

3.3.1. ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА



Исторически первая нормальная форма была создана для запрета создания многозначных и составных атрибутов и их комбинаций (т.н. «таблиц внутри таблиц»). Поскольку сегодня ни одна реляционная СУБД не позволит создать «таблицу внутри таблицы», акцент в определении первой нормальной формы смещается в сторону более глубокой трактовки понятия «атомарность».



Переменная отношения находится в **первой нормальной форме** (1НФ, 1NF¹⁶⁷) тогда и только тогда, когда каждый атрибут отношения содержит строго одно атомарное значение¹⁶⁸.

Упрощённо: каждый атрибут отношения атомарен (т.е. СУБД не должна оперировать никакой отдельной частью атрибута).

Рассмотрим примеры нарушения первой нормальной формы (см. рисунок 3.3.а).

Как только что было отмечено, ни одна современная реляционная СУБД не позволит в качестве типа поля таблицы указать «таблица», т.е. создать «таблицу внутри таблицы». И пусть объектно-реляционные СУБД позволяют в качестве значений атрибута использовать сложные структуры (фактически, объекты), мы вернёмся к классическим реляционным базам данных и рассмотрим подробнее случаи многозначных и составных атрибутов.

На рисунке 3.3.а многозначный атрибут **s_mark** представляет собой список значений оценок. Очевидно, что при выполнении таких типичных операций как поиск минимальной, максимальной и средней оценки СУБД будет вынуждена анализировать содержимое данного атрибута в каждой записи — иными словами, атрибут не атомарен.

¹⁶⁷ A relvar is in 1NF if and only if in every legal value of that relvar every tuple contains exactly one value for each attribute. (C.J. Date, “An Introduction to Database Systems”, 8th edition)

¹⁶⁸ Некоторые авторы явно добавляют к этому определению необходимость наличия у отношения первичного ключа (собственно, потому 1НФ, 2НФ, 3НФ, НФБК и называются «нормальными формами на основе первичных ключей»), но в подавляющем большинстве первоисточников этого требования нет. В любом случае, такие тонкости представляют скорее научный, а не практический интерес.

student

| PK s_id | s_result | |
|------------|--------------------|-----------------|
| 1731 | subject | mark |
| | Химия | 8 |
| | Физика | 9 |
| 1824 | ... | |

Комбинация вариантов
«многозначный атрибут» и
«составной атрибут»
(фактически, «таблица внутри
таблицы»)

student

| PK s_id | s_mark |
|------------|----------------------------------|
| 1731 | {3, 6, 8, 6, 4, 6, 7} |
| 1824 | ... |

Многозначный
атрибут

student

| PK s_id | s_address |
|------------|--|
| 1731 | г. Большой, ул. Новая, дом. 1 |
| 1824 | ... |

Составной
атрибут

Рисунок 3.3.a — *Нарушения первой нормальной формы*

Для приведения к 1НФ отношения с многозначными атрибутами необходимо вынести каждый такой атрибут в отдельное отношение и связать его с исходным отношением связью «один ко многим^[57]». Результат выполнения такой операции показан на рисунке 3.3.b.

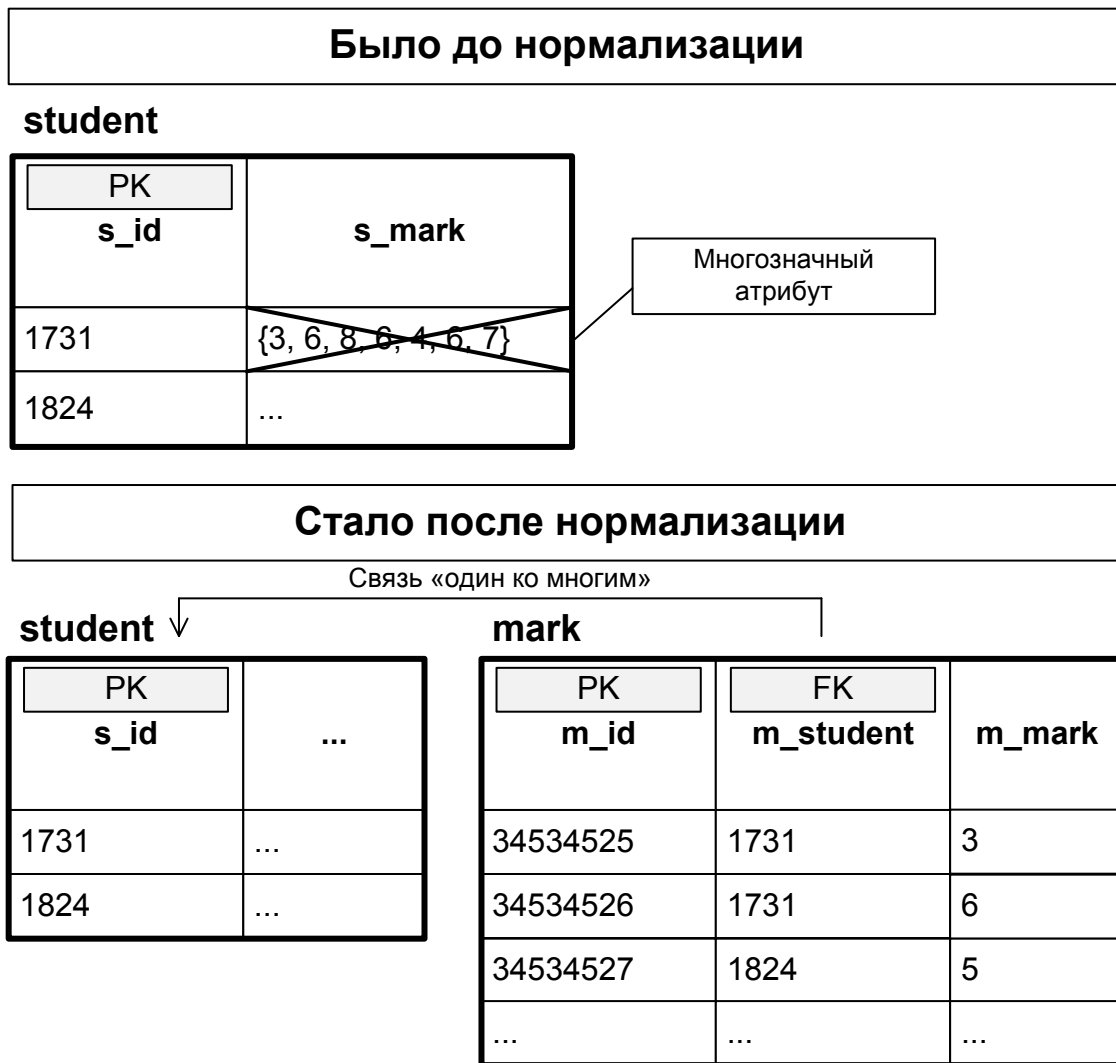


Рисунок 3.3.b — Приведение к 1НФ отношения с многозначными атрибутами

Если с многозначными атрибутами всё просто и очевидно, то с составными вопрос становится почти философским: как мы можем понять, что атрибут уже «достаточно атомарен»? В примере на рисунке 3.3.a всё достаточно тривиально: да, можно город, улицу и номер дома поместить в отдельные поля — результат выполнения такой операции показан на рисунке 3.3.c.

Было до нормализации

student

| PK s_id | s_address |
|------------|----------------------------------|
| 1731 | г. БОЛЬШОЙ, ул. Новая, дом. 1 |
| 1824 | ... |

Составной атрибут

Стало после нормализации

student

| PK s_id | s_addr_city | s_addr_street | s_add_building |
|------------|-------------|---------------|----------------|
| 1731 | Большой | Новая | 1 |
| 1824 | ... | ... | ... |

Рисунок 3.3.с — Приведение к 1НФ отношения с составными атрибутами

И теперь неожиданный вопрос: а нужно ли было выполнять эту операцию? Действительно ли атрибут **s_address** не был атомарным? Как правило, здесь большинство считает, что не был, и потому операция его разбиения на три отдельных атрибута оправдана.

А как насчёт следующих случаев:

- Стоит ли хранить в отдельных полях код оператора и номер телефона?
- Стоит ли хранить в отдельных полях имя пользователя и домен для адреса e-mail?
- Стоит ли хранить в отдельных полях год, месяц и день? Если да, то, может быть, стоит сохранить ещё и день недели? И номер недели в году? И признак того, выходной ли это?
- А для времени часы, минуты и секунды (а также их дробную часть) стоит хранить в отдельных полях?
- А серию и номер паспорта стоит хранить в отдельных полях?

Таких вопросов, т.е. примеров из предметных областей, можно придумать очень много. Их объединяет одно: сомнение в целесообразности разбиения исходного значения на составные части. Ведь если нам «не нужны» отдельные части (мы не заставляем СУБД их анализировать) такое разбиение как минимум бессмысленно, а как максимум ещё и вредно (нам придётся постоянно объединять эти фрагменты в единое целое при выполнении операций с данными).

Если подойти к этой ситуации бездумно, мы доведём её до абсурда и начнём каждый бит данных хранить в отдельном поле таблицы — теперь уж точно «атомарнее не бывает».

Но с практической точки зрения всё не так плохо: мы должны оценивать количество и сложность операций, при выполнении которых мы будем анализировать часть значения атрибута. Конечно, в идеале количество таких операций равно нулю, и тогда наше отношение полностью соответствует определению 1НФ. Но в реальности ради небольшого процента предельно редко и при том быстро выполняемых операций мы не станем «резать» атрибут и тем самым усложнять и замедлять все оставшиеся операции, которые оперируют значением этого атрибута как единым целым.

