

## Повторное использование структур баз данных

*Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ»*

Рассмотрен классический основанный на процессе нормализации подход к проектированию реляционных баз данных. Выделен и классифицирован ряд шаблонов структур баз данных, предназначенных для повторного использования в процессе проектирования. Изучены основные достоинства и недостатки описанных шаблонов.

**Ключевые слова:** проектирование баз данных, шаблоны баз данных, нормализация.

### Введение. Постановка задачи

С момента возникновения потребности в хранении электронных данных было разработано множество различных типов систем управления базами данных (СУБД). Основные из них по типам моделей хранимых данных можно подразделить на сетевые, иерархические, реляционные и постреляционные. Самой совершенной и популярной на сегодня является реляционная модель данных. Она предназначена для хранения связанных данных в табличном виде и опирается на формальный аппарат алгебры отношений и реляционного исчисления.

Применение реляционной модели позволяет выполнять следующие задачи:

- хранение данных;
- обеспечение целостности хранимых данных;
- сокращение избыточности хранимых данных;
- извлечение данных из хранилища по заданным критериям.

Для проектирования структур хранения данных в реляционных базах данных (БД) используется процесс нормализации [1].

Архитекторами БД было подмечено, что отдельные решения при проектировании структур данных в БД имеют тенденцию повторяться в различных задачах [2, 3].

**Цель работы** – выделить структуры данных (шаблоны), которые возможно применять повторно при проектировании БД.

### 1. Нормализация

Нормализация базы данных сводит к минимуму количество избыточной информации [1]. Ее целью является сохранение данных только один раз, но в нужном месте. Нормализация логической схемы базы данных включает в себя использование формальных методов (рис. 1) для разделения данных на несколько связанных таблиц.

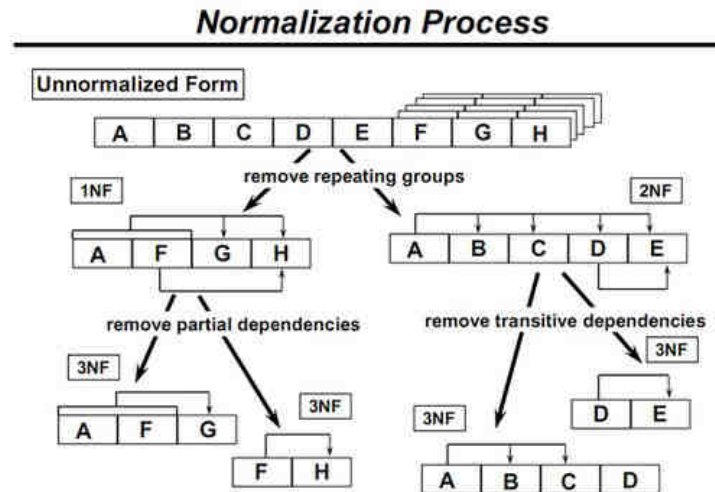


Рис. 1. Процесс нормализации структуры БД

При том, что идеи нормализации весьма полезны для проектирования баз данных, они отнюдь не являются универсальным или исчерпывающим средством повышения качества проекта БД. Это связано с тем, что существует слишком большое разнообразие возможных недостатков в структуре БД, которые нормализацией не устраняются.

Нормализованная БД имеет ряд преимуществ:

- увеличение связности и согласованности данных (повышение целостности данных);
- уменьшение избыточности и дублирования данных (в том числе и NULL-значений);
- эффективное использование индексов (уменьшение их количества, повышение производительности).

Наряду с существенными преимуществами нормализованной базе данных присущи определённые недостатки:

- рост числа таблиц и соединений между ними при выборке данных (снижение производительности);
- усложнение модификации структуры данных (изменение ключевого атрибута влечёт за собой изменение нескольких таблиц).

## 2. Повторное использование структур хранения данных в БД

Процесс проектирования БД не является формальным и тривиальным и не всегда полученный с помощью нормализации результат оптимален с точки зрения производительности либо архитектурной гибкости. Поэтому после процесса нормализации возможно проведение операции денормализации либо применение прочих ненормализованных структур при проектировании БД [4]. Также в целях упрощения процесса проектирования вместо нормализации часто используется декомпозиция.

