

Проектирование нормальных форм для РБД

Лекция 9

- **Нормализацией отношений** называют процесс устранения нежелательных функциональных зависимостей исходных отношений.
- **Нормализация** - это обратимый пошаговый процесс замены первоначальной схемы БД (или совокупности отношений) другой схемой, в которой отношения имеют более простую и регулярную структуру, сохраняя соответствие предметной области.

- *Процесс нормализации отношений* предусматривает последовательный переход от нормальной формы более низкого уровня к нормальной форме более высокого уровня (от первой до пятой формы).
- На каждом этапе процесса нормализации предполагается, что анализируемое отношение уже находится в нормальной форме достигнутого уровня, и производится проверка выполнения формальных правил для нормальной формы более высокого уровня.

- На основании анализа существующих отношений выявляются зависимости между данными, и осуществляется надлежащая перегруппировка атрибутов.
- Чем выше уровень нормальной формы, тем жестче накладываемые на отношения ограничения. В ходе нормализации отношения схемы базы данных становятся все менее подвержены различного рода аномалиям.
- Для начала процесса нормализации необходимо, чтобы все отношения базы данных находились в первой нормальной форме (1НФ). Нормальные формы более высокого порядка могут использоваться разработчиками по своему усмотрению.

Первая нормальная форма отношений. (1НФ)

- Отношение называется *нормализованным*, если в каждом кортеже каждый атрибут может иметь только единственное значение. В реляционной модели данных отношение всегда нормализовано, так как свойством отношения являются только простые (атомарные) значения каждого его атрибута.

Определение первой нормальной формы:

- Отношение находится в первой нормальной форме (1НФ), если все его атрибуты имеют простые (атомарные) значения, то есть значения в домене каждого атрибута отношения не являются ни списками, ни множествами простых или сложных значений.

Первая нормальная форма отношений. (1НФ)

Основные критерии:

- Все строки должны быть различными.
- Все элементы внутри ячеек должны быть атомарными (не списками). Другими словами, элемент является атомарным, если его нельзя разделить на части, которые могут использовать в таблице независимо друг от друга.

Определение понятия атомарности условно.

Например, значение атрибута **Дата рождения** предполагается использовать целиком – «5 марта 1987», то оно атомарно.

Если необходимо использовать один или несколько таких атрибутов как **День рождения**, **Месяц рождения**, **Год рождения**, то данные атрибута **Дата рождения** не являются атомарными.

Приведенная таблица не нормализованная, и не находится в 1НФ потому, что включает величины (Саша, Федор), являющиеся совокупностью атомарных значений.

Имя	Пол
(Александр, Федор)	Мужской
Ольга	Женский

Нормализованное отношение Семья

Имя	Пол
Александр	Мужской
Федор	Мужской
Ольга	Женский

В каждом кортеже имеются повторяющиеся данные (значение поля ПОЛ). Этим вносится избыточность данных, являющаяся источником ряда аномалий и в дальнейшем процессе нормализации должна быть устранена.

Пример не 1NF таблицы:

Категория	Товары
Книги	Война и Мир, Азбука
Игрушки	Юла

В этом примере в одной из ячеек содержится список из двух элементов: Война и Мир, Азбука, т.е. он является не атомарным.

Исправить можно так:

Категория	Товары
Книги	Война и Мир
Книги	Азбука
Игрушки	Юла

Методы приведения к 1NF:

- ✓ Устраните повторяющиеся группы в отдельных таблицах (одинаковые строки).
- ✓ Создайте отдельную таблицу для каждого набора связанных данных.
- ✓ Идентифицируйте каждый набор связанных данных с помощью первичного ключа (добавить уникальный id для каждой строки)

Вторая нормальная форма отношений. (2НФ)

- Приведение отношения ко второй и третьей нормальным формам выполняется с целью избежать аномалий обновления данных при работе с БД.
- Часто первичный ключ отношения включает несколько атрибутов. В связи с этим вводится понятие **полной функциональной зависимости**.

Определение:

- **Неключевой** атрибут ***функционально полно*** зависит от составного ключа если он функционально зависит от всего ключа **в целом**, а не от какой-либо части входящих в ключ атрибутов.

Определение второй нормальной формы:

- Отношение находится во 2НФ, если оно находится в 1НФ и каждый **неключевой** атрибут функционально полно зависит от ключа.
- Отношение, у которого первичный ключ включает только один атрибут, всегда находится во 2НФ.

Вторая нормальная форма отношений. (2НФ)

Основные критерии:

- ✓ Таблица должна находиться в первой нормальной форме.
- ✓ Любое её поле, не входящее в состав первичного ключа, функционально полно зависит от первичного ключа.

Если таблица приведена к первой нормальной форме и у нее установлен уникальный id для каждой строки, то она находится и во второй нормальной форме.

