

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 330, 331

А.А. Зарубин

РАЗРАБОТКА МОДЕЛИ ДАННЫХ ВИРТУАЛЬНОГО МУЗЕЯ УГЛЯ НА ОСНОВЕ CASE-ТЕХНОЛОГИЙ

Ускорившиеся темпы создания и накопления информации, усложнение и глобализация знаний сделали необходимым поиск инструментов, позволяющих обеспечить быстрый и эффективный доступ к различным знаниям, рассеянным по разным источникам. Одним из таких инструментов по праву считаются технологии электронных коллекций, развитие которых началось в 1990-х годах. Электронные коллекции представляют собой перспективные формы сложных распределенных информационных систем и рассматриваются как основа создания глобального распределенного хранилища знаний. Электронная коллекция – это распределенная информационная система, позволяющая надежно сохранять и эффективно использовать разнородные массивы объектов (текст, графика, аудио, видео и др.), доступные в удобном для пользователя виде через глобальные сети передачи данных [1]. Наиболее близка идея электронных коллекций к традиционным музеям – та же идея об интеграции различных типов информации, та же четко выделенная иерархия отделов и экземпляров, а также возможность одновременно принять достаточно большое количество посетителей [2].

Поэтому неудивительно, что сегодня в Интернете существует множество проектов электронных коллекций, в частности, электронных музеев. Примером могут служить элек музей Нобеля (www.nobel.se,

JavaScript [4]), Лувра (www.louvre.fr/louvrea.htm, HTML [5]). Не отстают от западных коллег и российские разработчики электронных музеев, например, Государственный Эрмитаж располагает собственным электронным музеем (www.hermitage.ru, CGI) [6]. На начало 2002 года в Интернете находятся порядка 400 ссылок на электронные музеи (по данным поисковой системы Яндекс). Одной из разновидностей виртуальных музеев являются геоинформационные музеи, предоставляющие данные о полезных ископаемых, флоре, и пр. По данным того же источника на них насчитывается порядка 150 ссылок. Это показывает достаточно широкую распространенность таких коллекций среди научных центров России.

Технология электронной коллекции основывается на объединении распределенных баз данных различных форматов. Доступ к данным может быть получен через язык запросов (SQL) или при помощи программирования с использованием различных языков программирования. Базы данных (БД) в электронных музеях играют главную, объединяющую роль, связывая их различные части. Обычно вся предоставляемая информация хранится в базах данных, и зачастую таблицы БД содержат множество взаимосвязей, индексов, выполняемых процедур и триггеров. Все это делает модель данных очень сложным объектом, и малейшее изменение в какой-либо из таб-

лиц может потребовать значительных усилий, в том числе и по перестройке программного кода, отвечающего за работу с БД. Вследствие этого поддержка подобных проектов требует высоких временных и ценовых затрат. CASE-средства позволяют свести издержки связанные с разработкой и поддержкой программного обеспечения к минимуму. Одним из поступатов CASE-средств является возможность в простой и наглядной форме смоделировать предметную область информационной системы, прежде чем начинать ее программную разработку, т.е. построить так называемые визуальные модели. На дальнейших этапах сложность разрабатываемой системы может возрастать, однако построенные модели информационных систем позволяют обеспечивать ясность представления выбранных архитектурных решений и понять разрабатываемую систему во всей ее полноте. Таким образом, уже на ранних этапах проектирования снижается риск возникновения ошибок, которые могут привести в дальнейшем к пересмотру все модели и, следовательно, к большим временным задержкам. CASE-средства позволяют сгенерировать модель данных, на которой отображены все объекты информационной схемы: таблицы, связи, триггеры, а также связать модель БД с объектной моделью и облегчить процесс создания программного кода.

Несмотря на явные преимущества CASE-моделирова-

ния, сегодня существует ничтожно малое количество электронных коллекций, созданных при помощи средств CASE. Это говорит о необходимости построения четкой формализованной модели, позволяющей создать электронный музей любого направления и с любой структурой за минимальное время и при минимальных изменениях модели и обладающей следующими свойствами:

- легкость и простота модификации;
- соответствие современным требованиям к оформлению интерфейса онлайнового музея;
- платформонезависимость;
- независимость от языка программирования и СУБД;
- небольшие материальные затраты на разработку и использование;
- максимальная автоматизация процесса разработки и поддержки музея.

