



Munich Personal RePEc Archive

# Database design and normalization of data based on the relational model

Meshcherjakova, Natalya

Financial University under the Government of the Russian  
Federation

5 February 2011

Online at <https://mpra.ub.uni-muenchen.de/63472/>  
MPRA Paper No. 63472, posted 06 Apr 2015 06:23 UTC

## **ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ И НОРМАЛИЗАЦИЯ ДАННЫХ, ОСНОВАННЫХ НА РЕЛЯЦИОННОЙ МОДЕЛИ**

*Мещерякова Н.А.*

*Финансовый университет при Правительстве Российской Федерации*

### **Database design and normalization of data based on the relational model**

*N.A. Meshcherjakova*

*Financial University under the Government of the Russian Federation*

**Аннотция:** Автор изучает проблемы структуризации, хранения и обработки больших объемов информации с использованием систем управления базами данных. Рассматривает вопрос проектирования реляционной модели базы данных, которая является наиболее распространенной.

**Ключевые слова:** системы управления базами данных, реляционная модель, уровни моделирования.

**Abstract:** The author studies the problem of structuring, storing and processing large amounts of data using database management systems. The question of designing a relational database model, since this model is the most common.

**Key words:** database management system, relational model, levels of modeling.

Информационное общество, в котором главными ценностями становятся знания и информация, характеризуется господством информационных технологий [3]. Параллельно с развитием ЭВМ, и их широким внедрением во все сферы деятельности увеличивались объемы, и усложнялась структура обрабатываемой информации [1].

Для решения проблемы структуризации, хранения и обработки больших объемов информации используют системы управления базами данных (СУБД) (database management system – DBMS). Основными моделями данных являются сетевая, иерархическая, реляционная, при этом реляционная модель изучается

более подробно, так как именно она является наиболее распространенной благодаря своей внешней простоте [2].

При проектировании реляционной модели базы данных решаются две основных проблемы:

Каким образом отобразить объекты предметной области в абстрактные объекты модели данных (таблицы, связи), чтобы это отображение не противоречило семантике предметной области и было по возможности лучшим (эффективным, удобным и т.д.)?

Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создание каких дополнительных структур (индексов и др.) потребовать и т.д.?

При разработке реляционной модели базы данных можно выделить несколько *уровней моделирования*:

- Модель предметной области или фаза формулирования требований;
- Логическая модель данных;
- Физическая модель данных;
- Собственно база данных и приложения.

На первый из поставленных выше вопросов отвечает *модель предметной области* и *логическая модель данных*, на второй – *физическая модель*. Рассмотрим подробнее построение модели предметной области и логической модели данных [5].

Каждая область человеческой деятельности связана с определенным видом информации, в связи с этим *фаза формулирования требований* состоит из построения концептуальной модели данных, обеспечивающей *однозначное описание структуры предметной области*; определения функциональных зависимостей, ограничений и правил. *Предметная область* – часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации. *Функциональные зависимости* отражают взаимосвязи, обнаруженные между объектами предметной области и являются

дополнительными ограничениями, определяемыми предметной областью. Таким образом, функциональная зависимость – семантическое понятие, она возникает, когда по значениям одних данных в предметной области можно определить значения других данных. Модель предметной области пытаются строить по аналогии с естественным языком. Результатом данной фазы является семантическая модель БД в виде модели "Сущность-Связи" (часто ее называют кратко ER-моделью).

Главное назначение *логической модели данных* – преобразование *требований к данным в структуры данных*. Производится систематизация разработанной информации и отражение ее свойств по содержанию, структуре, объему, связям, динамике с учетом удовлетворения информационных потребностей всех категорий пользователей. Построение логической модели ведется по этапам с постепенным приближением к оптимальному варианту в рамках конкретной точки зрения.

Проектирование логической модели данных рассматривается как процесс выделения информационных объектов и описания их атрибутов; задания элементов данных, их типов, длины и других свойств; выделения первичных ключей и зависимостей между атрибутами; установления связей между таблицами; нормализации [4].

Для правильной организации данных в таблицах используется понятие *нормализации*. *Нормализация данных* – процесс исключения избыточной информации, при котором достигается такое состояние БД, что каждый элемент информации запоминается только один раз. Алгоритм нормализации заключается в последовательной декомпозиции отношений для устранения функциональных зависимостей атрибутов от части составного первичного ключа (приведение к 2НФ) и устранения функциональных зависимостей неключевых атрибутов друг от друга (приведение к 3НФ). Функциональная зависимость - семантическое понятие. Она возникает, когда по значениям одних данных в предметной области можно определить значения других данных.

