

УДК 004.652.5

**А.И. РЯЗАНЦЕВ, В.В. НОВИКОВ, Е.А. ЗУБЕНКО**

*Северодонецкий технологический институт,  
Восточноукраинский национальный университет имени Владимира Даля, Украина*

## **ПРОБЛЕМЫ СОПРЯЖЕНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МОДЕЛЕЙ ДАНЫХ С РЕЛЯЦИОННЫМИ БАЗАМИ ДАНЫХ В ЗАЩИЩЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ**

В статье проанализированы различные механизмы сопряжения объектно-ориентированной модели системы и реляционного хранилища данных. На примере фрагмента системы безопасности рассмотрены подходы к моделированию объектно-ориентированных отношений средствами реляционных баз данных (РБД). Предложено решение в виде разработки каркаса приложений, обеспечивающего прозрачное для разработчика отображение объектно-ориентированной модели информационной системы на реляционную модель СУБД.

**реляционная база данных, объектно-ориентированная модель, система управления базами данных, object-relational mapping, информационная система**

### **Введение**

При проектировании трехзвенных информационных систем разработчик решает задачу взаимодействия объектной модели системы с реляционным хранилищем данных.

При этом возникает целый ряд проблем, вызванных следующим:

- недостаточной приспособленностью реляционных хранилищ данных к моделированию таких концепций объектно-ориентированного программирования (ООП), как наследование, полиморфизм, реализация ссылочных связей;

- слабой поддержкой программными средствами сопряжения таких функций, как динамическая упреждающая выборка, отгрузка дерева объектов сервера приложений, кэширование дерева объектов сервера приложений;

- отсутствием программных средств для организации отложенной записи, транзакций и блокировок одновременного доступа к объектной модели и сопряжение их с транзакциями реляционной системы управления базами данных;

- необходимостью написания большого объема однотипного кода загрузки и сохранения данных в реляционных базах данных.

Для решения отмеченных проблем разработан ряд стандартных подходов:

1. Реализация или использование одной из существующих объектно-ориентированных СУБД (ООСУБД).

Среди преимуществ данного подхода можно отметить:

- упрощение организации транзакций над объектной моделью ИС (в зависимости от возможностей конкретной ООСУБД);
- поддержка наследования и полиморфизма;
- меньший объем кода сопряжения.

Существенным недостатком названного подхода является тот факт, что реализация собственной объектно-ориентированной СУБД потребует несомерно огромных усилий.

2. Использование различных средств ORM.

Преимущества и недостатки данного подхода рассмотрены на примере типизированных наборов данных (typed datasets) ADO.NET [1].

Преимущества:

- автоматическая генерация кода классов типизированного набора данных по схеме БД;
- автоматическая поддержка загрузки и сохранения изменений в БД средствами ADO.NET;
- таблицы, строки представлены как классы с возможностью расширения функциональности;
- реализована возможность простейших запросов с условием отбора и навигация по отношениям между таблицами.

Недостатки:

- объектная модель ИС подгоняется под реляционную модель БД, так как типизированный набор данных является упрощенной автономной копией реальной БД;
- отсутствует поддержка динамической загрузки/отгрузки, все данные полностью кэшируются;
- отсутствует поддержка наследования и полиморфизма;
- реализация многопользовательского доступа, блокировок и транзакций сопряжена со значительными трудностями.

ADO.NET Entity Framework – новейшая ORM-технология от Microsoft (находится на стадии beta-тестирования) [2].

Функциональность схожа с типизированными наборами данных, однако есть ряд существенных преимуществ:

- поддержка наследования;
- расширенная поддержка запросов к модели (Entity SQL);
- улучшенная навигация по отношениям между таблицами РБД;
- более гибкие операции обновления и удаления и вставки данных в модель.

Недостатки те же, что и у типизированных наборов данных, за исключением поддержки наследования.

Кроме того, следует отметить тот факт, что все средства ORM (object-relational mapping) требуют ручной разработки схемы реляционной базы данных.

### 3. Ручная работа.

Недостаток – большой объем работы, множество рутинных операций и повторяющегося кода.

**Постановка задачи.** На основании анализа существующих механизмов взаимодействия объектно-ориентированной модели с РБД сделан вывод о том, что традиционные средства взаимодействия объектной модели системы с РБД обеспечивают решение ограниченного числа задач. В связи с этим, задача разработки адекватной реляционной базы данных и средств сопряжения на сегодняшний день, безусловно, является актуальной.

Таким образом, задачей исследования является концептуальное моделирование объектно-ориентированных отношений средствами РБД и разработка подходов к реализации средств сопряжения, в частности каркаса приложений (framework), обеспечивающего прозрачное для разработчика отображение объектно-ориентированной модели информационной системы на реляционную модель СУБД и генерацию БД и кода сопряжения в соответствии с объектной моделью системы.

## Моделирование ОО отношений средствами РБД

В качестве примера рассмотрен фрагмент системы безопасности информационной системы. Имеется следующий набор объектов (рис. 1).