Так, например, дата вполне может храниться в поле типа **DATE** как единое целое, если у нас нет постоянно возникающей необходимости искать записи по принципу «все события, произошедшие в апреле и августе любого года» или «все события, произошедшие первого числа любого месяца».

Краткий вывод по первой нормальной форме:

- сделать «таблицу в таблице» вам не позволит СУБД;
- каждый многозначный атрибут обязательно нужно убрать путём вынесения его в новое отношение;
- каждый составной атрибут стоит тщательно исследовать, чтобы понять, *действительно ли* он составной (не атомарный), и, если это так, разделить его на несколько новых отдельных атрибутов.



Задание 3.3.а: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся в первой нормальной форме? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.

3.3.2. ВТОРАЯ НОРМАЛЬНАЯ ФОРМА



Если по какой-то причине вы пропустили или забыли содержимое раздела, посвящённого теории зависимостей^{174}, сейчас — самое время освежить в памяти соответствующий материал (особенно стоит перечитать определения полной^{183} и частичной^{184} функциональных зависимостей).



{Здесь приведена т.н. «слабая формулировка» определения 2НФ. Она часто используется в различных учебных пособиях, но далее будет показано, почему 2НФ стоит рассматривать в «сильной формулировке» её определения.}

Переменная отношения находится во **второй нормальной форме** (2НФ, 2NF¹⁶⁹) тогда и только тогда, когда она находится в первой нормальной форме^{241}, и каждый её первичный атрибут^{23} функционально полно^{183} зависит от первичного ключа^{39}.

Упрощённо: ни один атрибут, не входящий в состав первичного ключа, не должен функционально зависеть от части первичного ключа.

Из такой формулировки определения 2НФ следует, что если первичный ключ отношения является простым^{40}, и отношение находится в 1НФ, то оно автоматически находится и во 2НФ. Казалось бы, всё очень хорошо и просто (достаточно отношению, находящемуся в 1НФ, добавить простой искусственный^{42} первичный ключ), но существует более полное определение 2НФ.



{Здесь приведена т.н. «сильная формулировка» определения 2НФ.}

Переменная отношение находится во **второй нормальной форме** (2НФ, 2NF¹⁷⁰) тогда и только тогда, когда она находится в первой нормальной форме^{241}, и каждый её неключевой атрибут^{23} функционально полно^{183} зависит от любого потенциального ключа^{37}.

Упрощённо: ни один атрибут, не входящий в состав потенциального ключа, не должен функционально зависеть от части какого бы то ни было из потенциальных ключей.

Для начала рассмотрим пример нарушения второй нормальной формы в слабой формулировке (см. рисунок 3.3.d).

Отношение **group** описывает учебную группу, и его атрибуты означают следующее:

- **g_number** — порядковый номер группы в рамках года поступления;
- **g_start_year** — год поступления (год, когда группа начала учёбу);
- **g_years** — годы обучения (через сколько лет после начала обучения группа должна закончить учёбу);
- **g_head** — идентификатор старосты группы (внешний ключ).

Поскольку в каждый год поступления может быть сформировано несколько групп, у данного отношения есть лишь один очевидный потенциальный ключ — комбинация **{g_number, g_start_year}**, который и был выбран в качестве первичного.

Атрибут **g_years** не может быть частью потенциального ключа (мы видим, что его комбинации с атрибутами **g_number** и **g_start_year** дублируются).

¹⁶⁹ A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R. («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

¹⁷⁰ A relation schema R is in 2NF if every nonprime attribute A in R is not partially dependent on any key of R. («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

Почему мы не можем рассматривать атрибут **gr_head** (один и тот же студент не может быть старостой нескольких групп, т.е. значение этого поля будет уникальным) как ещё один потенциальный ключ? Потому, что тогда мы были бы обязаны включить для этого поля свойство **NOT NULL** и сразу бы получили аномалию вставки¹⁶² — при формировании группы необходимо было бы сразу назначить старосту, что объективно не будет выполняться в реальности в большинстве случаев.

Рассмотрим имеющиеся в представленном на рисунке 3.3.d отношении зависимости:

- Полная функциональная зависимость $\{g_number, g_start_year\} \rightarrow \{g_head\}$ показывает, что идентификатор старосты зависит от первичного ключа целиком (действительно — чтобы узнать старосту группы, необходимо гарантированно идентифицировать группу).
- Частичная функциональная зависимость $\{g_number, g_start_year\} \rightarrow \{g_years\}$ показывает, что продолжительность обучения зависит только от года поступления, но никак не зависит от номера группы в рамках года поступления. При этом мы видим, что раньше обучение занимало пять лет (для поступивших по 1999-й год включительно), а потом оно стало занимать четыре года (для поступивших в 2000-м году и позднее).

Частичная зависимость от первичного ключа

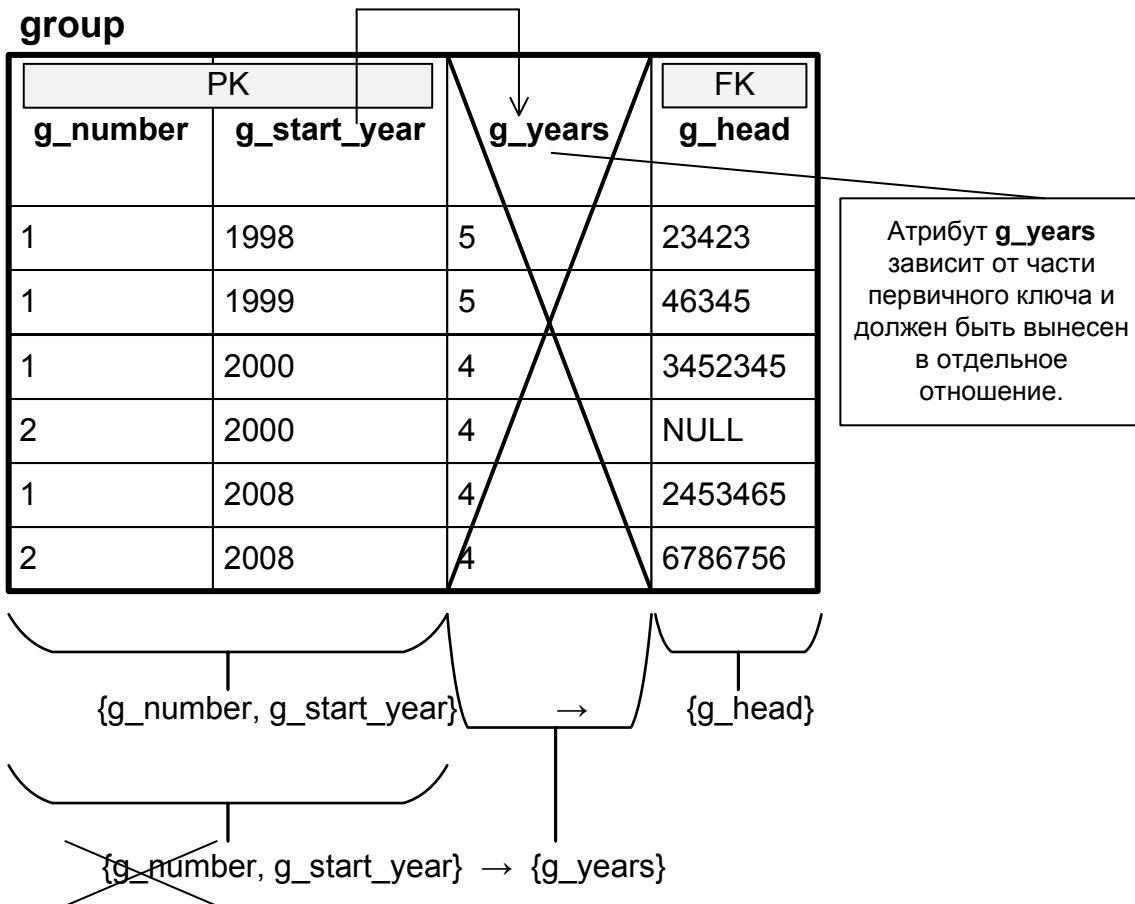


Рисунок 3.3.d — Нарушение второй нормальной формы (в «слабой формулировке»)

Итак, мы видим, что существует неключевой атрибут (**g_years**), который зависит от части первичного ключа, т.е. «слабая формулировка» определения второй нормальной формы нарушена, и отношение group не находится во второй нормальной форме.



«Слабая формулировка» определения 2НФ получила распространение по той причине, что отношения с составными альтернативными ключами в реальной жизни встречаются нечасто. Т.е. если таких ключей в отношении нет, можно руководствоваться «слабой формулировкой» при нормализации ко 2НФ. Но если такие ключи есть (как будет показано далее) — необходимо руководствоваться «сильной формулировкой».

Прежде, чем обсуждать последствия нарушения второй нормальной формы и способы приведения к ней переменных отношений, рассмотрим нарушение второй нормальной формы её «сильной формулировке». Для этого добавим в отношение group искусственный первичный ключ **g_id** (см. рисунок 3.3.e).

