



Нормализация отношений

Важным этапом создания реляционной базы данных является **нормализация отношения**.

Под ***нормализацией*** отношения подразумевается процесс декомпозиции (разбиения) исходных отношений БД на другие, более мелкие и простые отношения. При этом устанавливаются все возможные функциональные зависимости.

Зачем нужна нормализация

Для обеспечения достоверности и непротиворечивости хранимых данных каждый факт должен храниться в БД один-единственный раз. Причем эти свойства не должны утрачиваться в процессе работы с данными.

Для поддержания БД в таком устойчивом состоянии используется ряд механизмов, которые получили обобщенное название ***средств поддержки целостности***. К таким средствам, в частности, относится и нормализация.



Всего в реляционной теории насчитывается 6 НФ:

1. Первая нормальная форма (обычно обозначается 1НФ).
2. 2НФ
3. 3НФ
4. Нормальная форма Бойса-Кодда (НФБК).
5. 4НФ.
6. 5НФ.

На практике, как правило, ограничиваются третьей нормальной формой, ее оказывается вполне достаточно для создания надежной схемы БД.

Первая нормальная форма (1НФ)

Простой атрибут - атрибут, значения которого атомарны (неделимы).

Сложный атрибут - получается соединением нескольких атомарных атрибутов.

Отношение находится в первой нормальной форме, если все его атрибуты простые.

Это означает, что каждый атрибут отношения должен хранить одно-единственное значение и не являться ни списком, ни множеством значений.

Следует заметить, что однозначно определить понятие неделимости зачастую оказывается довольно затруднительно: **атрибут, который является атомарным в одном приложении, может оказаться составным в другом.**

Простейший пример: в БД отдела кадров предприятия в таблице, хранящей личные сведения о сотрудниках, имеется атрибут **"домашний адрес"**, в котором адрес хранится в формате: город, улица, дом, квартира. В данном случае адрес хранится в виде единой текстовой строки, поскольку маловероятно, чтобы потребовалось выбрать сотрудников, скажем, по номеру квартиры. Таким образом, в контексте БД отдела кадров адрес является атомарным понятием, и его деление на составные части не имеет смысла, т.к. только внесет в БД излишнюю громоздкость. Однако тот же адрес для приложения, предназначенного для сортировки почты в почтовом отделении, атомарным не является, поскольку желательно сгруппировать конверты в отдельные стопки по улицам, так как каждую улицу обслуживает свой почтальон. Кроме того, с целью оптимизации перемещений почтальона в пределах улицы, каждую стопку желательно отсортировать по номерам домов.

Приведение отношения к 1НФ

Для этого необходимо просмотреть схему отношения и разделить составные атрибуты на различные строки/столбцы. Повторить эту операцию несколько раз до тех пор, пока каждый из атрибутов не станет атомарным.

Пример.

СТУДЕНТ

Фамилия	Курс	Специальность	Спорт	
			Вид	Разряд
Иванов	2	Математика	плавание	м. с.
Петров	4	физика	футбол	к. м.с.
Сидоров	3	экономика	шахматы	1 разряд

Очевидно, что в данном случае атрибут "Спорт" является **СЛОЖНЫМ**.

Приведем это отношение к 1НФ, то есть избавимся от сложного атрибута:

СТУДЕНТ

Фамилия	Вид спорта	Курс	Специальность	Спорт_разряд
Иванов	плавание	2	математика	м. с.
Петров	футбол	4	физика	к. м.с.
Сидоров	шахматы	3	экономика	1 разряд

Теперь в отношении СТУДЕНТ все атрибуты простые, следовательно это отношение находится в 1НФ.

Другой пример. Пусть в БД имеется таблица, в которой хранятся следующие сведения:

Табл.1

Наименование агента	Город	Адрес	Вид агента	Контактные лица
Поршневой з-д	Владимир	Ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И., зам. дир., тел (3254)76-15-95 Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	Ул. Гоголя, 25	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владимир	Ул. Пушкинская, 37, оф. 565	Клиент	Васильев В.В., директор, тел (3254)74-57-45

Очевидно, что в данном случае атрибут "**контактные лица**" не является атомарным, поскольку в нем попадают списки из нескольких лиц.