На данный момент на рынке существует множество CASE-средств, позволяющие моделировать различные аспекты работы приложения. Одни служат для создания функциональной модели, т.е. описания функциональных возможностей программы (BPWin фирмы PLATINUM technology), другие позволяют строить объектные модели, отражающие предметную область (Rational Rose компании Rational Software), и, наконец, при помощи последних возможно создание модели баз данных (ERwin). Отметим, что функциональный и объектный подходы к разработке приложений являются в каком-то роде конкурентными, в то время как моделирование БД необходимо в обоих случаях. Во многих случаях в среде CASE присутствует поддержка сразу нескольких типов моделей. Примером такого средства может послужить продукт Rational Rose 2002 Enterprise – лидер на рынке продуктов для объектно-ориентированного анализа, мо-

делирования, проектирования и инструментальных средств разработки [7].

Начиная с версии 2000, в Rational Rose включены средства для разработки моделей баз данных - DataModeler. Ранее для создания модели БД ее либо приходилось проектировать в специализированном CASE-средстве (например, в ERWin), либо создавать объектную модель, а затем экспорттировать ее в тот же ERWin для доработки и создания БД. Это приводило к потере времени и средств. Теперь же Rational Rose Data Modeler позволяет использовать один продукт и один язык для написания проекта полностью, стирая грани между объектной моделью и моделью данных. В дополнение к преимуществам от использования UML для моделирования баз данных, Rational Rose Data Modeler дает следующие возможности [7]:

- **Соответствие между объектно-ориентированной структурой и моделью данных (object-relational mapping)** - позволяет отслеживать преобразования модели объекта в мо-

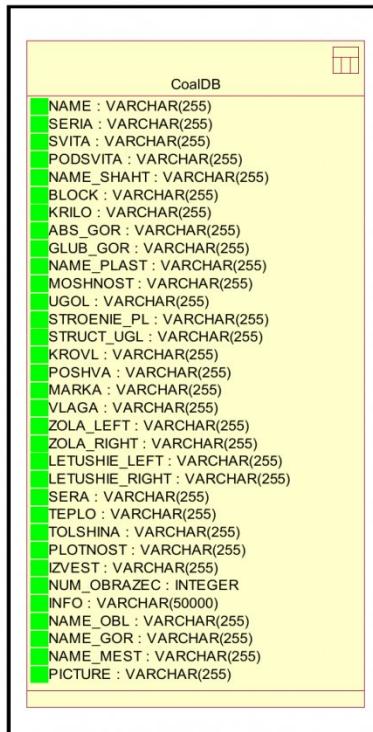


Рис. 1. Структура импортированной базы данных

дель базы данных. Эта форма соответствия позволяет проанализировать ссылки между приложением и базой данных и обновлять их на основе изменений, сделанных в процессе разработки.

- **Прямое и обратное проектирование базы данных (database round trip engineering).** Прямое и обратное проектирование модели данных и базы данных позволяет создать модель данных, основанную на структуре БД через прямое или обратное проектирование. Схема может быть сгенерирована прямо напротив базы данных или сохранена как скрипт для дальнейшего применения. Она будет включать таблицы, колонки, ограничения, индексы, триггеры и т.д.

- **Ссылочная целостность (referential integrity)** - сохраняет целостность базы данных благодаря автоматическому переносу первичных

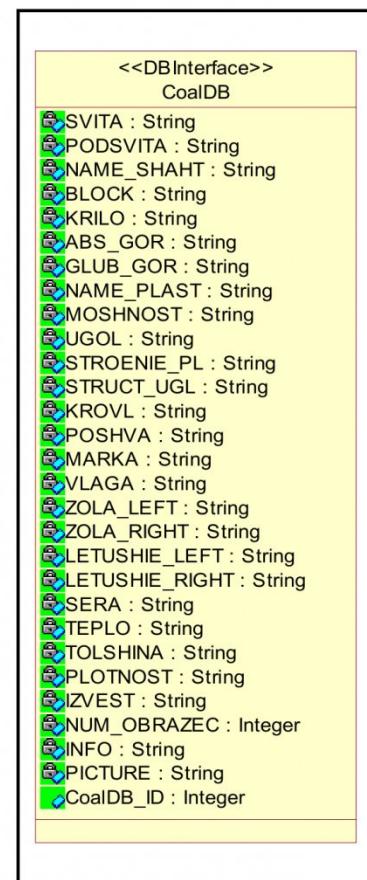


Рис. 2. Объект, импортированный из схемы

ключей главной таблицы (parent table) как внешних в зависимую таблицу (child table). Когда ссылка создана, пользователи могут выбрать, как обеспечить ссылочную целостность: либо через триггер, либо через декларативную целостность.