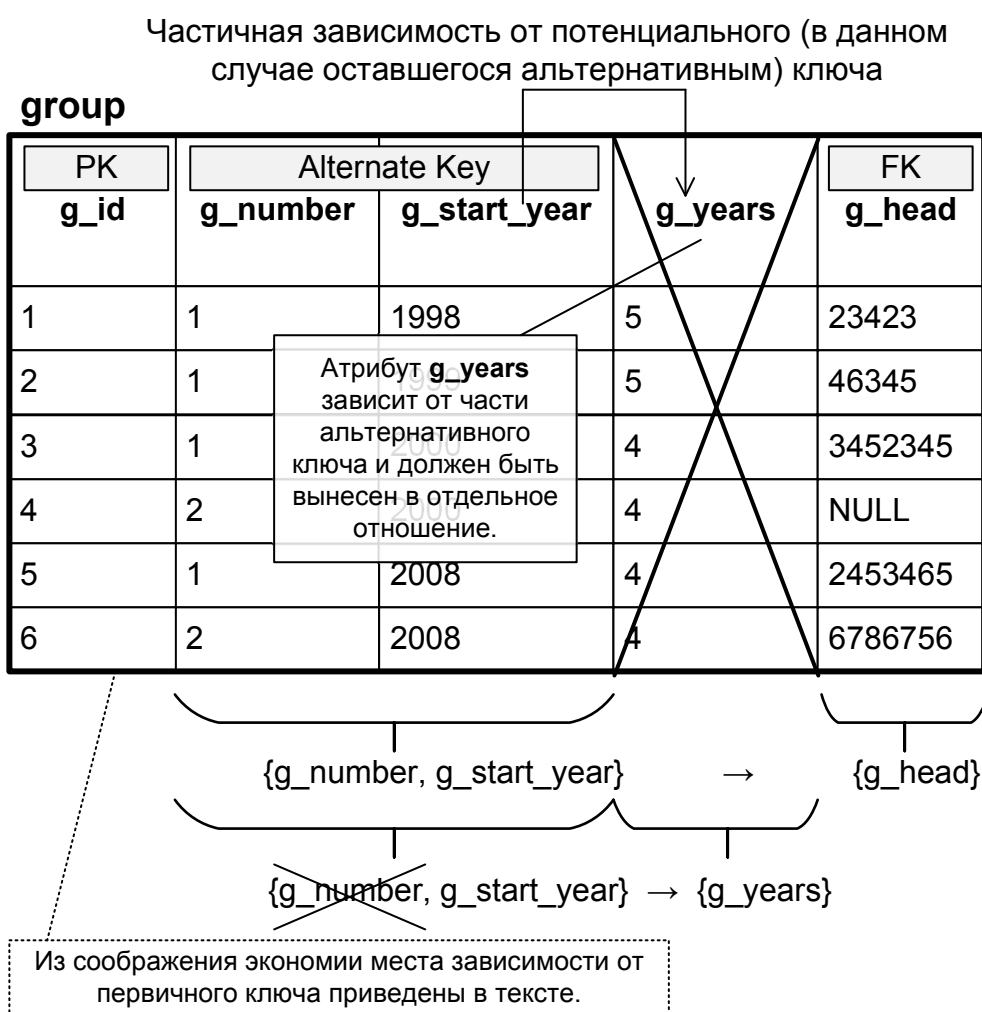


Рисунок 3.3.e — Нарушение второй нормальной формы (в «сильной формулировке»)

Рассмотрим имеющиеся в представленном на рисунке 3.3.e отношении зависимости:

- Полная функциональная зависимость $\{g_number, g_start_year\} \rightarrow \{g_head\}$ не изменилась по сравнению с ситуацией, представленной на рисунке 3.3.d.
- Частичная функциональная зависимость $\{g_number, g_start_year\} \rightarrow \{g_years\}$ не изменилась по сравнению с ситуацией, представленной на рисунке 3.3.d.
- Появилось четыре новых функциональных зависимости:
 - $\{g_id\} \rightarrow \{g_number\}$, т.к. уникальный идентификатор группы однозначно определяет её номер;
 - $\{g_id\} \rightarrow \{g_start_year\}$, т.к. уникальный идентификатор группы однозначно определяет год начала обучения;
 - $\{g_id\} \rightarrow \{g_years\}$, т.к. уникальный идентификатор группы однозначно определяет продолжительность обучения;
 - $\{g_id\} \rightarrow \{g_head\}$, т.к. уникальный идентификатор группы однозначно определяет её старосту.

Итак, все неключевые атрибуты функционально полно зависят от первичного ключа, т.е. «слабая формулировка» определения второй нормальной формы выполняется, но «сильная формулировка» нарушена, т.к. есть атрибут **g_years** зависящий от части потенциального ключа $\{g_number, g_start_year\}$.

И теперь настало время показать, чем грозит нарушение второй нормальной формы. Допустим, что в предметной области существует правило (оно в полной мере выполнено на рисунках 3.3.d и 3.3.e): «для поступивших по 1999-й год включительно обучение занимает пять лет, а для поступивших в 2000-м году и позднее — четыре года».

Но что мешает нам нарушить это правило, добавив новые или изменив текущие данные так, чтобы (например) для группы, начавшей обучение в 1998-м году срок обучения составил 4 года (или 3, или 8, или любое иное число лет)? В текущем состоянии схемы базы данных — ничто.

Интуитивное желание включить контроль этого правила через триггер^[350] может привести к обратной, но не менее опасной ситуации: что будет, если в будущем требования изменятся? Например, для поступивших «с 2015-го года и позднее» срок обучения снова должен стать пять лет.

Единственный вариант, гарантирующий как соблюдение взаимосвязи года поступления и срока обучения в настоящем, так и возможность корректировать это правило в будущем — создание отдельного отношения, как это показано на рисунке 3.3.f.

После выполнения нормализации единственный неключевой атрибут **g_head** отношения **group** функционально полно зависит от всех потенциальных ключей:

- $\{g_id\} \rightarrow \{g_head\}$, т.е. идентификатор группы однозначно определяет её старосту;
- $\{g_number, g_start_year\} \rightarrow \{g_head\}$, т.е. альтернативный способ гарантированно идентифицировать группу (альтернативный ключ^[38]) однозначно определяет её старосту.

В отношении **education_length** единственный неключевой атрибут **el_years** функционально полно зависит от единственного потенциального ключа (который за неимением альтернатив и выбран первичным):

- $\{el_start\} \rightarrow \{el_years\}$, т.е. год начала обучения однозначно идентифицирует продолжительность обучения.

Таким образом в обоих полученных отношениях нет неключевых атрибутов^{23}, частично зависящих^{184} от потенциальных ключей^{37}, что полностью соответствует требованиям «сильной формулировки» определения второй нормальной формы.

И теперь, указав год начала обучения для новой группы, мы гарантированно верно указываем её продолжительность обучения.

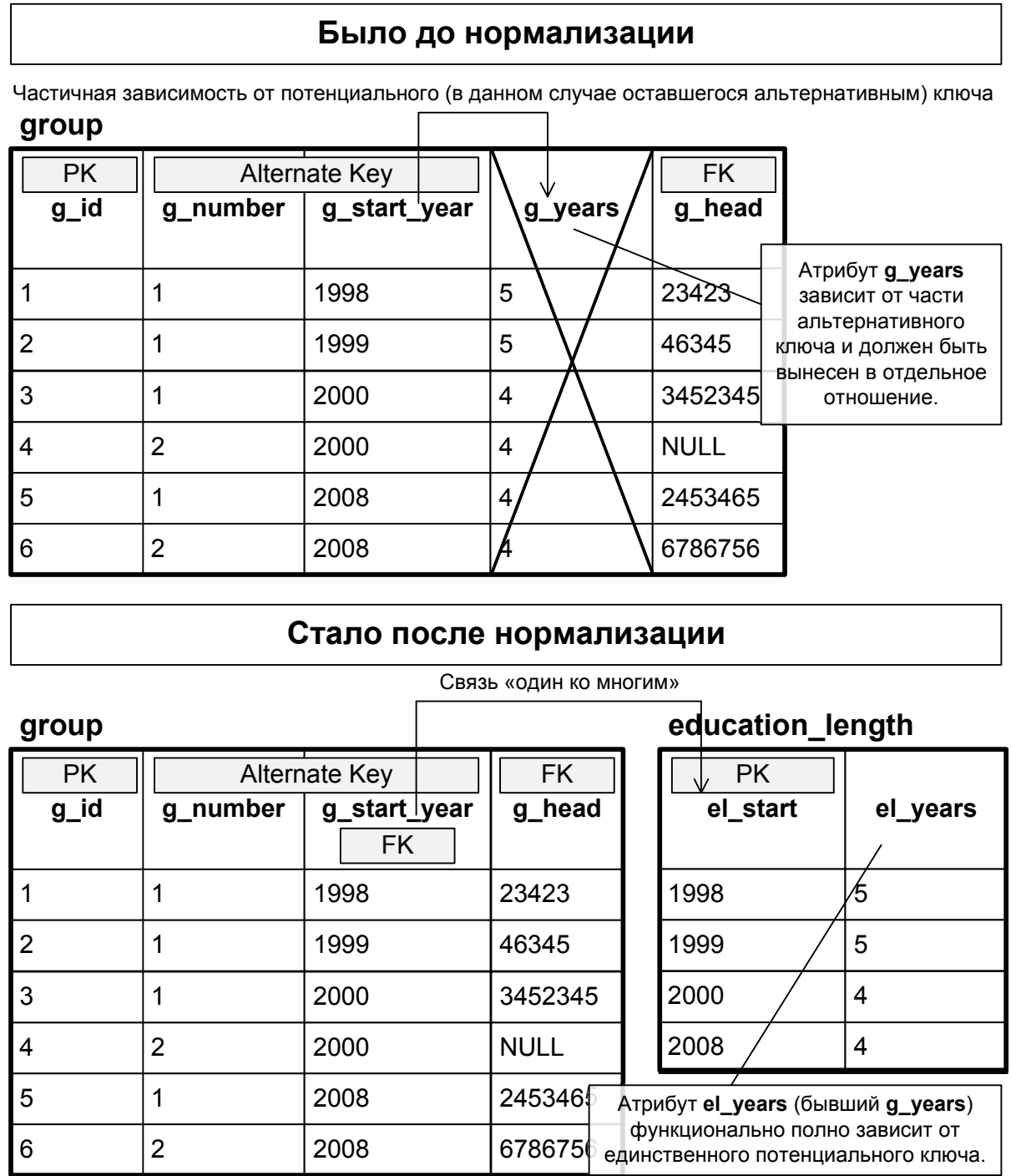


Рисунок 3.3.f — Приведение отношения ко второй нормальной форме

Краткий вывод по второй нормальной форме:

- следует руководствоваться «сильной формулировкой» определения 2НФ и искать частичные зависимости неключевых атрибутов именно от потенциальных ключей, а не только лишь от первичного ключа;
- каждый атрибут, частично зависящий от потенциального ключа, стоит перенести в отдельное отношение, сделав первичным ключом этого нового отношения ту «часть потенциального ключа», от которой переносимый атрибут зависел до проведения нормализации.