Существует множество шаблонов [2 – 5], которые возможно применять повторно при проектировании БД. Условно авторы статьи их разделяют на «примитивные шаблоны» и «domain-oriented шаблоны» (предназначенные для решения конкретной задачи).

### 3. Примитивные шаблоны

#### 3.1. Однотабличное наследование

Однотабличное наследование (SingleTableInheritance) – шаблон проектирования, который позволяет перенести модель объектно-ориентированного наследования на одну таблицу реляционной базы данных [3]. При использовании однотабличного наследования вся иерархия классов отображается в одной таблице (рис. 2), которая включает в себя колонки для всех свойств всех классов в иерархии. Конкретный подкласс представлен конкретными строками и может быть идентифицирован по значению столбца (type) с указанным типом подкласса.

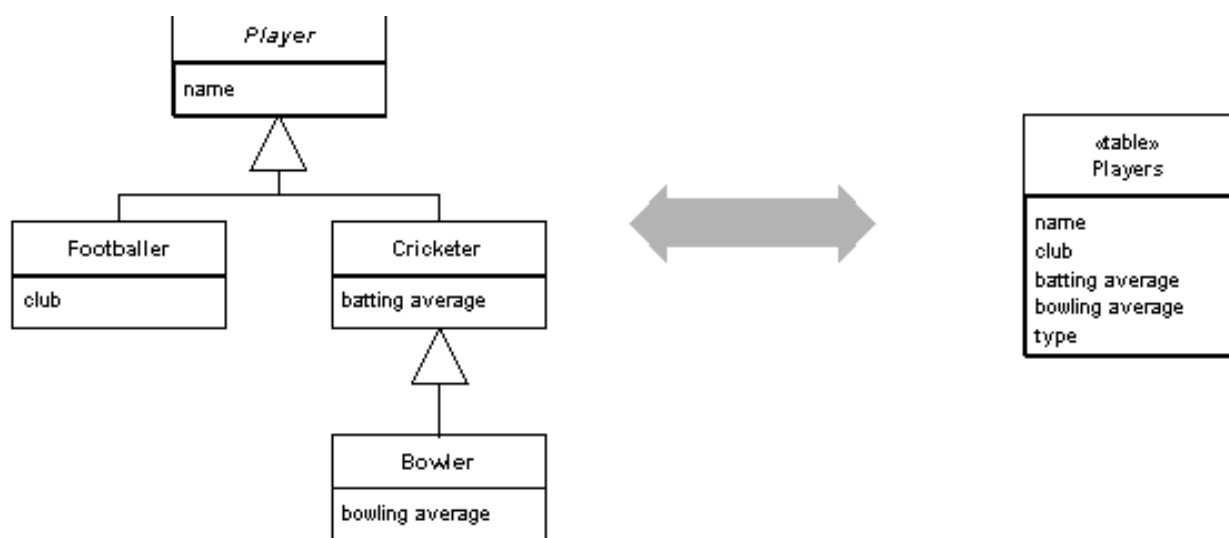


Рис. 2. Пример однотабличного наследования

К недостаткам такого шаблона следует отнести: 1) атрибуты, предназначенные для хранения свойств подклассов, могут содержать NULL-значения (это накладывает ограничения на целостность данных); 2) шаблон не нормализован, так как существует функциональная зависимость между неключевыми атрибутами (type – неключевой атрибут, который определяет, какие атрибуты принадлежат каким подклассам, что также накладывает ограничения на целостность данных).

#### 3.2. Наследование таблиц классов

Шаблон «Наследование таблиц классов» (ConcreteTableInheritance) переносит объектно-ориентированное наследование классов на несколько таблиц в БД [3]. Каждый класс/подкласс с объявленными свойствами имеет свою собственную таблицу. Таблицы для подклассов содержат столбцы только для неунаследованных свойств (свойств, объявленных в данном подклассе) вместе с первичным ключом, который также является внешним ключом таблицы базового класса (рис. 3), т.е. между родительской таблицей и дочерними таблицами реализована связь «один к одному».

