

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Р.Е.Алексеева

Кафедра «Информатика и системы управления»

БАЗЫ ДАННЫХ

НОРМАЛЬНЫЕ ФОРМЫ ОТНОШЕНИЙ

Методические указания к лабораторной работе №1 для студентов
специальности 230102, 230201

Нижний Новгород 2010

Составитель Балашова Т.И.

ББК

БАЗЫ ДАННЫХ. НОРМАЛЬНЫЕ ФОРМЫ ОТНОШЕНИЙ: учебно-методические указания к лабораторной работе №1 для студентов специальности 230102, 230201/ НГТУ; сост.: Балашова Т.И. Н.Новгород 2010. 9с.

В учебно-методических материалах изложены общие понятия работы с нормальными формами отношений, алгоритм нормализации отношений, его преимущества и недостатки

Подписано к печати . Формат 60x84 1/16. Бумага офсетная.
Печать офсетная. Усл.печ.л. Уч.-изд. л. .Тираж экз. Заказ .

Нижегородский государственный технический университет
им. Р.Е. Алексеева
Типография НГТУ им. Р.Е. Алексеева
Адрес университета и полиграфического предприятия:
603950, Нижний Новгород, ул. Минина, 24.

©Нижегородский государственный
технический университет
им. Р.Е.Алексеева, 2010

Цель работы: изучить нормальные формы отношений, научиться приводить отношения в соответствии с ними.

Краткие сведения из теории

1. Общие сведения

Сущность - это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Примерами сущностей могут быть такие классы объектов, как «Поставщик», «Сотрудник», «Накладная».

Экземпляр сущности - это конкретный представитель данной сущности. Например, представителем сущности «Сотрудник» может быть «Сотрудник Иванов». Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Примерами атрибутов сущности «Сотрудник» могут быть такие атрибуты, как «Табельный номер», «Фамилия», «Имя», «Отчество», «Должность», «Зарплата» и т.п.

Ключ сущности - это избыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Избыточность заключается в том, что удаление любого атрибута из ключа нарушает его уникальность. Сущность может иметь несколько различных ключей.

Первичный ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности (строку таблицы). Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

Внешний ключ - это атрибут или набор атрибутов зависимой (оперативной) таблицы, с помощью которого (которых) производится ссылка на первичный ключ другой таблицы (справочника), т.е. производится связывание таблиц. При этом справочником может быть как независимая, так и зависимая таблица – цепочка связей между таблицами ничем не ограничивается и может иметь чрезвычайно сложную структуру.

Связь - это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с нею. Например, связи между сущностями могут выражаться следующими фразами: «СОТРУДНИК может иметь несколько ДЕТЕЙ», «каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ». Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: «иметь», «принадлежать» и т.п. Каждое из наименований относится к своему концу связи.

Иногда наименования не пишутся ввиду их очевидности. Каждая связь может иметь один из следующих типов:

- связь типа «**один – к – одному**» означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь «**один – к – одному**» чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

- связь типа «**один – ко – многим**» означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны «один») называется родительской, правая (со стороны «многo») - дочерней.

- связь типа «**многие – ко – многим**» означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи «**многие – ко – многим**» является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа «**один – ко – многим**» путем создания промежуточной сущности.

Ключ связи — набор ключей сущностей, соединяемых данной связью.

2. Нормализация БД

Нормализация — разбиение исходного отношения на два или несколько, обладающих лучшими свойствами при включении, изменении и удалении данных.

Целью нормализации является получение БД, где каждый факт появляется только один раз, т.е. исключена избыточность, причём к этому стремятся не столько ради экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Нормализация выражается в приведении отношений в соответствии с некоторыми требованиями, называемыми нормальными формами. Сегодня известны шесть нормальных форм:

1. Первая нормальная форма (1 НФ);
2. Вторая нормальная форма (2 НФ);
3. Третья нормальная форма (3 НФ);
4. Нормальная форма Бойса – Кодда (НФБК);
5. Четвёртая нормальная форма (4 НФ);
6. Пятая нормальная форма (5 НФ).

В дальнейшем будем рассматривать только первые четыре нормальные формы.

3. Нормальные формы отношений

Будем рассматривать нормальные формы отношений на примере конкретной БД.

Пусть требуется спроектировать БД для сети продуктовых магазинов, в которой будут храниться данные о поступлениях товаров в магазины и их продажах.

Исходные данные:

1. Товар производится производителем. При этом: у каждого товара обязательно есть производитель, и каждый производитель производит хотя бы один товар.