Задание 3.3.b: Помимо возможности ошибочно указать продолжительность обучения в отношениях, представленных на рисунках 3.3.d и 3.3.e, там существует и вторая проблема: аномалии модификации данных. Рассмотрите внимательно эти отношения и определите, каким аномалиям они подвержены.



Задание 3.3.c: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся во второй нормальной форме? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.

3.3.3. ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА



Перед прочтением материала данной главы стоит повторить определения транзитивной^{187} и нетривиальной^{191} функциональных зависимостей).



{Здесь приведена т.н. «упрощённая формулировка» определения 3НФ.}

Переменная отношения находится в **третьей нормальной форме** (3НФ, 3NF¹⁷¹) тогда и только тогда, когда она находится во второй нормальной форме^{246}, и каждый её неключевой атрибут^{23} нетранзитивно^{187} зависит от первичного ключа^{39}.

Упрощённо: в отношении не должно быть атрибутов, не входящих в состав первичного ключа и при этом транзитивно зависящих от первичного ключа.



{Здесь приведена т.н. «каноническая формулировка» определения 3НФ.}

Переменная отношения R находится в **третьей нормальной форме** (3НФ, 3NF¹⁷²) тогда и только тогда, когда она находится во второй нормальной форме^{246}, и для каждой нетривиальной^{191} функциональной зависимости $\{X\} \rightarrow A$ в этой переменной отношения выполняется хотя бы одно из двух условий:

- а) $\{X\}$ является суперключом^{36} переменной отношения R;
- б) A является ключевым атрибутом^{23} переменной отношения R.

Упрощённо: если какой-то атрибут функционально зависит от множества других атрибутов, то должно выполняться хотя бы одно из двух условий:

- а) это множество атрибутов уникально определяет любую запись;*
- б) этот зависимый атрибут входит в состав потенциального ключа.*

В отличие от «слабой» и «сильной» формулировок определения 2НФ^{246} эти определения лишь показывают два взгляда на одну и ту же ситуацию, причём «упрощённая» формулировка следует из «канонической». Покажем это.

- Поскольку (по определению^{36}) любой суперключ уникально идентифицирует любую запись таблицы и (по определению^{39}) первичный ключ уникально идентифицирует любую запись таблицы, то одному значению первичного ключа всегда соответствует строго одно значение любого суперключа, т.е. (по определению^{180}) существует функциональная зависимость $\{\text{первичный ключ}\} \rightarrow \{\text{любой суперключ}\}$. Также можно вспомнить, что первичный ключ сам по себе является частным случаем суперключа.
- Поскольку зависимость $\{\text{первичный ключ}\} \rightarrow \{\text{суперключ}\} \rightarrow A$ является избыточной, приведённое в предыдущем пункте рассуждение позволяет в «канонической» формулировке понимать условие «а» как « $\{X\}$ является первичным ключом» (что соответствует «упрощённой» формулировке).
- Условие «б» в «канонической» формулировке можно выразить через отрицание, т.е. вместо «A является ключевым атрибутом» сказать «A не является неключевым атрибутом» (что также соответствует «упрощённой формулировке»).

¹⁷¹ A relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key. («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

¹⁷² A relation schema R is in 3NF if, whenever a nontrivial functional dependency $X \rightarrow A$ holds in R, either (a) X is a superkey of R, or (b) A is a prime attribute of R. («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

Рассмотрим пример нарушения третьей нормальной формы (рисунок 3.3.g). Здесь показаны два случая, соответствующие обоим формулировкам определениям 3НФ:

- по «упрощённой» формулировке определения получается, что существует транзитивная зависимость $\{g_id\} \rightarrow \{g_faculty\} \rightarrow \{g_dean\} \rightarrow \{g_dean_dob\}$, т.е. от первичного ключа **g_id** зависит неключевой атрибут **g_faculty** (факультет, к которому относится группа), и уже от факультета зависит его декан (**g_dean**) и вся относящаяся к декану информация (в данном случае мы ограничились одним полем **g_dean_dob**, хранящим дату рождения декана);
- по «канонической» формулировке определения получается, что существует транзитивная зависимость $\{g_number, g_start_year\} \rightarrow \{g_faculty\} \rightarrow \{g_dean\} \rightarrow \{g_dean_dob\}$, т.е. от альтернативного ключа **{g_number, g_start_year}** зависит неключевой атрибут **g_faculty** (факультет, к которому относится группа), и уже от факультета зависит его декан (**g_dean**) и вся относящаяся к декану информация (в данном случае мы ограничились одним полем **g_dean_dob**, хранящим дату рождения декана).

Неключевые атрибуты **g_dean** и **g_dean_dob** транзитивно зависят от потенциальных ключей и должны быть вынесены в отдельное отношение.

| group | | | | | |
|------------|--|------|-------------|--------------|------------|
| PK g_id | Alternate Key g_number g_start_year | | g_faculty | g_dean | g_dean_dob |
| 1 | 1 | 1998 | Physics | John Smith | 1974-01-12 |
| 2 | 1 | 1999 | Physics | John Smith | 1974-01-12 |
| 3 | 1 | 2000 | Mathematics | Joe Black | 1984-12-23 |
| 4 | 2 | 2000 | IT | Jane Dow | 1985-05-28 |
| 5 | 1 | 2008 | IT | Jane Dow | 1985-05-28 |
| 6 | 2 | 2008 | Chemistry | Walter White | 1969-02-20 |

$\{g_id\}$

\rightarrow

$\{g_faculty\}$

\rightarrow

$\{g_dean\}$

\rightarrow

$\{g_dean_dob\}$

$\{g_number, g_start_year\} \rightarrow \{g_faculty\} \rightarrow \{g_dean\} \rightarrow \{g_dean_dob\}$

Рисунок 3.3.g — *Нарушение третьей нормальной формы*

Нарушение третьей нормальной формы является нежелательным потому, что оно приводит к бессмысленному дублированию хранимой информации. Представьте, что на каждом факультете обучается по 100 групп: тогда мы будем для каждого факультета по 100 раз хранить имя декана и его дату рождения (а в совсем плохом случае и ещё несколько десятков значений данных, характеризующих декана).

Помимо бессмысленных затрат памяти, мы также получаем аномалию обновления^[161], т.к. при изменении какой-либо информации о декане её нужно будет обновить в записях всех групп, относящихся к данному факультету.



Часто можно услышать вопрос о том, в чём же принципиальное отличие нарушения 2НФ (обсуждая рисунок 3.3.e^[248], мы говорили, что можно ошибочно приписать не тот срок обучения некоторой группе) и нарушения 3НФ (ведь ничто не мешает, например, приписать некоторой группе не тот факультет).

Отличие состоит в том, что нарушение 2НФ позволяет отступить от некоего глобального правила предметной области (т.е. «любая группа, начавшая обучения в таком-то году, учится столько-то лет»), а нарушение 3НФ позволяет лишь ошибочно указать некое отдельное значение данных, не нарушая никакого глобального правила (да, мы можем ошибочно записать некую группу на не тот факультет, но этим мы не нарушим никакого правила вида «любая группа, соответствующая таким-то условиям, обучается на таком-то факультете», т.к. такого правила нет).

Рассмотрим приведение отношения к третьей нормальной форме (см. рисунок 3.3.h).

Ранее мы уже увидели, что атрибуты **g_dean** и **g_dean_dob** являются «лишними» в отношении **group** и должны быть перенесены в отдельное отношение, но как мы видим на рисунке 3.3.h, вместо одного нового отношения было создано два — **faculty** и **staff**.

У такого решения есть две причины.

Первая — тривиальная и очень практичная: логично предположить, что в базе данных будет храниться информация не только о деканах факультетов, но и об иных сотрудниках университета, лишь часть их которых является деканами. Нам было бы некуда поместить информацию о «не деканах», если бы мы не разделили описание факультетов и сотрудников.

Вторая — полу-интуитивная, но всё же заслуживающая упоминания. Если бы мы оставили информацию о деканах в отношении **faculty**, мы снова получили бы нарушение третьей нормальной формы, т.к. присутствовала бы транзитивная зависимость неключевого атрибута от потенциального ключа: {Факультет} → {Декан} → {Дата рождения декана}. А в отношении **staff** такой зависимости нет, т.к. дата рождения сотрудника не зависит от его имени.

