

БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Ст.преп. каф. ИТ Кеберле Наталья Геннадьевна

**Лекция 10. Нормализация отношений баз данных.
Нормальные формы. Нормализация через декомпозицию.**





На прошлой лекции

- ▶ Мы узнали о основной проблеме реляционной модели – избыточности хранения данных
- ▶ Мы узнали о том, что избыточности можно избежать, если разбить отношения БД на более мелкие
- ▶ Мы также узнали, что для этого можно использовать F-зависимости

- ▶ На этой лекции мы узнаем, как именно можно использовать F-зависимости

Нормализация

- ▶ Процедура нормализации состоит в постепенном формировании отношений, находящихся в так называемых «**нормальных формах**»
- ▶ Термин «нормальная форма» был предложен Коддом
- ▶ Существуют:
 - первая нормальная форма (1НФ),
 - вторая (2НФ),
 - третья (3НФ),
 - нормальная форма Бойса-Кодда (НФБК),
 - четвертая (4НФ)
 - и пятая (5НФ), связанные между собой так:

$1НФ \supset 2НФ \supset 3НФ \supset НФБК \supset 4НФ \supset 5НФ$

Первая нормальная форма (1НФ)

Схема отношения R находится в 1НФ, если для любого атрибута A из схемы R значения в $\text{dom}(A)$ являются атомарными (т.е. неделимыми на части)

семья (имя	пол)
	{Джон, Джеки, Иван}	м	
	{Анна, Мария}	ж	

Отношение *семья* (имя, пол) – не находится в 1НФ

Определить понятие атомарности трудно:

значение, атомарное в одном приложении, может быть неатомарным в другом

Считают, что значение **неатомарно**, если в приложении оно используется по частям

Первая нормальная форма (1НФ)

<i>семья1</i> (имя	пол)
	Джон	м	
	Джеки	ж	
	Иван	м	
	Анна	ж	
	Мария	м	

Отношение *семья1* (имя, пол)
– в 1НФ

Преимущества 1НФ:

- ❑ это способ выразить F-зависимости с нужной степенью детализации
- ❑ обновление данных находящихся в 1НФ - простое действие

Вторая нормальная форма (2НФ)

- ▶ Вторая и третья нормальные формы возникли для того, чтобы избежать аномалий обновления и избыточности данных.
- ▶ Будем рассматривать 2НФ и 3НФ на примере отношения с одним потенциальным ключом, который является и первичным ключом.



Частичная зависимость

Пусть

R - схема отношений,

$X, Y \subseteq R$

S - множество функциональных зависимостей для схемы R ,

$X \rightarrow Y$ – F-зависимость из S^+

Говорят, что **Y частично зависит от X относительно S** , если существует X' -собственное подмножество X

$X' \subset X$, такое, что

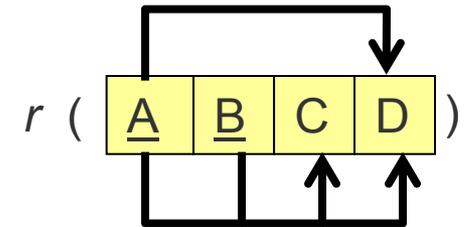
$(X' \rightarrow Y)$ - также является F-зависимостью из S^+ .

Иначе – **Y полностью зависит от X относительно S**



Частичная зависимость - пример

<u>A</u>	<u>B</u>	C	D
a1	b1	c1	d1
a1	b2	c2	d1
a2	b3	c1	d2



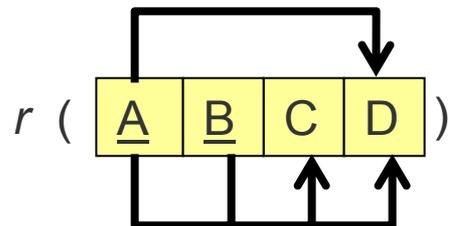
Дано отношение $r(\underline{AB}CD)$

т.к. AB – первичный ключ, выполняются такие F-зависимости: $AB \rightarrow C, AB \rightarrow D$

Однако, исходя из внешнего вида данного отношения, можно заметить и F-зависимость $A \rightarrow D$

Т.е. все множество F-зависимостей будет таким:
 $S = \{ AB \rightarrow C, AB \rightarrow D, A \rightarrow D \}$

Частичная зависимость - пример



Атрибут D частично зависит от AB

- ▶ Будем называть атрибут $A \in R$ **первичным** в R относительно множества F -зависимостей S , если A содержится в каком-либо потенциальном ключе схемы R .
- ▶ Иначе атрибут A - **непервичный**.