Применение этих и других возможностей приводит к уменьшению временных и цено-вых издержек процесса разработки модели и позволяет в некоторой степени его автоматизировать.

Создание модели данных было разделено на несколько этапов.

На первом этапе был произведен импорт структуры существующих баз данных для создания предварительной структуры модели. Сначала существующие БД разных форматов, содержащие качественные характеристики экземпляров раз-

личных типов, были объединены в одной схеме СУБД Oracle.

После создания в Rational Rose новой схемы данных с названием Museum для восстановления структуры существующих баз данных была использована возможность обратного моделирования (reverse engineer). Для импорта потребовалось лишь указать данные для соединения с БД и название схемы, все остальное было про-делано автоматически. В ре-зультате был создан скелет будущей модели данных. На рис. 1. приведена структура одной из баз.

Отметим, что поля со структурой VARCHAR(50000) – это двоичные поля BLOB. Полученная структура модели сильно напоминает схемы СУБД Oracle [3]. На диаграмме в по-нятной, визуальной форме ото-бражаются таблицы, связи меж-

ду ними, индексы, ключи. Видно, однако, что существующие таблицы необходимо было су-щественно доработать: добавить ключи, определить индексы, привести к нормализованной форме схему, связать таблицы между собой. Но в первую оче-редь нужно было объединить полученную модель данных с построенной объектной моде-лью музея.

Для этого на втором этапе была использована возможность автоматической генерации объ-ектной модели на основе физи-ческой модели данных (trans-form to object model). При гене-рации Data Modeler автоматиче-ски сопоставил каждой таблице объект. Все вновь созданные объек-ты были поме-щены в но-вый пакет классов с названием Databases. На рис. 2. изображен объек-т, соответствующий БД экземпляров углей.