Иными словами, атрибут, содержащий имя декана, играет разную роль в зависимости от того, в каком отношении он находится. В отношении **group** до нормализации значение этого атрибута отвечало на вопрос «Кто декан этого факультета, кто этот сотрудник?» и определяло собой всю информацию о сотруднике (включая дату рождения). В отношении **stuff** на вопрос «Кто этот сотрудник?» отвечает атрибут **s_id** (первичный ключ), и именно он определяет дату рождения и иные параметры сотрудника. А атрибут **s_name** отвечает только на вопрос «Каково имя этого сотрудника?» и на дату рождения никак не влияет: именно поэтому транзитивной зависимости {**s_id**} → {**s_name**} → {**s_dob**} нет, а есть лишь две функциональные зависимости {**s_id**} → {**s_name**} и {**s_id**} → {**s_dob**}, наличие которых не противоречит определению 3НФ.

Поскольку эта идея может быть неочевидной, но её понимание является важным для правильного приведения отношений к 3НФ, она показана графически на рисунке 3.3.i.



Рисунок 3.3.h — Приведение отношения к третьей нормальной форме

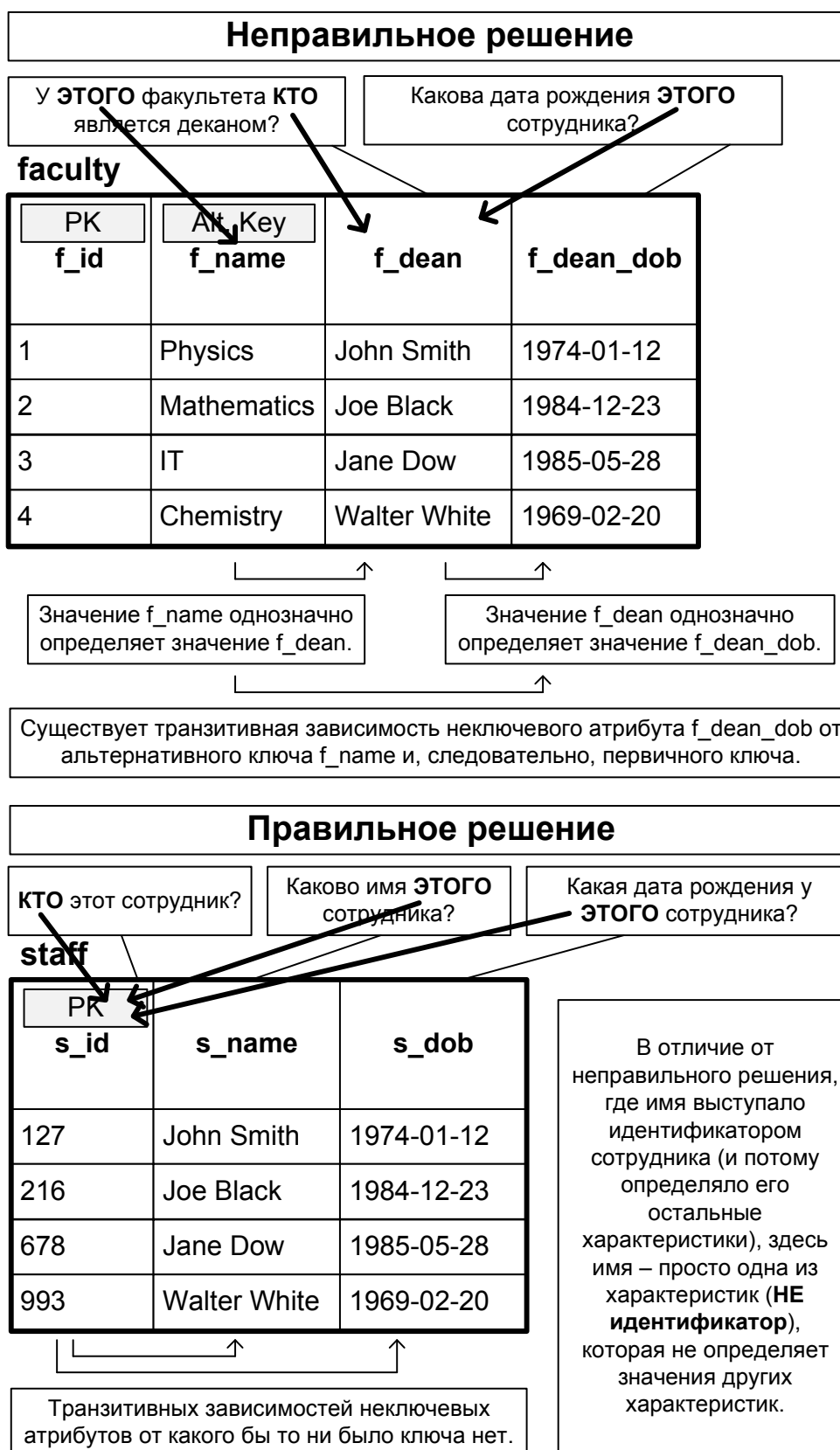


Рисунок 3.3.i — Изменение сути атрибута в зависимости от отношения, которому он принадлежит

Краткий вывод по третьей нормальной форме:

- «упрощённая» и «каноническая» формулировки определения 3НФ эквивалентны, и можно руководствоваться любой из них;
- 3НФ не защищает отношение от возможности ошибочного указания каких-то локальных данных (например, ошибочно вписать номер паспорта человека), но т.к. отношение уже находится во 2НФ, будут соблюдаться «глобальные правила», распространяющиеся на множество записей;
- ярким признаком нарушения 3НФ является бессмысленное дублирование одних и тех же данных в множестве строк таблицы (тогда атрибуты, значения которых бессмысленно дублируются, являются первыми кандидатами на перемещение в новое отдельное отношение).



Задание 3.3.d: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся в третьей нормальной форме? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.

3.3.4. НОРМАЛЬНАЯ ФОРМА БОЙСА-КОДДА



Перед прочтением материала данной главы стоит повторить определения транзитивной^[187] и нетривиальной^[191] функциональных зависимостей).



{Здесь приведена т.н. «упрощённая формулировка» определения НФБК.}

Переменная отношения находится в **нормальной форме Бойса-Кодда** (НФБК, BCNF) тогда и только тогда, когда она находится во второй нормальной форме^[246], и каждый её атрибут нетранзитивно^[187] зависит от первичного ключа^[39].

Упрощённо: в отношении не должно быть атрибутов, транзитивно зависящих от первичного ключа.



{Здесь приведена т.н. «каноническая формулировка» определения НФБК.}

Переменная отношения R находится в **нормальной форме Бойса-Кодда** (НФБК, BCNF¹⁷³) тогда и только тогда, когда она находится во второй нормальной форме^[246], и в каждой нетривиальной^[191] функциональной зависимости $\{X\} \rightarrow A$ в этой переменной отношения $\{X\}$ является суперключом^[36].

Упрощённо: если какой-то атрибут функционально зависит от множества других атрибутов, это множество атрибутов уникально определяет любую запись.

Отличие НФБК от 3НФ состоит в том, что 3НФ допускает транзитивную зависимость ключевых атрибутов от суперключа, а НФБК утверждает, что ни для какого атрибута (в т.ч. ключевого) не должно существовать таких зависимостей.

Это отличие хорошо видно, если расположить определения рядом:

| | 3НФ | НФБК |
|---------------------------|--|---|
| Упрощённая формулировка | Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и каждый её неключевой атрибут нетранзитивно зависит от первичного ключа. | Переменная отношения находится в нормальной форме Бойса-Кодда тогда и только тогда, когда она находится во второй нормальной форме, и каждый её неключевой атрибут нетранзитивно зависит от первичного ключа. |
| Каноническая формулировка | Переменная отношения R находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и для каждой нетривиальной функциональной зависимости $\{X\} \rightarrow A$ в этой переменной отношения выполняется хотя бы одно из двух условий: а) $\{X\}$ является суперключом переменной отношения R ; б) A является ключевым атрибутом переменной отношения R . | Переменная отношения R находится в нормальной форме Бойса-Кодда тогда и только тогда, когда она находится во второй нормальной форме, и для каждой нетривиальной функциональной зависимости $\{X\} \rightarrow A$ в этой переменной отношения выполняется хотя бы одно из двух условий: а) $\{X\}$ является суперключом переменной отношения R ; б) A является ключевым атрибутом переменной отношения R. |

¹⁷³ A relation schema R is in **BCNF** if whenever a nontrivial functional dependency $X \rightarrow A$ holds in R , then X is a superkey of R . («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

Существует ещё две формулировки определения НФБК, которые могут оказаться более простыми для понимания и запоминания.



Переменная отношения находится в **нормальной форме Бойса-Кодда** (НФБК, BCNF¹⁷⁴) тогда и только тогда, когда она находится в третьей нормальной форме^[252], и не содержит пересекающихся потенциальных ключей.

Упрощённо: отношение должно находиться в 3НФ, и в нём не должно быть потенциальных ключей^[37], имеющих общие атрибуты.

Переменная отношения находится в **нормальной форме Бойса-Кодда** (НФБК, BCNF¹⁷⁵) тогда и только тогда, когда детерминанты^[180] всех её функциональных зависимостей являются потенциальными ключами.

Упрощённо: в любой функциональной зависимости $\{X\} \rightarrow A$ множество $\{X\}$ должно быть потенциальным ключом^[37].

Рассмотрим пример нарушения НФБК (см. рисунок 3.3.j). Отношение **gift** описывает подарочные наборы. При этом (в рамках предметной области) существуют договорённости:

- ни в каком наборе не может быть двух и более предметов одного типа;
- ни в каком наборе не может быть двух и более одноимённых предметов;
- для каждого типа предметов допустим только один вариант предмета.

