



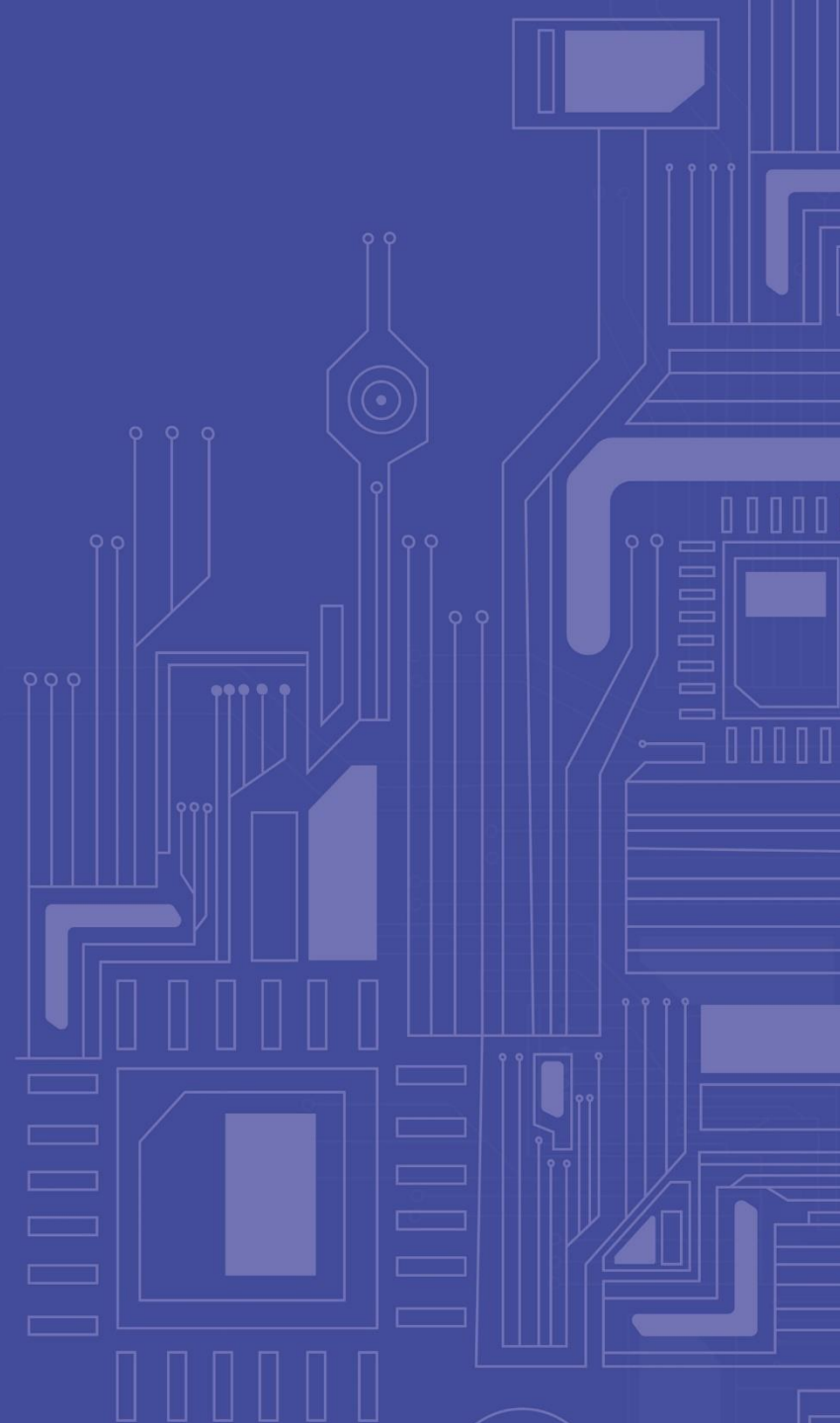
МИНОБРНАУКИ  
РОССИИ



Передовые  
инженерные  
школы

# СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

*Лекция 2*



- Введение в реляционную модель данных
  - Современные модели данных
    - Основные понятия РМД
      - Свойства отношений
        - Описание РМД
          - Реляционная алгебра Кодда

# НЕФОРМАЛЬНОЕ ВВЕДЕНИЕ В РЕЛЯЦИОННУЮ МОДЕЛЬ ДАННЫХ



# РЕЛЯЦИОННЫЕ СТРУКТУРЫ ДАННЫХ (1/2)



Идея реляционной модели данных – Эдгар Кодд 1969г.

Удобство для приложений + ненавигационные операции над данными (сети и иерархии не подходят - сами по себе навигационные структуры).

Кодд в качестве родовой структуры данных предложил «таблицы», строки и столбцы которых неупорядоченны.

Таблица - множество столбцов  $\{A_1, A_2, \dots, A_n\}$

Каждый столбец  $A_i$  может содержать значения из множества

$T_i = \{v_{i1}, v_{i2}, \dots, v_{im}\}$  (все множества конечны)

Таблица - отношение над множествами  $\{T_1, T_2, \dots, T_n\}$

## Математическое отношение:

Отношением над множествами  $\{T_1, T_2, \dots, T_n\}$  называется подмножество декартова произведения этих множеств, т.е. некоторое множество кортежей  $\{\{v_1, v_2, \dots, v_n\}\}$ , где  $v_i \in T_i$ .

Кодд для «таблиц» (родовых структур данных) стал использовать термин *отношение*, а для элементов отношения - термин *кортеж*. Соответственно, модель данных получила название реляционной модели.

# РЕЛЯЦИОННЫЕ СТРУКТУРЫ ДАННЫХ (2/2)



Сложно сейчас будет

Схема БД в реляционной модели данных – это набор именованных *заголовков отношений* вида

$$H_i = \{ \langle A_i^1, T_i^1 \rangle, \langle A_i^2, T_i^2 \rangle, \dots, \langle A_i^{ni}, T_i^{ni} \rangle \}.$$

$T_i$  называется *доменом* атрибута  $A_i$ .

По Кодду, каждый домен  $T_i$  является подмножеством значений некоторого базового типа данных  $T_i^+$ , а значит, к его элементам применимы все операции этого базового типа.

Реляционная база данных в каждый момент времени представляет собой набор именованных отношений, каждое из которых обладает заголовком, таким как он определен в схеме БД, и телом. Имя отношения  $R_i$  совпадает с именем заголовка этого отношения  $H_{R_i}$ .

Тело отношения  $B_{R_i}$  – это множество кортежей вида  $\{ \langle A_i^1, T_i^1, v_i^1 \rangle, \langle A_i^2, T_i^2, v_i^2 \rangle, \dots, \langle A_i^{ni}, T_i^{ni}, v_i^{ni} \rangle \}$ , где  $v_i^j \in T_i^j$ .

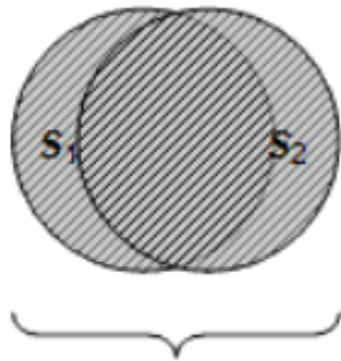
Во время жизни БД тела отношений могут изменяться, но все содержащиеся в них кортежи должны соответствовать заголовкам соответствующих отношений.

В реляционной модели данных заголовки и тело любого отношения представляют собой множества, поэтому к ним применимы операции над множествами.

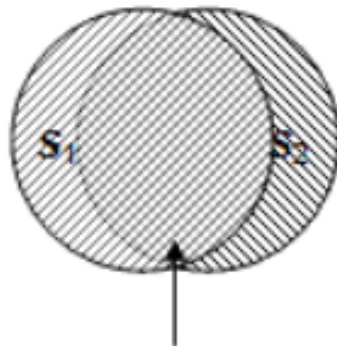
Результат объединения двух множеств  $S_1 \{s_1\}$  и  $S_2 \{s_2\}$  - это множество  $S \{s\}$ :  $s \in S_1$  **или**  $s \in S_2$ .

Результат пересечения двух множеств  $S_1 \{s_1\}$  и  $S_2 \{s_2\}$  - это множество  $S \{s\}$ :  $s \in S_1$  **и**  $s \in S_2$ .

