

Основные положения реляционной модели данных

Преподаватель :

канд. тех. наук, доц. Озерова Г.П.

Как все начиналось

В 60-х годах прошлого столетия появилась необходимость в надежной модели хранения и обработки данных. В первую очередь эти данные требовались банками и финансовыми организациями.

В то время не существовало единых стандартов работы с данными и моделями, да и работа с данными заключалась в ручном упорядочении и организации хранящейся информации.

Как все начиналось

Банки записывали информацию о транзакциях в виде файлов определенной структуры.

У каждой организации было собственное понимание того, какова будет структура данных и каким образом ее обрабатывать.

В файлах часто встречались дубликаты данных клиентов и их транзакций, с этой проблемой «боролись» вручную.

Использование отдельных файлов

Пример. Учет успеваемости студентов в вузе.

Для каждой группы и каждой сессии выделен один файл, в котором в табличном виде храниться список группы, а также по столбцам перечень предметов по каждой сессии. В эти файлы заносятся информация по сданным зачетам и экзаменам для каждого студента.

Использование отдельных файлов

Пример. Учет успеваемости студентов в вузе.

	A	B	C	D	E
1	Ведомость				
2	группа № 111 1 семестр 2021				
3	№ п.п	Фамилия И.О.	Мат. анализ	Информат ика	История экон. учен.
4	1	Адоньева Е.П.	4	5	5
5	2	Власова Л.В	5	4	5
6	3	Грибков С.П.	4	3	3
7	4	Дуров Б.Р.	3	2	3
8	5	Жилин Т.Л.	4	4	3
9	6	Золотов В.П.	4	4	5
10	7	Ипатов Б.Г.	5	5	5
11	8	Кривцова Т.Т.	4	5	4
12	9	Лукьянов О.Д.	3	4	3
13	10	Носов Б.Л.	2	4	3
14					

Использование отдельных файлов

Пример. Учет успеваемости студентов в вузе.

После того, как исходные файлы с данными о сдаче сессии подготовлены, разрабатывался перечень программ по их обработке: начисление стипендии, формирование вкладыша в диплом и т.д.

Эти программы можно было написать на любом языке программирования или использовать уже готовые системы, например электронные таблицы.

Использование отдельных файлов

Пример. Учет успеваемости студентов в вузе.

	A	B	C	D	E	F	G	H	I	J	K
1	Ведомость										
2	группа № 111 1 семестр 2021 года										
3											
4	№ п.п	Фамилия И.О.	Мат. анализ	Информатика	История экон. учен.	Среднее	отличников	двоечников	хорошистов	Стипендия	
5	1	Адоньева Е.П.	4	5	5	4,6667	0	0	1	4 500,0р.	
6	2	Власова Л.В	5	4	5	4,6667	0	0	1	4 500,0р.	
7	3	Грибков С.П.	4	3	3	3,3333	0	0	0	3 600,0р.	
8	4	Дуров Б.Р.	3	2	3	2,6667	0	1	0	0,0р.	
9	5	Жилин Т.Л.	4	4	3	3,6667	0	0	0	3 600,0р.	
10	6	Золотов В.П.	4	4	5	4,3333	0	0	1	3 960,0р.	
11	7	Ипатов Б.Г.	5	5	5	5	1	0	0	5 400,0р.	
12	8	Кривцова Т.Т.	4	5	4	4,3333	0	0	1	3 960,0р.	
13	9	Лукьянов О.Д.	3	4	3	3,3333	0	0	0	3 600,0р.	
14	10	Носов Б.Л.	2	4	3	3	0	1	0	0,0р.	
15		Среднее по дисциплине	3,8	4	3,9	3,9	1	2	4	33 120,00р.	
17	Статистическая информация					Начисление стипендии					
18	Отличников		1	Минимальная		3 600,0р.					
19	Хорошистов		4	Увеличение							
20	Троечников		3	Отличники		50%					
21	Двоечников		2	Средний балл выше 4,5		25%					
22	Всего в группе		10	Хорошисты		10%					
24	Число студентов, получающих стипендию										
25	Минимальную			3 600,00р.		3					
26	Повышенную		50%	5 400,00р.		1					
27	Повышенную		25%	4 500,00р.		2					
28	Повышенную		10%	3 960,00р.		2					
29	Всего					8					

