

ТРЕБОВАНИЯ К СИСТЕМЕ РЕСТРУКТУРИРОВАНИЯ БАЗЫ ДАННЫХ

Д. П. Лакомов

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д.39
<dmitrii_lakomov@mail.ru>

УДК 665.63

Д. П. Лакомов. **Требования к системе реструктурирования базы данных** // Труды СПИИРАН. Вып. 2, т. 1. — СПб.: СПИИРАН, 2004.

Аннотация. В статье описаны требования к системе поддержки принятия решения при перепроектировании базы данных и сформулированы задачи, которые она должна решать. — Библ. 4 наим.

UDC 665.63

D. P. Lakomov. **Requirements for database refactoring system** // SPIIRAS Proceedings Issue 2, vol. 1. — SPb.: SPIIRAS, 2004.

Abstracts. *Requirements for Decision Support System for relational database refactoring are proposed and problems that this System should solve are formulated.* — Bibl. 4 items.

1. Введение

В настоящее время активно ведутся исследования и разрабатываются программные средства для адаптивного улучшения архитектуры проекта (рефакторинга) в процессе его эксплуатации и сопровождения. Наиболее широко теория рефакторинга разработана для улучшения кода программ при сопровождении. Предметом настоящей статьи являются приемы последовательного улучшения архитектуры части программного проекта, относящейся к структуре базы данных.

Каким бы хорошим ни признавалось решение о структуре базы данных при начальном проектировании, оно устаревает. Пути развития системы не всегда укладываются в модель расширения, заложенную при начальном проектировании, и зачастую требуется внести изменения в архитектуру проекта.

Программное средство реструктуризации реляционной базы данных должно решать задачи сохранения свойств системы, поддержки новых требований и проверки соответствия целям оптимизации.

2. Существующие системы проектирования БД

Системы проектирования БД позволяют создавать структуру БД и предоставляют средства ее анализа. Однако, при модернизации логической модели и физической структуры БД задачи сравнения исходной и полученной структур, а также задача переноса существующих данных такими системами не решаются.

Наиболее распространенной системой проектирования БД является ER-Win фирмы Computer Associations. Основной его чертой является использование только моделей «Сущность — Связь» (также называемых ER-диаграммами) как средства логического моделирования данных и построение логических моделей только одного уровня детализации. Реализована концепция «Subject Area» — «предметных областей», под которыми понимается группа взаимосвязанных сущностей (таким образом, реализуется многомодельность).

Средство поддерживает конструкции языка SQL и позволяет строить физическую модель БД. Реализован механизм Reverse Engeneereng — построение схемы БД и модели данных по физической структуре, представленной скриптом создания БД или выделенной из файла БД или источника данных ODBC [2, 4].

Средство PowerDesigner разработано фирмой SyBase. Поддерживается несколько типов моделей (UML, CDM и др.) и уровней моделирования — как по степени детализации, так и по типу моделируемых объектов и процессов. Классические ER диаграммы заменены CDM-диаграммами (Conceptual Data Modeling), предоставляющих возможность построения нереляционных моделей данных [3, 4].

3. Оптимизация как повышение производительности

Одним из наиболее важных критериев оценки БД является производительность, под которой понимают время исполнения наиболее важных запросов на чтение, изменение и добавление данных. Отметим, что управление производительностью в значительной мере определяется используемой СУБД. Повышение производительности может достигаться по следующим направлениям.

- изменение физической структуры БД: добавление индексов и сохранение результатов исполнения запросов (материализованных представлений);
- использование настройки СУБД: указывают, если есть возможность, стратегию исполнения и оптимизации запросов;
- регулирование настроек самой СУБД как программы: увеличение используемых ресурсов компьютера;
- изменение структуры таблиц. Таблицы приводятся к такому виду, что запросы на чтение, изменение и добавление обрабатывают быстрее.

Подход с изменением структуры таблиц важен по ряду причин:

- не все СУБД поддерживают материализованные представления как средства хранения промежуточных результатов запросов, и указание иной структуры таблиц есть единственный (кроме индексов) способ оптимизации;
- выбор наложения индексов и управления исполнением запросов определяется структурой таблиц;
- изменение априорной информации о задачах, решаемых БД, будет приводить к необходимости менять структуру таблиц.