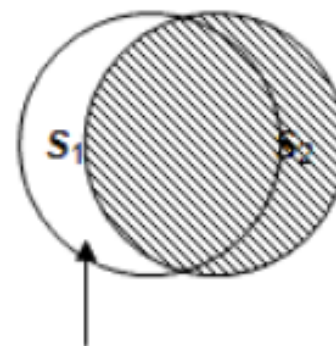
Результат вычитания двух множеств  $S_1 \{s_1\}$  и  $S_2 \{s_2\}$  - это множество  $S \{s\}$ :  $s \in S_1$  **и**  $s \notin S_2$ .



Объединение



Пересечение



Вычитание

## ДЕСЯТЬ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ КОДДА

- объединение (UNION)
- пересечение (INTERSECT)
- вычитание (MINUS)
- взятие расширенного декартова произведения (TIMES)
- переименование атрибутов (RENAME)
- проекция (PROJECT)
- ограничение (WHERE)
- соединение (  $\theta$  - JOIN)
- деление (DIVIDE BY)
- присваивание ( $:=$ )

1. При выполнении операции *объединения* (UNION) двух отношений с одинаковыми заголовками производится отношение, включающее все кортежи, входящие хотя бы в одно из отношений-операндов.
2. Операция *пересечения* (INTERSECT) двух отношений с одинаковыми заголовками производит отношение, включающее все кортежи, входящие в оба отношения-операнда.
3. 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.

## ДЕСЯТЬ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ КОДДА

4. При выполнении *декартова произведения* (TIMES) двух отношений, пересечение заголовков которых пусто, производится отношение, кортежи которого производятся путем объединения кортежей первого и второго операндов.
5. Операция *переименования* (RENAME) производит отношение, тело которого совпадает с телом операнда, но имена атрибутов изменены; эта операция позволяет выполнять первые три операции над отношениями с «почти» совпадающими заголовками (совпадающими во всем, кроме имен атрибутов) и выполнять операцию TIMES над отношениями, пересечение заголовков которых не является пустым.
6. Результатом *ограничения* (WHERE) отношения по некоторому условию является отношение, включающее кортежи отношения-операнда, удовлетворяющее этому условию.
7. При выполнении *проекции* (PROJECT) отношения на заданное подмножество множества его атрибутов производится отношение, кортежи которого являются соответствующими подмножествами кортежей отношения-операнда.



## ДЕСЯТЬ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ КОДДА

8. При *Θ-соединении* ( $\Theta$ -JOIN) двух отношений по некоторому условию ( $\Theta$ ) образуется результирующее отношение, кортежи которого производятся путем объединения кортежей первого и второго отношений и удовлетворяют этому условию.
9. 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.
10. Операция *присваивания* ( $:=$ ) позволяет сохранить результат вычисления реляционного выражения в существующем отношении БД.

# ЦЕЛОСТНОСТЬ ДАННЫХ



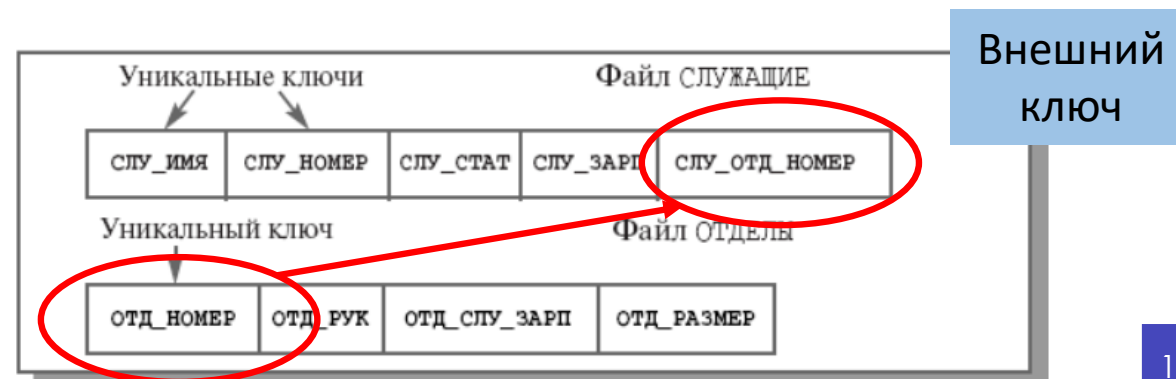
## Ограничение целостности сущности или ограничение первичного ключа

Для заголовка любого отношения базы данных должен быть явно или неявно определен первичный ключ, являющийся таким минимальным подмножеством заголовка отношения, что в любом теле этого отношения, которое может появиться в базе данных, значение первичного ключа в любом кортеже этого тела является уникальным, т.е. отличается от значения первичного ключа в любом другом кортеже.

Внешним ключом отношения  $R_1$ , ссылающимся на отношение  $R_2$ , называется подмножество заголовка  $H_{R_1}$ , которое совпадает с первичным ключом отношения  $R_2$  (с точностью до имен атрибутов).

## Ограничение ссылочной целостности или ограничение внешнего ключа

Ограничение ссылочной целостности реляционной модели данных можно сформулировать следующим образом: в любом теле отношения  $R_1$ , которое может появиться в базе данных, для «не пустого» значения внешнего ключа, ссылающегося на отношение  $R_2$ , в любом кортеже этого тела должен найтись кортеж в теле отношения  $R_2$ , которое содержится в базе данных, с совпадающим значением первичного ключа.



# СОВРЕМЕННЫЕ МОДЕЛИ ДАННЫХ



Год	Веха	Суть
1989	Манифест систем объектно-ориентированных баз данных	Набор требований к ООСУБД: преодоление несоответствий типов данных в ЯП и СУБД, а также возможность хранить данные произвольно сложной структуры.
1991	ODMG (Object Database Management Group), стандарт ODMG 3.0	Создание стандарта объектно-ориентированной модели данных на выверенном академическом уровне.
1990	Манифест систем баз данных третьего поколения	Против ODMG, за эволюцию SQL-ориентированных СУБД, в защиту инвестиций крупных компаний, владеющих СУБД.
1999	Стандарт SQL:1999	Фиксация новых особенностей языка, придающих ему черты полноценной модели данных.

# ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ДАННЫХ (1/2)



Литеральные типы	<p><b><u>Атомарные литералы</u></b></p> <ul style="list-style-type: none"><li>• Скалярные числовые</li><li>• Символьные</li><li>• Булевские</li></ul>	<p><b><u>Структуры</u></b></p> <ul style="list-style-type: none"><li>• Содержат сложные объекты с обращением по OID (аналог указателей)</li><li>• Могут содержать объекты (OID на OID)</li></ul>	<p><b><u>Коллекции</u></b></p> <ul style="list-style-type: none"><li>• Множества</li><li>• Мультимножества (неупорядоченные наборы элементов, возможны дубликаты)</li><li>• Списки (упорядоченные наборы элементов, возможны дубликаты)</li><li>• Словари (множества пар &lt;ключ, значение&gt;, причем все ключи в этих парах должны быть различными)</li></ul>
Объектные типы	<p>Обязательно есть операции создания и инициализации нового объекта данного типа, создается OID для обращения к операциям внутри данного типа (как классы)</p>	<p><b><u>Атомарный объектный тип</u></b></p> <ul style="list-style-type: none"><li>• Состоят из литералов и объектов</li><li>• Содержат атрибуты и связи</li><li>• Связи инверсивны и могут указывать только на объект</li><li>• Экстенты – связанные с данным объектом множества объектов</li></ul>	<p><b><u>Объектные типы коллекций</u></b></p> <ul style="list-style-type: none"><li>• Множества, Мультимножества, Списки, Словари</li><li>• Не может содержать элементы того же типа коллекций</li></ul>

# ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ДАННЫХ (2/2)



Типы и структуры данных

Манипуляция данными

Ограничения целостности

## Манипулирование данными

Язык OQL (Object Query Language), опирается на ODMG, очень формальный

Получить номера руководителей отделов и тех служащих их отделов, зарплата которых превышает 20000 руб.

```
SELECT DISTINCT STRUCT ( ОТД_РУК: D.ОТД_РУК,  
СЛУ: ( SELECT E  
FROM D.CONSISTS_OF AS E  
WHERE E.СЛУ_ЗАП > 20000.00 ) )  
FROM ОТДЕЛЫ D
```

У ОТДЕЛ есть экстент ОТДЕЛЫ, по связи происходит переход к служащим, и выгружается усеченное ограничение множество

## Ограничения целостности

- 1 **Ограничение целостности сущности** отсутствует, потому что равенство объектов возможно лишь при равенстве OID (тот же самый объект)
- 2 **Ограничение ссылочной целостности** есть при наличии связи один ко многим, СУБД воспринимает экстент как потомков и следит за сохранностью связи.