Атрибуты C и D - непервичные, A и B – первичные (т.к. части первичного ключа)

Определение 2НФ

Схема отношения R находится в 2НФ относительно множества F -зависимостей S , если она находится в 1НФ и каждый непервичный атрибут полностью зависит от каждого ключа для R .

<u>A</u>	B	C	D
a1	b1	c1	d1
a1	b2	c2	d1
a2	b3	c1	d2

Отношение $r(ABCD)$ не находится в 2НФ, т.к. в нём есть частичная зависимость $A \rightarrow D$

Транзитивные зависимости

Пусть R - схема отношения,
 S - множество F-зависимостей для схемы R ,
 $X \subseteq R$ (подмножество R),
 $A \in R$ (атрибут из R).

Говорят, что **атрибут A транзитивно зависит от X** в R , если существует подмножество атрибутов Y
 $Y \subseteq R$, такое, что
 $X \rightarrow Y$, $\neg(Y \rightarrow X)$, $Y \rightarrow A$ относительно S ,
и $A \notin X$, $A \notin Y$
(хотя X и Y могут иметь общие атрибуты).



Транзитивная зависимость - пример

<u>A</u>	<u>B</u>	C	D	E
a1	b1	c1	d1	e1
a1	b2	c2	d1	e1
a2	b3	c1	d2	e2
a2	b2	c3	d1	e1

Дано отношение $r(\underline{A}\underline{B}CDE)$

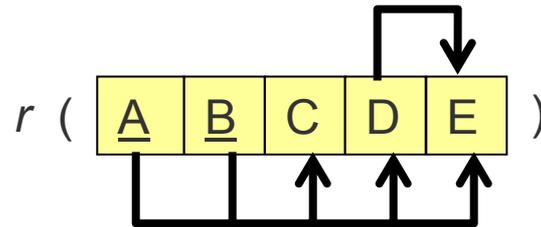
AB – первичный ключ, выполняются такие F-зависимости: $AB \rightarrow C, AB \rightarrow D, AB \rightarrow E$

Исходя из внешнего вида данного отношения, можно заметить и F-зависимости $D \rightarrow E$ и $E \rightarrow D$

Т.е. все множество F-зависимостей будет таким:

$$S = \{ AB \rightarrow CDE, D \rightarrow E, E \rightarrow D \}$$

Транзитивная зависимость - пример



Атрибут E транзитивно зависит от AB , т.к. существует атрибут D , такой что:

$$AB \rightarrow D, \neg(D \rightarrow AB), D \rightarrow E, E \notin (A \cup B \cup D)$$

Атрибут D транзитивно зависит от AB по аналогичным соображениям

Определение 3НФ

Схема отношения R находится в 3НФ относительно множества F -зависимостей S , если она находится в 2НФ и ни один из первичных атрибутов в R не является транзитивно зависящим от ключа для R

r (<u>A</u>	<u>B</u>	C	D	E)
	a1	b1	c1	d1	e1	
	a1	b2	c2	d1	e1	
	a2	b3	c1	d2	e2	
	a2	b2	c3	d1	e1	

Отношение $r(ABCDE)$ не находится в 3НФ, т.к. в нём есть транзитивные зависимости $D \rightarrow E$ и $E \rightarrow D$

Промежуточный итог

- ▶ В нормализованном отношении $r(R)$ не разрешаются никакие F-зависимости, кроме F-зависимостей вида $K \rightarrow A$, где K - потенциальный ключ для схемы R , а A – непервичный атрибут.
- ▶ Если же схема R удовлетворяет F-зависимости $K' \rightarrow B$, и K' не является потенциальным ключом, то в отношении $r(R)$ обязательно будет избыточность данных.
- ▶ Значит, процедура нормализации отношений со схемой R должна отсекают из всего множества F-зависимостей S для этой схемы частичные и транзитивные зависимости от ключей



Способы нормализации

▶ Существует два способа:

■ Синтез

- исходные данные: все F-зависимости, установленные между атрибутами предполагаемых отношений.
- алгоритм синтеза: конструирует отношения, заведомо находящиеся в 3НФ (при этом используется понятие неприводимого минимального покрытия T множества функциональных зависимостей S).

