots,k)$, а общая матрица смежности A несвязного графа (при соответствующей группировке его вершин и ребер) имеет блочную диагональную (квазидиагональную) форму:

$$A=\begin{pmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & A_{kk} \end{pmatrix}$$

Подобно другим отраслям информатики, в реляционной теории нет универсальных рецептов для проектирования надежной и эффективной в использовании базы данных. Разработчик волен выбирать различные инструменты и методы проектирования. Независимо от того, каким образом создается реляционная модель, зачастую представляется возможным повысить качество проекта, реализуя определенные типы ограничений. Обычно эти ограничения вытекают из анализа предметной области, которая моделируется информационной системой. Одним из средств формализации информации, полученной в результате такого анализа, являются функциональные зависимости между данными.

Функциональные зависимости играют важную роль при нахождении ключей отношения, проверке отношения на принадлежность к той или иной нормальной форме и, как следствие, при проведении декомпозиции отношения как одного из способов устранения аномалий, заключающейся в разбиении реляционной схемы (т.е. множества атрибутов) на две более «мелкие» схемы.

Реляционная схема подвержена минимуму аномалии избыточности, если все ее отношения соответствуют нормальной форме.

Виды нормальных форм (NF) для реляционной схемы разработаны таким образом, что любая схема в нормальной форме гарантированно обладает определенными качественными характеристиками.

Всего в реляционной теории насчитывается 6 нормальных форм: 1NF, 2NF, 3NF, BCNF (нормальная форма Бойса-Кодда), 4NF, 5NF. Каждое из последующих правил (NF) дополняет предыдущее (что, собственно, и позволяет разбить процесс преобразования исходной БД к нормальной форме на этапы и производить его однократно, не возвращаясь к предыдущим этапам).

В результате приведения отношения к форме BCNF схема базы данных обычно находится в виде, наиболее подходящем для физической реализации. Очень часто на практике ограничиваются 3NF.

Следует отметить, что при реализации абстрактной схемы БД в виде реальной базы данных разработчики иногда вынуждены сделать шаг назад – провести денормализацию с целью повышения эффективности, т. к. идеальная с точки зрения теории структура может оказаться слишком накладной на практике.

Сформулируем алгоритм проведения декомпозиции отношения на основе орграфа его функциональных зависимостей.

Алгоритм

1. Провести стягивание подграфов в деревья.
2. Найти все узлы, удовлетворяющие всем трем правилам точки декомпозиции. Допустим, это узлы v_1 и v_2 .
3. Выбрать один из найденных узлов в качестве точки декомпозиции.
4. В первое отношение, которое всегда будет удовлетворять BCNF, записать атрибуты, соответствующие узлам-потомкам точки декомпозиции и самой точке декомпозиции.
5. Из орграфа удалить:

a) все узлы-потомки точки декомпозиции;

b) все дуги, исходящие из точки декомпозиции.

Точка декомпозиции и все дуги, входящие в нее, остаются в орграфе.

6. Во второе отношение, которое может и не удовлетворять *BCNF*, записать все атрибуты, не вошедшие в первое отношение, и атрибуты, соответствующие точке декомпозиции.
7. Повторить алгоритм для второго отношения.

Алгоритм является итерационным и повторяется до тех пор, пока в графе не останется ни одного узла, удовлетворяющего всем трем правилам точки декомпозиции.

В соответствие данному алгоритму можно поставить матричный алгоритм, который позволит оптимизировать процесс декомпозиции.

Две вершины v_i и $v_j \in V$ графа $G = (V, E)$ называются смежными, если они являются граничными вершинами ребра $e_k \in E$. Смежность представляет собой отношение между однородными объектами (вершинами).

Граф можно представить матрицей смежности. Строки и столбцы этой матрицы соответствуют вершинам графа, а ее (i, j)-элемент равен 1, если вершины v_i и v_j являются смежными (или если вершина v_j является концом дуги, соединяющей вершины v_i и v_j в орграфе).

Алгоритм преобразования матрицы смежности.

1. Провести стягивание строк.

a) найти строку a_{ij} , содержащую всего одну единицу;

b) определить столбец t_j , в котором находится эта единица;

c) если в столбце t_j больше нет единиц, то логически прибавить ее к строке или строкам, содержащим атрибут с именем, равным имени столбца t_j ;

d) удалить *i*-ю строку из матрицы *A*. Удалить элемент s_i из матрицы *S*. Повторить эти шаги для каждой строки с единственной единицей.

2. Найти строку, содержащую единицы только в тех столбцах, имена которых отсутствуют среди элементов массива S. Если такой строки нет, перейти к шагу 6.
3. Включить в новое отношение все атрибуты, соответствующие столбцам с единицами в найденной строке, и атрибуты, соответствующие этой строке.
4. Удалить из матрицы столбцы с единицами в найденной строке и саму строку.
5. Повторить действия 2-5 до тех пор, пока в матрице не останется строк, содержащих единицы только в тех столбцах, имена которых отсутствуют среди элементов массива S, или до тех пор, пока в матрице не останется одна строка, полностью состоящая из единиц.
6. Все атрибуты, соответствующие столбцам и строкам оставшейся матрицы, включить в последнее отношение.

Данный алгоритм позволяет провести декомпозицию группы алгоритмов на подгруппы (реляционные отношения) за один проход; автоматизировать процесс проверки реляционного отношения на принадлежность к нормальной форме; свести аномалию избыточности к минимуму.

В основе алгоритма лежат разреженные матрицы, для которых существуют методы компактного хранения данных. Матрицы поддаются распараллеливанию коэффициентом эффективности, следовательно, алгоритм может быть применен к базам данных с большим числом атрибутов.

Литература

1. Шичкина, Ю. А. Автоматизация процесса нормализации реляционных баз данных / Ю. А. Шичкина // Наука. Технологии. Инновации : материалы всерос. науч. конф. молодых ученых. Новосибирск, 2004. – Ч. 1. – С. 77-84.

2. Шичкина, Ю. А**.** Разреженные матрицы в базах данных / Ю. А. Шичкина // Естественные и инженерные науки – развитию регионов : материалы межрег. науч.- техн. конф. – Братск, 2004. С. 53-59.