## Типы и структуры данных

## Манипуляция данными

## Ограничения целостности

**SQL-ориентированная база данных** представляет собой набор таблиц, каждая из которых в любой момент времени содержит некоторое мультимножество строк, соответствующих заголовку таблицы. Столбцы упорядочены (отличие таблиц от отношений).

### Традиционные таблицы

Множество столбцов с указанными типами данных

### Типизированные таблицы

OID	1	...	n
-----	---	-----	---

Атрибуты объекта структурного типа

Самоссылающийся столбец

### Типы данных

- точные числовые типы
- приближенные числовые типы
- типы символьных строк
- типы битовых строк
- типы дат и времени
- типы временных интервалов
- булевский тип (*false, true, unknown*)
- типы коллекций (массивы и мультимножества)
- анонимные строчные типы
- типы, определяемые пользователем
- ссылочные типы

### Пользовательские типы данных

**Индивидуальный тип** — переименованный от предопределенного (операции явно надо прописывать, не наследуются)

**Структурный тип** — это именованный тип данных, включающий один или более атрибутов любого из допустимых типов данных

Типы и структуры данных

Манипуляция данными

Ограничения целостности

```
SELECT [ ALL | DISTINCT ] select_item_commalist  
FROM table_reference_commalist  
[ WHERE conditional_expression ]  
[ GROUP BY column_name_commalist ]  
[ HAVING conditional_expression ]  
[ ORDER BY order_item_commalist ]
```

Оператор SELECT может применяться как к таблицам, хранимым в памяти, так и произвольным таблицам. Выполнение запроса начинается с формирования декартова произведения таблиц запроса

WHERE

GROUP BY

HAVING

ORDER BY

Фильтрация  
декартова  
произведения по  
заданным критериям

Группировка по  
заданным атрибутам

Фильтрация групп по  
критерию

Сортировка  
результата по  
атрибутам



Типы и структуры данных

Манипуляция данными

Ограничения целостности

## Ограничения целостности сущности

Поддержка мультимножеств строк противоречит поддержке ограничений целостности сущности в смысле ИРМ.

Наличие таблиц без первичного ключа – тоже противоречит ИРМ.

Таблицы с первичным ключом  
- ОК

## Ссылочная целостность

Поддерживается, но не всегда в терминах истинной реляционной модели, что часто связано с использованием неопределенных значений (*unknown/NULL*)

## Явное задание ограничений целостности

- ↳ На уровне столбца
- ↳ На уровне таблицы
- ↳ На уровне базы данных

# ИСТИННАЯ РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ (1/2)



Типы и структуры данных

Манипуляция данными

Ограничения целостности



Режим повышенной строгости

Скалярный тип данных

Инкапсулированный тип, структура которого скрыта от пользователей.  
Предопределенный скалярный тип – булевский с двумя значениями *true* и *false*.

Кортежный тип

Безымянный тип данных, определяемый с помощью генератора типа `TUPLE` с указанием множества пар `<имя_атрибута, тип_атрибута>` (заголовок кортежа).  
Значением кортежного типа является кортеж, множество триплетов `<имя_атрибута, тип_атрибута, значение_атрибута>`, которое соответствует заголовку кортежа этого кортежного типа.

Тип отношения

Безымянный тип данных, определяемый с помощью генератора типа `RELATION` с указанием некоторого заголовка кортежа. Значением типа отношения является заголовок отношения, совпадающий с заголовком кортежа этого типа отношения, и тело отношения, представляющее собой множество кортежей, соответствующих этому заголовку.

