

## Теория проектирования реляционных баз данных

При проектировании базы данных решаются две основные проблемы:

1. Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области, и было, по возможности, лучшим (эффективным, удобным и т. д.)? Часто эту проблему называют проблемой логического проектирования баз данных.
2. Как обеспечить эффективность выполнения запросов к базе данных? Эту проблему обычно называют проблемой физического проектирования баз данных.

В случае реляционных баз данных нет общих рецептов по части физического проектирования. Здесь слишком много зависит от используемой СУБД. Поэтому ограничимся только существенными вопросами логического проектирования реляционных баз данных. Более того, не будем касаться определения ограничений целостности общего вида, а ограничимся ограничениями первичного и внешнего ключей. Будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том, из каких отношений должна состоять база данных, и какие атрибуты должны быть у этих отношений.

Классический подход к проектированию реляционных баз данных заключается в том, что сначала предметная область представляется в виде одного или нескольких отношений, а далее осуществляется процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая. Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером набора ограничений является ограничение первой нормальной формы – значения всех атрибутов отношения атомарны. Поскольку требование первой нормальной формы является базовым требованием классической реляционной модели данных, будем считать, что исходный набор отношений уже соответствует этому требованию.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

1. первая нормальная форма (1НФ или 1NF);
2. вторая нормальная форма (2НФ или 2NF);
3. третья нормальная форма (3НФ или 3NF);
4. нормальная форма Бойса-Кодда (НФБК или BCNF);
5. четвертая нормальная форма (4НФ или 4NF);
6. пятая нормальная форма, или нормальная форма проекции-соединения (5НФ или 5NF или PJ/NF).

Основные свойства нормальных форм такие:

1. каждая следующая нормальная форма в некотором смысле лучше предыдущей;
2. при переходе к следующей нормальной форме свойства предыдущих нормальных свойств сохраняются.

Процесс проектирования реляционной базы данных на основе метода нормализации преследует две основные цели:

1. избежать избыточности хранения данных;
2. устранить аномалии обновления отношений.

Эти цели являются актуальными для информационных систем оперативной обработки транзакций (On-Line Transaction Processing – OLTP), которым свойственны частые обновления базы данных, и потому аномалии обновления могут сильно вредить

эффективности приложения. В информационных системах оперативной аналитической обработки (On-Line Analytical Processing – OLAP), в частности, в системах поддержки принятия решений, базы данных в основном используются для выборки данных. Поэтому аномалиями обновления можно пренебречь. Из этого не следует, что принципы нормализации непригодны при проектировании баз данных OLAP-приложений. Даже если схема такой базы данных должна быть денормализована по соображениям эффективности, то чтобы получить правильную денормализованную схему, нужно сначала понять, как выглядит нормализованная схема.

В основе метода нормализации лежит декомпозиция отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих требованиям следующей нормальной формы. Считаются правильными такие декомпозиции отношения, которые обратимы, т. е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации.

Наиболее важные на практике нормальные формы отношений основываются на фундаментальном в теории реляционных баз данных понятии функциональной зависимости.

### Декомпозиция без потерь

Декомпозицией отношения  $R$  называется замена  $R$  на совокупность отношений  $\{R_1, R_2, \dots, R_n\}$  такую, что каждое из них есть проекция  $R$ , и каждый атрибут  $R$  входит хотя бы в одну из проекций декомпозиции.

Например, для отношения  $R$  с атрибутами  $\{a, b, c\}$  существуют следующие основные варианты декомпозиции:

- $\{a\}, \{b\}, \{c\}$
- $\{a\}, \{b, c\}$
- $\{a, b\}, \{c\}$
- $\{b\}, \{a, c\}$
- $\{a, b\}, \{b, c\}$
- $\{a, b\}, \{a, c\}$
- $\{b, c\}, \{a, c\}$
- $\{a, b\}, \{b, c\}, \{a, c\}$

Рассмотрим теперь отношение  $R'$ , которое получается в результате операции естественного соединения (NATURAL JOIN), применённой к отношениям, полученным в результате декомпозиции  $R$ .

Декомпозиция называется декомпозицией без потерь, если  $R'$  в точности совпадает с  $R$ .

Неформально говоря, при декомпозиции без потерь отношение «разделяется» на отношения-проекции таким образом, что из полученных проекций возможна «сборка» исходного отношения с помощью операции естественного соединения.

