

Лекция 2

Модели данных

Модель данных

– набор родовых понятий и признаков, которыми должны обладать все конкретные СУБД и управляемые ими БД, если они основываются на этой модели.

- Структурная часть
- Манипуляционная часть
- Целостная часть

Модель данных

- Структурная часть модели данных содержит основные логические структуры данных, которые могут применяться на уровне пользователя при организации БД.
- Манипуляционная часть содержит спецификацию одного или нескольких языков, предназначенных для написания запросов к БД. Эти языки могут быть абстрактными или законченными производственными языками.
- Основное назначение манипуляционной части модели данных – это обеспечить эталонный «модельный» язык БД, который должен поддерживаться в реализациях СУБД, соответствующих данной модели.
- Целостная часть модели данных специфицирует механизмы ограничений целостности, которые обязательно должны поддерживаться во всех реализациях СУБД, соответствующих данной модели.

Целостность и согласованность

- Целостность данных означает соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам. Каждое правило, налагающее некоторое ограничение на возможное состояние базы данных, называется ограничением целостности.
- Примеры таких правил:
 - возраст должен быть положительным;
 - возраст родителей не может быть меньше возраста их ребёнка;
- Целостность обеспечивает правдоподобность информации

Ранние модели данных

- Иерархическая модель
- Сетевая модель
- Модель данных инвертированных таблиц

Использовалась явная навигация в БД, оптимизация производилась пользователем, требовалось создание собственных интерфейсов.

Модель данных инвертированных таблиц

- Структурная часть
 - БД - это набор таблиц. Однако пользователям видны и хранимые таблицы, и индексы. Строки таблиц упорядочиваются системой в некотором порядке, видимом пользователям. Для каждой таблицы можно определить произвольное число ключей поиска, для которых строятся индексы.
- Манипуляционная часть
 - Операции позиционирования и выборки/изменения
 - Типичные операции: LOCATE FIRST, LOCATE NEXT, RETRIEVE, DELETE, ...
- Целостная часть
 - Поддержка целостности – обязанность приложения -> плохо!

Иерархическая модель

- Структурная часть

- Упорядоченный набор деревьев.
- Тип дерева состоит из одного «корневого» типа записи и упорядоченного набора типов поддеревьев, каждое из которых является некоторым типом дерева. Тип дерева представляет собой иерархически организованный набор типа записей.



- Целостная часть

- Автоматически поддерживается целостность ссылок между предками и потомками
- Основное правило: никакой потомок не может существовать без своего родителя (потомок без родителя не может по определению быть в схеме иерархической БД).

Иерархическая модель

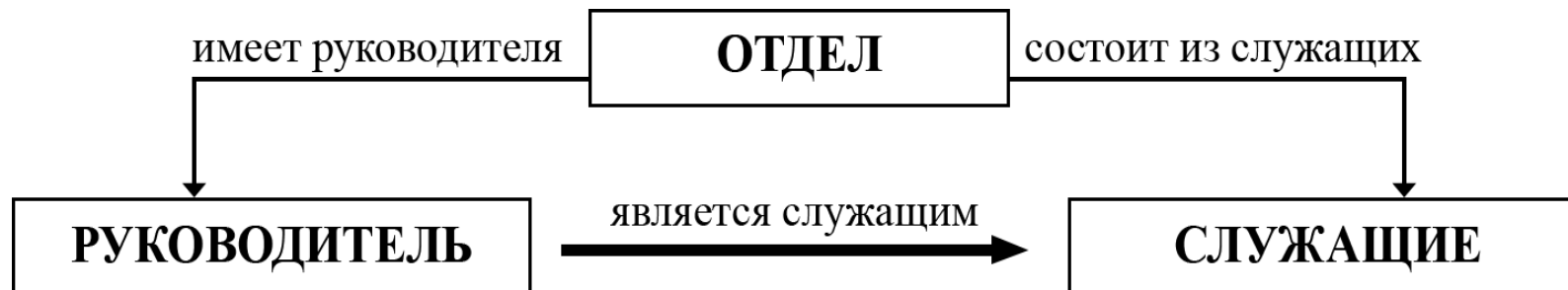
Для иерархической БД определяется полный порядок обхода дерева сверху-вниз, слева-направо.

