

7 Реляционная модель данных

реляционная модель данных - совокупность данных, состоящая из набора двумерных таблиц.

структура модели:

- отношение (relation) - двумерная таблица. каждая строка таблицы содержит данные о конкретном объекте или его части. каждый столбец таблицы описывает какой-либо атрибут объекта. строки называют кортежами, столбцы - атрибутами. атрибуты попарно различны.
- схема отношения - перечень имен атрибутов данного отношения с указанием домена, к которому они относятся. множество всех атрибутов отношения.
- домен - множество допустимых значений для атрибута, обладающих необходимыми свойствами. скалярные константы.
- первичный ключ - группа из одного или более атрибутов, которая уникальным образом идентифицирует строку.
- внешний ключ - набор атрибутов в подчиненном отношении соответствующих первичному ключу основного отношения.

ограничения целостности модели:

- ячейки таблицы должны содержать одиночные значения, в качестве значений не допускаются ни повторяющиеся группы, ни массивы.
- все записи в одном столбце должны соответствовать одному и тому же домену.
- каждый атрибут должен иметь уникальное имя, порядок атрибутов несущественен.
- двух одинаковых кортежей быть не может, порядок несущественен.

операции реляционной алгебры:

- обновление данных
 - добавление - вставка нового кортежа в отношение может быть не выполнено, если:
 1. кортеж не соответствует схеме отношения
 2. некоторые значения кортежа не принадлежат соответствующим доменам
 3. кортеж совпал по ключу с уже находящимся в отношении
 - удаление кортежа из отношения может быть не выполнено, если:
 1. удаляемого кортежа нет в отношении
 2. удаляемый кортеж не соответствует схеме отношения
 - изменение (обновление) кортежа из отношения может быть не выполнено, если:
 1. изменяемого кортежа нет в отношении
 2. изменяемый кортеж не соответствует схеме отношения
- операции над множествами:
 - объединение (UNION) - в результат включаются кортежи, входящие в первый или второй аргумент

- пересечение (INTERSECT) - в результат включатся только кортежи, входящие в первый и второй аргументы
- разность (EXCEPT) - в результат включатся кортежи, входящие в первый и не входящие во второй аргумент.
условия выполнения: схемы отношений должны быть эквивалентны.
- еще операции:
 - проекция (PROJ) - в результат включится подмножество атрибутов, переданных в качестве аргументов. проекция отношения R на набор атрибутов B - отношение со схемой, соответствующей B, содержащее кортежи исходного отношения Rm из которого удалены значения, не принадлежащие атрибутам из набора B. исключает возникновение дубликатов.
 - селекция (FILTER) - операция выбора только тех кортежей, которые имеют заданное значение на заданном атрибуте. условие составления: атрибут можно сравнить с константой или другим атрибутом этого отношения, условия можно соединять логическими операторами AND, OR, NOT и заключать в круглые скобки.
 - соединение (JOIN) отношений R1 (A, B, C) и R2 (B, C, D) - отношение R (A, B, C, D), полученное по условию равенства общих атрибутов связи (B, C), из которого исключены дубли атрибутов. предназначено для соединения двух отношений с разными схемами. выполнится, если заданные атрибуты есть в схеме отношений.

1НФ (первая нормальная форма) - любое отношение, удовлетворяющее ограничениям целостности (все атрибуты одномерны, то есть не являются списками и т.д.)

2НФ (вторая нормальная форма) - если отношение находится в 1НФ, и в нем отсутствуют неполные функциональные зависимости (все неключевые атрибуты зависят от ключа)

функциональная зависимость: $X \rightarrow Y$ атрибут X функционально определяет атрибут Y, если каждому значению атрибута X соответствует единственное значение атрибута Y.

Пример [\[править \]](#) [\[править код \]](#)

Пример приведения отношения ко второй нормальной форме

Пусть в следующем отношении **первичный ключ** образует пара атрибутов {*Филиал компании*, *Должность*}:

<i>R</i>			
<u>Филиал компании</u>	<u>Должность</u>	Зарплата	Наличие компьютера
Филиал в Томске	Уборщик	20000	Нет
Филиал в Москве	Программист	40000	Есть
Филиал в Томске	Программист	25000	Есть

Допустим, что зарплата зависит от филиала и должности, а наличие компьютера зависит только от должности.