Далеко не всякая декомпозиция является декомпозицией без потерь. Проиллюстрируем это на примере отношения  $R$  с атрибутами  $\{a, b, c\}$ , приведённом выше. Пусть отношение  $R$  имеет вид:

<b>a</b>	<b>b</b>	<b>c</b>
Москва	Россия	столица
Томск	Россия	не столица
Берлин	Германия	столица

Декомпозиция  $R_1 = \{a\}$ ,  $R_2 = \{b, c\}$  имеет вид:

<b>a</b>	<b>b</b>	<b>c</b>
Москва	Россия	столица
Томск	Россия	не столица
Берлин	Германия	столица

Результат операции соединения этих отношений:  $R' = R_1 \text{ NATURAL JOIN } R_2$

<b>a</b>	<b>b</b>	<b>c</b>
Москва	Россия	столица
Москва	Россия	не столица
Москва	Германия	столица
Томск	Россия	столица
Томск	Россия	не столица
Томск	Германия	столица
Берлин	Россия	столица
Берлин	Россия	не столица
Берлин	Германия	столица

Очевидно, что  $R'$  не совпадает с  $R$ , а значит такая декомпозиция не является декомпозицией без потерь. Рассмотрим теперь декомпозицию  $R_1 = \{a, b\}$ ,  $R_2 = \{a, c\}$ :

<b>a</b>	<b>b</b>	<b>a</b>	<b>c</b>
Москва	Россия	Москва	столица
Томск	Россия	Томск	не столица
Берлин	Германия	Берлин	столица

Такая декомпозиция является декомпозицией без потерь, в чём читатель может убедиться самостоятельно.

В некоторых случаях отношение вообще невозможно декомпозировать без потерь. Существуют также примеры отношений, для которых нельзя выполнить декомпозицию без потерь на две проекции, но которые можно подвергнуть декомпозиции без потерь на три или большее количество проекций.

### Первая нормальная форма

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице. Например, есть таблица «Автомобили»:

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Нарушение нормализации 1НФ происходит в моделях BMW, т.к. в одной ячейке содержится список из 3 элементов: M5, X5M, M1, т.е. он не является атомарным. Преобразуем таблицу к 1НФ:

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

### Вторая нормальная форма

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

Например, дана таблица:

Модель	Фирма	Цена	Скидка
M5	BMW	5500000	5 %
X5M	BMW	6000000	5 %
M1	BMW	2500000	5 %
GT-R	Nissan	5000000	10 %

Таблица находится в первой нормальной форме, но не во второй. Цена машины зависит от модели и фирмы. Скидка зависит от фирмы, то есть зависимость от первичного ключа неполная. Исправляется это путем декомпозиции на два отношения, в которых не ключевые атрибуты зависят от ПК.

Модель	Фирма	Цена	Фирма	Скидка
M5	BMW	5500000	BMW	5 %
X5M	BMW	6000000	Nissan	10 %
M1	BMW	2500000		
GT-R	Nissan	5000000		

### Третья нормальная форма

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Рассмотрим таблицу:

Модель	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

Таблица находится во 2НФ, но не в 3НФ.

В отношении атрибут «Модель» является первичным ключом. Личных телефонов у автомобилей нет, и телефон зависит исключительно от магазина.

Таким образом, в отношении существуют следующие функциональные зависимости: Модель → Магазин, Магазин → Телефон, Модель → Телефон.

Зависимость Модель → Телефон является транзитивной, следовательно, отношение не находится в 3НФ.

В результате разделения исходного отношения получаются два отношения, находящиеся в 3НФ:

Магазин	Телефон	Модель	Магазин
Риал-авто	87-33-98	BMW	Риал-авто
Некст-Авто	94-54-12	Audi	Риал-авто
		Nissan	Некст-Авто

### Нормальная форма Бойса-Кодда (НФБК) (частная форма третьей нормальной формы)

Определение 3НФ не совсем подходит для следующих отношений:

- 1) отношение имеет два или более потенциальных ключа;
- 2) два и более потенциальных ключа являются составными;
- 3) они пересекаются, т.е. имеют хотя бы один общий атрибут.

Для отношений, имеющих один потенциальный ключ (первичный), НФБК является 3НФ.

Отношение находится в НФБК, когда каждая нетривиальная и неприводимая слева функциональная зависимость обладает потенциальным ключом в качестве детерминанта.

Предположим, рассматривается отношение, представляющее данные о бронировании стоянки на день:

Номер стоянки	Время начала	Время окончания	Тариф
1	09:30	10:30	Бережливый
1	11:00	12:00	Бережливый
1	14:00	15:30	Стандарт
2	10:00	12:00	Премиум-В
2	12:00	14:00	Премиум-В
2	15:00	18:00	Премиум-А

Тариф имеет уникальное название и зависит от выбранной стоянки и наличии льгот, в частности:

- «Бережливый»: стоянка 1 для льготников
- «Стандарт»: стоянка 1 для не льготников
- «Премиум-А»: стоянка 2 для льготников
- «Премиум-В»: стоянка 2 для не льготников.

Таким образом, возможны следующие составные первичные ключи: {Номер стоянки, Время начала}, {Номер стоянки, Время окончания}, {Тариф, Время начала}, {Тариф, Время окончания}.

