

РАЗРАБОТКА УНИВЕРСАЛЬНОГО МЕХАНИЗМА РАБОТЫ С ИЕРАРХИЧЕСКИМИ ДАННЫМИ

Петухов А.С.

*Тюменский государственный университет, лаборатория автоматизации медицинских
учреждений*

Тюмень, Россия

При создании информационных систем самой различной направленности всё большую популярность получают иерархические структуры хранения данных. Рост их популярности обусловлен, прежде всего, наглядностью организации данных и возможностью выделения из иерархической совокупности определённого множества, соответствующего тому или иному уровню вложенности. Однако, наряду с очевидными удобствами иерархических структур у них имеется и ряд недостатков, вызванных сложностью работы с ними в реляционной базе данных.

Лабораторией автоматизации медицинских учреждений Тюменского государственного университета в течение нескольких лет разрабатывается медицинская информационная система для Тюменского кардиологического центра. В этой системе иерархическую организацию имеют электронная история болезни пациентов и данные в некоторых справочных таблицах, таких как варианты оплаты пациентом услуги госпитализации, виды высокотехнологической медицинской помощи, виды амбулаторных медицинских услуг и некоторые другие. Данные в этих справочных таблицах должны редактироваться пользователем информационной системы. Следовательно, программный комплекс должен быть снабжён универсальным механизмом работы с любыми иерархическими данными, содержащимися в какой-либо таблице базы данных.

Поставленная задача осложняется тем, что данные, содержащиеся в справочной таблице, делятся на группы по различным признакам. Данные в каждой группе организованы в виде дерева. Так, например, данные в справочной таблице «виды высокотехнологических методов» делятся на высокотехнологические методы (ВТМ), оказываемые при финансировании госпитализации пациента из ОМС; ВТМ, оказываемые при финансировании госпитализации пациента из Федерального бюджета и соответственно, ВТМ, выполняемые при платной госпитализации пациента. В общем случае число групп, на которые делятся данные в иерархическом справочнике, может быть произвольным.

Для выбора нужной группы и её подгрупп в совокупности иерархических данных был создан класс, позволяющий динамически фильтровать произвольную таблицу базы данных по любому критерию. Для хранения критериев фильтрации и выполнения операций над ними

был создан вспомогательный класс, организованный в виде массива, содержащего параметры фильтрации и их значения. При фильтрации данных допускается использование SQL-функций. Таким образом, механизм работы с иерархическими данными обращается не напрямую к таблице базы данных, а взаимодействует с описанным классом динамической фильтрации данных.

Однако, при создании универсального механизма работы с иерархическими данными, хранимыми в таблице реляционной базы данных, возникает проблема доступа к полям этой таблицы. Очевидно, что, например, ключевое поле в разных таблицах может быть описано под разными именами. Аналогичная ситуация возникает с полем, хранящим код родителя записи и с полем для выбора данных. Наряду с этими полями в справочнике, безусловно, могут присутствовать поля для хранения идентификаторов групп. В связи с этим для хранения информации о полях таблицы был создан класс, содержащий массив полей и информацию об их роли в организации иерархии (ключевое поле, поле указателя на родителя, справочное поле). В классе-массиве полей должна присутствовать информация и о полях, хранящих идентификаторы групп совместно с теми значениями этих групп, по которым была отфильтрована таблица. Значения групп необходимы для автоматического добавления записей в эту таблицу механизмом редактирования иерархии, т. к. при добавлении новой записи таблицы, содержащей иерархические данные, необходимо описать к какой группе относятся добавляемые данные.

Между тем, иерархические данные могут быть организованы не только на базе реляционной таблицы, но и в любом другом формате, например, в XML-строке. Для работы с иерархическими данными, представленными в разных форматах, необходимо разделение визуальной части создаваемого механизма и части, выполняющей операции над данными. Следовательно, при замене части, включающей в себя операции над данными, появляется возможность работы с иерархическими данными, представленными в принципиально другой форме. Следует заметить, что подобная замена происходит прозрачно для пользователя, поскольку визуальная часть, с которой он работает, не претерпевает никаких изменений.

Независимо от формата организации иерархических данных, механизм работы с ними выполняет добавление нового узла в иерархию, редактирование данных, содержащихся на любом уровне иерархии и удаление узла и всех его потомков.

Универсальность созданного механизма и единая визуальная форма работы с иерархическими данными, представленными в различных форматах, значительно облегчает обучение пользователей и качественно повышает гибкость программного комплекса.