# ИСТИННАЯ РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ (2/2)



Типы и структуры данных

Манипуляция данными

Ограничения целостности

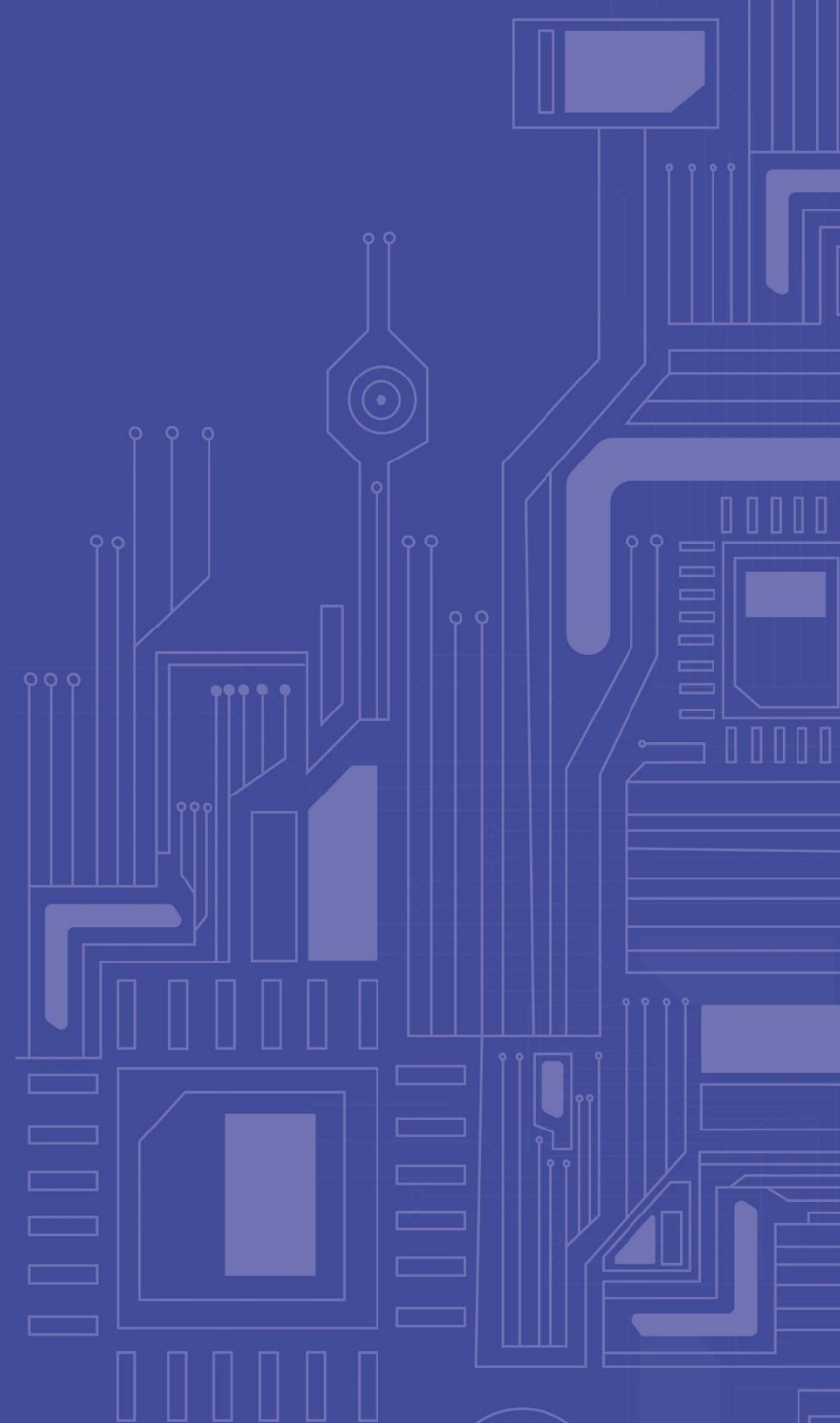
Манипулирование данными

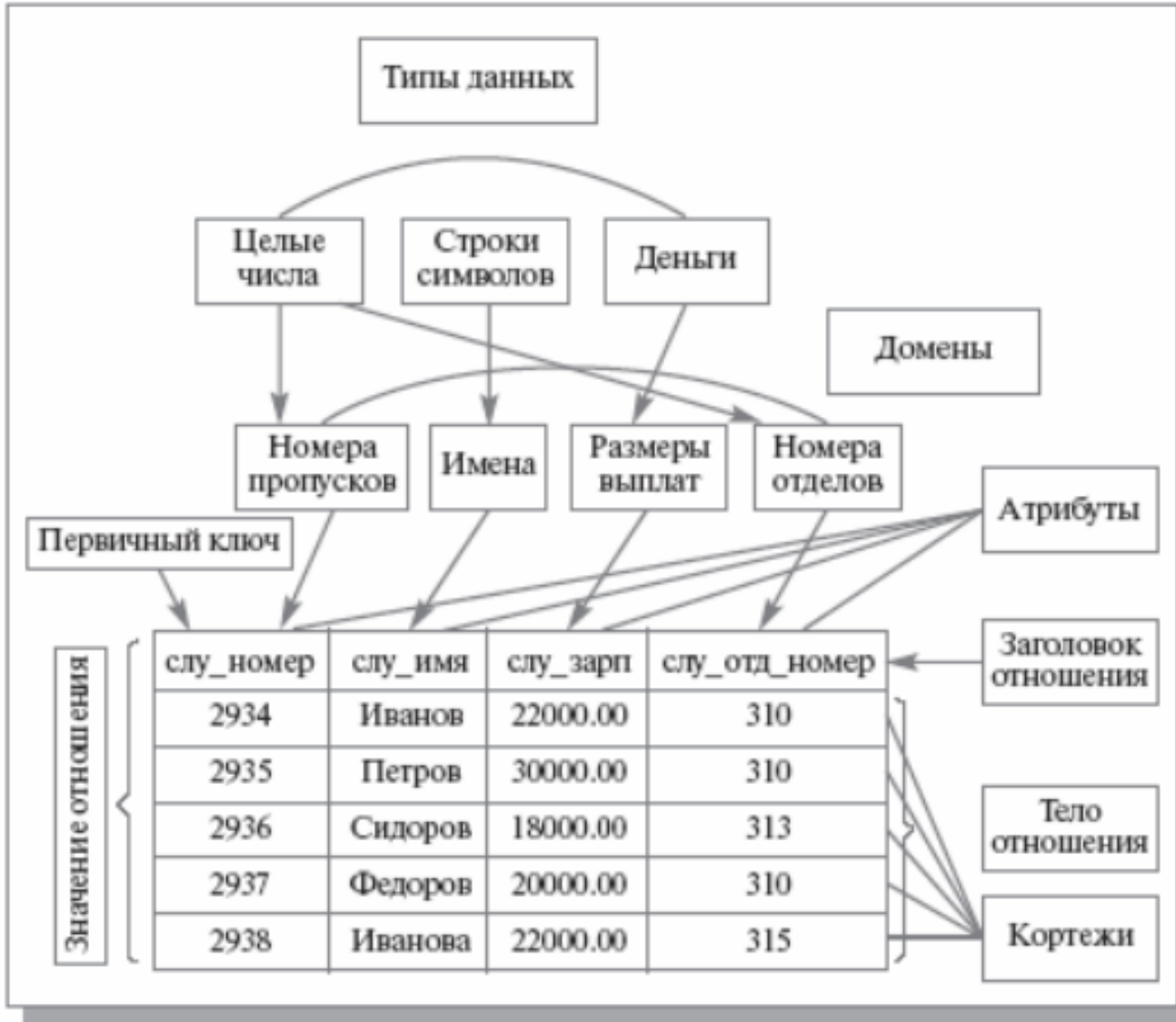
- Алгебра Кодда
- Алгебра А Дейта и Дарвена, основанная на реляционных аналогах булевских функций *конъюнкции*, *дизъюнкции* и *отрицания*.
- Через алгебру А можно выразить все операции алгебры Кодда.

Ограничения целостности

- Обязательно определение хотя бы одного ключа для каждого отношения
- Любое условное выражение, которое является (или логически эквивалентно) замкнутой правильно построенной формулой (WFF) реляционного исчисления, должно быть допустимо в качестве спецификации ограничения целостности
- Ограничения ссылочной целостности носят рекомендательный характер

# ОСНОВНЫЕ ПОНЯТИЯ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ





## Преимущества реляционного подхода

1. Возможность описать все известные предметные области (гибкость)
2. Отсутствие необходимости знать, как расположены данные в физической памяти
3. Математическая точность и логика, возможность формально определить используемые абстракции.

## ТИП ДАННЫХ

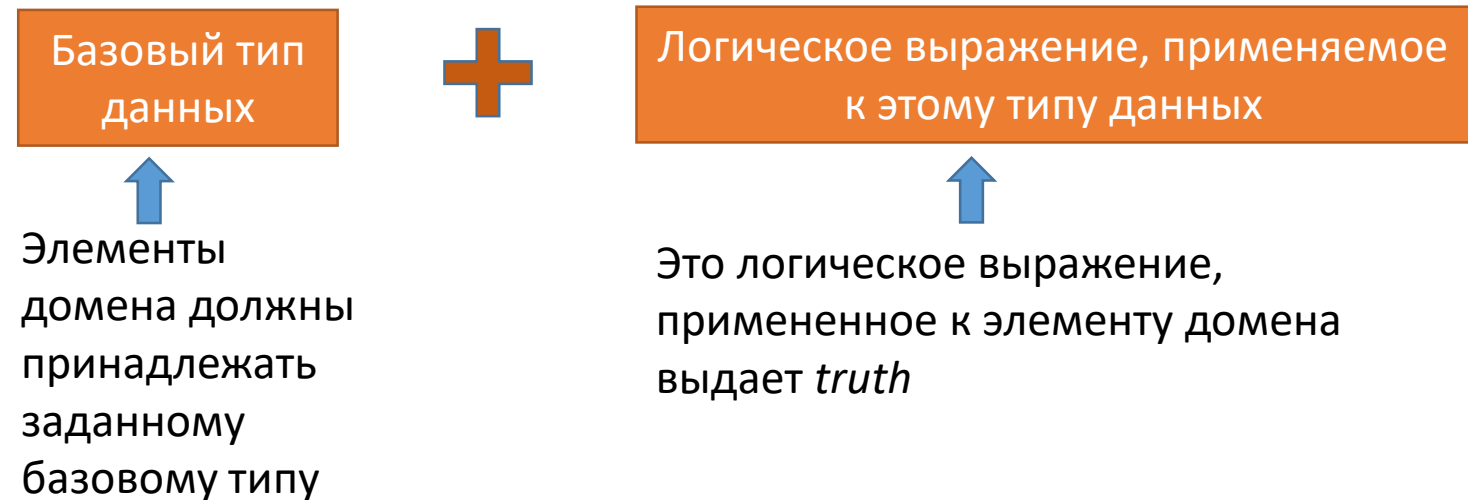
- ↳ Множество значений
- ↳ Виды применимых операций
- ↳ Внешнее представление (литералы)

### Какие типы данных бывают:

1. Символьные
2. Числовые
3. Специализированные (деньги)
4. Темпоральные (дата, время)
5. Пользовательские

## ДОМЕН

В Третьем манифесте Дейт и Дарвен ликвидируют различия между доменом и типом данных



Пример. Русский алфавит, «ц» принадлежит, «z» - нет

Сравнивать можно элементы одного домена. Русская и английская «а» несравнимы. Номер пропуска и номер отдела несравнимы.

# ОСНОВНЫЕ ПОНЯТИЯ (2/3)



Отношение – родовое понятие реляционной БД. *Relational* от *relation* – отношение.