Первая договорённость позволяет нам считать комбинацию значений полей `{g_set, g_type}` уникальной и объявить соответствующий составной первичный ключ.

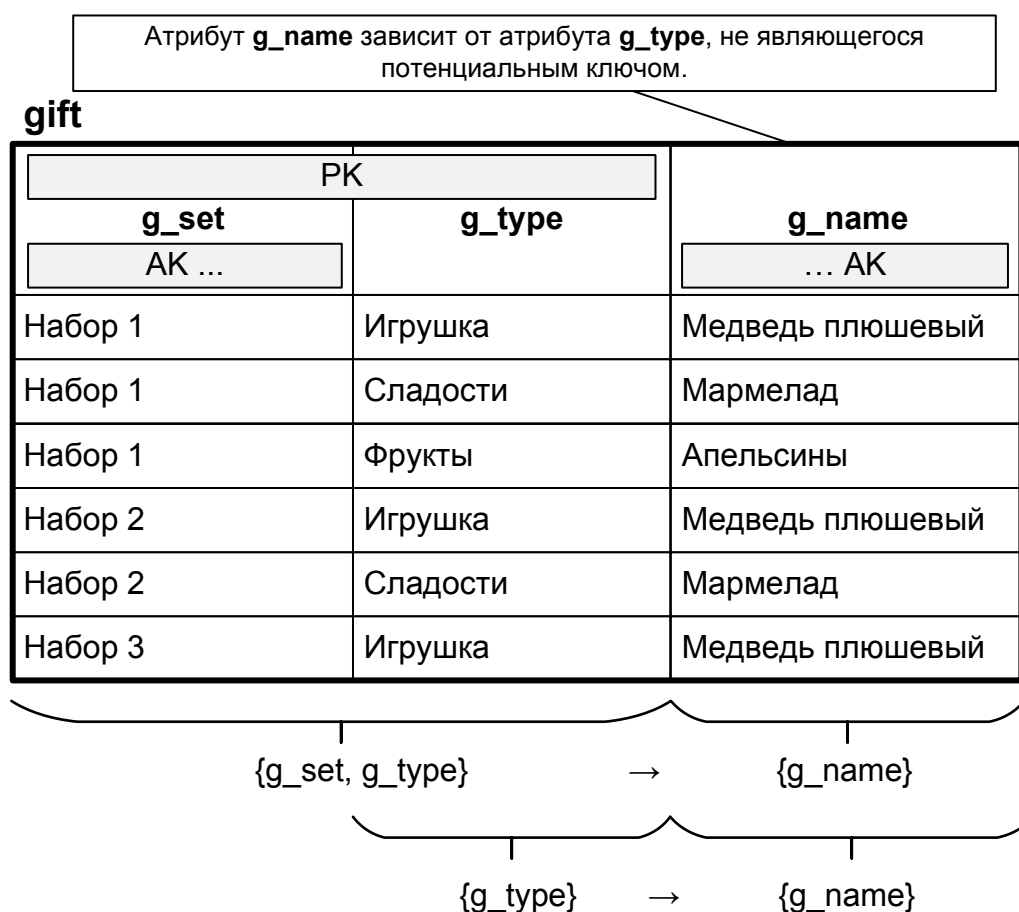
Вторая договорённость позволяет нам также считать уникальной комбинацию значений полей `{g_set, g_name}`, но т.к. первичный ключ уже выбран, эта комбинация останется альтернативным ключом. И всё же атрибут `g_name` является ключевым (поэтому отношение **gift** не противоречит второй нормальной форме^[246]: да, атрибут `g_name` зависит от части первичного ключа, но он сам является ключевым).

Третья договорённость даёт нам функциональную зависимость `{g_type} → {g_name}`, но мы видим (см. рисунок 3.3.j), что атрибут `g_type` не обладает свойством уникальности, т.е. детерминант функциональной зависимости не является потенциальным ключом, и потому отношение **gift** не находится в нормальной форме Бойса-Кодда.

Нарушение НФБК опасно тем же набором аномалий, что и нарушение второй нормальной формы: например, можно для набора подарков «Набор 2» перепутать наименования предметов и записать «игрушка — мармелад, сладости — медведь плюшевый», а потом в «Наборе 1» тоже что-то перепутать и записать «сладости — апельсины». Кроме того, что здесь перепутаны данные мы ещё получаем и нарушение только что упомянутой третьей договорённости, согласно которой для каждого типа предметов допустим лишь один вариант предмета, а у нас получилось, что «сладости» — это и «медведь плюшевый», и «апельсины».

¹⁷⁴ A 3NF table that does not have multiple overlapping candidate keys is guaranteed to be in BCNF. «A Note on Relation Schemes Which Are in 3NF But Not in BCNF» (Vincet, M.W. and B. Srinivasan)

¹⁷⁵ Relvar R is in BCNF if and only if every FD that holds in R is implied by some superkey. («The New Relational Database Dictionary», C.J. Date)

Рисунок 3.3.j — *Нарушение нормальной формы Бойса-Кодда*

Чтобы привести это отношение к НФБК, его атрибуты необходимо распределить по новым отношениям.

Отметим, что здесь может быть сложность, т.к. не каждое распределение позволит потом получить исходное отношение через операцию объединения¹⁷⁶ (**JOIN**), но в общем случае достаточно «разрезать» исходное отношение по «проблемному» полю (являющемуся детерминантом функциональной зависимости и при этом не являющемуся потенциальным ключом; в нашем случае — это поле **g_type**).

Результат такого преобразования показан на рисунке рисунок 3.3.k.

¹⁷⁶ См. подробности в главе 15.5 книги «Fundamentals of Database Systems» (Ramez Elmasri, Shamkant Navathe).

Было до нормализации

gift

| PK | | |
|---------|--|------------------|
| g_set | g_type | g_name |
| AK ... | | ... AK |
| Набор 1 | Игрушка | Медведь плюшевый |
| Набор 1 | Атрибут g_name зависит от атрибута g_type , не являющегося потенциальным ключом. | Мармелад |
| Набор 1 | Фрукты | Апельсины |
| Набор 2 | Игрушка | Медведь плюшевый |
| Набор 2 | Сладости | Мармелад |
| Набор 3 | Игрушка | Медведь плюшевый |

Стало после нормализации

set

| PK | |
|---------|----------|
| s_name | s_o_type |
| | FK |
| Набор 1 | Игрушка |
| Набор 1 | Сладости |
| Набор 1 | Фрукты |
| Набор 2 | Игрушка |
| Набор 2 | Сладости |
| Набор 3 | Игрушка |

Связь «1-М»

object

| PK | |
|----------|------------------|
| o_type | o_name |
| Игрушка | Медведь плюшевый |
| Сладости | Мармелад |
| Фрукты | Апельсины |

Рисунок 3.3.k — Приведение отношения к нормальной форме Бойса-Кодда

Краткий вывод по третьей нормальной форме:

все формулировки определения НФБК эквивалентны, и можно руководствоваться любой из них;

- как и 3НФ, НФБК не защищает отношение от возможности ошибочного указания каких-то локальных данных (например, можно ошибочно вписать название игрушки), но позволяет уменьшить дублирование данных и защищает отношение от нарушения «глобальных правил», распространяющихся на множество записей и несколько атрибутов;
- ярким признаком нарушения НФБК является наличие в отношении пересекающихся потенциальных ключей.



Задание 3.3.е: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся нормальной форме Бойса-Кодда? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.

3.3.5. ЧЕТВЁРТАЯ НОРМАЛЬНАЯ ФОРМА



Перед прочтением материала данной главы стоит повторить определения многозначной^{193} зависимости и её подвидов — тривиальной^{194} и нетривиальной^{194}.



Переменная отношения R находится в **четвёртой нормальной форме** (4НФ, 4NF¹⁷⁷) тогда и только тогда, когда для любой её нетривиальной многозначной зависимости $X \twoheadrightarrow Y$ множество X является суперключом^{36} R .

Упрощённо: отношение должно находиться в НФБК^{258}, и в нём не должно быть нетривиальных^{194} многозначных зависимостей.

Хорошим способом пояснения 4НФ является логика вывода упрощённого определения из строгого.

Почему отношение должно находиться в НФБК (см. упрощённую формулировку определения)?

Начнём с того, как связаны функциональная зависимость^{180} и многозначная зависимость^{193}. Фактически, функциональная зависимость — это многозначная зависимость, в которой количество значений функции ограничено единицей, т.е. $A \twoheadrightarrow B$ эквивалентно $A \rightarrow B$, когда мощность B равна единице¹⁷⁸.

Примем во внимание тот факт, что (см. строгую формулировку определения) многозначная зависимость $X \twoheadrightarrow Y$ должна быть нетривиальной, потому что функциональная зависимость $X \rightarrow Y$ также должна быть нетривиальной^{191}, т.е. Y не может быть подмножеством X .

Это в полной мере соответствует определению НФБК^{258}, которое для данного случая можно сформулировать так: «отношение R находится в НФБК тогда и только тогда, когда оно находится во второй нормальной форме, и для каждой не-тривиальной функциональной зависимости $X \rightarrow A$ в этом отношении X является суперключом отношения R ».

Почему в отношении не должно быть нетривиальных многозначных зависимостей (см. упрощённую формулировку определения)?