Значение второго правила можно понять на примере, когда первичный ключ таблицы состоит из нескольких полей. То есть каждой строке соответствует уникальный набор из нескольких значение полей таблицы.

Пусть имеется отношение

ПОСТАВКИ(*N_ПОСТАВЩИКА*, ТОВАР, ЦЕНА).

Ключ отношения - "*N_поставщика + товар*".

Поставщик может поставлять различные товары, и один и тот же товар может поставляться разными поставщиками.

В случае поставки товара всеми поставщиками по одной и той же цене имеем следующие функциональные зависимости:

N_поставщика, товар \square *цена*
товар \square *цена*

Неполная функциональная зависимость атрибута "*цена*" от ключа приводит к следующей аномалии:

при изменении цены товара необходимо просмотреть всё отношение и изменить все записи о поставках этого товара.

Аномалия является следствием наличия в одной структуре данных двух семантических фактов: поставщик-товар и товар-цена.

Следующее разложение представляет отношения во 2НФ:

ПОСТАВКИ (N_ПОСТАВЩИКА, ТОВАР)

ЦЕНА_ТОВАРА (ТОВАР, ЦЕНА)

Например. Эта таблица находится в первой нормальной форме, но не во второй.

Категория	Дата	Скидка	Товар
Книги	10.10.2008	10%	PHP for dummies
Ноутбуки	11.10.2008	20%	Acer
Книги	10.10.2008	10%	Windows XP

- В этой таблице первичный ключ составляют первые два столбца (Категория и Дата).
- Скидка функционально полно зависит от них обоих, так как определяется одновременно категорией и датой.
- А вот с товаром проблемы. Она зависит только от категории и не зависит от даты (в любой день будет лежать там, пока не купят).
- Поэтому, говорят, что он зависит функционально не полно от всего первичного ключа и функционально полно от его части (Категория). Это нарушает требования второй нормальной формы.

Исправляется это разделением этой таблицы на две другие:

Категория	Дата	Скидка
Книги	10.10.2008	10%
Ноутбуки	11.10.2008	20%
Книги	10.10.2008	10%

Категория	Товар
Книги	PHP for dummies
Ноутбуки	Acer
Книги	Windows XP

Методы приведения к 2NF:

- Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям (Выше мы это сделали).
- Свяжите эти таблицы с помощью внешнего ключа (В нашем случае – это поле Категория).

Третья нормальная форма отношений

Определение транзитивной функциональной зависимости:

Если три атрибута X , Y , Z некоторого отношения находятся в функциональных зависимостях. $X \square Y$ и $Y \square Z$, но отсутствуют обратные зависимости, т.е. $Z \not\rightarrow Y$ и $Y \not\rightarrow X$, то Z транзитивно зависит от X .

Пусть имеется отношение

ХРАНЕНИЕ(ФИРМА, СКЛАД, ОБЪЕМ),

которое содержит информацию о фирмах, складах, с которых они получают товары, и объемах этих складов.

Ключевой атрибут - "**ФИРМА**". Если каждая фирма может получать товар только с одного склада, то в данном отношении имеются следующие функциональные зависимости:

фирма \square *склад*,
склад \square *объем*

При этом возникают аномалии:

- *в базу данных нельзя ввести данные об объеме склада, из которого ни одна фирма не получает товар (т.к. не определен ключевой атрибут)*
- *при изменении объема склада необходимо просмотреть всё отношение и внести изменения в кортежи для всех фирм, связанных с данным складом.*

Для устранения этих аномалий следует выполнить декомпозицию исходного отношения на два:

ХРАНЕНИЕ (ФИРМА, СКЛАД)

ОБЪЕМ_СКЛАДА (СКЛАД, ОБЪЕМ)

Определение третьей нормальной формы:

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Третья нормальная форма отношений

Основные критерии:

- Таблица находится во второй нормальной форме.
- Любой её не ключевой атрибут функционально зависит только от первичного ключа.

Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Например, есть у нас таблица:

Имя шпиона	Государство
Джеймс Бонд	Великобритания
Ким Филби	СССР
Штирлиц	СССР

В этой таблице ключом является имя шпиона.

А не ключевым полем – государство, на которое он работает.

Вполне логично предположить, что в этой таблице государства могут быть одинаковыми для нескольких записей. И для того, чтобы эта таблица находилась в третьей нормальной форме, не обходимо ее разделить на две:

ID	Государство
1	Великобритания
2	СССР

Имя шпиона	Государство
Джеймс Бонд	1
Ким Филби	2
Штирлиц	2

Третья нормальная форма отношений

Благодаря этому правилу, при удалении какого-то государства, имена шпионов не будут утеряны

Вообще, говоря, на практике, совершенствовать таблицы заканчивают на этом этапе (приведя их в третью нормальную форму).

Методы приведения к 3NF:

- Удаление полей не зависящих от ключа

Домашнее задание:

- Подготовить примеры нормализаций для 1, 2 и 3й нормальных форм