

Лекция № 11.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель данных

Согласно К. Дейту реляционная модель данных (РМД), как любая другая модель, состоит из трех частей: структурной, целостной и манипуляционной.

Именно в таком порядке мы будем рассматривать РМД.

В реляционных базах данных (далее – РБД) *единственной структурой данных* является ***нормализованное n-арное отношение***.

(Отсюда и название "реляционные", т. к. по-английски отношение "relation".)

Отношение называется **нормализованным**, если значения всех его атрибутов (аргументов) являются атомарными, т.е. имеют простой тип и не включают множества значений.

Реляционная модель данных

В математике отношение R определяется следующим образом:

Отношение R , заданное на n множествах $S1, S2, \dots S_n$,
есть набор кортежей вида $\langle s_1^i, s_2^i, \dots s_n^i \rangle$,
таких что $s_1^i \in S1, s_2^i \in S2, \dots s_n^i \in S_n$.

Заметим, что не обязательно всем множествам быть различными.

В зависимости от n отношения называются бинарными, тернарными и т.п.

Самый простой способ представления n -арного отношения – это таблица, включающая n столбцов. Поэтому в РБД отношения называют еще и **таблицами**.

Реляционная модель данных

1. Основные понятия

Основными понятиями РБД являются *тип данных, домен, атрибут, кортеж, отношение, первичный ключ*.

Тип данных — как и в традиционном программировании — это множество значений и совокупность операций над ними.

Типами данных в РБД являются числа (целые и вещественные), строки (символы), логические значения, даты и другие специальные типы, как, например, деньги.

Домен — это подмножество значений данного типа или подтип данного типа, полученный из него путем накладывания ограничений на его значения (в виде логических выражений или перечислением конкретных значений).

Реляционная модель данных

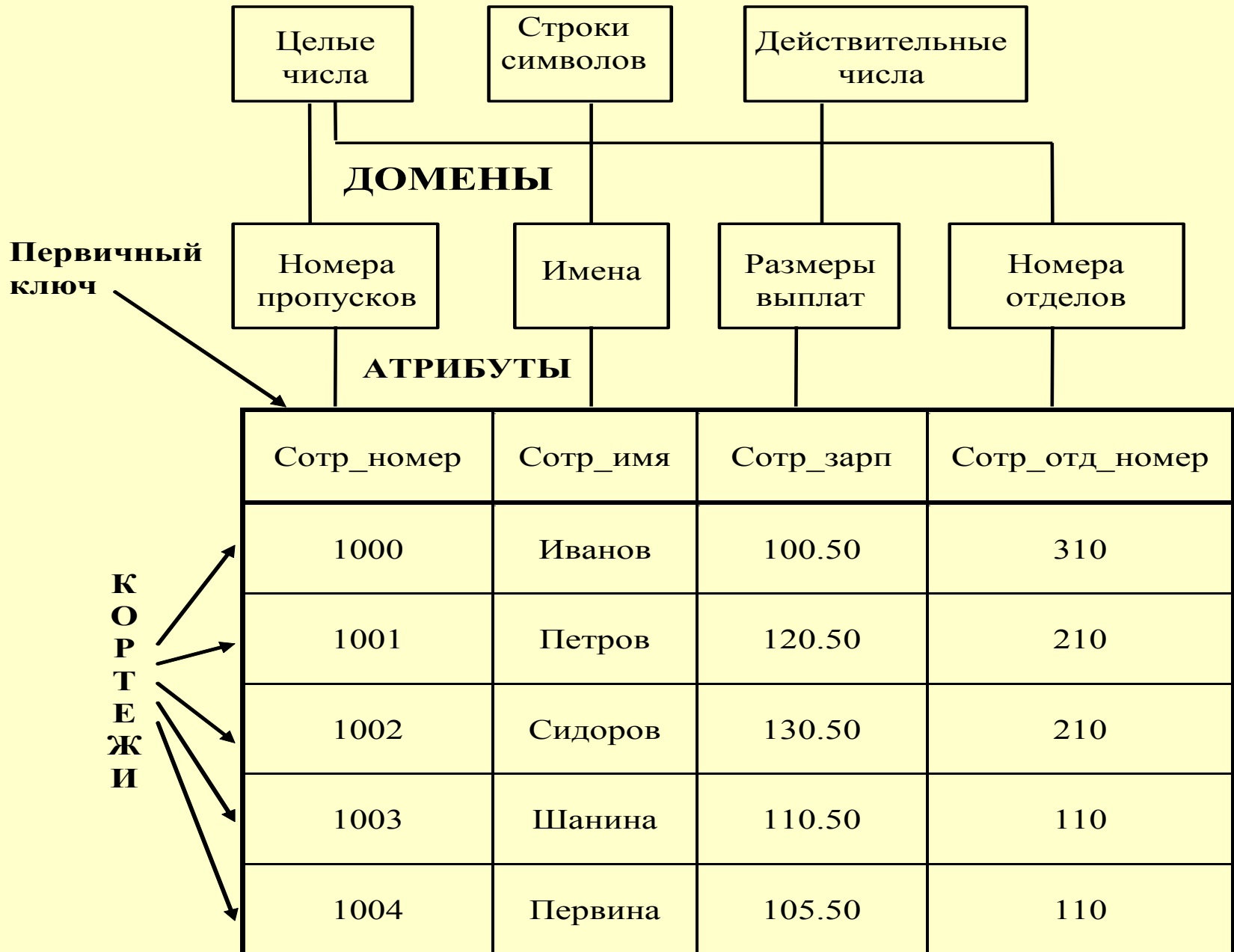
Например, домен "Размеры выплат" получен из типа данных "Действительные числа" путем накладывания на его значения ограничения вида $V \leq 1000$.