Заголовок (схема) отношения

$$H_r = \{ \langle A_r^1, T_r^1 \rangle, \langle A_r^2, T_r^2 \rangle, \dots, \langle A_r^n, T_r^n \rangle \}$$

Имя атрибута

Базовый тип,  
или ранее  
определенный  
домен

Тело отношения

$$B_r = \{ t_{r1}, t_{r2}, \dots, t_{rm} \}$$

Множество кортежей, пусть  $m$

Кортеж

$$t_{ri} = \{ \langle A_r^1, T_r^1, v_{ri}^1 \rangle, \dots, \langle A_r^n, T_r^n, v_{ri}^n \rangle \}$$

Элемент  
заголовка  
отношения

Допустимое  
значение типа  
данных или домена

Значение отношения

$$v_r = \langle H_r, B_r \rangle$$

Переменная отношения ( $VAR_r$ ) - называется именованный контейнер, который может содержать любое допустимое значение  $v_r$ . Иногда просто называем отношением.

**Реляционная база данных** – набор пар  $\langle VAR_r, H_r \rangle$

**Первичный ключ** переменной отношения является такое подмножество  $S$  множества атрибутов ее заголовка, что в любое время значение первичного ключа (составное, если в состав первичного ключа входит более одного атрибута) в любом кортеже тела отношения отличается от значения первичного ключа в любом другом кортеже тела этого отношения, а никакое собственное подмножество  $S$  этим свойством не обладает.

Обычным житейским представлением отношения является *таблица*, заголовком которой является схема отношения, а *строками* – кортежи отношения-экземпляра; в этом случае имена атрибутов соответствуют именам *столбцов* данной таблицы. Поэтому иногда говорят про «столбцы таблицы», имея в виду «атрибуты отношения».



У обычных таблиц и строки, и столбцы упорядочены, тогда как атрибуты и кортежи отношений являются элементами неупорядоченных множеств



# ФУНДАМЕНТАЛЬНЫЕ СВОЙСТВА ОТНОШЕНИЙ



# ФУНДАМЕНТАЛЬНЫЕ СВОЙСТВА (1/3)



1. Так как тело отношения – множество кортежей, то по определению множества там нет дубликатов.
2. Промежуточные отношения, возникающие в результате выполнения запросов, могут содержать дубликаты, они называются *мультимножества*.
3. Из свойства отсутствия дубликатов вытекает неперменное наличие минимального ключа отношения, который можно считать *первичным*.
4. Из нескольких возможных один первичный выбирает архитектор БД. Остальные остаются *возможными*.

①	A	🏠
②	A	○
③	B	□
④	C	●
⑤	D	◇
⑥	E	⊖
⑦	F	⊗
⑧	D	⬆

Первичный  
ключ

Возможный  
ключ

①	A	🏠
②	A	◇
③	B	□
④	C	●
⑥	D	◇
⑥	E	⊖
⑦	F	⊗
②	D	⬆

Любая комбинация 2х  
столбцов – ключ, сами  
выбираем первичный

## Отсутствие упорядоченности кортежей

1. Отсутствие упорядоченности кортежей является следствием того, что отношение – это множество кортежей.
2. Некоторые хотели бы, чтобы кортежи лежали в памяти по порядку, но это затрудняет изменения в БД.
3. Неупорядоченность добавляет гибкости в хранении.
4. Это не противоречит выведению результатов запросов SQL в виде упорядоченных (отсортированных) отношений.

## Отсутствие упорядоченности атрибутов

1. Отсутствие упорядоченности атрибутов является следствием того, что это множество пар <имя атрибута, домен (базовый тип)>.
2. Можно привести аналогию со структурным типом языка C.

`STRUCT {integer a, char b, integer c} d;`

Обращение к b будет не

`&d + sizeof(integer) ,`

а `d.b`, потому что хранятся переменные могут не подряд.

3. В заголовке отношения атрибуты неупорядоченны по той же причине. Это облегчает модификацию схем отношений.

## Атомарность значений атрибутов

- Значения атрибутов являются атомарными (скалярными), определенными на типе домена
- Можно использовать пользовательские типы
- Операции с атрибутами определяются на уровне домена
- Главное, чтобы СУБД явно не отображала структуру значений

## Первая нормальная форма отношений

НОМЕР_ОТДЕЛА	ОТДЕЛ		
	СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
310	2934	Иванов	22000.00
	2935	Петров	30000.00
	2937	Федоров	20000.00
313	2936	Сидоров	18000.00
315	2938	Иванова	22000.00