Использование отдельных файлов

Проблемы подобной структуры обработки данных:

1. **Высокая избыточность данных.** *Для нашего случая: список каждой группы фактически в неизменном виде, за некоторыми исключениями, будет храниться как минимум восемь раз...*
2. **Сложность совместной обработки данных.** *Как решить проблему: студент несколько курсов учился в одной группе, а затем перевелся в другую?*
3. **Зависимость программ от данных.** В нашем случае существенное значение играет количество сдаваемых в сессию предметов ...
4. **При возникновении новых потребностей** в использовании системы приходится пересматривать не только программы обработки, но и исходную информацию. *Как, например, выбрать всех студентов, получающих стипендию и проживающих в общежитии? Потребуется новая характеристика студентов, которая у нас не предусмотрена.*

Новое представление данных

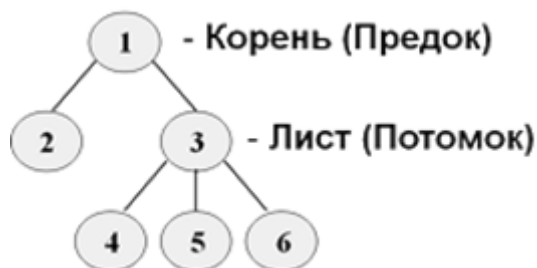
Были попытки создания моделей, позволяющих навести порядок в данных и их обработке.

Иерархическая модель данных

Данные в иерархической модели были организованы в виде древовидной структуры.

Иерархическая модель была востребованной, но не гибкой.

В ней каждая запись могла иметь только одного «предка», даже если отдельные записи могли иметь несколько «потомков».

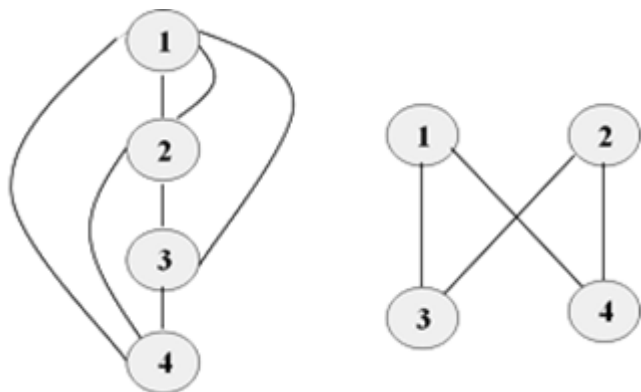


Первая иерархическая СУБД называлась IMS от IBM.

Сетевая модель данных

Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода.

В сетевой структуре у потомка может иметься любое число предков.

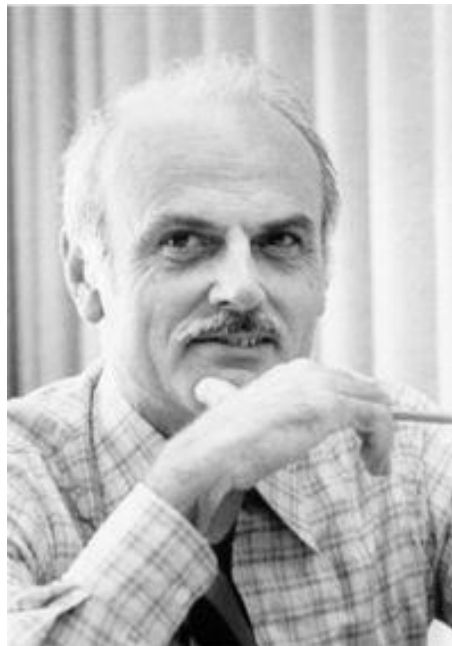


Недостаток сетевой модели - высокая сложность и жесткость схемы базы данных: если необходимо изменить структуру данных, то нужно изменить и программу ее обработки.