Домен "Номера отделов" является подмножеством значений целых чисел и может быть задан перечислением вида {110, 210, 310}.

Домен "Имена" определен на базовом типе "Строки символов", но включает только те строки, которые могут изображать имя.

Семантическая нагрузка домена состоит в том, что данные сравнимы, если они относятся к одному и тому же домену. Это вводит дополнительный контроль за правильным использованием данных и манипулированием ими.

ТИПЫ ДАННЫХ



Реляционная модель данных

Схема отношения – это именованное множество пар вида *<Имя атрибута, Имя домена>* или *<Имя атрибута, Имя типа>*.

Степень или "арность" схемы отношения – мощность множества таких пар.

Схема 4-арного отношения *СОТРУДНИКИ*, приведенного на рисунке, имеет вид:

СОТРУДНИКИ
{
 <Сотр_номер, Номера_пропусков>,
 <Сотр_имя, Имена>,
 <Сотр_зарп, Размеры_выплат>,
 <Сотр_отд_номер, Номера_отделов> }

Сокращенный способ описания схемы отношения:

СОТРУДНИКИ
(*Сотр_ном, Сотр_имя, Сотр_зарп, Сотр_отд_номер*).

Реляционная модель данных

Схема БД – это набор именованных отношений.

Кортеж – это множество пар *<Имя атрибута, Значение>*, содержащее только одно вхождение каждого *имени атрибута*, принадлежащего схеме отношения. *Значение* является допустимым значением домена данного атрибута.

Арность кортежа совпадает с арностью схемы отношения. На практике имя атрибута в кортеже опускается, тогда его роль играет номер позиции, которую атрибут занимает в схеме отношения.

Проще говоря, *кортеж* – это набор именованных значений заданного типа.

Отношение – это множество кортежей, соответствующих одной схеме отношения.

Реляционная модель данных

Обычным представлением отношения является плоская *двумерная таблица*, заголовком которой является *схема отношения*, а строками – *кортежи отношения*. В этом случае *атрибуты* именуют *столбцы* этой таблицы.

Реляционная БД – это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

Ключ – это атрибут или набор атрибутов, используемых для идентификации кортежа.

Первичный или **основной ключ** – это атрибут или набор атрибутов, значения которых однозначно определяют кортеж отношения.

Важным требованием к первичному ключу является его *минимальность* или *отсутствие избыточности*: никакой атрибут нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации кортежей.

Реляционная модель данных

Возможно наличие нескольких первичных ключей в отношении. Обычно используют тот из них, который имеет минимальное число атрибутов.

В схеме отношения атрибуты, составляющие первичный ключ, как правило, подчеркиваются:

СОТРУДНИКИ (Сотр номер, Сотр_имя, Сотр_зарп,
Сотр_отд_номер).

Ключ, который используется для выделения всех кортежей, обладающих определенным свойством, называется *дополнительным* или **вторичным ключом**.

Например, атрибут *Сотр_отд_номер* может использоваться для поиска всех сотрудников из отдела номер 310.