Примеры операций манипулирования:

- найти указанный экземпляр типа дерева БД
- перейти от одного экземпляра типа дерева к другому
- перейти от одной записи к другой в порядке обхода иерархии (сверху-вниз, слева-направо)
- создать / уничтожить / модифицировать запись

Сетевая модель

- Расширение иерархического подхода: у потомка может иметься любое число предков (в том числе 0)
- Имеется возможность потребовать для конкретного типа связи отсутствие потомков (как и в иерархической модели)



Сетевая модель

Условия связи предок-потомок:

- предок может быть только в одном экземпляре связи
- если потомок имеет предка, то он единственный
- потомок в одной связи может быть предком в другой связи
- два типа записи могут быть предком-потомком в одной связи и потомком-предком в другой

Сетевая модель

Набор операций манипулирования данными:

- найти конкретную запись в наборе однотипных записей
- перейти от предка к потомку по некоторой связи
- перейти от потомка к предку по некоторой связи
- создать / уничтожить / модифицировать запись
- включить в связь / исключить из связи
- переставить в другую связь

Реляционная модель

- Основные идеи были изложены Э. Коддом ещё в 1969 году
- Развивалась в 2 направлениях:
 - Эдгар Кодд. IBM System R. Возникает язык SQL и фактически свой стандарт.
 - Кристофер Дейт, Хью Дарвен. В 1990-е годах пришло понимание, что SQL искажает реляционную модель данных и они разработали свою альтернативу.

Реляционные структуры данных

- Навигационная природа ранних баз данных заставляла разработчиков писать слишком много деталей реализации
- Основной подход: создание такой структуры данных, которая была бы понятна большинству разработчиков БД (не математику и не специалисту), но в то же время несла чёткий математический смысл
- В качестве такой структуры была выбрана **неупорядоченная таблица** (без порядка строк и столбцов)

Реляционные структуры данных

- Таблица с множеством столбцов $\{A_1, A_2, \dots, A_n\}$, где каждый столбец A_i может содержать значения из конечного множества $T_i = \{v_{i1}, v_{i2}, \dots, v_{im}\}$, в математическом смысле представляет собой отношение над множествами $\{T_1, T_2, \dots, T_n\}$
- В математике отношением над множествами $\{T_1, T_2, \dots, T_n\}$ называется подмножество декартова произведения этих множеств, т.е. множество кортежей $\{\{v_1, v_2, \dots, v_n\}\}$, где $v_i \in T_i$

Структура отношения

- Отношение – фундаментальное понятие в реляционном подходе к организации баз данных. Характеризуется парой множеств: телом и заголовком.
- Заголовком (или схемой) отношения называется конечное множество H упорядоченных пар вида $\langle A, T \rangle$, где A – имя атрибута, а T – обозначение некоторого базового типа или ранее определенного домена. Также все имена атрибутов в заголовке должны быть различные.

Структура отношения

- Тип данных. Аналогично типу данных в ЯП. Для типа данных определяется множество значений данного типа, множество операций, применимых к значениям типа, способ внешнего представления значений типа.
- Домен – определяется путем задания некоего базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу этого типа данных. Данные считаются сравнимыми тогда и только тогда, когда они относятся к одному и тому же домену.

Структура отношения

- Кортежем t , соответствующим заголовку H , называется множество упорядоченных триплетов вида $\langle A, T, v \rangle$, по одному такому триплету для каждого атрибута в H . v – допустимое значение типа данных или домена T .
- Телом отношения называется произвольное **множество** кортежей. В общем случае, могут существовать такие кортежи, которые соответствуют заголовку, но не входят в тело отношения.