Нормируем

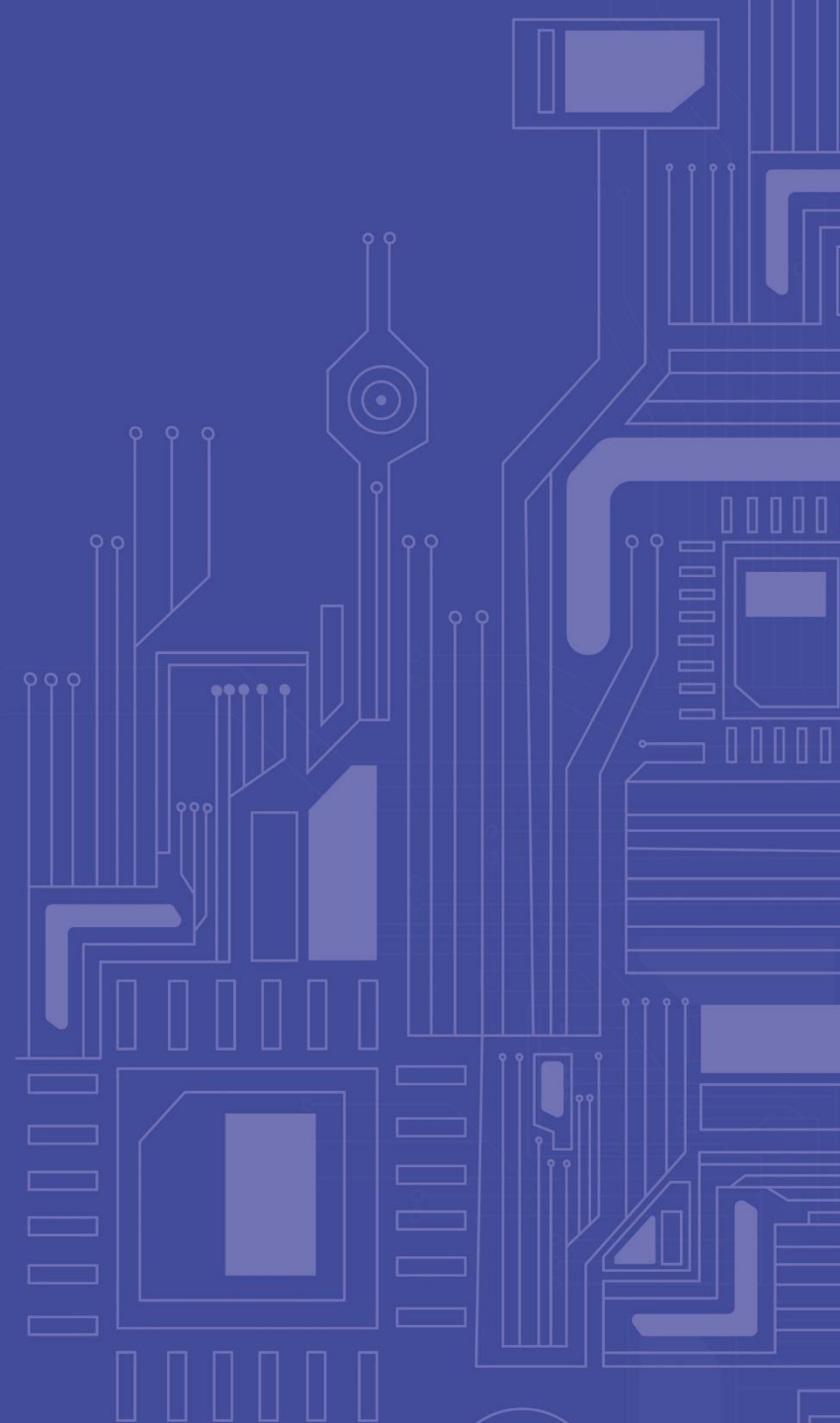


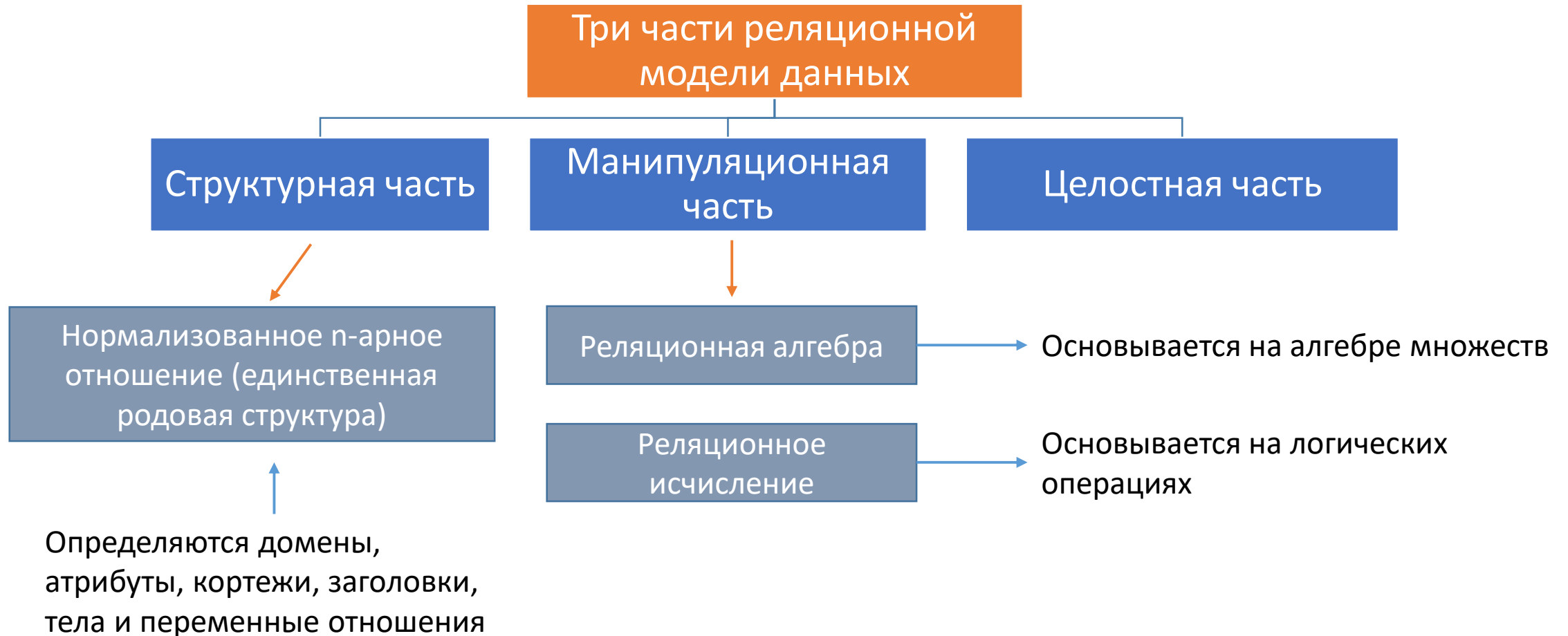
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

По сути, это вид плоской таблицы

По сути, это бинарное отношение  
(атрибут НОМЕР\_ОТДЕЛА –  
отношение ОТДЕЛ)

# ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ





Кодд придумал, Дейт популяризировал

## Требование целостности сущности

У любой переменной отношения должен существовать первичный ключ, и никакое значение первичного ключа в кортежах значения-отношения переменной отношения не должно содержать неопределенных значений.

## Что такое NULL?

Если  $a$  – это значение некоторого типа данных или NULL,  $op$  – любая двуместная «арифметическая» операция этого типа данных (например,  $+$ ), а  $lop$  – операция сравнения значений этого типа (например,  $=$ ), то по определению:

```
 $a \ op \ NULL = NULL$   
 $NULL \ op \ a = NULL$   
 $a \ lop \ NULL = unknown$   
 $NULL \ lop \ a = unknown$ 
```

```
NOT  $unknown = unknown$   
 $true \ AND \ unknown = unknown$   
 $true \ OR \ unknown = true$   
 $false \ AND \ unknown = false$   
 $false \ OR \ unknown = unknown$ 
```

Unknown – третье значение  
булевого типа

Внешний ключ отношения – это ссылочная связь на другое отношение, где этот ключ является первичным.

## Требование целостности ссылок

1

Если внешний ключ не NULL, то должен существовать такой же первичный ключ в связанном отношении (если у служащего прописан отдел, то такой отдел должен существовать).

2

При удалении связанного значения могут быть три варианта действий:

- Удалить все связанные записи (удалили отдел, удалить из базы всех служащих этого отдела)
- Всем связанным записям поставить внешний ключ NULL
- Запретить удаление кортежа, если на него есть ссылки (сначала удалить все ссылки, а потом сам кортеж)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

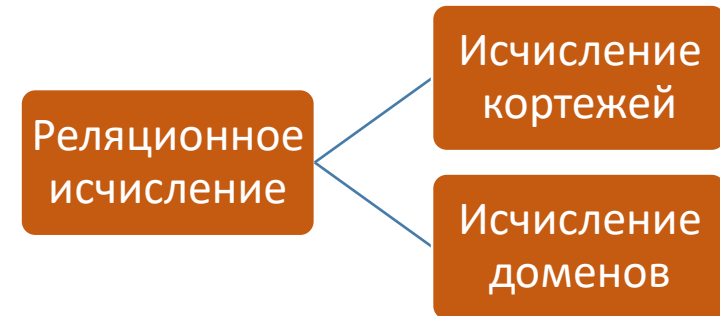
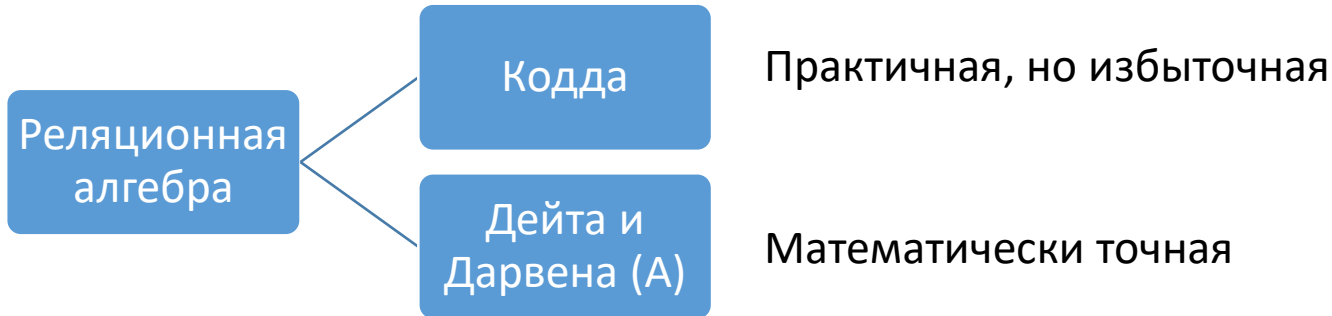


Внешний  
ключ



# БАЗИСНЫЕ СРЕДСТВА МАНИПУЛИРОВАНИЯ РЕЛЯЦИОННЫМИ ДАНЫМИ: АЛГЕБРА КОДДА





И реляционное исчисление и реляционная алгебра **замкнуты** относительно отношений. Проводя операции с отношениями, получаем отношения, над которыми вновь можно проводить операции.

Реляционное исчисление и реляционная алгебра **эквивалентны**. Для любого выражения реляционной алгебры можно построить формулу реляционного исчисления с тем же результатом.

Выражения реляционной алгебры имеют **процедурную интерпретацию**. Из выражения однозначно вытекает последовательность действий, ведущих к результату (как математическая формула со скобками).

Языки реляционного исчисления **декларативны**. Формула реляционного исчисления описывает результат (условия, которым удовлетворяют кортежи результирующего отношения). Пути к результату могут быть различны.

Так как отношения – это множества, алгебра основана на традиционных теоретико-множественных операциях. В дополнение к ним приняты специальные операции для реляционных баз данных.

## Теоретико-множественные операции

1. Объединение отношений;
2. Пересечение отношений;
3. Взятие разности отношений;
4. Взятие декартова произведения отношений.

## Специальные реляционные операции

5. Ограничение отношения;
6. Проекция отношения;
7. Соединение отношений;
8. Деление отношений.



