

# Базы данных нормальные формы

В.Н.Лукин

04.11.2020

# Функциональная зависимость

Цели, которым служит база данных, — снижение избыточности и повышение надежности хранения информации. Любое знание об ограничениях на данные, может служить этим целям.

Один из способов формализации этих знаний — установление зависимости между данными, отражающей их семантику, которая выражается в функциональных зависимостях схемы.

Функциональная зависимость имеет место, если значение кортежа на одном множестве атрибутов однозначно определяет его значение на другом.

# Функциональная зависимость

## определение

Пусть  $R$  — схема отношения,  $X, Y \subseteq R$ . Множество атрибутов  $Y$  функционально зависит от  $X$  ( $X \rightarrow Y$ ) тогда и только тогда, когда в любой момент времени для равных значений  $X$  значения  $Y$  тоже равны:  
$$\forall r(R), t_i, t_j \in r: t_i(X) = t_j(X) \Rightarrow t_i(Y) = t_j(Y).$$

Равнозначное утверждение:  $t_i(Y) \neq t_j(Y) \Rightarrow t_i(X) \neq t_j(X)$ .  
Левая часть функциональной зависимости ( $X$ ) — *детерминант*.

# Функциональная зависимость

## пример

Задан график ежедневного выполнения авиарейсов, на которые назначается первый пилот. Зададим его отношением со следующей схемой:

***график*** (*Пилот, Рейс, Дата, Время*).

Зависимость атрибутов определяется ограничениями:

- для каждого рейса определено одно время вылета;
- пилот в один день и в одно время выполняет один рейс;
- на рейс в заданную дату назначается один пилот.

Функциональные зависимости:

- *Рейс* → *Время*
- (*Пилот, Дата, Время*) → *Рейс*
- (*Рейс, Дата*) → *Пилот*

# Нормализация (1)

Некоторые функциональные зависимости могут быть нежелательными в конкретной схеме, так как они при модификации базы данных приводят к трудностям, *аномалиям модификации* (добавления, изменения и удаления).

Для приведения схемы в корректный вид используется замена одного множества отношений другим, сохраняющим её эквивалентность.

Такое преобразование составляет суть процесса нормализации.

## Нормализация (2)

В результате исходное небольшое число таблиц с «большой» схемой и непривлекательными свойствами, заменяется бóльшим числом таблиц с «меньшей» схемой без этих свойств.

Говорят, что полученные отношения удовлетворяют некоторой *нормальной форме*.

Нормальные формы составляют иерархию, в ней формы с большими номерами не обладают некоторыми нежелательными свойствами форм с меньшими номерами. Для реляционных БД существует шесть уровней нормализации.

# Нормализация (3)

Проектирование данных не всегда сопровождается процессом нормализации. Обычно при создании информационной модели проектировщик, при естественном порядке построения, строит её сразу в третьей нормальной форме.

К. Дейт вообще утверждает, что отношения, полученные при проектировании, будут сразу в пятой нормальной форме, если проектировщик не злонамерен.

Некоторые специалисты считают, что достаточно третьей или четвёртой формы.

# 1 Нормальная форма

Согласно определению отношения, все его атрибуты атомарны, то есть не могут быть разделены семантически на более мелкие элементы.

Отношение, обладающее этим свойством, называется нормализованным или, что то же самое, находящимся в первой нормальной форме (1НФ).

*Отношение находится в первой нормальной форме (1НФ), если все значения его атрибутов атомарны, то есть для каждого отношения  $r(R)$  и  $A \in R$ ,  $atom(A, r)$  атомарен.*



# 1 Нормальная форма

## пример

Расписание авиарейсов задано отношением **Расписание** со схемой (*Рейс, Пункт назначения, Вылет*). Атрибут *Вылет* обозначает день недели и время вылета.

( <i>Рейс</i> ,	<i>Пункт назначения</i> ,	<i>Вылет</i> )
1	Владивосток	пн, 9:40
1	Владивосток	ср, 9:40
347	Уфа	пн, 20:00
347	Уфа	вт, 20:00

Запрос «Выдать рейсы до Уфы во вторник» невозможен: день недели — часть атомарного атрибута. Заменим атрибут *Вылет* парой (*День, Время*):  
**расписание1**(*Рейс, Пункт назначения, День, Время*).