Потому, что (см. строгую формулировку определения) в зависимости $X \twoheadrightarrow Y$ множество X должно быть суперключом, т.е. в отношении не может находиться двух и более кортежей, значение совокупности атрибутов X в которых совпадает. А раз значение X может встретиться только один раз, то ему может соответствовать не более одного значения Y .

И это — как раз случай вырождения нетривиальной многозначной зависимости в тривиальную (см. рисунок 3.2.p^{195}), т.е. нетривиальные многозначные зависимости в рамках 4НФ существовать не могут (что и сказано в упрощённой формулировке определения 4НФ).

Осталось привести графические примеры.

На рисунке 3.3.1 представлен случай нарушения четвёртой нормальной формы. Так, если предметная область допускает подачу документов каждым абитуриентом на несколько факультетов, а на каждом факультете существует несколько вступительных экзаменов, то существуют зависимости $\{ue_applicant\} \twoheadrightarrow \{ue_faculty\}$ и $\{ue_faculty\} \twoheadrightarrow \{ue_exam\}$, что можно записать как $\{ue_applicant\} \twoheadrightarrow \{ue_faculty\} \mid \{ue_exam\}$.

¹⁷⁷ Relvar R is in **fourth normal form**, 4NF, if and only if every MVD that holds in R is implied by some superkey of R — equivalently, if and only if for every nontrivial MVD $X \twoheadrightarrow Y$ that holds in R , X is a superkey for R (in which case the MVD $X \twoheadrightarrow Y$ effectively degenerates to the FD $X \rightarrow Y$). («The New Relational Database Dictionary», C.J. Date)

¹⁷⁸ См. подробное пояснение в разделе 13.2 книги «An Introduction to Database Systems (8th edition)» (C.J. Date).

Одному абитуриенту может соответствовать несколько факультетов, на которые он подал документы (т.е. существует зависимость $\{ue_applicant\} \rightarrow \{ue_faculty\}$).

Одному факультету может соответствовать несколько экзаменов, которые нужно сдать абитуриентам (т.е. существует зависимость $\{ue_faculty\} \rightarrow \{ue_exam\}$).

university_entrance

| PK | | | |
|--------------|----------------|--------------|---|
| ue_applicant | ue_faculty | ue_exam | |
| Иванов И.И. | Математический | История КПСС | |
| Иванов И.И. | Математический | Математика | Появление нового абитуриента обязывает нас добавить в таблицу столько строк, сколько экзаменов сдаётся на соответствующем факультете. (+ См. в тексте.) |
| Иванов И.И. | Физический | История КПСС | |
| Иванов И.И. | Физический | Физика | |
| Петров П.П. | Математический | История КПСС | |
| Петров П.П. | Математический | Математика | |
| Сидоров С.С. | Физический | История КПСС | |
| Сидоров С.С. | Физический | Физика | |

$\{ue_applicant\} \rightarrow \{ue_faculty\} | \{ue_exam\}$

Рисунок 3.3.1 — Нарушение четвёртой нормальной формы

Отношение **university_entrance** находится в НФБК (у него нет неключевых атрибутов или пересекающихся потенциальных ключей), но оно не находится в четвёртой нормальной форме.

Нарушение четвёртой нормальной формы чревато возникновением серии аномалий операций с данными^{161}:

- при появлении нового абитуриента запись о нём необходимо продублировать столько раз, сколько в совокупности будет экзаменов на всех факультетах, на которые он подал документы;
- при появлении нового экзамена на некотором факультете придётся добавить соответствующую запись столько раз, сколько абитуриентов хочет поступить на данный факультет;
- при добавлении нового факультета... мы не сможем его добавить, т.к. на него пока никто не подал документы, а поле **ue_applicant** входит в состав первичного ключа;
- аналогичные проблемы будут при обновлении и удалении данных, т.е. представленное на рисунке 3.3.1 отношение является отличным примером реализации практически любой аномалии операций с данными.

Логика приведения отношения **university_entrance** к четвёртой нормальной форме показана на рисунке 3.3.m.

Было до нормализации

university_entrance

| PK | | |
|--------------|----------------|--------------|
| ue_applicant | ue_faculty | ue_exam |
| Иванов И.И. | Математический | История КПСС |
| Иванов И.И. | Физический | Математика |
| Иванов И.И. | Физический | История КПСС |
| Иванов И.И. | Физический | Физика |
| Петров П.П. | Математический | История КПСС |
| Петров П.П. | Математический | Математика |
| Сидоров С.С. | Физический | История КПСС |
| Сидоров С.С. | Физический | Физика |

Одному абитуриенту может соответствовать несколько факультетов, на которые он подал документы (т.е. существует зависимость {ue_applicant} → {ue_faculty}).

Одному факультету может соответствовать несколько экзаменов, которые нужно сдать абитуриентам (т.е. существует зависимость {ue_faculty} → {ue_exam}).

Стало после нормализации

application

| PK | |
|--------------|----------------|
| a_applicant | a_faculty |
| Иванов И.И. | Математический |
| Иванов И.И. | Физический |
| Петров П.П. | Математический |
| Сидоров С.С. | Физический |

exam

| PK | |
|----------------|--------------|
| e_faculty | e_exam |
| Математический | История КПСС |
| Математический | Математика |
| Физический | История КПСС |
| Физический | Физика |

Рисунок 3.3.т — Приведение отношения к четвёртой нормальной форме

Отношения **application** и **exam** находятся в 4НФ, т.к. у них нет неключевых атрибутов, пересекающихся потенциальных ключей и «лишних» атрибутов, которые могли бы составить множество Z из определения нетривиальной многозначной зависимости^[194].

Однако, такой результат приведения к 4НФ обладает очень неприятным недостатком: мы не можем провести между отношениями **application** и **exam** никакой связи, т.к. установка связи привела бы к миграции первичного ключа родительского отношения в дочернее, что дало бы нам исходное отношение **university_entrance**.

Потому с точки зрения практической реализации более выгодным является конечная схема, показанная на рисунке 3.3.п.

Стало после нормализации

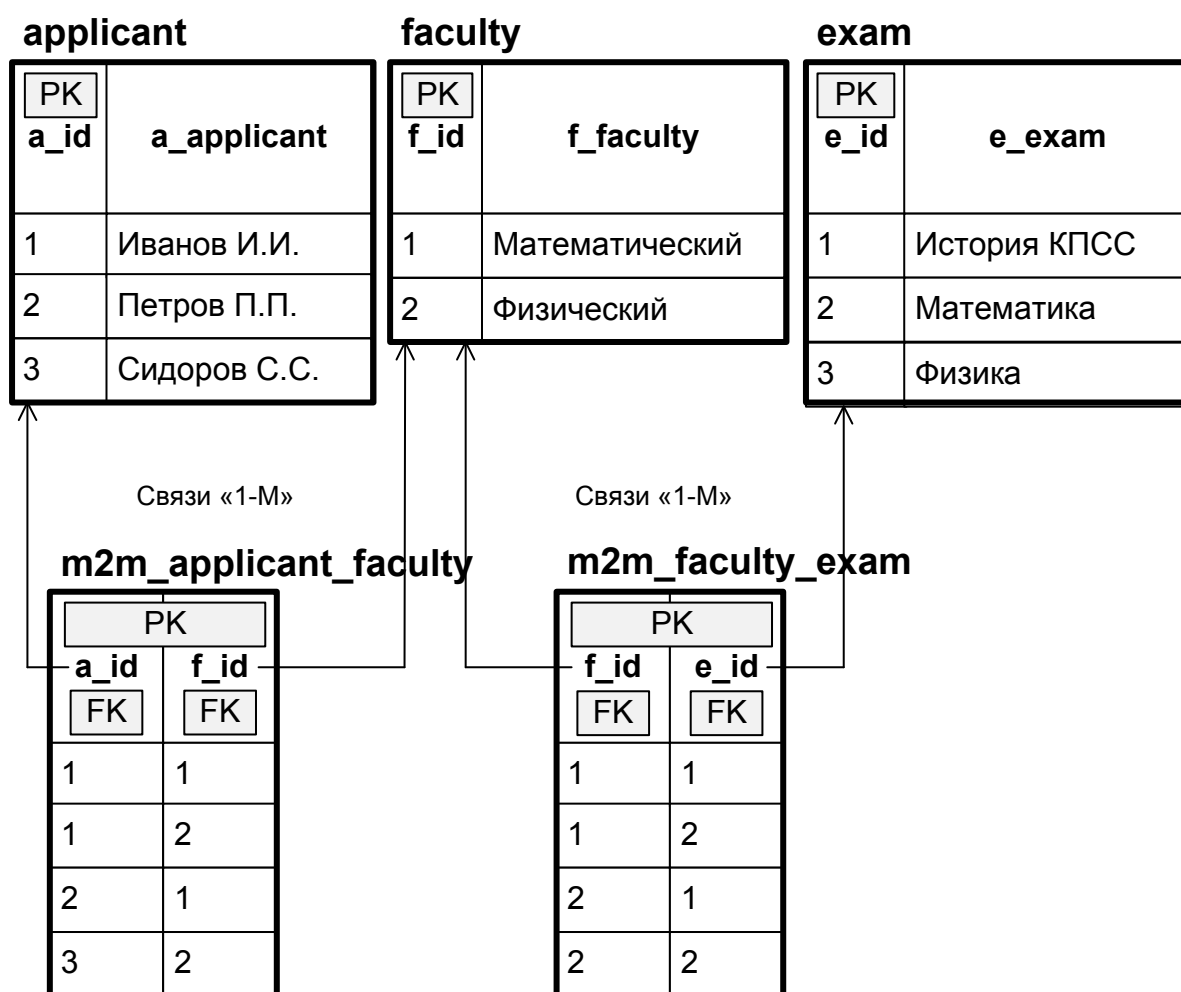


Рисунок 3.3.n — Более эффективное приведение отношения к четвёртой нормальной форме

Пусть вариант, показанный на рисунке 3.3.n и содержит значительно больше отношений (т.е. при выполнении операций объединения СУБД придётся оперировать большим количеством объектов), он всё же позволяет задействовать механизм обеспечения ссылочной целостности^[72], и потому может считаться более предпочтительным, чем показанный на рисунке 3.3.m.