Отношение находится в 3НФ. Требования второй нормальной формы выполняются, так как все атрибуты входят в какой-то из потенциальных ключей, а неключевых атрибутов в отношении нет. Также нет и транзитивных зависимостей, что соответствует требованиям третьей нормальной формы. Тем не менее, существует функциональная зависимость Тариф → Номер стоянки, в которой левая часть (детерминант) не является потенциальным ключом отношения, то есть отношение не находится в нормальной форме Бойса — Кодда.

Недостатком данной структуры является то, что, например, по ошибке можно приписать тариф «Бережливый» к бронированию второй стоянки, хотя он может относиться только к первой стоянке.

Можно улучшить структуру с помощью декомпозиции отношения на два и добавления атрибута Имеет льготы, получив отношения, удовлетворяющие НФБК (подчёркнуты атрибуты, входящие в первичный ключ.):

Тариф	Время начала	Время окончания
Бережливый	09:30	10:30
Бережливый	11:00	12:00
Стандарт	14:00	15:30
Премиум-В	10:00	12:00
Премиум-В	12:00	14:00
Премиум-А	15:00	18:00

Тариф	Номер	Имеет
Бережливый	1	Да
Стандарт	1	Нет
Премиум-А	2	Да
Премиум-В	2	Нет

#### Четвертая нормальная форма

Отношение находится в 4НФ, если оно находится в НФБК и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от ее потенциальных ключей.

В отношении R (A, B, C) существует многозначная зависимость  $R.A \twoheadrightarrow R.B$  в том и только в том случае, если множество значений B, соответствующее паре значений A и C, зависит только от A и не зависит от C.

Предположим, что рестораны производят разные виды пиццы, а службы доставки ресторанов работают только в определенных районах города. Составной первичный ключ соответствующей переменной отношения включает три атрибута: {Ресторан, Вид пиццы, Район доставки}.

Ресторан	Вид пиццы	Район доставки
A1 Пицца	Толстая корка	Springfield
A1 Пицца	Толстая корка	Shelbyville
A1 Пицца	Толстая корка	Столица
A1 Пицца	Фаршированная корочка	Springfield
A1 Пицца	Фаршированная корочка	Shelbyville
A1 Пицца	Фаршированная корочка	Столица
Элитная пицца	Толстая корка	Столица
Элитная пицца	Фаршированная корочка	Столица
Пицца Винченцо	Толстая корка	Springfield
Пицца Винченцо	Толстая корка	Shelbyville
Пицца Винченцо	Толстая корка	Springfield
Пицца Винченцо	Толстая корка	Shelbyville

Такая переменная отношения не соответствует 4НФ, так как существует следующая многозначная зависимость:

{Ресторан}  $\twoheadrightarrow$  {Вид пиццы}

{Ресторан}  $\twoheadrightarrow$  {Район доставки}

То есть, например, при добавлении нового вида пиццы придется внести по одному новому кортежу для каждого района доставки. Возможна логическая аномалия, при которой определенному виду пиццы будут соответствовать лишь некоторые районы доставки из обслуживаемых рестораном районов.

Для предотвращения аномалии нужно декомпонировать отношение, разместив независимые факты в разных отношениях. В данном примере следует выполнить декомпозицию на {Ресторан, Вид пиццы} и {Ресторан, Район доставки}.

Однако, если к исходной переменной отношения добавить атрибут, функционально зависящий от потенциального ключа, например цену с учётом стоимости доставки ({Ресторан, Вид

пиццы, Район доставки} → Цена), то полученное отношение будет находиться в 4НФ и его уже нельзя подвергнуть декомпозиции без потерь.

Ресторан	Район доставки	Ресторан	Вид пиццы
A1 Пицца	Springfield	A1 Пицца	Толстая корка
A1 Пицца	Shelbyville	A1 Пицца	Фаршированная корочка
A1 Пицца	Столица	Элитная пицца	Толстая корка
Элитная пицца	Столица		Фаршированная корочка
Пицца Винченцо	Springfield	Пицца Винченцо	Толстая корка
Пицца Винченцо	Shelbyville		

### Пятая нормальная форма

Отношение находится в пятой нормальной форме (иначе — в проекционно-соединительной нормальной форме) тогда и только тогда, когда каждая нетривиальная зависимость соединения в нём определяется потенциальным ключом (ключами) этого отношения.