## 2 Нормальная форма

Распространенная ошибка при проектировании — объявить в качестве первичного ключа суперключ. Это приводит к значительным неприятностям.

*Функциональная зависимость  $X = (A_1, A_2, \dots, A_k) \rightarrow B$  полная, если  $B$  зависит от каждого  $A_i \in X$ . Если же существует  $X' \rightarrow B$ ,  $X' \subset X$ , функциональная зависимость  $X \rightarrow B$  неполная*

Отношение находится во *второй* нормальной форме (2НФ), если оно находится в 1НФ и каждый непервичный атрибут функционально полно зависит от ключа.

## 2 Нормальная форма

### пример (1)

Рассмотрим задачу поставок товара поставщиками, если определены следующие ограничения:

- товар могут поставлять разные поставщики;
- поставщик может поставлять разные товары;
- цена одинаковых товаров одинакова.

Первые два ограничения говорят о том, что ключ отношения **поставки** составляет пара атрибутов (Поставщик, Товар):

**поставки**(Поставщик, Товар, Цена),

что соответствует функциональной зависимости

Поставщик, Товар  $\rightarrow$  Цена

- Третье ограничение порождает функциональную зависимость  
Товар  $\rightarrow$  Цена

## 2 Нормальная форма

### пример (2)

Цена функционально зависит от части ключа, то есть отношение не находится во второй нормальной форме.

**Аномалия включения:** новый товар не включается в БД, если поставщик неизвестен.

**Аномалия удаления:** последний поставщик прекращает поставлять товар — теряются сведения о товаре.

**Аномалия обновления:** изменение цены влечет полный пересмотр таблицы.

Преобразование:

*поставки*(Поставщик, Товар)

*цена товара*(Товар, Цена)

## 2 Нормальная форма

### пример (3)

Проектируется учебная база данных, которая должна содержать список студентов. Предлагается схема

***студент***(*Группа*, *Номер*, *ФИО*),

где *Группа* — номер группы, а *Номер* — номер студента в списке группы. Выясняется, что работать с номерами студентов неудобно, и атрибут *Номер* заменяется на *Номер зачетки*:

***студент***(*Группа*, *Номер зачетки*, *ФИО*).

Но *Номер зачетки* однозначно определяет студента! Таким образом, ключ превратился в суперключ, а таблица перестала быть во 2НФ, что привело к появлению аномалий (например, студента нельзя включить в список, если он не распределен в конкретную группу).

### 3 Нормальная форма

У отношения во 2НФ, могут быть другие аномалии. Они обнаруживаются, если есть функциональная зависимость атрибута от непервичных.

Атрибут  $A$  *транзитивно зависит* от  $X$ , если есть  $Y$  такой, что выполняется условие

$$(X \rightarrow Y) \ \& \ (Y \rightarrow A) \ \& \ (X \rightarrow A), \ A \notin XY$$

и  $X$  функционально не зависит от  $Y$ .

Отношение находится в третьей нормальной форме (3НФ), если оно находится в 1НФ и в нём нет транзитивной зависимости атрибутов от первичных.

# 3 Нормальная форма

## пример (1)

Решается задача определения складов для отделений больницы.

Для руководства больницы важно, чтобы у каждого отделения был склад определенного объема, причем только один.

Считается, что один склад может предназначаться для разных отделений.

У отношения со схемой **хранение**(Отделение, Склад, Объем) есть одна функциональная зависимость:

Отделение → Склад

Но если нужно следить и за состоянием складов отдельно, возникает функциональная зависимость Склад → Объем, и отношение оказалось не в 3НФ.

# 3 Нормальная форма

## пример (2)

**Аномалия включения:** нет возможности добавить сведения о новом складе, для которого нет отделения.

**Аномалия удаления:** ликвидируется последнее отделение, работающее с некоторым складом, — данные об этом складе становятся недоступными.