Возможно использование нескольких вторичных ключей.

Базовые свойства отношений

Из приведенных определений следуют следующие свойства отношений:

1) Отсутствие кортежей дубликатов

Это следует из определения отношения как множества кортежей (в классической теории множеств каждое множество состоит из различных элементов).

Из этого свойства вытекает *наличие у каждого отношения первичного ключа*. (По крайней мере, полный набор атрибутов является первичным ключом.)

2) Отсутствие упорядоченности кортежей

Это свойство также вытекает из свойств множества.

Отсутствие требования порядка дает дополнительную гибкость СУБД при хранении данных во внешней памяти и обработке запросов к БД.

Базовые свойства отношений

3) Отсутствие упорядоченности атрибутов

Атрибуты не упорядочены, так как по определению схема отношения есть множество пар вида *<Имя атрибута, Значение>* (для ссылки на значение атрибута всегда используется его имя). Это свойство позволяет модифицировать схемы существующих отношений не только путем добавления новых атрибутов, но и удалением существующих. Позволяет также соединять отношения, выдавать атрибуты в любом порядке, хотя часто на практике используется неявный порядок атрибутов.

4) Атомарность значений атрибутов

Это свойство следует из определения домена как потенциального множества значений простого типа данных.

Ограничения целостности

В любой реляционной СУБД должны удовлетворяться два базовых требования целостности:

- 1) требование целостности сущностей;
- 2) требование целостности по ссылкам.

Ограничения целостности

1. Целостность сущностей

Сущности, или объекту реального мира в РБД соответствуют кортежи отношений.

Любой кортеж любого отношения должен быть отличим от любого другого кортежа этого же отношения.

Другими словами, *любое отношение должно обладать первичным ключом*.

Это требование автоматически удовлетворяется, если в системе не нарушаются *базовые свойства отношений*.

Ограничения целостности

2. Целостность по ссылкам

При соблюдении нормализованности отношений сложные сущности реального мира представляются в РБД в виде нескольких кортежей нескольких отношений.

Например, сущность *ОТДЕЛ* имеет следующие характеристики: *номер отдела, количество сотрудников и список сотрудников отдела*. При правильном проектировании эта сущность представляется в виде двух отношений:

ОТДЕЛЫ (Отд_номер, Отд_кол).

***СОТРУДНИКИ (Сотр_номер, Сотр_имя, Сотр_зарп,
Сотр_отд_номер).***

В отношении *СОТРУДНИКИ* атрибут *Сотр_отд_номер* является не только собственным свойством сотрудника, но и обеспечивает при необходимости возможность восстановления полной сущности *ОТДЕЛ*.

Ограничения целостности

ОТДЕЛЫ (*Отд_номер*, *Отд_кол*).

СОТРУДНИКИ (*Сотр_номер*, *Сотр_имя*, *Сотр_зарп*,
Сотр_отд_номер).

Значение атрибута *Сотр_отд_номер* в любом кортеже отношения *СОТРУДНИКИ* должно соответствовать значению атрибута *Отд_номер* в некотором кортеже отношения *ОТДЕЛЫ*.

Такой атрибут, как *Сотр_отд_номер* называется **внешним ключом**, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают его *первичный ключ*).

Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут (набор атрибутов) является первичным ключом.

Ограничения целостности

С учетом сказанного выше, *требование целостности по ссылкам* можно сформулировать как *требование внешнего ключа*:

для каждого значения внешнего ключа из ссылающегося отношения в отношении, на которое ведется ссылка, должен найтись кортеж с таким же значением первичного ключа. В противном случае значение внешнего ключа должно быть *неопределенным*.

В нашем примере: если для сотрудника указан номер отдела, то такой отдел должен существовать.

Ограничения целостности

Ограничение целостности по ссылкам автоматически поддерживается следующим образом:

- 1) **при добавлении новых кортежей** достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа;
- 2) **при удалении кортежей** из отношения, на которое ведет ссылка, имеется три способа поддержания целостности:
 - а) запрещается удалять кортеж, на который существует ссылка (сначала нужно разобраться со ссылающимися кортежами);
 - б) при удалении кортежа, на который есть ссылка, во всех ссылающихся кортежах значение внешнего ключа устанавливается не-определенным;
 - в) при удалении такого кортежа удаляются все ссылающиеся на него кортежи (каскадное удаление).