Существует функциональная зависимость $Должность \rightarrow Наличие\ компьютера$, в которой левая часть (детерминант) является лишь частью первичного ключа, что нарушает условие второй нормальной формы.

Для приведения к 2NF исходное отношение следует декомпозировать на два отношения:

<i>R1</i>		
<u>Филиал компании</u>	<u>Должность</u>	Зарплата
Филиал в Томске	Уборщик	20000
Филиал в Томске	Программист	25000
Филиал в Москве	Программист	40000

<i>R2</i>	
<u>Должность</u>	Наличие компьютера
Уборщик	Нет
Программист	Есть

ЗНФ (третья нормальная форма) - если отношение находится в 2НФ, и в нем отсутствуют транзитивные зависимости.

$R(A, B, C) : A \rightarrow B \rightarrow C$

Пример [\[править \]](#) [\[править код \]](#)

Рассмотрим в качестве примера переменную отношения R1:

<i>R1</i>		
<u>Сотрудник</u>	<u>Отдел</u>	Телефон
Гришин	Бухгалтерия	11-22-33
Васильев	Бухгалтерия	11-22-33
Петров	Снабжение	44-55-66

Каждый сотрудник относится исключительно к одному отделу; каждый отдел имеет единственный телефон. Атрибут *Сотрудник* является первичным ключом. Личных телефонов у сотрудников нет, и телефон сотрудника зависит исключительно от отдела.

В примере существуют следующие функциональные зависимости: $Сотрудник \rightarrow Отдел$, $Отдел \rightarrow Телефон$, $Сотрудник \rightarrow Телефон$.

Переменная отношения R1 находится во **второй нормальной форме**, поскольку каждый атрибут имеет неприводимую функциональную зависимость от потенциального ключа *Сотрудник*.

Зависимость $Сотрудник \rightarrow Телефон$ является транзитивной, следовательно, отношение не находится в третьей нормальной форме.

В результате разделения R1 получаются две переменные отношения, находящиеся в 3NF:

Исходное отношение R1 при необходимости легко получается в результате **операции соединения** отношений R2 и R3.

НФБК/BCNF (нормальная форма Бойса — Кодда) - если каждая нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта некоторый потенциальный ключ.

то есть если есть зависимость атрибута В от атрибута А, то А может быть ключом отношения.

Пример [\[править | править код \]](#)

Предположим, рассматривается отношение, представляющее данные о бронировании теннисных кортов на день:

Бронирования кортов			
Номер корта	Время начала	Время окончания	Тариф
1	09:30	10:30	«Корт 1 для членов клуба»
1	11:00	12:00	«Корт 1 для членов клуба»
1	14:00	15:30	«Корт 1 для не членов клуба»
2	10:00	11:30	«Корт 2 для не членов клуба»
2	11:30	13:30	«Корт 2 для не членов клуба»
2	15:00	16:30	«Корт 2 для членов клуба»

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

Отношение соответствует второй (2NF) и третьей (3NF) нормальной форме. 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 корта**, 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 «Корт 1 для членов клуба» к бронированию второго корта, хотя он может относиться только к первому корту.

Можно улучшить структуру с помощью декомпозиции отношения на два, получив отношения, удовлетворяющие BCNF (подчёркнуты атрибуты, входящие в первичный ключ). Для большей наглядности к информации о тарифах добавлен атрибут **Для членов клуба**:

Тарифы			Бронирования		
Тариф	Номер корта	Для членов клуба	Тариф	Время начала	Время окончания
«Корт 1 для членов клуба»	1	Да	«Корт 1 для членов клуба»	09:30	10:30
«Корт 1 для не членов клуба»	1	Нет	«Корт 1 для членов клуба»	11:00	12:00
«Корт 2 для членов клуба»	2	Да	«Корт 1 для не членов клуба»	14:00	15:30
«Корт 2 для не членов клуба»	2	Нет	«Корт 2 для не членов клуба»	10:00	11:30
			«Корт 2 для не членов клуба»	11:30	13:00
			«Корт 2 для членов клуба»	15:00	16:30