Разделим эти строки таким образом, чтобы каждая строка содержала данные только об одном лице:

Табл.2

Наименование агента	Город	Адрес	Вид агента	Контактные лица
Поршневой з-д	Владимир	Ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И., зам. дир., тел (3254)76-15-95
Поршневой з-д	Владимир	Ул. 2-я Кольцевая, 17	Поставщик	Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	Ул. Гоголя, 25	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владимир	Ул. Пушкинска я, 37, оф. 565	Клиент	Васильев В.В., директор, тел (3254)74-57-45

Несколько лучше, хотя при ближайшем рассмотрении оказывается, что атрибут "контактные лица" снова может быть назван атомарным лишь с натяжкой, поскольку содержит разнородные данные, хотя и об одном лице.

Разобьем атрибут «контактные лица» на несколько атрибутов:

Табл.3

Наименование агента	Город	Адрес	Вид агента	Фамилия	Должность	Телефон
Поршневой з-д	Владими р	Ул. 2-я Кольцевая, 17	Поставщик	Иванов И.И.	зам. директора	(3254)76-15-95
Поршневой з-д	Владими р	Ул. 2-я Кольцевая, 17	Поставщик	Петров П.П.	нач. отд. сбыта	(3254)76-15-35
ООО Вымпел	Курск	Ул. Гоголя, 25	Клиент	Сидоров С.С.	директор	(7634)66-65-38
ИЧП Альфа	Владими р	Ул. Пушкинская, 37, оф. 565	Клиент	Васильев В.В.	директор	(3254)74-57-45

Теперь можем считать, что каждое значение каждого из атрибутов нашего отношения является атомарным и, следовательно, **отношение находится в 1НФ.**

Вторая нормальная форма (2НФ)

Определение. Отношение находится во второй нормальной форме, если оно находится в 1НФ и каждый неключевой атрибут функционально полно зависит от составного ключа.

Замечание: вышесказанное относится к отношениям с **составным ключом**. Отношение с простым ключом (состоящим из единственного атрибута), приведенное к 1НФ, находится во 2НФ по определению и в данном этапе нормализации не нуждается.

Приведение отношения к 2НФ

Для иллюстрации используем отношение ПРЕПОДАВАТЕЛЬ-ПРЕДМЕТ с составным ключом «Личный номер» и «Название предмета» из предыдущей лекции

ПРЕПОДАВАТЕЛЬ - ПРЕДМЕТ

Личный номер	Название предмета	Кол-во часов	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Информатика	36	Фролов	доцент	3900	ПМ	23-33-15
201	ИТ	24	Фролов	доцент	3900	ПМ	23-33-15
202	Физика	48	Костин	профессор	5600	Физика	23-45-19
401	Экономика	36	Глазов	ассистент	1999	Экономика	23-56-90
401	Бухучет	16	Глазов	ассистент	1999	Экономика	23-56-90

Отметим, что:

Оно находится в 1НФ, поскольку не содержит составных атрибутов.

В этом отношении можно отметить частичную зависимость атрибутов «Фамилия», «Должность», «Оклад», «Кафедра», «Телефон» от части «Личный номер» составного ключа.

Такая частичная зависимость приводит к следующим проблемам:

1. Имеет место дублирование данных о преподавателе, поскольку преподаватель может читать несколько предметов.
2. Существует проблема контроля избыточности данных, так как изменение, например, оклада влечет за собой необходимость поиска и изменения значений окладов во всех записях с данным преподавателем.
3. Возникает проблема с преподавателями, которые в данное время не ведут предметы, а именно, преподавателя без предмета невозможно включить в отношение и наоборот, если преподаватель увольняется и удаляется из отношения, то будет удален и предмет, хотя предмет должен продолжать читаться.