**Аномалия обновления:** изменение объема склада влечет полный пересмотр таблицы.

Преобразование:

**хранение**(Отделение, Склад)

**объем склада**(Склад, Объем)



# Нормальная форма Бойса–Кодда (НФБК)

Нормальная форма Бойса–Кодда — одно из базовых понятий в теории нормальных форм, но оно вызывает неоднозначное отношение исследователей.

Некоторые считают НФБК частным случаем 3НФ, другие выделяют её как самостоятельную.

Существуют эквивалентные определения НФБК.

**Определение.** *Отношение находится в нормальной форме Бойса–Кодда (НФБК), если оно находится в 1НФ и в нем отсутствует зависимость первичных атрибутов от непервичных.*

# Нормальная форма Бойса–Кодда

## пример (1)

Студенты занимаются спортом, для каждого вида спорта секция единственная. В секции может быть более одного тренера, но каждый тренер работает только в одной секции. По одному виду спорта студент тренируется только у одного тренера, но может заниматься разными видами спорта.

Очевидно, что существует функциональная зависимость

Студент, Вид спорта → Тренер

Казалось бы, схема отношения должна быть такой:

**тренировка**(Студент, Вид спорта, Тренер)

Однако из условия единственности вида спорта, которым занимается тренер, следует функциональная зависимость:

Тренер → Вид спорта

# Нормальная форма Бойса–Кодда

## пример (2)

**Аномалия включения:** тренер не попадает в БД, пока нет занимающихся у него студентов.

**Аномалия удаления:** последний студент, который занимается у данного тренера, ушёл — тренера в базе данных не стало.

**Аномалия модификации:** замена тренера требует пересмотра всей таблицы.

Преобразование:

*тренировка*(Студент, Тренер)

*тренер*(Тренер, Вид спорта)

# Многозначная зависимость

Ошибка при проектировании баз данных — наличие в отношении атрибутов, связанных зависимостью «один ко многим» или «многие ко многим», где детерминант содержит один из возможных ключей.

*Для  $X, Y, Z \subset R$  существует многозначная зависимость  $Y$  от  $X$  (обозначается  $X \twoheadrightarrow Y$ ), если каждому значению  $X$  соответствует множество значений  $Y$ , не зависящих от  $Z$ .*

То есть, множество значений  $Y$ , соответствующее паре  $(X, Z)$ , зависит от  $X$  и не зависит от  $Z$ . Многозначная зависимость тривиальна, если либо  $X \subseteq Y$ , либо  $Z = \emptyset$ .

# Многозначная зависимость

## пример

Для базы данных учебной работы задали отношение

***преподаватель***(Код преп, ФИО, Должность, Читаемый курс)

В нём указан уникальный ключ преподавателя и читаемые им курсы. Атрибуты Код преп и Читаемый курс находятся в отношении 1:М, каждый из этих атрибутов первичен.

Читаемый курс не зависит от других атрибутов, т. е. имеет место многозначная зависимость Код преп → читаемый курс.

Если здесь же регистрировать аспирантов, отношение примет вид

***преподаватель***(Код преп, ФИО, Должность, Читаемый курс,  
Код асп)

В нем, по аналогичным соображениям, появится еще одна многозначная зависимость Код преп → Код асп.

## 4 Нормальная форма

### определение

*Отношение находится в четвертой нормальной форме (4НФ), если оно находится в 3НФ и в нем для любой нетривиальной зависимости  $X \rightarrow Y$  множество атрибутов  $X$  содержит возможный ключ.*

Определения 4НФ у разных авторов несколько различаются. В некоторых работах требуется, чтобы отношение находилось в НФБК, в других этого требования нет, но есть дополнительные условия, следствием которых будет 3НФ или НФБК.

## 4 Нормальная форма

### пример

Зависимость между преподавателем, курсами и аспирантами из предыдущего примера приводит не только к избыточности, но и к тому, что при появлении нового аспиранта приходится добавлять столько кортежей, сколько курсов читает этот преподаватель, а при добавлении курса следует добавить столько кортежей, сколько у преподавателя аспирантов.