■ Декомпозиция

- исходные данные: набор отношений и множества F-зависимостей для каждого отношения.
- алгоритм: исходные отношения дробятся на более мелкие, но находящиеся в 3НФ, отношения



Нормализация декомпозицией

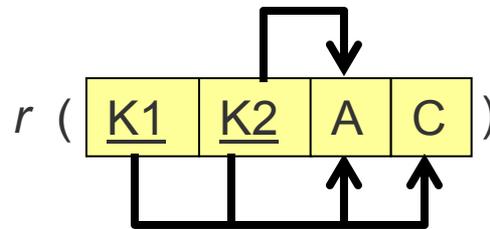
При декомпозиции рассматривают всего два основных случая.

Первый случай

Отношение $r(R)$ имеет составной первичный ключ $(K1, K2)$,

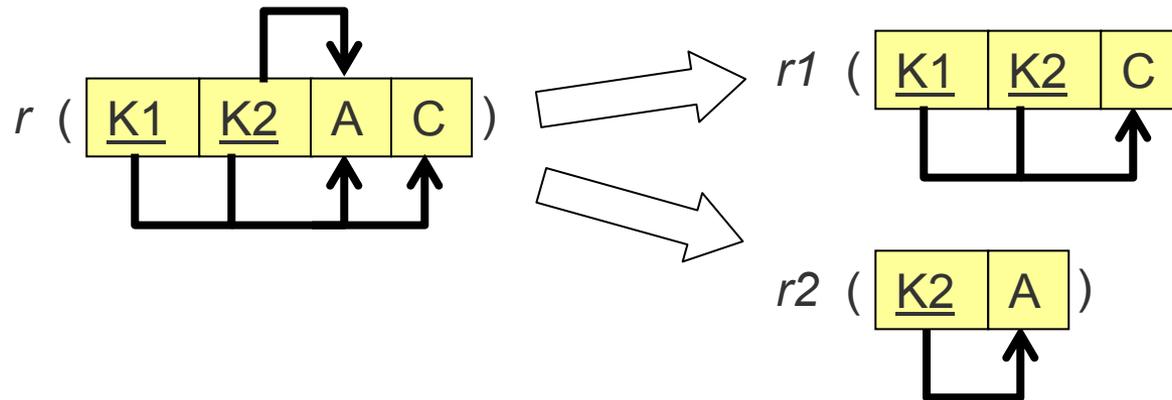
атрибут **A**, **частично** зависящий от первичного ключа,

и атрибут **C**, **полностью** зависящий от первичного ключа:



Нормализация декомпозицией

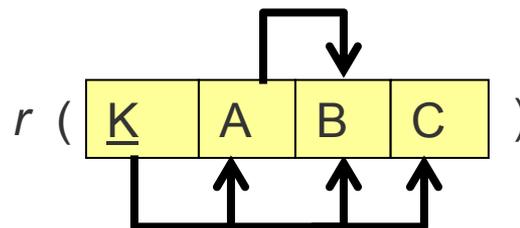
При декомпозиции $r(\underline{K1}, \underline{K2}, A, C)$ разбивается на два отношения, $r1(\underline{K1}, \underline{K2}, C)$ и $r2(\underline{K2}, A)$