4. Нормализация как одно из направлений реструктурирования

В процессе развития системы выявляются новые атрибуты, отношения и зависимости. В отличие от добавления новых атрибутов и отношений, поддержка зависимостей требует обеспечения поддержки целостности (избежание потерь данных, обеспечение их однозначности), реализуемого довольно сложными алгоритмами, и может потребовать существенного обновления данных (например, в первичный ключ потребовалось добавить уже существующее в таблице поле, в котором уже есть NULL-значения).

Одним из стандартных способов устранения трудностей с соблюдением зависимостей является нормализация, то есть замена отношений новыми от-

ношениями, для которых эта зависимость уже не выполняется либо носит иной характер. В качестве упрощения схемы зависимости между атрибутами может служить приведение отношения в третью нормальную форму методом формирования для зависимости отдельного отношения. Первичным ключом нового отношения является детерминант зависимости. В результате реструктуризации зависимость остается, но она переходит в зависимость от первичного ключа. В этом случае не требуется проверки выполнения удаленной зависимости.

Однако в общем случае нормализация приводит к некоторому замедлению исполнения запросов, а современные OLAP технологии непосредственно включают в себя денормализацию как элемент технологии именно в целях максимального ускорения выборки данных и операций над ними. Таким образом, необходимо некоторое средство, которое позволит варьировать структуры БД по направлению нормализация-денормализация и сравнивать их по времени исполнения важнейших запросов.

5. Хранение зависимостей

Под зависимостями понимаются как функциональные, так и многозначные зависимости между группами атрибутов. Зависимость может включать атрибуты разных отношений (бизнес-правила, не выражаемые средствами стандартного SQL). Кроме того, надо включить во множество зависимостей вычисляемые атрибуты (значение атрибута вычисляется из значений других атрибутов) [1].

Учет зависимостей важен для более полного описания характера информации о моделируемом объекте, в первую очередь как существенное дополнение к уже реализованной структуре БД. Необходимо учитывать зависимости, выявленные разработчиком БД на основании анализа предметной области, так и сформированные при денормализации отношений.

6. Хранение запросов

Под запросами понимаются SQL выражения запросов на чтение, добавление, вставку и удаление данных. Запросы не входят непосредственно в структуру БД и содержатся в коде клиентских приложений, а в самой БД они содержатся в виде представлений (view) и внутри хранимых процедур и функций. Представления являются объектами структуры БД и потому учитываются и преобразуются вместе со всей структурой БД. Запросы, не являющиеся представлениями, следует сохранять отдельно для гарантированного изменения выражения запроса при изменении структуры.

Хранение должно отвечать следующим требованиям:

- фиксировать связи запроса с другими запросами и объектами БД;
- поддерживать хранение эквивалентных формулировок запроса.

7. Реструктуризация

Реструктуризация включает следующие стадии:

- выбор пользователем отношений и зависимостей, по которым будет проведена реструктуризация

- изменение структуры БД по отношениям и зависимостям. Возможно изменение структуры 2 способами — как путем формирования новой БД, так и путем изменения рабочей версии БД;
- формирование механизма изменения и переноса данных;
- переформулировка запросов с учетом изменений структуры;
- отслеживание различий между структурами;
- оценка новой структуры.

8. Заключение

Необходимо разработать новое инструментальное средство для поддержки принятия решений о выборе структуры на указанных стадиях реструктуризации. В его задачи должны входить фиксирование информации о структуре данных, порождение предложений по реструктурированию и поддержка автоматического изменения SQL-выражений для работы с данными при реструктурировании. Для решения данных задач и поддержки выше перечисленных стадий реструктурирования, эта система должна удовлетворять следующим требованиям

- хранение зависимостей
- обеспечивать удаление зафиксированных зависимостей и добавление зависимостей, возникших при реструктуризации.
- обеспечивать хранение изменений структуры
- обеспечивать хранение SQL-выражений, с учетом их связи с объектами БД
- при изменении структуры автоматически порождать новые выражения для всех сохраненных SQL-выражений
- возможность порождать скрипт новой структуры и скрипт на изменение данных.
- обеспечивать наработку статистики времени исполнения запросов.

Литература

- [1] Мейер Д. Теория реляционных баз данных. М: Мир, 1987. 608 с.
 [2] Erwin. <<http://www.cia.com>>.
 [3] PowerDesigner. <<http://www.sybase.com>>.
 [4] Федоров А., Елманова Н. Введение в базы данных // КомпьютерПресс. 2001. № 1.