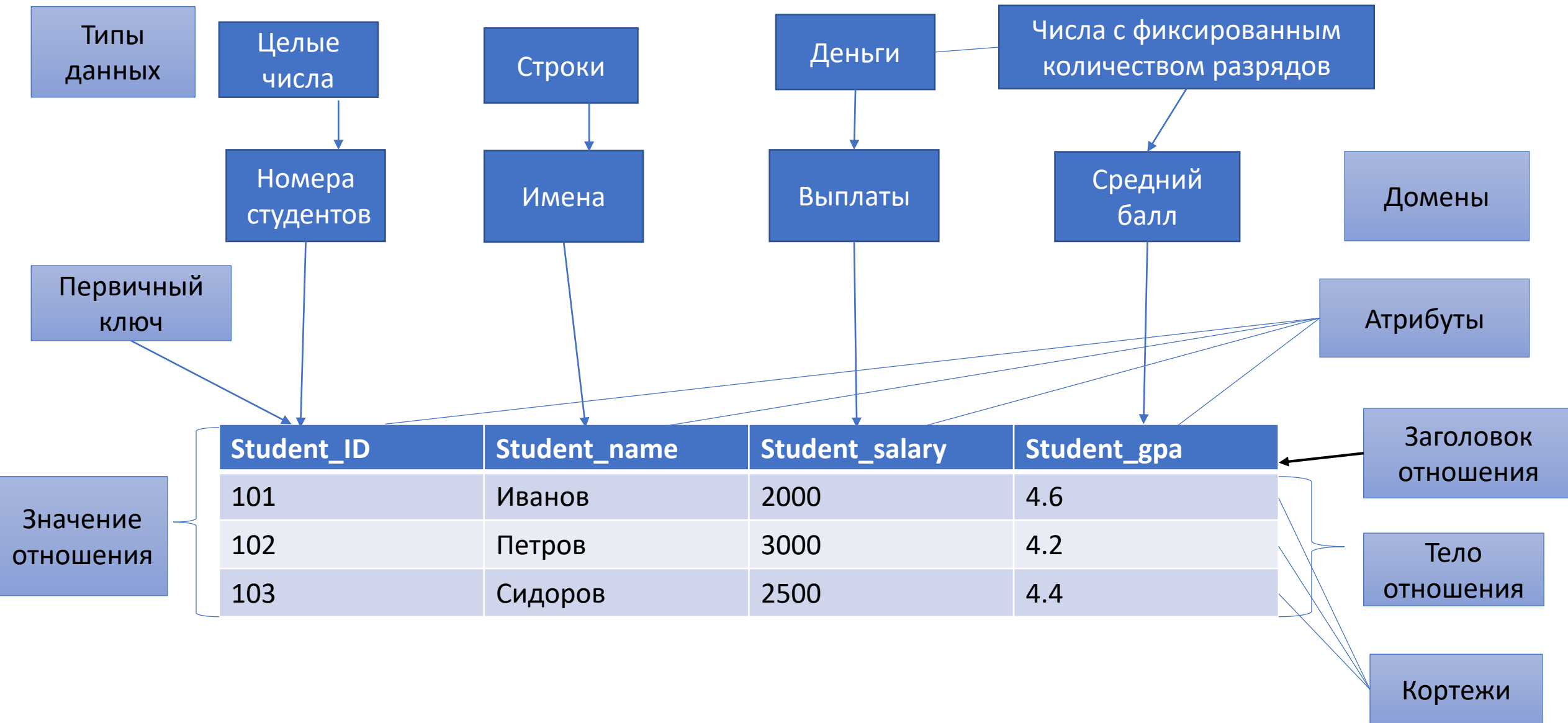
Реляционные структуры данных

Примеры описанных множеств:

Заголовок	$\langle A_1, T_1 \rangle$	$\langle A_2, T_2 \rangle$...	$\langle A_n, T_n \rangle$
Тело	$\langle A_1, T_1, v_{11} \rangle$	$\langle A_2, T_2, v_{12} \rangle$...	$\langle A_n, T_n, v_{1n} \rangle$
	$\langle A_1, T_1, v_{21} \rangle$	$\langle A_2, T_2, v_{22} \rangle$		$\langle A_n, T_n, v_{2n} \rangle$

$\langle \text{Имя, string} \rangle$	$\langle \text{дата рождения, date} \rangle$
$\langle \text{Имя, string, Вася} \rangle$	$\langle \text{дата рождения, date, 6/10/1980} \rangle$
$\langle \text{Имя, string, Петя} \rangle$	$\langle \text{дата рождения, date, 1/19/1991} \rangle$

Реляционные базы данных (понятия)



Манипулирование данными в РМД

- В качестве средства манипулирования реляционными данными был предложен определенный набор операций, который всегда возвращает отношения. Такой набор операций называется реляционной алгеброй Кодда.
- Кодд ввел 10 операций: объединение (UNION), пересечение (INTERSECT), вычитание (MINUS), взятие расширенного декартова произведения (TIMES), переименование атрибутов (RENAME), проекция (PROJECT), ограничение (WHERE), соединение (Θ -JOIN), деление (DIVIDE BY) и присваивание.

Алгебра Кодда

- При выполнении операции объединения (UNION) двух отношений с одинаковыми заголовками производится отношение, включающее все кортежи, входящие хотя бы в одно из отношений-операндов.
- Операция пересечения (INTERSECT) двух отношений с одинаковыми заголовками производит отношение, включающее все кортежи, входящие в оба отношения операнда.
- Отношение, являющееся разностью (MINUS) двух отношений с одинаковыми заголовками, включает все кортежи, входящие в отношение-первый операнд, такие, что ни один из них не входит в отношение, являющееся вторым операндом.
- При выполнении расширенного декартова произведения (TIMES) двух отношений, пересечение заголовков которых пусто (без общих атрибутов в заголовках), производится отношение, кортежи которого производятся путем объединения кортежей первого и второго операндов.