Нормализация декомпозицией

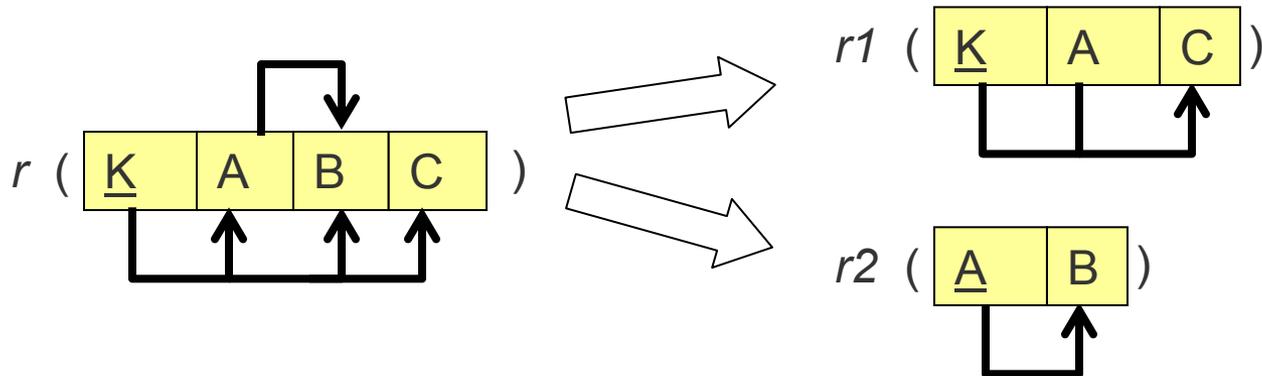
Второй случай

Отношение $r(R)$ имеет первичный ключ K , атрибуты A и B , такие, что существует функциональная зависимость $A \rightarrow B$ (т.е. A транзитивно зависит от ключа), атрибут C , полностью зависящий от первичного ключа



Нормализация декомпозицией

При декомпозиции $r(\underline{K}, A, B, C)$ разбивается на два отношения, $r1(\underline{K}, A, C)$ и $r2(\underline{A}, B)$



Сравнение методов нормализации

- ▶ **Метод синтеза** не требует какой-либо логической схемы базы данных. Достаточно лишь, изучив предметную область, предоставить **все** интересующие нас функциональные зависимости между свойствами сущности (связей). Однако реализация синтеза достаточно сложна и требует обработки множества функциональных зависимостей S и создания минимального покрытия T .



Сравнение методов нормализации

- ▶ **Метод декомпозиции** предполагает, что в результате исследования предметной области была построена исходная концептуальная схема, а затем ее преобразовали в набор реляционных таблиц.
- ▶ Реализация метода декомпозиции проста. Поэтому на практике чаще используют декомпозицию.
- ▶ Построив концептуальную модель, и корректно отобразив ее в логическую схему базы данных, разработчик должен проверить с помощью нормализации все отношения реляционной базы данных, а если нужно, то проделать декомпозицию.



Недостатки метода декомпозиции

- ▶ Временная сложность – неполиномиальна.
- ▶ Проверка атрибута схемы на первичность – NP-полная задача (которая решается простым перебором).
- ▶ Число порожденных процессом схем может быть больше, чем необходимо для 3НФ.
- ▶ При декомпозиции могут возникнуть новые частичные зависимости, устранение которых потребует создания новых отношений.
- ▶ Для построенной схемы базы данных во множестве F-зависимостей S может отыскаться F-зависимость, которая не будет выполняться в новых отношениях.

Недостатки декомпозиции - пример

Дано отношение $r(\underline{A} B C D E)$

$S = \{A \rightarrow BCDE, CD \rightarrow E, EC \rightarrow B\}$

Исключив транзитивную зависимость E от A через CD , получаем два отношения $r1(\underline{A} B C D)$ и $r2(\underline{C D} E)$. Функциональная зависимость $EC \rightarrow B$ уже не применима ни к отношению $r1$, ни к отношению $r2$.

Исключив транзитивную зависимость B от A через EC , получаем два таких отношения:

$r1(\underline{A} E C D)$ и $r2(\underline{E C} B)$. Тогда не выполняется функциональная зависимость $CD \rightarrow E$.



Недостатки декомпозиции

- ▶ Могут появиться “скрытые” транзитивные зависимости
- ▶ Например:

Дано отношение $r(A B C D)$

$S = \{AD \rightarrow ADCB, A \rightarrow B, B \rightarrow C\}$

Исключая частичную зависимость B от AD (через $A \rightarrow B$) получаем $r1(A D C)$ и $r2(A B)$. Несмотря на то, что формально $r1$ и $r2$ находятся в 3НФ, в $r1$ появляется скрытая транзитивная зависимость C от AD через B .



Что узнали сегодня?

- ▶ В реляционной модели данных предлагается решение проблемы возможной избыточности данных в БД – нормализация схемы БД
- ▶ Степень нормализованности схемы – нормальная форма
- ▶ Нормальных форм – шесть (от 1НФ к 5НФ)
- ▶ Основанием для нормализации служат особые шаблоны F-зависимостей: частичная и транзитивная.
- ▶ Исключая частичные зависимости, получаем 2НФ
- ▶ Исключая транзитивные зависимости, получаем 3НФ