1. Объекты пользователей (дескриптор безопасности (SID), имя, выводимое имя и набор групп, в которые он входит).
2. Объекты групп (дескриптор безопасности (SID), имя и набор входящих в нее дочерних пользователей).

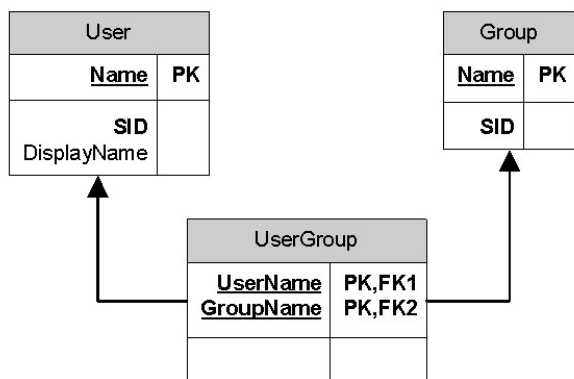


Рис. 1. БД пользователей и групп безопасности. Модель 1

При необходимости включения в существующие группы кроме пользователей других групп, в модель 1 достаточно добавить таблицу-связку, и реляционная модель останется адекватной требованиям задачи. Однако при необходимости работы со смешанными наборами пользователей и групп, например, требовании запрашивать имена и дескрипторы безопасности всех пользователей и групп или пользователей и групп, входящих в какую-либо группу, модель 1 перестает обеспечивать полиморфное поведение рассматриваемых объектов. Задача усложняется, если тип объекта заранее неизвестен, например, необходимо добавить объект с указанным дескриптором SID в какую-либо группу.

Данный набор задач, как правило, решается в частности. Например, выборку имён и дескрипторов всех групп и пользователей можно осуществить при помощи объединяющего (union) запроса. Вставка объекта с указанным SID осуществляется следующим образом.

1. Выполняются запросы к таблицам User и Group для определения типа и имени объекта.
2. В зависимости от типа объекта осуществляется вставка либо в таблицу UserGroup, либо в GroupGroup.

В целом возникает необходимость выделить базовый класс для сущностей User и Group, содержащий общие свойства для пользователей и групп – такие как имя и SID – назовём его SecurityObject, а

сущности User и Group будут производными классами.

На рис. 2 показан пример представления исследуемой иерархии при помощи реляционных таблиц.

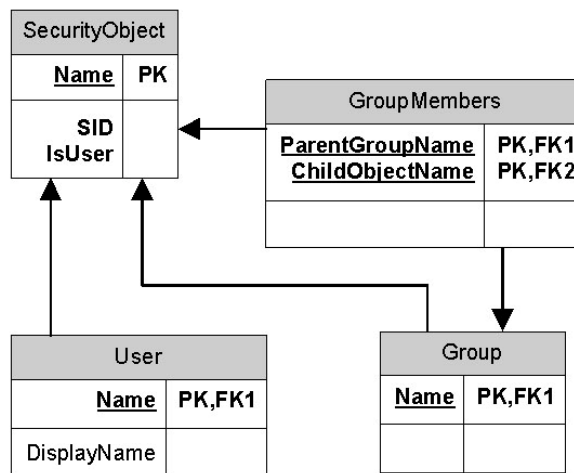


Рис.2. БД пользователей и групп безопасности. Модель 2

Исходя из приведенной выше схемы можно определить основной подход к моделированию иерархии классов, средствами РБД.

1. Для каждого класса объектов вводится своя таблица в БД.
2. Таблицы производных классов связываются с таблицами базовых классов связью один-к-одному по первичному ключу базового класса.
3. В таблице базового класса вводится признак, идентифицирующий тип объекта.

Для ссылки на объект какого-либо класса используется связь один-ко-многим по первичному ключу таблицы класса.

В общем случае первичного ключа может и не быть или он может быть достаточно объёмным, поэтому для унификации данного подхода можно ввести понятие идентификатора объекта и выполнять связь по нему.

Для приведения в соответствие данной модели с моделью объектов Microsoft .NET вводится понятие базового класса Object, который будет содержать в себе 2 поля (идентификатор объекта и идентификатор типа объекта), а также дополнительный класс

объектов-коллекций. Объект коллекции моделируется двумя таблицами – таблицей объекта (в которой содержится идентификатор объекта и дополнительные обще-объектные данные, например, количество элементов) и таблицей элементов.

С учетом приведенных положений схема данных для исследуемого примера объектной модели принимает вид, представленный на рис. 3.