Плюс дополнительные операции, свойственные языкам программирования – присваивание и переименование. Присваивание нужно для формирования результата, а переименование – для формирования заголовка результирующего отношения.

- При выполнении операции *объединения* (UNION) двух отношений с одинаковыми заголовками производится отношение, включающее все кортежи, которые входят хотя бы в одно из отношений-операндов.
- Операция *пересечения* (INTERSECT) двух отношений с одинаковыми заголовками производит отношение, включающее все кортежи, которые входят 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, которое является вторым операндом.
- При выполнении *декартова произведения* (TIMES) двух отношений, пересечение заголовков которых пусто, производится отношение, кортежи которого производятся путем объединения кортежей первого и второго операндов.

- 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.
- При *соединении* (JOIN) двух отношений по некоторому условию 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.

- Операция *переименования* (RENAME) производит отношение, тело которого совпадает с телом операнда, но имена атрибутов изменены.
- Операция *присваивания* ( $:=$ ) позволяет сохранить результат вычисления реляционного выражения в существующем отношении БД.

Вместо отношения-операнда может использоваться реляционное выражение (вложенное реляционное выражение). Это следствие замкнутости реляционных выражений над отношениями.

## Приоритеты операций

RENAME  $\geq$  WHERE = PROJECT  $\geq$  TIMES = JOIN = INTERSECT = DIVIDE BY  $\geq$  UNION = MINUS

## Замкнутость и операция переименования

Чтобы было возможно выполнить операцию (произвести отношение и обеспечить замкнутость) в отношениях-операндах не должны встречаться одноименные атрибуты, определенные на одних и тех же доменах.  
Для решения конфликтов используется *переименование*.

Операция	Приоритет
RENAME	4
WHERE	3
PROJECT	3
TIMES	2
JOIN	2
INTERSECT	2
DIVIDE BY	2
UNION	1
MINUS	1

# ОСОБЕННОСТИ ОПЕРАЦИЙ (1/4)



Здесь говорим только о теоретико-множественных операциях.

Вообще, все нижесказанное для операции объединения верно для пересечения и взятия разности

## Определения из теории множеств

- Результатом объединения двух множеств  $A\{a\}$  и  $B\{b\}$  является такое множество  $C\{c\}$ , что для каждого  $c$  либо существует такой элемент  $a$ , принадлежащий множеству  $A$ , что  $c=a$ , либо существует такой элемент  $b$ , принадлежащий множеству  $B$ , что  $c=b$ ;
- Пересечением множеств  $A$  и  $B$  является такое множество  $C\{c\}$ , что для любого  $c$  существуют такие элементы  $a$ , принадлежащий множеству  $A$ , и  $b$ , принадлежащий множеству  $B$ , что  $c=a=b$ ;
- Разностью множеств  $A$  и  $B$  является такое множество  $C\{c\}$ , что для любого  $c$  существует такой элемент  $a$ , принадлежащий множеству  $A$ , что  $c=a$ , и не существует такой элемент  $b$ , принадлежащий  $B$ , что  $c=b$ .

*Совместимость по объединению* (пересечению, взятию разности) равносильна равенству заголовков схем отношений-операндов.

*Некоторые отношения можно привести к совместимости через операцию переименования.*

Наличие в алгебре одновременно операций объединения, пересечения и взятия разности избыточно, так как операцию пересечения можно выразить через операцию взятия разности. Но, все для интуитивного удобства пользователя.

$$A \text{ INTERSECT } B = A \text{ MINUS } (A \text{ MINUS } B) = B \text{ MINUS } (B \text{ MINUS } A)$$

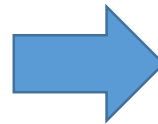
# ОСОБЕННОСТИ ОПЕРАЦИЙ (2/4)



## ПРИМЕР

СЛУЖАЩИЕ_В_ПРОЕКТЕ_1			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

СЛУЖАЩИЕ_В_ПРОЕКТЕ_2			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2939	Сидоренко	18000.00	313
2940	Федоренко	20000.00	310
2941	Иваненко	22000.00	315



СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 UNION СЛУЖАЩИЕ_В_ПРОЕКТЕ_2			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2939	Сидоренко	18000.00	313
2940	Федоренко	20000.00	310
2941	Иваненко	22000.00	315
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 INTERSECT СЛУЖАЩИЕ_В_ПРОЕКТЕ_2			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310

СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 MINUS СЛУЖАЩИЕ_В_ПРОЕКТЕ_2			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315



# ОСОБЕННОСТИ ОПЕРАЦИЙ (3/4)



## Теория множеств:

Декартовым произведением множеств  $A\{a\}$  и  $B\{b\}$  является такое множество пар  $C\{<c_1, c_2>\}$ , что для каждого элемента  $<c_1, c_2>$  множества  $C$  существуют такой элемент  $a$  множества  $A$ , что  $c_1=a$ , и такой элемент  $b$  множества  $B$ , что  $c_2=b$ .

Пары кортежей не являются отношением, поэтому вводится понятие **расширенного декартова произведения**.

Пусть имеются два отношения  $R1\{a_1, a_2, \dots, a_n\}$  и  $R2\{b_1, b_2, \dots, b_m\}$ . Тогда результатом операции  $R1 \text{ TIMES } R2$  является отношение  $R\{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$ , тело которого является множеством кортежей вида  $\{r_{a1}, r_{a2}, \dots, r_{an}, r_{b1}, r_{b2}, \dots, r_{bm}\}$  таких, что  $\{r_{a1}, r_{a2}, \dots, r_{an}\}$  входит в тело  $R1$ , а  $\{r_{b1}, r_{b2}, \dots, r_{bm}\}$  входит в тело  $R2$ .

Два отношения **совместимы по взятию расширенного декартова произведения** в том и только в том случае, если пересечение множеств имен атрибутов, взятых из их схем отношений, пусто. Любые два отношения всегда могут стать совместимыми по взятию декартова произведения, если применить операцию переименования к одному из этих отношений.