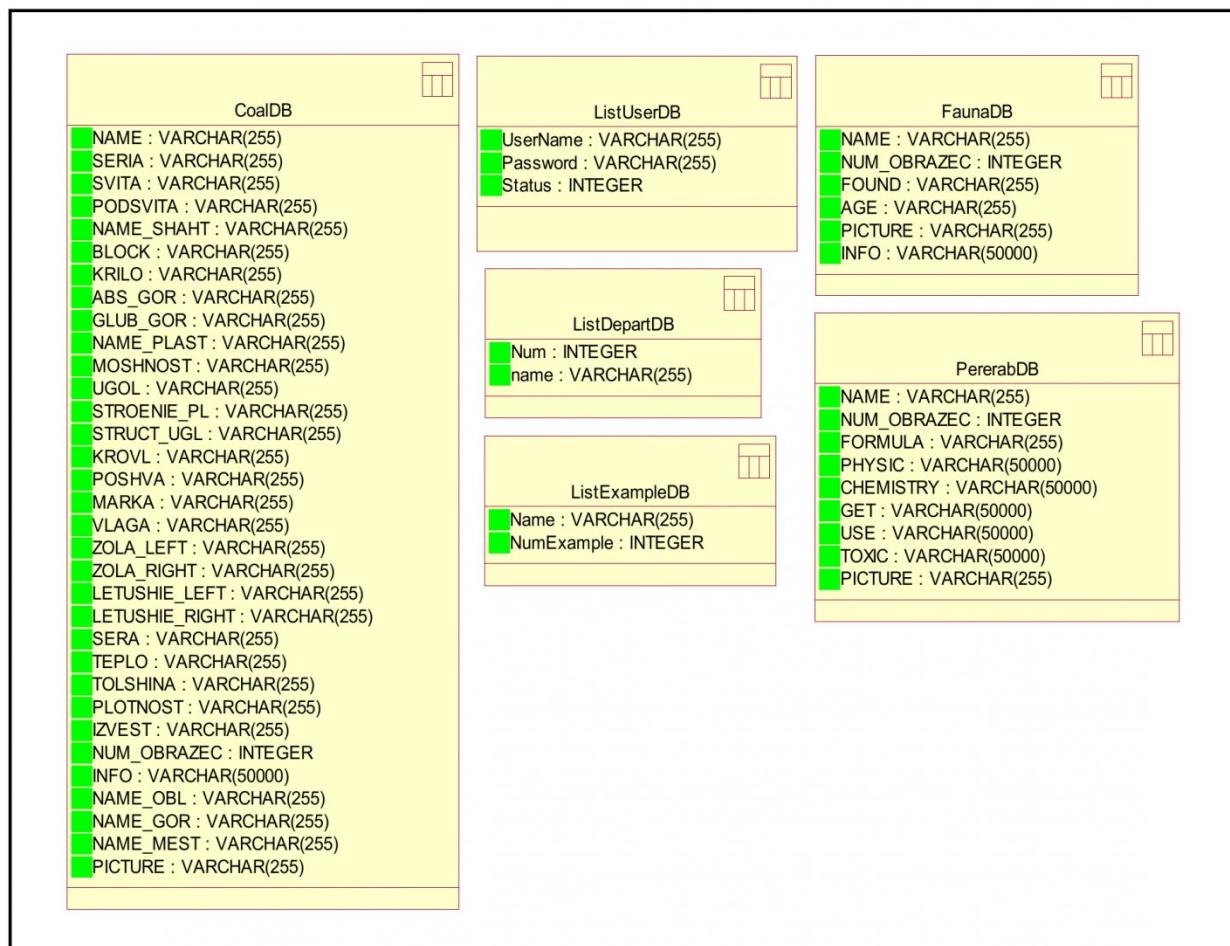


Рис. 3. Схема с добавленными базами

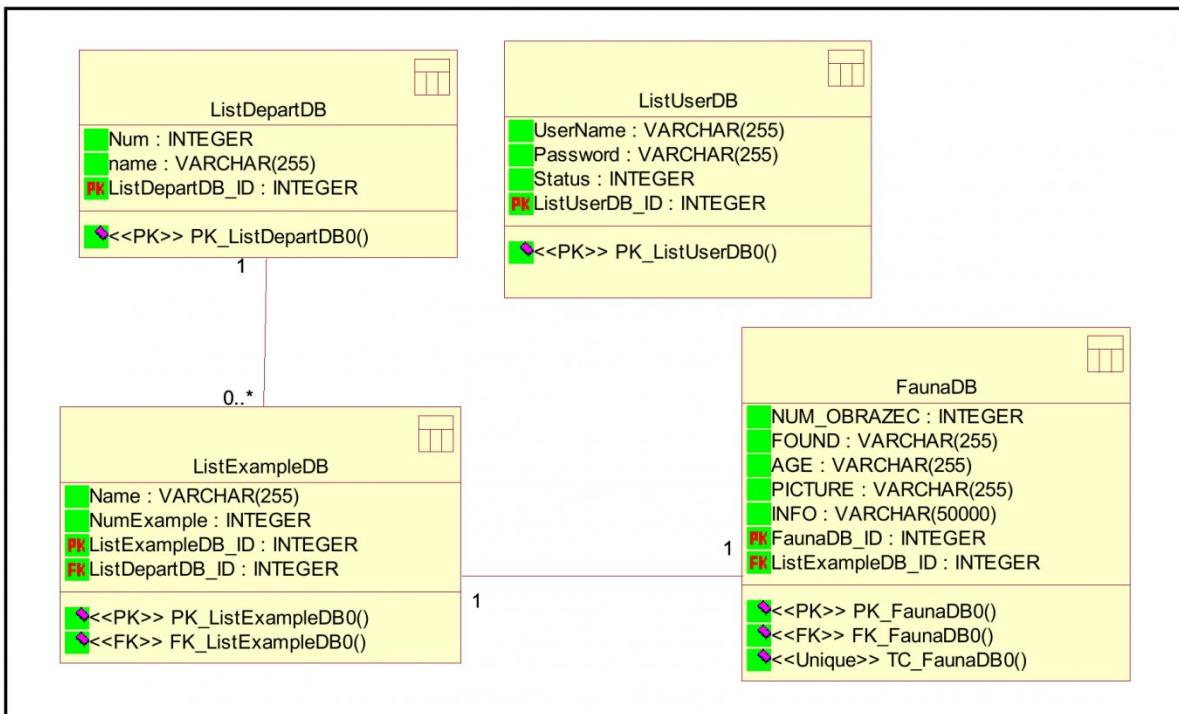


Рис. 4. Связи между таблицами

После этого стало возможным привести модель данных в соответствие с логикой музея. Для этого сначала на диаграмму пакета были добавлены несколько новых классов: ListUserDB, соответствующий базе данных с паролями и правами посетителей; ListDepartDB, отражающий БД отделов музея и ListExampleDB, служащий для объединения всех экземпляров различных отделов музея. После определения их структуры, схему данных была обновлена с использованием возможности трансформации объектной модели в модель БД. Результат изображен на рис.3.

В любой схеме БД важнейшую роль играют связи между таблицами. Они отражают логику работы приложения. Сначала для всех таблиц были созданы первичные ключи, после чего на схеме были отображены связи между ними, при этом в каждой из подчиненных таблиц автоматически был сгенерирован внешний ключ. Это служит для обеспечения целостности базы данных. Для каждой из связей задается кардинальность и роли таблиц, что также отражается на модели. Результат

продемонстрирован на рис. 4.

Как можно видеть, между базами данных ListDepartDB и ListExampleDB существует связь «один к многим», т.е. каждому отделу может соответствовать множество экземпляров, а в свою очередь, с таблицей экземпляров связаны отношениям «один к одному» три таблицы, содержащие их качественные характеристики. Таким образом разрозненные базы данных были объединены через базу ListExampleDB, содержащую только название, номер экземпляра, первичный ключ и внешний ключ для связи с БД ListDepartDB. Подобная схема позволяет вводить новые отделы в схему музея с небольшими временными затратами.

Далее модель была приведена к нормальной форме, т.е. из схемы были удалены многозначные зависимости. Для этого из таблицы CoalDB были выделены в отдельные таблицы поля Name_Obl, Name_Gor, Name_Mest, Seria, Svita, Podsvita, Name_Shaht, Block, Krilo, Name_Plast, Stroenie_Pl, Struct_Ugl, Krovl, Poschva, Marka, Izvest. Полностью привести схему данных не пред-