Данный подход был предложен как спецификация модели CODASYL в рамках рабочей группы DBTG (Data Base Task Group).

Реляционная модель данных

Упростить задачу сбора и обработки данных смог Франк Кодд (Edgar F. Codd).



Его фундаментальная работа привела к появлению реляционных баз данных, которые до сих пор нужны практически всем отраслям.

Реляционная модель данных

Реляционная модель определяет:

- способ представления данных (структуру данных);
- методы защиты данных (целостность данных);
- операции, которые можно выполнять с данными (манипулирование данными).

Реляционная модель данных

Основные принципы:

- все данные на концептуальном уровне представляются в виде объекта, заданного в виде строк и столбцов, называемого отношением, более распространенное название – таблица;
- в пересечение строки и столбца таблицы можно занести только одно значение;
- все операции выполняются над целыми отношениями и результатом этих операций является отношение.

Реляционная модель данных

Пусть даны n множеств D_1, D_2, \dots, D_n .

Тогда R есть **отношение** над этими множествами, если R есть множество упорядоченных наборов вида

$$\langle d_1, d_2, \dots, d_n \rangle,$$

где d_1 - элемент из D_1 , d_2 - элемент из D_2 , \dots , d_n - элемент из D_n .

При этом наборы вида $\langle d_1, d_2, \dots, d_n \rangle$ называются **кортежами**, а множества D_1, D_2, \dots, D_n - **доменами**.

Каждый кортеж состоит из элементов, выбираемых из своих доменов.

Эти элементы называются **атрибутами**, а их значения - **значениями атрибутов**.

Реляционная модель данных

ДОМЕН	Множество, состоящее из различных элементов
	Примеры: $A = \{ 1, 2, 3, 4, 5, 6 \}$ $B = \{ H, T \}$ $C = \{ R, V \}$ $D = \{ \text{Машина, Самолет, Поезд} \}$
Декартово произведение	Множество, элементами которого являются все возможные упорядоченные последовательности элементов исходных множеств.
	Пример: $B \times C = \{ \langle H, R \rangle, \langle H, V \rangle, \langle T, R \rangle, \langle T, V \rangle \}$
Отношение (таблица)	Любое подмножество элементов декартова произведения
	Примеры: $Q = \{ \langle H, R \rangle, \langle H, V \rangle \}$ $Q_1 = \{ \langle H, R \rangle \}$

Реляционная модель данных

$$Q = \{ \langle H, R \rangle, \langle H, V \rangle \}$$

Кортеж (запись)	Каждая строка, содержащая данные
	Пример: $\langle H, R \rangle$ - кортеж
Мощность	Количество кортежей в отношении
	Пример: мощность - 2
Атрибут (столбец)	Совокупность значений, находящихся на определенном месте кортежа
	Пример: атрибут – $[R, V]$
Размерность	Количество атрибутов в таблице
	Пример: размерность – 2

Пример

Домены

$A = \{1, 2, 3, 4, 5, 6\}$ $B = \{H, T\}$ $C = \{R, B\}$ $D = \{12-12, 11-01, 07-89\}$

Построить декартово произведение

$B \times C \times D = \{ \langle H, R, 12-12 \rangle, \langle H, B, 12-12 \rangle, \langle H, R, 11-01 \rangle, \langle H, B, 11-01 \rangle, \langle H, R, 07-89 \rangle, \langle H, B, 07-89 \rangle, \langle T, R, 12-12 \rangle, \langle T, B, 12-12 \rangle, \langle T, R, 11-01 \rangle, \langle T, B, 11-01 \rangle, \langle T, R, 07-89 \rangle, \langle T, B, 07-89 \rangle \}$