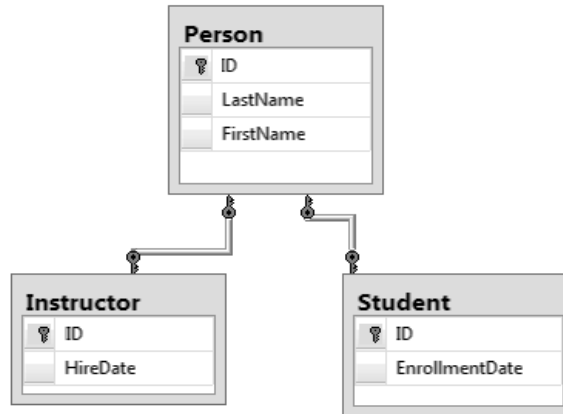


Рис. 3. Использование шаблона «Наследование таблиц класса»

Основными преимуществами этого шаблона являются: 1) нормализованная схема (соблюдается обеспечение целостности данных); 2) простое изменение схемы данных (модификация базового класса или добавление нового подкласса решается изменением/добавлением одной таблицы).

К недостаткам следует отнести снижение производительности из-за роста числа таблиц для его реализации и соответствующего роста числа соединений в запросах.

#### 4. «Domain-oriented» шаблоны

##### 4.1. Денормализованная таблица

*Денормализация* – намеренное приведение структуры базы данных в состояние, не соответствующее критериям нормализации из-за добавления избыточных данных.

В запросах к полностью нормализованной базе часто приходится соединять большое количество таблиц. А каждое соединение – ресурсоёмкая операция. Для уменьшения количества соединений несколько нормализованных таблиц объединяют в одну – денормализованную (denormalised table, flat table) (рис. 4). Это позволяет повысить производительность путём привнесения некоторой избыточности хранимых данных.

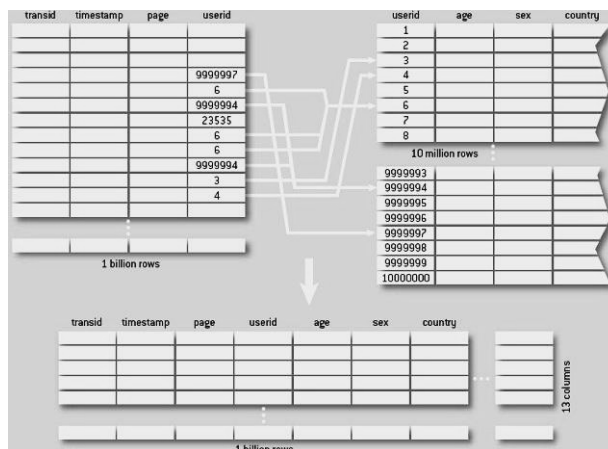


Рис. 4. Денормализованная таблица

К недостаткам такого подхода следует отнести: 1) избыточность данных; 2) необходимость дополнительной поддержки целостности данных (усложняется разработка приложения для работы с БД); 3) необходимость дополнительной поддержки связанных данных.

#### 4.2. «Сущность-атрибут-значение»

Шаблон сущность-атрибут-значение (entity-attribute-value, EAV) представляет собой модель данных для описания объектов с большим числом потенциально возможных атрибутов, при том, что реально применяются только некоторые из них (рис. 5). В математике эта модель известна как разреженная матрица. Реализация паттерна EAV связана с определёнными трудностями [2], однако он позволяет эффективно использовать хранилище данных при непрерывном росте числа атрибутов-параметров хранимых объектов.