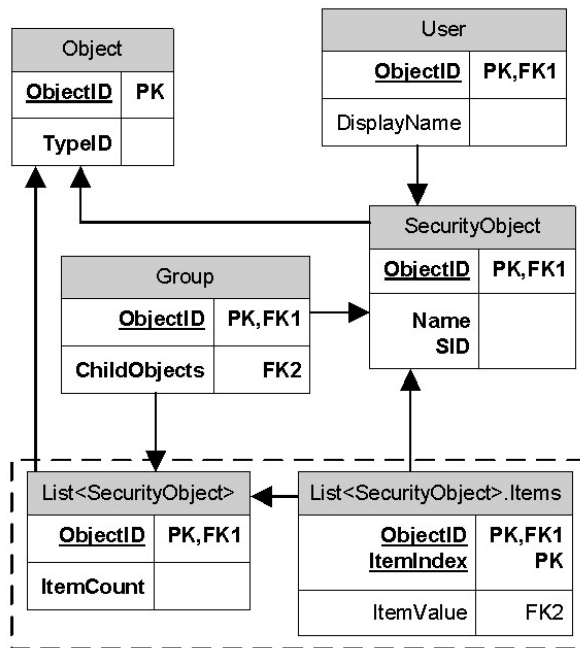


Рис. 3. БД пользователей и групп безопасности. Модель 3

### Подходы к абстрагированию от конкретной СУБД

Для решения задачи абстрагирования от конкретной СУБД необходимо ввести в разрабатываемый каркас приложений абстрактный компонент «Источник хранилища».

В состав интерфейса компонента входят:

1. Функции манипулирования метаданными:

- чтение информации о типах и их структуре;
- создание новых типов (в том числе производных классов и коллекций) и удаление существующих; при выполнении данной операции будут автоматически создаваться таблицы и отношения в ниже лежащей РБД;

- модификация структуры существующих типов (добавление и удаление полей). При выполнении данной операции будут автоматически модифицироваться структура таблиц и отношений в ниже лежащей РБД.

2. Функции манипулирования объектным деревом (на уровне структур объектов):

- создание объектов в хранилище;
- регистрация объектов и получение зарегистрированных объектов. В общем случае объектное дерево может быть довольно сложным и разветвленным, но приложение обычно начинает работу не со всем объектным деревом, а с конкретного объекта – корня дерева. При регистрации объекта ему присваивается строковое имя, по которому можно в будущем получить этот объект;

- запросы на чтение данных объектов. Включают в себя чтение значений отдельных полей конкретных объектов, чтение группы полей набора объектов, принадлежащих одному классу, чтение отдельных элементов и набора элементов объектов-коллекций;

- запросы на запись значений полей объектов и элементов объектов-коллекций, вставка и удаление элементов объектов-коллекций;

- пакетная запись изменений в объектном дереве.

3. Функции поиска и удаления недостижимых объектов (т. н. «сборка мусора») [3]. Объект считается достижимым, если к нему можно прийти, двигаясь по графу объектного дерева, начиная с корней. Передвижение по графу может осуществляться через поля ссылочного типа или через элементы объектов-коллекций ссылочного типа.

Реализация данного компонента выполняется отдельно под каждую конкретную систему управления базами данных.

## Выводы

На основании проведенного анализа и разработки основных подходов к моделированию и к абстрагированию от конкретной СУБД сформулированы основные требования к разрабатываемому каркасу приложений, обеспечивающему прозрачное отображение ОО модели информационной системы на реляционную модель СУБД.

В состав запланированных возможностей разрабатываемого каркаса приложений необходимо включить следующие:

- возможность пользователя создавать собственные объекты на .NET-совместимом языке, исполняемые в среде CLR, значения полей которых будут сохраняться в БД. Данный механизм должен быть как можно менее навязчивым для разработчика;
- загрузка дерева объектов не целиком, а по мере использования. То же самое относится и к выгрузке объектов. Данная функциональность должна быть настраиваемой;
- реализация гибкой системы кэширования наиболее часто используемых объектов;
- возможность подключения модулей, осуществляющих фоновую упреждающую загрузку и отгрузку дерева объектов;
- сборка мусора в хранилище;
- поддержка транзакций над объектной моделью (синхронизация одновременного доступа и отложенная запись состояния, восстановление после сбоя);

- удобочитаемая структура нижележащей РБД для работы в автономном режиме;
- поддержка коллекций объектов. Массовая загрузка элементов. Запросы к коллекции (аналог SQL-запросов к таблицам) [4];
- поддержка подключения слоя безопасности инфраструктурой framework'a;
- модульность и расширяемость компонентов Framework'a и абстракция от конкретной СУБД.

## Литература

1. Малик С. Microsoft ADO.NET 2.0 для профессионалов: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2006. – 560 с.
2. José Blakeley, David Campbell, Jim Gray, S. Muralidhar, Anil Nori. Next-Generation Data Access: Making the Conceptual Level Real: Visual Studio. – 2005. – Technical Articles, June 2006.
3. Рихтер Дж. Программирование на платформе Microsoft .NET Framework: Пер. с англ. – 2-е изд., испр. – М.: Издательско-торговый дом «Русская редакция», 2003. – 512 с.
4. Don Box, Anders Hejlsberg. The LINQ Project: .NET Development Technical Articles. – September 2005. – 512 p.

*Поступила в редакцию 22.02.2007*

**Рецензент:** д-р техн. наук, проф. И.А. Фурман, Харьковский национальный технический университет сельского хозяйства им. П. Василенко, Харьков.