2. У товара есть тип. Каждый товар обязательно относится к какому-либо типу, причем, только к одному. В магазине могут быть в наличии товары не всех типов. Товаров одного типа может быть несколько.

3. Экземпляр товара находится в магазине или продан. Экземпляр товара обязательно находится в каком-то магазине, и в каждом магазине обязательно есть товары. Каждый экземпляр товара может быть только в одном магазине, при этом в каждом магазине может быть множество экземпляров товаров.

Составим теперь универсальное отношение – отношение, включающее все атрибуты и содержащее все данные БД. Таблица будет иметь 5 полей: «Производитель» со всеми данными о производителе, «Магазин» со всеми данными о магазине, «Тип товара», «Товар», «Экземпляр товара».

С учётом полученных сведений можно заключить, что будет иметь место дублирование данных (рис. 1,б), равно как и отсутствие значений некоторых полей в записях.

Универсальное отношение
Производитель
Магазин
Тип Товара
Товар
Экземпляр товара

а - универсальное отношение

Производитель	Магазин	Тип Товара	Товар	Экземпляр товара
Kraft Foods	Eurospar	Конфеты	Alpen Gold Composition	Alpen Gold Composition ассорти из тем
Kraft Foods	Eurospar	Конфеты	Alpen Gold Composition	Alpen Gold Composition ассорти из мо
Kraft Foods	Eurospar	Конфеты	Alpen Gold Composition	Alpen Gold Composition С ликером Ам
Kraft Foods	Eurospar	Конфеты	Alpen Gold Composition	Alpen Gold Composition Вишня в конь
Kraft Foods	Eurospar	Конфеты	Alpen Gold Composition	Alpen Gold Composition конфеты с ко
*				

б- дублирование данных

Рис.1. дублирование данных в универсальном отношении;
а- универсальное отношение; б- дублирование данных

Теперь рассмотрим процесс нормализации отношений и нормальные формы.

3.1. Первая нормальная форма

Все атрибуты отношения простые, то есть их значения неделимы. В нашем случае все атрибуты не являются простыми. Так, например, данные о производителе можно разделить на название, телефон, адрес и т.д. Иными словами, первая нормальная форма содержит правило об исключении повторяющихся групп, т.е. атрибут некоторого экземпляра сущности должен содержать одно значение, а не список значений. Например, атрибут «Цена» сущности «Товар» должен содержать только одно значение цены для каждого экземпляра этой сущности.

Первая НФ
Название производителя
Адрес производителя
Телефон производителя
Название магазина
Адрес магазина
Телефон магазина
Тип товара
Название товара
Количество товара
Цена товара
Дата поступления
Продано

Рис. 2. Первая НФ

После приведения отношения к первой нормальной форме можно получить, например, такое отношение (рис. 2).

3.2. Вторая нормальная форма

Каждое неключевое поле связано полной функциональной зависимостью с первичным ключом.

Полной функциональной зависимостью называется ситуация, при которой значение данного атрибута определяется значением некоторого составного атрибута и не определяется значением любой его части.

Ключ может быть составным, то есть состоять из нескольких полей. Поэтому вторая нормальная форма фактически добавляет требование о том, чтобы значение любого неключевого поля зависело от значения всего ключа и не зависело от значения его части.

В примере таким ключом могла бы быть совокупность полей: «Название производителя», «Название магазина», «Тип товара», но например, значение поля «Адрес производителя» зависит от части ключа «Название производителя».

Решением проблемы является введение дополнительного атрибута «Код», являющегося *первичным ключом*. После этого отношение будет выглядеть так, как показано на рис. 3.

Код
Название производителя
Адрес производителя
Телефон производителя
Название магазина
Адрес магазина
Телефон магазина
Тип товара
Название товара
Количество товара
Цена товара
Дата поступления
Продано

Рис. 3. 2НФ

3.3. Третья нормальная форма

Любой неключевой атрибут нетранзитивно зависит от ключа.

Транзитивная зависимость – ситуация, при которой в некоторой тройке атрибутов первый зависит от второго, второй от третьего, а третий от первого.

В рассмотренном отношении транзитивные зависимости присутствуют.

Исключить транзитивные зависимости можно, разбив данное отношение на пять отношений: «Производитель», «Магазин», «Тип товара», «Товар» и «Экземпляр товара». При этом в каждое из них следует добавить поле, являющееся первичным ключом отношения, например, поле с именем «Код».