**Физическое проектирование** – определение особенностей хранения данных, методов доступа и т.д.

Создание собственно **базы данных** состоит из создания таблиц, отношений, индексов, ограничений, форм, запросов, отчетов, заполнения таблиц, тестирования средствами конкретной СУБД. Примерами СУБД для персональных компьютеров являются Microsoft Access, Microsoft FoxPro, Borland Paradox, Borland dBase, Oracle, DB2 фирмы IBM и др. Продуманность пользовательского интерфейса MS Access делает его особенно привлекательным в качестве средства решения задач организации и обработки данных для специалистов в области экономики и финансов, не имеющих квалификации или опыта в профессиональном программировании, или с минимальным таким опытом. СУБД MS Access служит для проектирования двумерных таблиц с разными типами данных; установления связей между таблицами с поддержкой целостности данных; ввода, хранения, просмотра, сортировки, модификации и выборки данных из таблиц с использованием различных средств контроля информации, индексирования таблиц; создания, модификации и использования форм, запросов и отчетов.

В качестве примера для рассмотрения процесса **проектирования базы данных** будет использоваться предметная область по созданию автоматизированного рабочего места (АРМ) администратора театра. Заметим, что здесь не ставится цель всеобъемлющего отслеживания данного процесса, поэтому рассматриваемая в дальнейшем модель данных не содержит всех необходимых объектов и их атрибутов.

АРМ администратора театра дает возможность автоматизировать комплекс работ по обслуживанию зрителей, а именно:

- создает удобную систему ведения репертуара театра;
- дает быстрое и наглядное получение всех необходимых данных о любом спектакле из репертуара (программа спектакля);
- печать любой информации на билете.

Модель предметной области опишем следующим неформальным текстом:

1. В театре в один день, в одно время проходит один спектакль.
2. Каждый спектакль имеет номер, название, режиссера, жанр, количество действий.
3. О каждом актере необходимо хранить уникальный табельный номер, ФИО и адрес.
4. Каждый актер может играть роль в одном или нескольких спектаклях, или временно не участвовать ни в каком спектакле.
5. В одном спектакле один актер играет одну роль.
6. На каждый спектакль продаются билеты, которые характеризуются номером ряда, номером места и ценой.

В ходе логического моделирования на первом шаге предложено хранить данные в одном отношении *Спектакли\_Актеры\_Роли\_Билеты*, имеющем следующие атрибуты:

<i>Номер спектакля</i>	Название спектакля
Количество действий	ФИО режиссера
Жанр	<i>Табельный номер актера</i>
ФИО актера	Роль
Адрес	<i>Дата</i>
<i>Время</i>	<i>Ряд</i>
<i>Место</i>	Цена билета

Курсивом выделены атрибуты, которые согласно описанию предметной области в совокупности однозначно определяют запись в таблице, т.е. составляют первичный ключ. При заполнении таблицы отношения очевидно, что данные хранятся в ней с *большой избыточностью*, в строках повторяются ФИО режиссера, жанр, ФИО актеров и т.д. Кроме того, в данном отношении хранятся вместе независимые друг от друга данные - и данные о спектаклях, и об актерах, и о билетах и о ролях актеров в спектаклях. Пока никаких действий с отношением не производится, это не страшно. Но как только состояние предметной области изменяется, то, при попытках соответствующим образом изменить состояние базы данных, возникает большое количество проблем, которые получили название *аномалии вставки (INSERT)*, *аномалии обновления*

(UPDATE), аномалии удаления (DELETE). В нашем примере аномалии возникают при попытке *ввести* актера, который пока не играет ни в одном спектакле (ТН\_акт входит в состав первичного ключа, и, следовательно, не может содержать null-значений). *Обновление* жанра, названия спектакля и др. невозможно выполнить одним действием. Невозможно *удалить*, например, снятый с репертуара спектакль, если есть актер, который играет только в этом спектакле (он тоже удалится). Таким образом, логическая модель данных не адекватна модели предметной области.