Личный номер	Название предмета	Кол-во часов	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Информатика	36	Фролов	доцент	3900	ПМ	23-33-15
201	ИТ	24	Фролов	доцент	3900	ПМ	23-33-15
202	Физика	48	Костин	профессор	5600	Физика	23-45-19
401	Экономика	36	Глазов	ассистент	1999	Экономика	23-56-90
401	Бухучет	16	Глазов	ассистент	1999	Экономика	23-56-90

Чтобы устранить частичную зависимость и привести рассматриваемое отношение к 2НФ, необходимо разбить его на два отношения:

В итоге получим два отношения – ПРЕДМЕТ и ПРЕПОДАВАТЕЛЬ, находящиеся в 2НФ.

ПРЕДМЕТ

Личн ый номер	Название предмета	Кол- во часов
201	Информатика	36
201	ИТ	24
202	Физика	48
401	Экономика	36
401	Бухучет	16

ПРЕПОДАВАТЕЛЬ

Личн ый номер	Фамилия	Должность	Оклад	Кафедра	Телефон
201	Фролов	доцент	3900	ПМ	23-33-15
201	Фролов	доцент	3900	ПМ	23-33-15
202	Костин	профессор	5600	Физика	23-45-19
401	Глазов	ассистент	1999	Экономика	23-56-90
401	Глазов	ассистент	1999	Экономика	23-56-90


Третья нормальная форма (3НФ)

Определение. Отношение находится в **третьей нормальной форме**, если оно находится в 2НФ и в нем отсутствуют транзитивные зависимости неключевых атрибутов от ключа.

В полученном отношении ПРЕПОДАВАТЕЛЬ имеются **транзитивные функциональные зависимости**, например:

Личный номер → Кафедра → Телефон

Личный номер → Должность → Оклад



Наличие транзитивных зависимостей порождает следующие неудобства (на примере атрибута «Телефон»):

1. Имеет место дублирование информации о телефоне для преподавателей одной кафедры, это порождает проблему контроля избыточности, поскольку изменение номера телефона кафедры влечет за собой необходимость поиска и изменения номеров всех преподавателей этой кафедры.
2. Нельзя включить данные о новой кафедре (название и номер телефона), если на данный момент еще отсутствуют преподаватели.

Для решения этих проблем необходимо разбить отношение ПРЕПОДАВАТЕЛЬ на три следующих отношения:

ПРЕПОДАВАТЕЛЬ

Личный номер	Фамилия	Должность	Кафедра
201	Фролов	1	1
202	Костин	2	2
401	Глазов	3	3

КАФЕДРА


Номер	Кафедра	Телефон
1	ПМ	23-33-15
2	Физика	23-45-19
3	Экономика	23-56-90

ДОЛЖНОСТЬ

Номер	Должность	Оклад
1	доцент	3900
2	профессор	5600
3	ассистент	1999

Каждое из этих отношений находится в 3НФ

3 НФ освобождает от избыточности данных и аномалий выполнения операций включения, удаления и изменения записей.



Резюме: нормализация увеличивает число отношений в БД и тем самым может возрасти время обработки данных. Но благодаря корректности и устранению дублирования данных ускоряется выполнение операций доступа к данным.

На практике дело обычно обстоит не так. Во-первых, вряд ли кому придет в голову начинать проектирование базы данных с таблиц, нарушающих 1НФ (возможно, за исключением случаев, когда нужно занести в БД данные, подготовленные табличным процессором вроде Excel).

С составными ключами на практике тоже доводится встречаться не так уж часто. Обычно составной ключ подменяют ключом искусственного происхождения, который обычно генерируется самой СУБД (например, тип Счетчик), которая и гарантирует его уникальность.

Таким образом, на практике приводить отношения к 1НФ и 2НФ приходится не столь уж часто. Скорее вполне достаточно оказывается проглядеть отношения, чтобы убедиться, что они не нарушают эти нормальные формы.

С 3НФ ситуация несколько сложнее, так как транзитивная зависимость не всегда так явно бросается в глаза. Поэтому нарушение 3НФ - наиболее частая проблема, с которой приходится иметь дело.