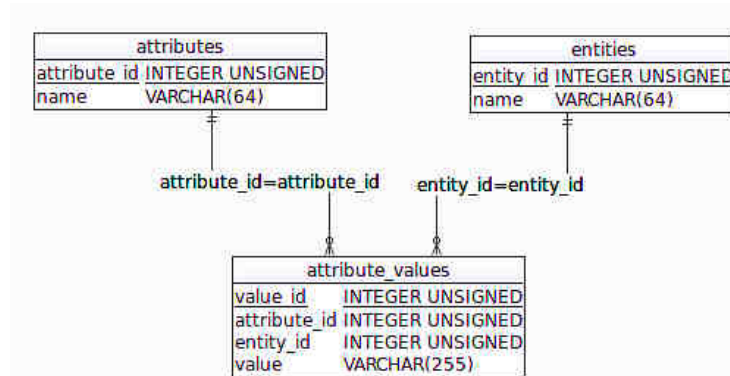


Рис. 5. Модель «сущность-атрибут-значение»

Суть шаблона EAV заключается в использовании трёх таблиц [5]: 1) сущность (entity) – содержит по одной строке на каждый объект; 2) атрибут (attribute) – содержит по одной строке на каждый атрибут объекта; 3) значение (value) – содержит по одной строке на каждое существующее значение атрибута объекта.

Применение подхода с использованием шаблона EAV приводит к следующим недостаткам:

- невозможность применения встроенных средств обеспечения целостности (каждый атрибут соответствует строке в таблице, а не колонке, как того требует реляционная модель);
- ограничение на использование SQL-типов данных (классический шаблон EAV предполагает единственный тип данных для хранимых значений – строковый, однако, существуют способы обхода этого ограничения).

При существующих недостатках шаблон проектирования EAV имеет ряд преимуществ:

- уменьшение временных затрат на проектирование и разработку простых структур данных по сравнению с классическим процессом нормализации;
- лёгкость расширения словаря хранимых атрибутов данных, которые могут добавляться "на лету" без перепроектирования схемы данных;
- возможность написания универсальных интерфейсных компонентов доступа к данным.

## Выводы

Проведя исследование популярных подходов к хранению данных в реляционных БД можно сделать вывод, что не существует универсального подхода для хранения данных. Наряду с классическим проектированием БД с применением нормализации шаблоны хранения данных позволяют улучшить различные характеристики БД. Каждый из представленных подходов позволяет решить поставленную задачу, но при этом имеет ряд присущих ему недостатков, таких, как снижение производительности, избыточность данных либо невозможность использования механизмов обеспечения целостности данных. Исходя из вышеизложенного, задача выбора структур данных БД в зависимости от решаемых задач хранения данных и требований к разрабатываемому проекту является актуальной и требует дальнейших исследований.

## Список литературы

1. Normalization [Электронный ресурс] // MSDN. – Режим доступа до ресурсу: [https://technet.microsoft.com/en-us/library/ms191178\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms191178(v=sql.105).aspx).
2. Karwin B. SQL Antipatterns [Text] // Bill Karwin, 2010.– 334 с. – (The Pragmatic Bookshelf).– (1-934356-55-7).
3. Fowler M. Patterns of Enterprise Application Architecture [Text] // Martin Fowler., 2003. – (Addison Wesley Longman).
4. Stathopoulou E. Design Patterns for Relational Databases [Электронный ресурс] / E. Stathopoulou, P. Vassiliadis // Operational Database Management Systems. – 2013.– Режим доступа до ресурсу: <http://www.odbms.org/wp-content/uploads/2013/11/PP2.pdf>.
5. Tropashko. V. SQL Design Patterns [Text] // Vadim Tropashko.. – Kittrell, NC, USA, 2006. – (Rampant Techpress).

Поступила в редакцию: 14.11.2015

## Повторне використання структур баз даних

Розглянуто класичний оснований на процесі нормалізації підхід до проектування реляційних баз даних. Виділено і класифіковано ряд шаблонів структур баз даних, призначених для повторного використання у процесі проектування. Проаналізовано основні переваги і недоліки описаних шаблонів.

**Ключові слова:** проектування баз даних, шаблони баз даних, нормалізація.

## Reuse of the Database Structures

The article describes the classic approach to the relational databases design, which based on the normalization process. The series of database structure templates for repeated use in the design process have been identified and classified. The main advantages and disadvantages of the described patterns were study.

**Keywords:** database design, database templates, normalization.