Это очень жесткое требование, которое можно выполнить лишь при дополнительных условиях. На практике трудно найти пример реализации этого требования в чистом виде. Предположим, что нужно хранить данные об ассортименте нескольких продавцов, торгующих продукцией нескольких фирм (номенклатура товаров фирм может пересекаться):

Продавец	Фирма	Товар
Иванов	Рога и Копыта	Пылесос
Иванов	Рога и Копыта	Хлебница
Петров	Безенчук&Ко	Сучкорез
Петров	Безенчук&Ко	Пылесос
Петров	Безенчук&Ко	Хлебница
Петров	Безенчук&Ко	Зонт
Сидоров	Безенчук&Ко	Пылесос
Сидоров	Безенчук&Ко	Телескоп
Сидоров	Рога и Копыта	Пылесос
Сидоров	Рога и Копыта	Лампа
Сидоров	Геркулес	Вешалка

Если дополнительных условий нет, то данное отношение, которое находится в 4-й нормальной форме, является корректным и отражает все необходимые ограничения.

Теперь предположим, что нужно учесть следующее ограничение: каждый продавец имеет в своём ассортименте ограниченный список фирм и ограниченный список типов товаров и предлагает товары из списка товаров, производимые фирмами из списка фирм.

То есть продавец не имеет право торговать какими угодно товарами каких угодно фирм. Если продавец П имеет право торговать товарами фирмы Ф, и если продавец П имеет право



торговать товарами типа Т, то в этом случае в ассортимент продавца П входят товары типа Т фирмы Ф при условии, что фирма Ф производит товары типа Т.

Такое ограничение может быть вызвано, например, тем, что список типов товаров продавца ограничен имеющимися у него лицензиями, либо знаниями и квалификацией, необходимыми для их продажи, а список фирм каждого продавца определён партнёрскими соглашениями.

В рассматриваемом примере, в частности, предполагается, что продавец Иванов имеет право торговать товарами только фирмы «Рога и Копыта», продавец Петров — товарами только фирмы «Безенчук & Ко», зато продавец Сидоров не имеет права торговать хлебницами и сучкорезами и т. д.

Предложенное выше отношение не может исключить ситуации, при которых данное ограничение будет нарушено. Ничто не препятствует занести данные о торговле товаром, который данная фирма вообще не выпускает, либо данные о торговле товарами той фирмы, которую данный продавец не обслуживает, либо данные о торговле таким типом товара, который данный продавец не имеет право продавать.

Отношение не находится в 5NF, поскольку в нём есть нетривиальная зависимость соединения  $*\{\{\text{Продавец, Фирма}\}, \{\text{Фирма, Товар}\}, \{\text{Продавец, Товар}\}\}$ , однако подмножества  $\{\text{Продавец, Фирма}\}, \{\text{Фирма, Товар}\}, \{\text{Продавец, Товар}\}$  не являются суперключами исходного отношения.

В данном случае для приведения к 5NF отношение должно быть разбито на три:  $\{\text{Продавец, Фирма}\}, \{\text{Фирма, Товар}\}, \{\text{Продавец, Товар}\}$ .

Продавец	Фирма	Фирма	Товар	Продавец	Товар
Иванов	Рога и Копыта	Рога и Копыта	Пылесос	Иванов	Пылесос
Петров	Безенчук&Ко	Рога и Копыта	Хлебница	Иванов	Хлебница
Сидоров	Безенчук&Ко	Рога и Копыта	Лампа	Петров	Сучкорез
Сидоров	Рога и Копыта	Безенчук&Ко	Сучкорез	Петров	Пылесос
Сидоров	Геркулес	Безенчук&Ко	Пылесос	Петров	Хлебница
		Безенчук&Ко	Хлебница	Петров	Зонт
		Безенчук&Ко	Зонт	Сидоров	Телескоп
		Безенчук&Ко	Телескоп	Сидоров	Пылесос
		Геркулес	Вешалка	Сидоров	Лампа
				Сидоров	Вешалка

## Шестая нормальная форма

Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6НФ, также находится и в 5НФ.

Идея «декомпозиции до конца» выдвигалась до начала исследований в области хронологических данных, но не нашла поддержки. Однако для хронологических баз данных максимально возможная декомпозиция позволяет бороться с избыточностью и упрощает поддержание целостности базы данных.

Для хронологических баз данных определены U\_операторы, которые распаковывают отношения по указанным атрибутам, выполняют соответствующую операцию и упаковывают полученный результат. В данном примере соединение проекций отношения должно производится при помощи оператора U\_JOIN.

Таб.№	Время	Должность	Домашний адрес
6575	01-01-2000:10-02-2003	слесарь	ул.Ленина,10
6575	11-02-2003:15-06-2006	слесарь	ул.Советская,22
6575	16-06-2006:05-03-2009	бригадир	ул.Советская,22

Переменная отношения «Работники» не находится в БНФ и может быть подвергнута декомпозиции на переменные отношения «Должности работников» и «Домашние адреса работников».

Таб.№	Время	Должность	Таб.№	Время	Домашний адрес
6575	01-01-2000: 10-02-2003	слесарь	6575	01-01-2000: 10-02-2003	ул.Ленина,10
6575	16-06-2006: 05-03-2009	бригадир	6575	11-02-2003: 15-06-2006	ул.Советская, 22