Количество элементов = 12

Найти ошибки в декартовом произведении

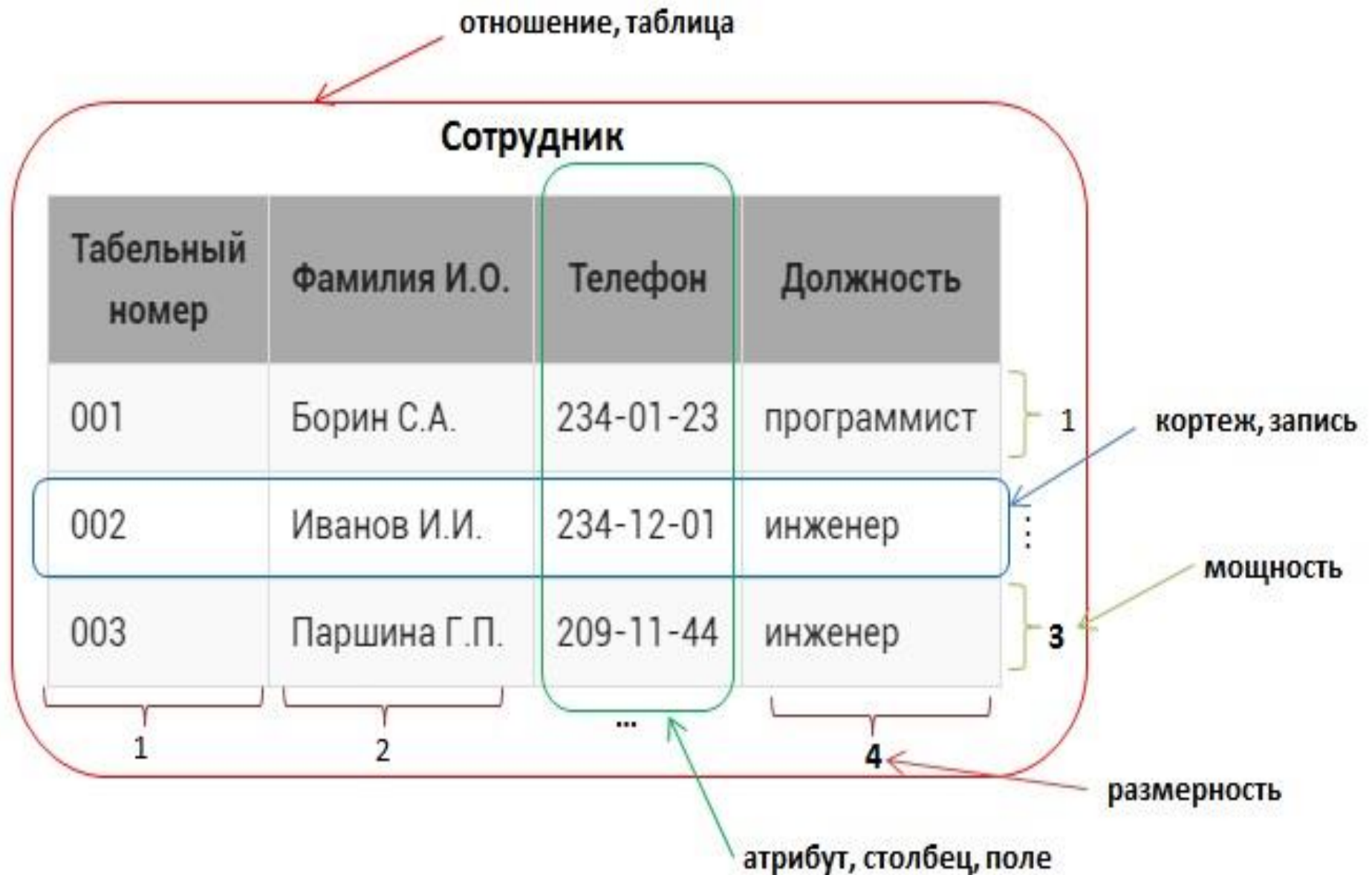
$A \times B \times C =$

$\{ \langle \cancel{H}, R, 1 \rangle, \langle 1, H, R \rangle, \langle \cancel{7}, H, R \rangle, \langle 1, \cancel{H}, H \rangle, \langle \cancel{T}, R \rangle, \langle 4, T, B \rangle \dots \}$