ставляется возможным ввиду ее громоздкости, продемонстрируем принцип на рис.5. на примере поля Name_Obl.

Таблица Name_OblDB, как видно, содержит все названия областей и связана с таблицей CoalDB отношением «один к одному» через внешний ключ Name_Obl. Таким образом из БД выделяются и остальные вышеуказанные поля. После окончания процесса нормализации модель данных опять была приведена в соответствие с объектной моделью, при этом на основе указанных ключей и связей автоматически сформировались отношения между классами и новые атрибуты.

И, наконец, на третьем этапе разработки модели был сгенерирован DDL-скрипт, который содержит SQL-инструкции по созданию структуры баз данных и может быть выполнен в любой SQL-ориентированной СУБД. Ниже приведен фрагмент скрипта, генерирующий таблицу Name_OblDB.

```

CREATE TABLE Name_OblDB
(Name_Obl_ID INTEGER NOT NULL,
NAME VARCHAR ( 255 ) NOT NULL,
  
```

*CONSTRAINT K_Name_OblDB0
PRIMARY KEY (Name_Obl_ID);*

Полученная в итоге схема данных сильно отличается от исходных источников данных,

связи и таблицы, в том числе зависимые таблицы-справочники (*Name_GorDB* и т.п.).

В результате была построена типовая схема данных элек-

тельный музей. Одним из преимуществ использования CASE-средств является то, что модель данных не зависит от СУБД, что обеспечивает высокую переносимость электронной коллекции. При мененная структура имеет высокую гибкость, необходимость добавления нового атрибута или даже нового отдела не поставит

разработчика перед необходимостью значительной перестройки программного кода музея. Использованный инструментарий позволил обеспечить концептуальное соответствие моделирования данных и объектной модели. Все это обеспечивает кросс-платформенность, легкость обновления информации и

низкие временные и ценовые затраты на разработку и поддержку коллекции.