Кроме того, чтобы данные не потеряли связи между собой, нужно добавить в таблицы вторичные ключи. Этими ключами будут: в таблице «Товар» - «Производитель», «Тип»; в таблице «Экземпляр товара» - «Товар». В результате получим схему данных, изображенную на рис. 4.

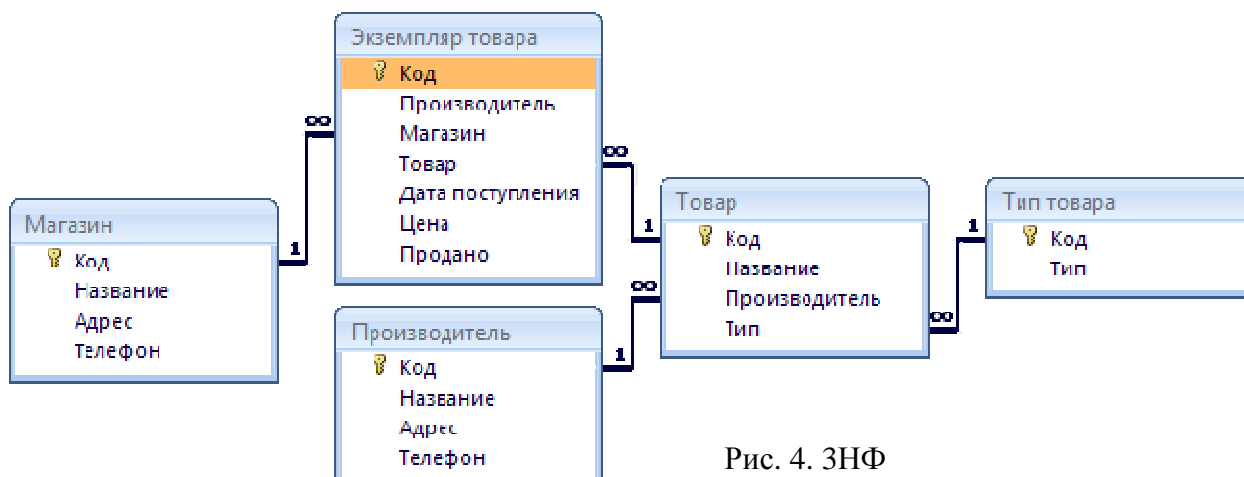


Рис. 4. ЗНФ

Иными словами, третья нормальная форма содержит правило о том, что каждый атрибут, не являющийся ключевым, должен зависеть только от ключа сущности и не зависеть от других атрибутов. Требование о независимости атрибутов друг от друга означает, что изменение одного атрибута, не входящего в ключ сущности, никак не отразится на другом.

3.4. Третья усиленная нормальная форма (*Нормальная форма Бойса – Кодда*)

Каждый детерминант отношения является возможным ключом. Иначе говоря, каждое поле, по значению которого можно однозначно идентифицировать запись, по требованиям соответствует первичному ключу, хотя может им и не являться. Требования НФБК наши отношения не нарушают.

Примечание: каждая последующая нормальная форма является недостижимой, пока не будут достигнуты все предыдущие.

Моделирование структуры базы данных при помощи алгоритма нормализации, имеет серьезные **недостатки**:

1. Первоначальное размещение всех атрибутов в одном отношении является очень неестественной операцией. Интуитивно разработчик сразу проектирует несколько отношений в соответствии с обнаруженными сущностями. Даже если создать одно или несколько отношений, включив в них все предполагаемые атрибуты, то совершенно неясен смысл полученного отношения.

2. Невозможно сразу определить полный список атрибутов. Пользователи имеют привычку называть разными именами одни и те же вещи, или наоборот, называть одними именами разные вещи.

3. Для проведения процедуры нормализации необходимо выделить зависимости атрибутов, что тоже очень нелегко, так как необходимо *явно* *выписать все зависимости*, даже те, которые являются очевидными.

Задание: составить описание отношений проектируемой базы данных согласно алгоритму нормализации. Привести отношения к нормальной форме Бойса – Кодда.

Контрольные вопросы:

1. Понятие первичного и вторичного ключей.
2. Что такое нормализация? Каковы её цели?
3. Сколько известно нормальных форм? По какому принципу они строятся?
4. Что такое первая нормальная форма?
5. Что такое вторая нормальная форма?
6. Что такое третья нормальная форма?
7. Что такое нормальная форма Бойса – Кодда?
8. Недостатки моделирования структуры БД с помощью алгоритма нормализации.