Краткий вывод по четвёртой нормальной форме:

- отношения, находящиеся в НФБК, но не находящиеся в 4НФ — большая редкость (на практике они появляются очень редко);
- как и НФБК, 4НФ не защищает отношение от возможности ошибочного указания каких-то локальных данных (например, можно ошибочно вписать название экзамена или приписать его «не тому» факультету), но позволяет уменьшить дублирование данных и защищает отношение от нарушения «глобальных правил», распространяющихся на множество записей и несколько атрибутов;
- ярким признаком нарушения 4НФ является наличие в отношении группы из трёх и более атрибутов, в которой атрибуты попарно зависят друг от друга, но при этом не зависят от значений остальных атрибутов.



Задание 3.3.f: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся четвёртой нормальной форме? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.

3.3.6. ПЯТАЯ НОРМАЛЬНАЯ ФОРМА



Перед прочтением материала данной главы стоит повторить определение зависимости соединения^{196}.



Переменная отношения находится в **пятой нормальной форме** (5НФ¹⁷⁹, 5NF¹⁸⁰) тогда и только тогда, когда она находится в 4НФ^{263}, и для каждой существующей в ней нетривиальной зависимости соединения^{196} $JD(R_1, R_2, \dots, R_n)$ каждый набор атрибутов R_i является суперключом исходной переменной отношения.

Упрощённо: необходимо знать все потенциальные ключи отношения и все существующие в нём зависимости соединения, а также удостовериться, что каждая проекция, определяющая любую из зависимостей соединения, содержит потенциальный ключ.

Если вернуться к примерам наличия и отсутствия зависимости соединения (см. рисунки 3.2.r^{198} и 3.2.s^{200}), легко заметить что в реальной жизни крайне редко встречаются ограничения предметной области наподобие представленного ранее: «Если преподаватель ведёт некоторый предмет **П**, и на некотором факультете **Ф** преподаётся этот предмет **П**, и преподаватель ведёт хотя бы один любой предмет на этом факультете **Ф**, то он обязан вести на этом факультете **Ф** и предмет **П**».

Именно в силу редкости и сложности обнаружения в предметной области таких громоздких и нетривиальных правил говорят, что нормализация до 5НФ почти никогда не выполняется.

И всё же рассмотрим признаки нарушения 5НФ и способ её достижения. 5НФ нарушена, если (все условия должны быть выполнены одновременно):

- в отношении существуют зависимости соединения^{196};
- хотя бы для одной такой зависимости существует проекция, не включающая в себя потенциальный ключ отношения.

Продолжим рассмотрение отношения, ранее представленного на рисунке 3.2.r^{198}. Оно находится в 4НФ, т.к. здесь нет многозначных зависимостей^{193}, однако с учётом только что процитированного правила про связь преподавателей, факультетов и предметов можно утверждать, что здесь есть зависимость соединения $JD((w_tutor, w_subject), (w_subject, w_faculty), (w_tutor, w_faculty))$.

Таким образом, первое условие нарушения 5НФ выполнено — зависимости соединения в этом отношении есть.

Выполнение второго условия нарушения 5НФ ещё более очевидно: ни одна из проекций $(w_tutor, w_subject)$, $(w_subject, w_faculty)$, $(w_tutor, w_faculty)$ не содержит в себе потенциальных ключей (что легко проверить по представленным на рисунке 3.3.о данным).

¹⁷⁹ 5НФ иногда называют «нормальной формой проекции-соединения» (project-join normal form (PJNF)).

¹⁸⁰ A relation schema R is in fifth normal form (5NF) (or project-join normal form (PJNF)) with respect to a set F of functional, multivalued, and join dependencies if, for every nontrivial join dependency $JD(R_1, R_2, \dots, R_n)$ in F^+ (that is, implied by F), every R_i is a superkey of R . («Fundamentals of Database Systems», Ramez Elmasri, Shamkant Navathe)

workload

| PK | | |
|--------------|-------------|----------------|
| w_tutor | w_subject | w_faculty |
| Иванов И.И. | Математика | Точных наук |
| Иванов И.И. | Информатика | Естествознания |
| Петров П.П. | Математика | Естествознания |
| Сидоров С.С. | Информатика | Кибернетики |
| Петров П.П. | Физика | Точных наук |
| Петров П.П. | Математика | Точных наук |
| Иванов И.И. | Математика | Естествознания |

Присутствует зависимость соединения:

JD((w_tutor, w_subject),
(w_subject, w_faculty),
(w_tutor, w_faculty))

Ни одна из проекций
(w_tutor, w_subject),
(w_subject, w_faculty),
(w_tutor, w_faculty) не
содержит в себе
потенциальных ключей.

Рисунок 3.3.о — *Нарушение пятой нормальной формы*

Нарушение 5НФ, наличие которого мы только что доказали, говорит о том, что данное отношение можно декомпозировать^{180} без потерь на несколько отдельных отношений. Результат такой декомпозиции представлен на рисунке 3.3.р.

Было до нормализации

workload

| ПК | | |
|--------------|-------------|----------------|
| w_tutor | w_subject | w_faculty |
| Иванов И.И. | Математика | Точных наук |
| Иванов И.И. | Информатика | Естествознания |
| Петров П.П. | Математика | Естествознания |
| Сидоров С.С. | Информатика | Кибернетики |
| Петров П.П. | Физика | Точных наук |
| Петров П.П. | Математика | Точных наук |
| Иванов И.И. | Математика | Естествознания |

Присутствует зависимость соединения:
 $JD((w_tutor, w_subject), (w_subject, w_faculty), (w_tutor, w_faculty))$

Ни одна из проекций
 $(w_tutor, w_subject)$,
 $(w_subject, w_faculty)$,
 $(w_tutor, w_faculty)$ не
 содержит в себе
 потенциальных ключей.

Стало после нормализации

workload_ts

| ПК | |
|--------------|-------------|
| w_tutor | w_subject |
| Иванов И.И. | Математика |
| Иванов И.И. | Информатика |
| Петров П.П. | Математика |
| Сидоров С.С. | Информатика |
| Петров П.П. | Физика |

workload_sf

| ПК | |
|-------------|----------------|
| w_subject | w_faculty |
| Математика | Точных наук |
| Информатика | Естествознания |
| Математика | Естествознания |
| Информатика | Кибернетики |
| Физика | Точных наук |

workload_tf

| ПК | |
|--------------|----------------|
| w_tutor | w_faculty |
| Иванов И.И. | Точных наук |
| Иванов И.И. | Естествознания |
| Петров П.П. | Естествознания |
| Сидоров С.С. | Кибернетики |
| Петров П.П. | Точных наук |

Рисунок 3.3.p — Приведение отношения к пятой нормальной форме

Как и в случае с 4НФ^[263] мы получили вариант, корректный с точки зрения реляционной теории, но крайне неудобный на практике: мы не можем провести между полученными в процессе нормализации отношениями никакой связи, т.к. установка связи привела бы к миграции первичного ключа родительского отношения в дочернее, что дало бы нам исходное отношение **workload**.

Потому с точки зрения практической реализации более выгодным является конечная схема, показанная на рисунке 3.3.q.



Рисунок 3.3.q — Более эффективное приведение отношения к пятой нормальной форме

Вариант, показанный на рисунке 3.3.q содержит значительно больше отношений (т.е. при выполнении операций объединения СУБД придётся оперировать большим количеством объектов), но он позволяет задействовать механизм обеспечения ссылочной целостности^[72], и потому может считаться более предпочтительным, чем показанный на рисунке 3.3.p. Также данный вариант позволяет более эффективно использовать триггеры^[350] для контроля выполнения бизнес-правила о связи между преподавателями, предметами и факультетами.

В завершении этой главы отметим, что если бы в предметной области не существовало правила о связи между преподавателями, предметами и факультетами, в отношении **workload** не было бы зависимости соединения, и оно уже изначально находилось бы в 5НФ.

Краткий вывод по пятой нормальной форме:

- отношения, находящиеся в 4НФ, но не находящиеся в 5НФ — большая редкость (на практике они почти никогда не встречаются);
- как и 4НФ, 5НФ не защищает отношение от возможности ошибочного указания каких-то локальных данных (например, можно ошибочно вписать название факультета или приписать его «не тому» предмету), но позволяет уменьшить дублирование данных и защищает отношение от нарушения «глобальных правил», распространяющихся на множество записей и несколько атрибутов;
- признаком нарушения 5НФ является наличие в отношении группы из трёх и более атрибутов, в которой все входящие в эту группу атрибуты находятся во взаимосвязи друг от друга;
- 5НФ является «финальной» нормальной формой, т.е. находящееся в 5НФ отношение уже невозможно без потерь декомпонировать дальше (за исключением случаев хронологических отношений — см. 6НФ^{273}).



Задание 3.3.g: существуют ли в базе данных «Банк»^{408} схемы отношений, не находящиеся пятой нормальной форме? Если вы считаете, что «да», внесите в модель соответствующие правки для приведения таких схем отношений к соответствующей нормальной форме.