Манипулирование данными в РМД

- Операция переименования (RENAME) производит отношение, тело которого совпадает с телом операнда, но имена атрибутов изменены. Эта операция позволяет выполнять UNION, INTERSECT и MINUS над отношениями с почти совпадающими заголовками (совпадающими во всем, кроме имен атрибутов) и выполнять операцию TIMES над отношениями, пересечение заголовков которых не является пустым.
- Результатом ограничения (WHERE) отношения по некоторому условию является отношение, включающее кортежи отношения-операнда, удовлетворяющее этому условию.
- При выполнении проекции (PROJECT) отношения на заданное подмножество множества его атрибутов производится отношение, кортежи которого являются соответствующими подмножествами кортежей отношения-операнда.

Манипулирование данными в РМД

- При Θ -соединении (Тета-соединении) (Θ -JOIN) двух отношений по некоторому условию (Θ) образуется результирующее отношение, кортежи которого производятся путем объединения кортежей первого и второго отношений и удовлетворяют этому условию.
- У операции реляционного деления (DIVIDE BY) два операнда – бинарное и унарное отношения. Результирующее отношение состоит из унарных кортежей, включающих значения первого атрибута кортежей первого операнда таких, что множество значений второго атрибута (при фиксированном значении первого атрибута) включает множество значений второго операнда.
 - Пример: `Product[manufacturer, category] DIVIDE BY Product[category]` даст производителей, которые производят все категории товаров
- Операция присваивания (`:=`) позволяет сохранить результат вычисления реляционного выражения в существующем отношении БД.

Целостность реляционной модели

- Ограничение целостности сущности (первичного ключа)
 - для заголовка любого отношения БД должен быть явно или неявно задан первичный ключ
- Ограничение ссылочной целостности (внешнего ключа)

Ограничение целостности сущности

- Для заголовка любого отношения БД должен существовать **первичный ключ**, являющийся таким минимальным подмножеством заголовка отношения, что в любом теле этого отношения, которое *может* появиться в БД, значение первичного ключа в любом кортеже этого тела уникально.
 - **Первичный ключ** – минимальное подмножество заголовка отношения, предназначенное для уникальной идентификации каждой записи таблицы.
- Под минимальностью первичного ключа понимается то, что если к множеству атрибутов первичного ключа добавить хотя бы один атрибут, то ограничение целостности нарушится, т.е. могут появляться тела отношений, которые не допускались исходным первичным ключом.

Ограничение ссылочной целостности

- Атрибут (возможно, составной) называется **внешним ключом** (foreign key), если его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают значения их первичного ключа). Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом.

Ограничение ссылочной целостности

Student_ID	Student_name	Student_group	Student_lead
101	Иванов	1	Иванов
102	Петров	1	Иванов
103	Сидоров	2	Сидоров

Student_group	Student_lead
1	Иванов
2	Сидоров

Student_ID	Student_name	Student_group
101	Иванов	1
102	Петров	1
103	Сидоров	2

Объектно-ориентированная модель данных

- В 1991 г. возник консорциум ODMG (Object Database Management Group), задачей которого была разработка стандарта ОО-модели данных.
- Основное отличие ОО-модели данных от модели данных SQL и истинной реляционной модели:
 - В модели данных SQL и истинной реляционной модели данных БД представляет собой набор именованных контейнеров данных одного родового типа: таблиц или отношений соответственно.
 - В ОО-модели данных БД – это набор объектов (контейнеров данных) произвольного типа.

Объектно-ориентированная модель данных

В объектной модели данных вводятся две разновидности типов:

- литеральные (т.е. с явным определением области значений)
- объектные, которые делятся на два вида: атомарные объектные типы и объектные типы коллекций

Неформально говоря, любой тип данных обладает тремя характеристиками: множеством значений, набором операций и литералами.

Литеральные типы данных

– это обычные типы данных, принятые в традиционных языках программирования.

Они подразделяются на:

- базовые скалярные числовые типы
- символьные и булевские типы (атомарные литералы)
- конструируемые типы записей (структуры) и коллекций

Атомарный объектный тип

При определении атомарного объектного типа указывается его внутренняя структура (набор свойств – атрибутов и связей) и набор операций, которые можно применять к объектам этого типа.

Для определения атомарного объектного типа можно использовать механизм наследования, расширяя набор свойств и/или переопределяя существующие и добавляя новые операции.

Атомарный объектный тип

Атрибутом называется свойство объекта, значение которого можно получить по OID объекта. Значениями атрибутов могут быть и литералы, и другие объекты, но только тогда, когда не требуется обратная ссылка.

Связи – это инверсные свойства. В этом случае значением свойства может быть только объект. Связи определяются только между атомарными объектными типами. В объектной модели ODMG поддерживаются только бинарные связи. Связи могут быть «один-к-одному», «один-ко-многим» и «многие-ко-многим» в зависимости от того, сколько экземпляров соответствующего объектного типа может участвовать в связи.