Построить произвольное отношение на основе декартова произведения $A \times B \times C \times D$

A	B	C	D
2	H	B	07-89
2	T	B	11-01
6	H	R	12-12

Отношение (таблица)

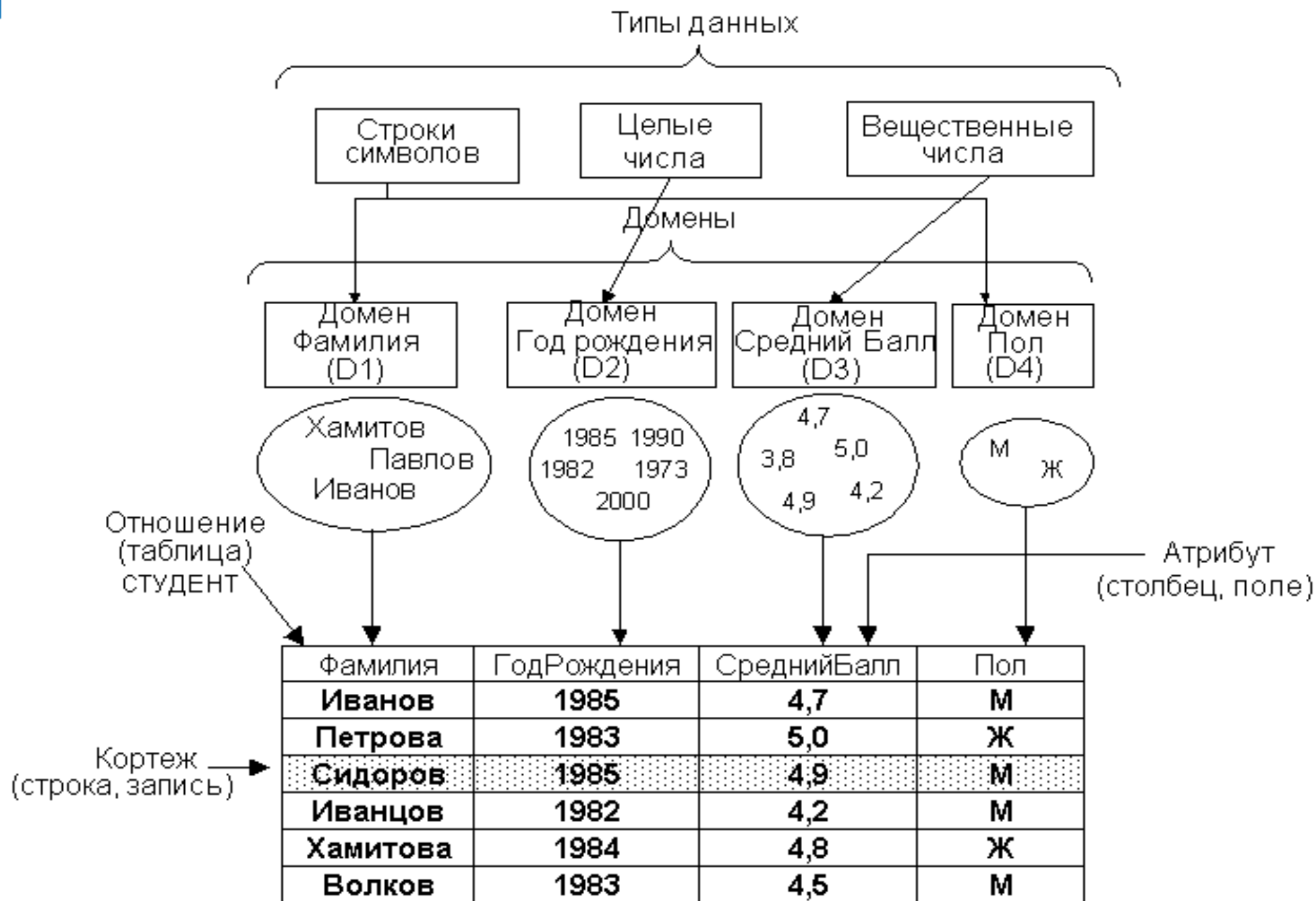


Таблица

Таблица в реляционной базе данных:

- состоит из совокупности столбцов;
- каждый столбец имеет уникальное имя;
- в каждом столбце хранится информация одного типа;
- строки таблицы имеют одинаковую структуру;
- в таблице не может быть одинаковых строк;
- последовательность строк в таблице не существенна.

Таблица



Предметная область «Книжный склад»

На некотором складе продаются книги.

Каждая книга имеет название, написана одним автором.

Каждая книга имеет цену. На складе хранится несколько экземпляров одной книги (все экземпляры имеют одинаковую цену).

Предметная область

Реляционная модель данных используется для **формального описания некоторой предметной области реального мира**, например, склада, учебного процесса и пр.

Является информационной моделью предметной области, то есть описывает не все ее аспекты, а только те, которые являются существенными для процесса обработки данных.

Предметная область «Книжный склад»

Название	Автор	Цена, руб	Количество
Мастер и Маргарита	Булгаков М.А.	670.99	3
Белая гвардия	Булгаков М.А.	540.50	5
Идиот	Достоевский Ф.М.	460	10
Братья Карамазовы	Достоевский Ф.М.	799.01	2

Проектирование информационного объекта

Цель: привести полученную таблицу к виду, который может быть описан отношением в реляционной модели.

Проектирование информационного объекта

- Выбрать имя таблицы.
- Определить структуру:
 - количество столбцов и их назначение;
 - название каждого столбца;
 - тип данных каждого столбца.
- Добавить ключевой столбец.

Имя информационного объекта

Требования:

- может включать английские буквы, цифры и знак подчеркивания, должно начинаться с буквы;
- имя должно быть уникальным в пределах базы данных.

Рекомендации:

- имя - существительное в единственном числе, записывается строчными буквами;
- имя должно быть понятным и соответствовать тому объекту, который оно описывает;
- имя должно быть как можно короче, максимум до 10 символов.

Имя информационного объекта, примеры

Верные имена:

store, book, book_store, books, goods, my_book

Неверные имена:

1_book, 123, book store, my.book

Предметная область «Книжный склад»

Имя таблицы: **book**

Название	Автор	Цена, руб	Количество
Мастер и Маргарита	Булгаков М.А.	670.99	3
Белая гвардия	Булгаков М.А.	540.50	5
Идиот	Достоевский Ф.М.	460	10
Братья Карамазовы	Достоевский Ф.М.	799.01	2

Имя столбца (поля)

Требования:

- может включать английские буквы, цифры и знак подчеркивания, должно начинаться с буквы;
- имя должно быть уникальным в пределах таблицы.

Рекомендации:

- имя должно быть понятным и соответствовать тем данным, которые хранятся в поле;
- имя может состоять из нескольких слов, тогда слова разделяются подчеркиванием, после подчеркивания слово пишется с маленькой буквы.

Имена столбцов, примеры

Верные имена:

title, book_name, price, author, author_name, amount,
good_name, book_price

Неверные имена:

2_title, 1_2, book name, author.name

Предметная область «Книжный склад»

Таблица **book**

title	author	price	amount
Мастер и Маргарита	Булгаков М.А.	670.99	3
Белая гвардия	Булгаков М.А.	540.50	5
Идиот	Достоевский Ф.М.	460	10
Братья Карамазовы	Достоевский Ф.М.	799.01	2