Преобразование:

***преп\_курс**(Код преп, Читаемый курс)*

***преп\_асп**(Код преп, Код асп)*

***преп**(Код преп, ФИО, Должность)*

.

# 5 Нормальная форма

Пусть для схемы  $R$  заданы подмножества  $R_1, R_2, \dots, R_k \subseteq R$ . Будем называть их множество  $\{R_1, R_2, \dots, R_k\}$  полной нетривиальной декомпозицией, если оно отлично от  $\{R, \emptyset\}$  и  $\{R, R\}$  и при этом  $R_1 \cup R_2 \cup \dots \cup R_k = R$ .

Зависимость по соединению говорит о том, что отношение может быть восстановлено без потерь соединением его проекций на элементы полной нетривиальной декомпозиции. Для отношения  $r(R)$ :

$$\pi_{R_1}(r) \parallel \pi_{R_2}(r) \parallel \dots \parallel \pi_{R_k}(r) = r(R).$$

**Определение.** Отношение находится в 5НФ, если оно находится в 1НФ и любая зависимость по соединению определяется его возможными ключами.



# 5 Нормальная форма

Здесь отношение не разбивается на другие. Оно просто может быть разбито на более удобные отношения и восстановлено, то есть, его проекции состоят из полностью соединимых кортежей.

Пример — факторизация отношения при работе с распределенными базами данных. Когда разделенное оператором фактора отношение будет соединено, мы должны быть уверены в его корректности.

Исследование 5НФ достаточно сложно, в ряде работ говорится об отсутствии ясной её интерпретации и сомнительной практической применимости. Однако это финальная форма: отношение, находящееся в ней, больше не нормализуется.

# 5 Нормальная форма

## пример (1)

Рассмотрим отношение  $R1$ . Оно находится в 4НФ: нет многозначных зависимостей. Посмотрим, находится ли оно в 5НФ. Если мы обнаружим не полностью соединимые проекции, значит, оно не в 5НФ.

**$R1$ :**

<u><math>A</math></u>	<u><math>B</math></u>	<u><math>C</math></u>
$a_1$	$b_1$	$c_1$
$a_1$	$b_1$	$c_2$
$a_2$	$b_1$	$c_1$
$a_2$	$b_1$	$c_2$
$a_3$	$b_1$	$c_1$
$a_3$	$b_1$	$c_2$
$a_3$	$b_2$	$c_1$
$a_3$	$b_2$	$c_2$

# 5 Нормальная форма

## пример (2)

Для проекций  $R2(\underline{A}, \underline{B})$  и  $R3(\underline{B}, \underline{C})$  условие  $R2 \parallel R3 = R1$  выполняется.

Но в соединении проекций  $R4(\underline{A}, \underline{C})$  и  $R3$  уже появляются лишние кортежи, например,  $\langle a_1, b_2, c_1 \rangle$ . Значит, исходное отношение не находится в 5НФ.

**$R2: (\underline{A}, \underline{B})$**

$a_1$	$b_1$
$a_2$	$b_1$
$a_3$	$b_1$
$a_3$	$b_2$

**$R3: (\underline{B}, \underline{C})$**

$b_1$	$c_1$
$b_1$	$c_2$
$b_2$	$c_1$
$b_2$	$c_2$

**$R4: (\underline{A}, \underline{C})$**

$a_1$	$c_1$
$a_1$	$c_2$
$a_2$	$c_1$
$a_2$	$c_2$
$a_3$	$c_1$

# Литература

1. *Атре Ш.* Структурный подход к организации баз данных.
2. *Дейт К.* Руководство по реляционной СУБД DB2.
3. *Каратыгин С. А., Тихонов А. Ф., Тихонова Л. Н.* Visual FoxPro 7.
4. *Кренке Д.* Теория и практика построения баз данных.
5. *Кузнецов С. Д.* Основы баз данных: учебное пособие.
6. *Марков А. С., Лисовский К. Ю.* Базы данных. Введение в теорию и методологию.
7. *Мейер Д.* Теория реляционных баз данных.