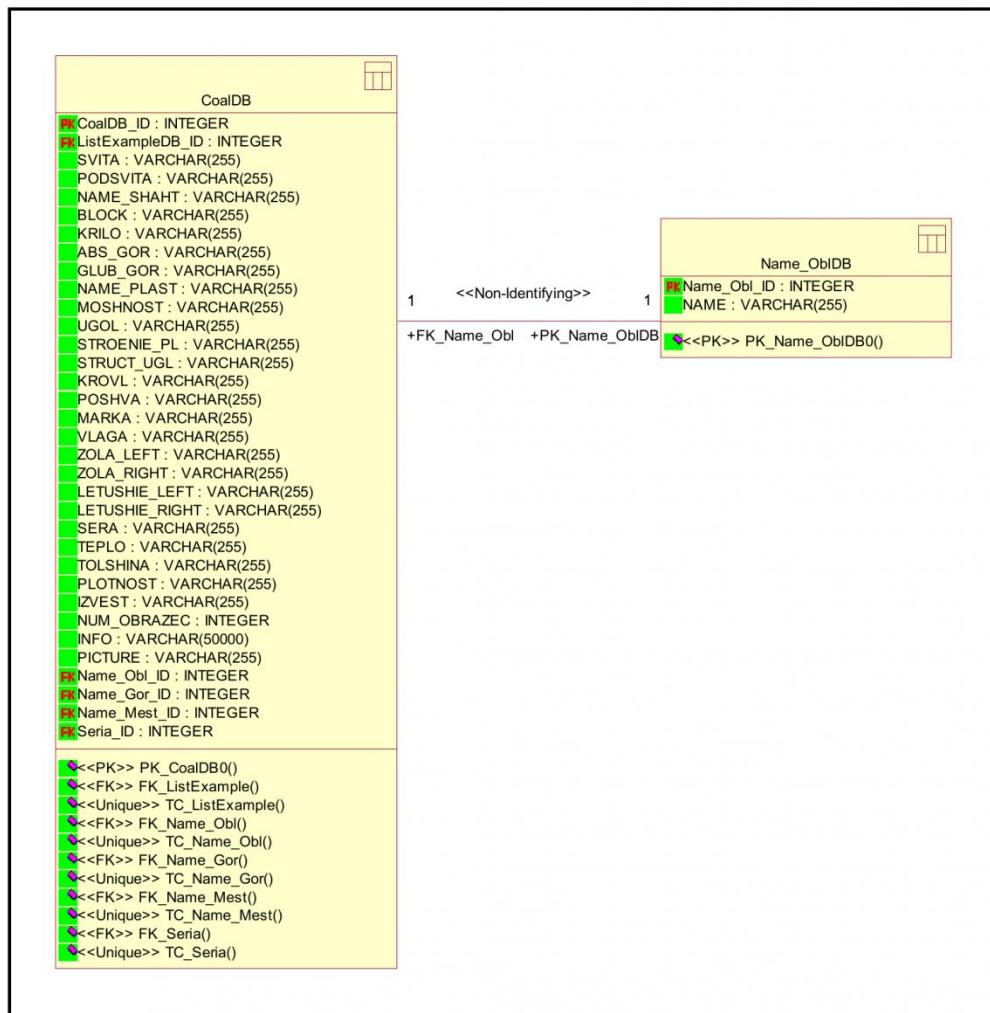


Рис. 5 Нормализация таблицы

поэтому для переноса данных была написана небольшая программа на языке Java, автоматически создающая необходимые

тронной коллекции, на основе которой при минимальных изменениях можно в короткие сроки построить любой вирту-

СПИСОК ЛИТЕРАТУРЫ

1. Фролов А. и Фролов Г. Базы данных в Интернет. – М.: «Русская Редакция», 2000 – 432 с.
2. Научный электронный журнал "Электронные библиотеки" www.iis.ru/el-bib/
3. Научное издание "Oracle 8. Энциклопедия пользователя". – Киев. Изд. «ДиаСофт», 1998 – 864 с.
4. Электронный музей Нобеля. www.nobel.se
5. Официальный сайт Лувра. www.louvre.fr
6. Эрмитаж. www.hermitage.ru
7. www.interface.ru

□ Автор статьи:

Зарубин

Алексей Анатольевич
- аспирант Института Угля и
Углемеханики СО РАН