

ХРАНЕНИЕ ДРЕВОВИДНЫХ СТРУКТУР В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

Романенко Дмитрий, канд. техн. наук, доцент, зав. кафедры информатики и веб-дизайна,

Боровик Алексей, студент группы I-ИСиТ,

Белорусский государственный технологический университет, Беларусь

Сегодня большинство хранилищ данных, как простых так и сложных, построены на основе реляционных баз данных. В них используются либо файл-серверные системы (dBase, Paradox, Clipper, FoxPro, Access), либо SQL-серверы (Oracle, Informix, Sybase, Borland InterBase, MS SQL и т. д.). Реляционные базы данных в большинстве случаев удовлетворяют требования какой-либо предметной области данных, но часты и случаи когда требуется представление и хранение данных в иерархическом виде. Варианты построения таких представлений в реляционной модели и будут рассмотрены в данной статье.

Существует несколько шаблонов проектирования баз данных для хранения и обработки иерархических структур:

- Ссылка на родителя
- Хранение вспомогательных связей
- Составной путь
- Интервальное дерево

Самый простой способ представления дерева в реляционной базе — это помещение матрицы смежности в таблицу. Каждая запись такой таблицы соответствует узлу дерева и хранит его уникальный идентификатор и ссылку на родительский узел.

Имея для каждого узла только идентификатор родителя, возникает необходимость прибегать к рекурсии для обхода всего дерева. Избежать этого нежелательного явления можно, если хранить все связи между конкретным узлом и всеми узлами на пути от него к вершине дерева. В таком случае нам потребуется две таблицы. Первая — будет хранить список узлов дерева. Вторая таблица будет хранить связи между каждым узлом дерева и узлами, находящимся на пути от него к вершине, включая и саму вершину дерева. Связь задается тремя числами: идентификатором узла, от которого начинается путь, идентификатором конечного узла на пути и расстоянием — целым числом, равным количеству дуг между ними. Т.е. расстояние между потомком и родителем равно 1, между потомком и родителем родителя равно 2 и т.д. вплоть до самого корня.

Не обязательно для хранения списка узлов на пути к корню дерева использовать отдельную таблицу. Можно объединить упорядоченный от корня список идентификаторов в строку, используя произвольный символ-разделитель, и хранить ее непосредственно в записи таблицы узлов.

Проблемы, которые возникают при использовании ранее описанных методов: использование рекурсии, дополнительной таблицы или громоздкого

строкового поля были вызваны тем, что язык SQL по своей природе — это язык, предназначенный для работы с множествами, а не с графами.

Для того чтобы представить древовидную структуру в виде вложенных множеств необходимо для каждого узла ввести два целочисленных параметра: левую и правую границу, так чтобы выполнялись следующие условия:

- Правая граница узла больше, либо равна левой;
- Левые границы потомков больше чем левая граница родительского узла;
- Правые границы потомков меньше, либо равны правой границе родительского узла;
- Интервалы, определенные левыми и правыми границами узлов, имеющих одного родителя, не должны пересекаться.

Каждая, из предложенных структур, имеет свои сильные и слабые стороны, которые должны оцениваться применительно к требованиям конкретной задачи. Очевидно, что при небольшом размере дерева различия в производительности при применении того, либо иного метода будут минимальны и несущественны. Интервальное дерево позволяет достичь наилучших результатов при извлечении данных, но одновременно весьма требовательно к вычислительным ресурсам при вставке новой записи или перемещении узлов. При большом уровне вложенности, применение второй и третьей методик приведет к значительному росту базы данных из-за необходимости хранения информации обо всех связях между узлами на пути к вершине. При выборе способа хранения древовидных структур в реляционных базах данных прежде всего необходимо определить какой из критериев является более критичным для проектируемой системы: скорость извлечения информации или размер, занимаемый данными на диске. Следует помнить, что нет «плохих» или «хороших» методов: оценки и выбор необходимо производить, исходя из условий конкретной задачи.

Список использованной литературы

1. Дейт К. Руководство по реляционной СУБД DB2 // Дейт К. - М.: Финансы и статистика, 1998. - 320 с.
2. Мейер М. Теория реляционных баз данных // Мейер М. -М.: Мир, 2007. - 608 с.