Типы данных

Тип данных	Описание	Пример
<code>INT</code> <code>INTEGER</code>	Целое число, могут принимать значения от -2 147 483 648 до 2 147 483 647	-567 1205
<code>DECIMAL</code> <code>NUMERIC</code>	Вещественное число, в скобках указывается максимальная длина числа (включает символы слева и справа от десятичной запятой) и количество знаков после запятой. Можно использовать оба этих типа, они эквивалентны, принимают значения в диапазоне $-10^{38}+1$ до $10^{38}-1$. <code>DECIMAL(4,1)</code> <code>NUMERIC(6,3)</code>	34.6 -3.294
<code>DATE</code>	Дата в формате ГГГГ-ММ-ДД <i>26 июля 2020 года</i> <i>3 января 2021 года</i>	2020-07-26 2021-01-03
<code>VARCHAR</code>	Строка длиной не более 255 символов, в скобках указывается максимальная длина строки, которая может храниться в поле <code>VARCHAR(10)</code>	пример описание

Предметная область «Книжный склад»

Имя таблицы: **book**

- **title** - строка текста - VARCHAR(50);
- **author** - строка текста - VARCHAR(30);
- **price** - для описание денежного значения используется числовой тип данных с двумя знаками после запятой - DECIMAL(8,2);
- **amount** - целое число - INT.

Ключевой столбец (поле)

Ключевое поле является ОБЯЗАТЕЛЬНЫМ ЭЛЕМЕНТОМ каждой реляционной таблицы.

Ключевое поле является уникальным для каждой записи таблицы, однозначно определяет запись и в дальнейшем будет использоваться для связей с другими таблицами.

Рекомендации по именованию ключевых полей:

имя должно состоять из двух частей:

- начинаться с названия таблицы, которой поле принадлежит;
- затем через подчеркивание необходимо указать **id**.

Описание ключевого поля:

INT PRIMARY KEY AUTO_INCREMENT

Это означает, что

- в поле будут записываться различные целые числа,
- при этом они будут автоматически генерироваться (каждая следующая строка будет иметь значение ключа на 1 больше предыдущего).

Предметная область «Книжный склад»

Таблица **book**

book_id	Title	author	Price	amount
INT PRIMARY KEY AUTO_INCREMENT	VARCHAR(50)	VARCHAR(30)	DECIMAL(8,2)	INT
1	Мастер и Маргарита	Булгаков М.А.	670.99	3
2	Белая гвардия	Булгаков М.А.	540.50	5
3	Идиот	Достоевский Ф.М.	460.00	10
4	Братья Карамазовы	Достоевский Ф.М.	799.01	2
5	Стихотворения и ПОЭМЫ	Есенин С.А.	650.00	15

Создание таблицы book

```
CREATE TABLE book(  
    book_id INT PRIMARY KEY AUTO_INCREMENT,  
    title VARCHAR(50),  
    author VARCHAR(30),  
    price DECIMAL(8, 2),  
    amount INT  
);
```

Наполнение таблицы book

```
INSERT INTO book (title, author, price, amount)
VALUES
('Мастер и Маргарита', 'Булгаков М.А.', 670.99, 3),
('Белая гвардия', 'Булгаков М.А.', 540.50, 5),
('Идиот', 'Достоевский Ф.М.', 460, 10),
('Братья Карамазовы', 'Достоевский Ф.М.', 799.01, 2),
('Стихотворения и поэмы', 'Есенин С.А.', 650.00, 15);
```

Реляционная модель - модель, основанная на табличных методах и средствах представления и обработки информации.

Основные понятия реляционной модели:

- ✓ **Отношение** – таблица;
- ✓ **Атрибуты** – столбцы таблицы;
- ✓ **Кортежи** – записи (строки) таблицы.

Информационный объект – это таблица базы данных.

При **проектировании** информационного объекта необходимо:

- ✓ Выбрать имя таблицы.
- ✓ Определить структуру:
 - количество столбцов и их назначение;
 - название каждого столбца;
 - тип данных каждого столбца.
- ✓ Добавить ключевой столбец.



Спасибо за внимание!