Выделим следующие функциональные зависимости:

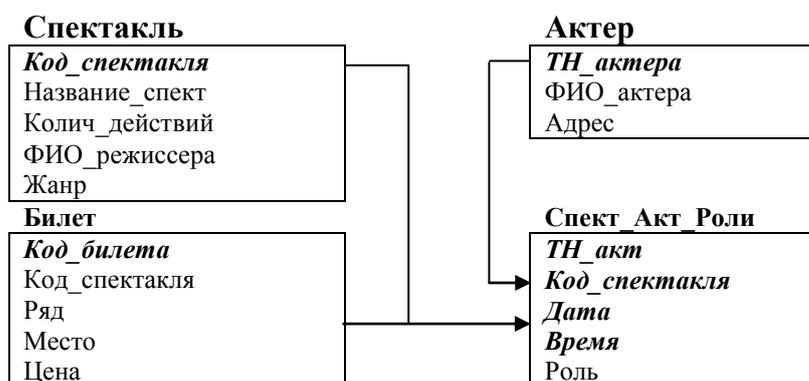
Код_спектакля ТН_акт Дата Время Ряд Место	→	Название_спектакля
Код_спектакля ТН_акт Дата Время Ряд Место	→	Колич_действий
Код_спектакля ТН_акт Дата Время Ряд Место	→	ФИО_режиссера
Код_спектакля ТН_акт Дата Время Ряд Место	→	Жанр
Код_спектакля ТН_акт Дата Время Ряд Место	→	ФИО_актера
Код_спектакля ТН_акт Дата Время Ряд Место	→	Роль
Код_спектакля ТН_акт Дата Время Ряд Место	→	Адрес
Код_спектакля ТН_акт Дата Время Ряд Место	→	Цена билета

Отношение ***Спектакли\_Актеры\_Роли\_Билеты*** не находится в 2НФ, т.к. есть атрибуты, функционально зависящие от части составного первичного ключа:

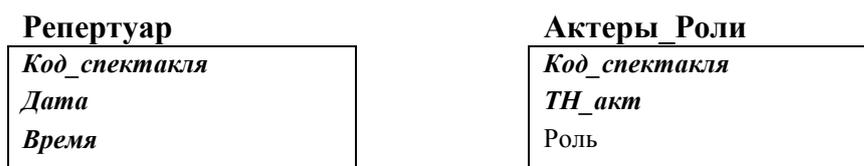
Код_спектакля	←	Название_спект Колич_действий
	←	ФИО_режиссера Жанр
ТН_акт	←	ФИО_актера Адрес
Код_спектакля ТН_акт Дата Время	←	Роль
Код_спектакля Ряд Место	←	Цена билета

Для того чтобы устранить зависимость атрибутов от части составного первичного ключа, нужно произвести *декомпозицию* отношения. При этом *те атрибуты, которые зависят от части составного ключа*, выносятся в отдельное отношение.

Отношение ***Спектакли\_Актеры\_Роли\_Билеты*** декомпозируется на отношения ***Спектакль, Актер, Билет, Спектакли\_Актеры\_Роли***.



Отношения *Спектакль*, *Актер*, *Билет*, полученные в результате декомпозиции, находятся в 2НФ, т.к. они имеют простые первичные ключи. Отношение *Спектакли\_Актеры\_Роли* имеет составной ключ и по-прежнему имеет аномалии вставки – нельзя назначить актера на роль без определения даты и времени спектакля. Производим дальнейшую декомпозицию отношения:



Таким образом, все обнаруженные аномалии обновления устранены, отношения находятся в третьей нормальной форме.

### Литература

1. Бурмистрова Н.А. Математическое моделирование и всеобщая компьютеризация или имитационные модели // Информационные технологии в образовании. VIII Международная конференция-выставка: сборник трудов. – М.: Изд-во МИФИ, 1998. – С. 20–22.
2. Бурмистрова Н.А. Математическое моделирование как творческий процесс // Естественные науки и экология: межвузовский сборник научных трудов. – Омск: Изд-во ОмГПУ, 1998. – С. 3–5.
3. Мещерякова Н.А. Методические аспекты обучения информатике в вузах экономического профиля // Математика и информатика: наука и образование: межвузовский сборник научных трудов: Ежегодник. Вып. 3. Омск: Изд-во ОмГПУ, 2003. С. 198-204.
4. Остринская Л.И. Автоматизированные информационные системы и технологии в экономике. Учебное пособие. – Омск: Изд-во СибАДИ, 2005. – 270 с.
5. Экономическая информатика. Учебник / Под ред. В.П. Косарева и Л.В. Ереминой – М.: Финансы и статистика, 2002. – 592 с.

**Опубликовано:** Мещерякова Н.А. Проектирование баз данных и нормализация данных, основанных на реляционной модели // Физико-математические и компьютерные науки в образовании: сборник научных трудов / Отв. ред. Л.Г. Кузнецова. – Нижневартовск: Изд-во НГГУ, 2011. – С. 133-137 (Доступна [электронная версия](#))