## Ассоциативность и коммутативность теоретико-множественных операций

$$(A \text{ OP } B) \text{ OP } C = A \text{ OP } (B \text{ OP } C)$$

$A \text{ OP } B = B \text{ OP } A$  – кроме взятия разности

# ОСОБЕННОСТИ ОПЕРАЦИЙ (4/4)



## ПРИМЕР РАСШИРЕННОГО ДЕКАРТОВА ПРОИЗВЕДЕНИЯ

ПРОЕКТЫ

ПРОЕКТ_НАЗВ	ПРОЕКТ_РУК
ПРОЕКТ 1	Иванов
ПРОЕКТ 2	Иваненко

СЛУЖАЩИЕ\_В\_ПРОЕКТЕ\_1

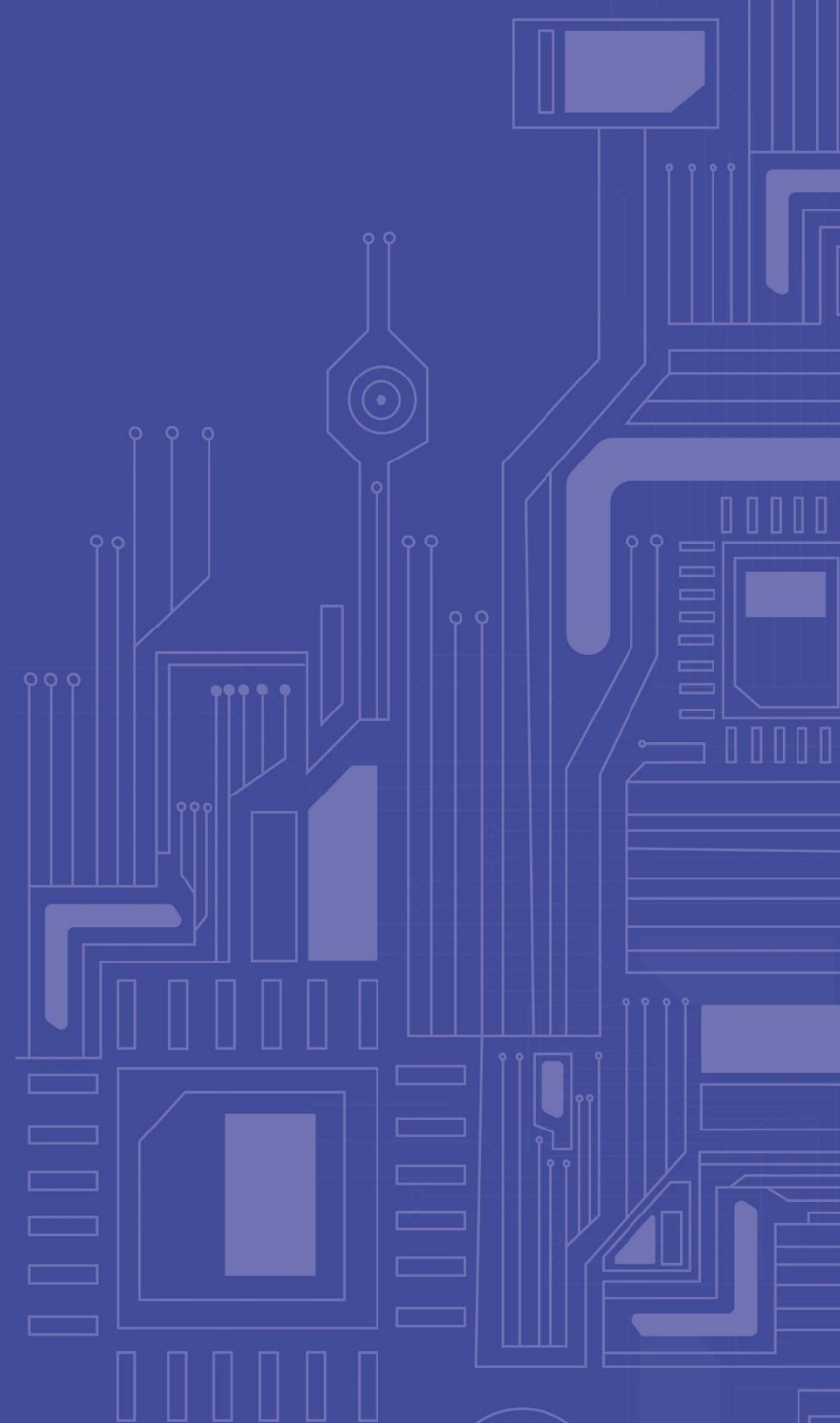
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

СЛУЖАЩИЕ\_В\_ПРОЕКТЕ\_1 TIMES ПРОЕКТЫ

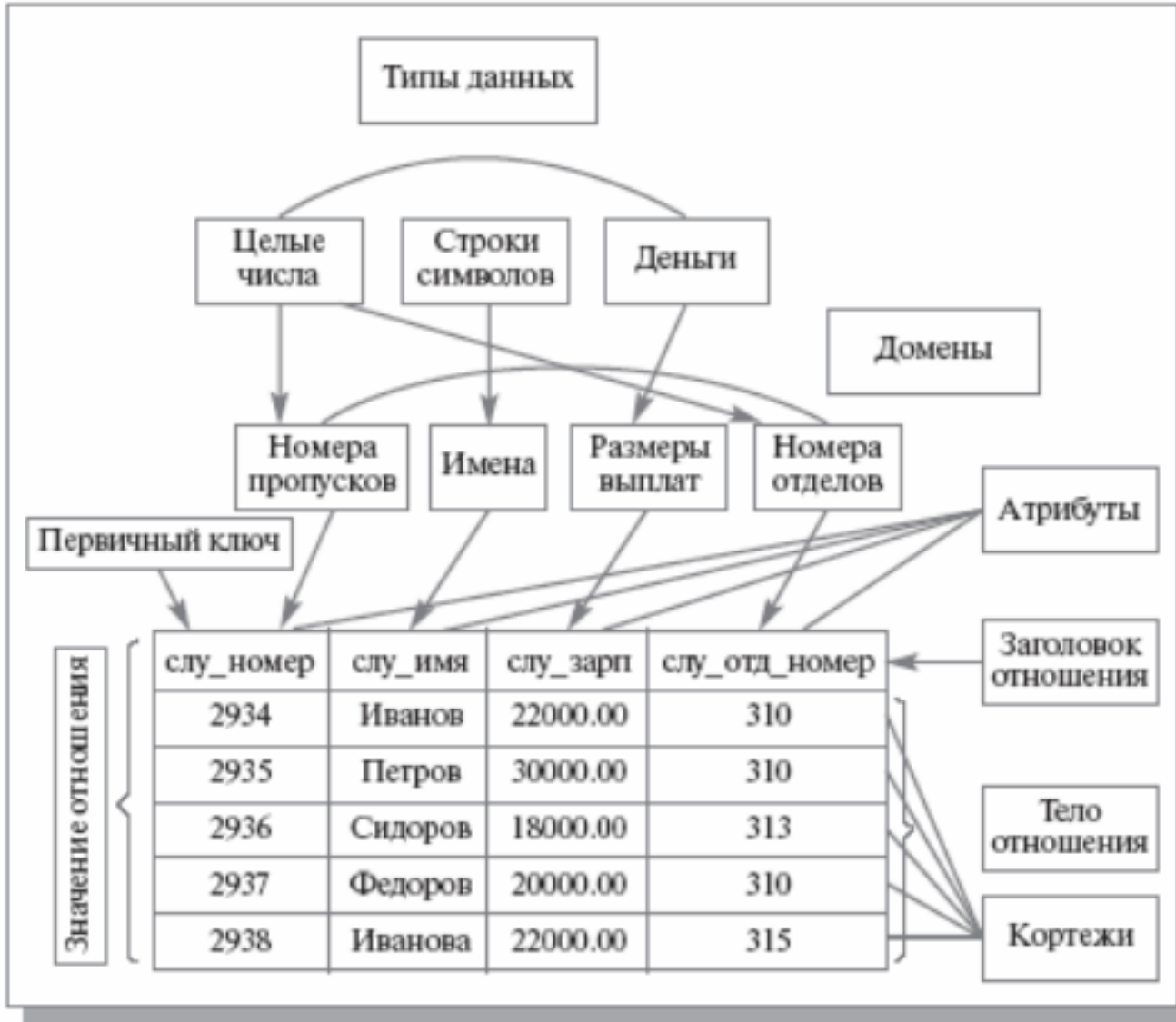
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР	ПРОЕКТ_НАЗВ	ПРОЕКТ_РУК
2934	Иванов	22000.00	310	ПРОЕКТ 1	Иванов
2935	Петров	30000.00	310	ПРОЕКТ 1	Иванов
2936	Сидоров	18000.00	313	ПРОЕКТ 1	Иванов
2937	Федоров	20000.00	310	ПРОЕКТ 1	Иванов
2938	Иванова	22000.00	315	ПРОЕКТ 1	Иванов
2934	Иванов	22000.00	310	ПРОЕКТ 2	Иваненко
2935	Петров	30000.00	310	ПРОЕКТ 2	Иваненко
2936	Сидоров	18000.00	313	ПРОЕКТ 2	Иваненко
2937	Федоров	20000.00	310	ПРОЕКТ 2	Иваненко
2938	Иванова	22000.00	315	ПРОЕКТ 2	Иваненко



СЕМИНАР



# ПРИМЕР КОРТЕЖЕЙ



Кортежем  $tr$ , соответствующим заголовку  $Hr$ , называется множество упорядоченных триплетов вида  $\langle A, T, v \rangle$ , по одному такому триплету для каждого атрибута в  $Hr$ . Третий элемент –  $v$  – триплета  $\langle A, T, v \rangle$  должен являться допустимым значением типа данных или домена  $T$ .

Заголовку отношения СЛУЖАЩИЕ соответствуют, например, следующие кортежи:  $\{\langle \text{слу\_номер}, \text{номера\_пропусков}, 2934 \rangle, \langle \text{слу\_имя}, \text{имена}, \text{Иванов} \rangle, \langle \text{слу\_зарп}, \text{размеры\_выплат}, 22.000 \rangle, \langle \text{слу\_отд\_номер}, \text{номера\_отделов}, 310 \rangle\}$ ,  $\{\langle \text{слу\_номер}, \text{номера\_пропусков}, 2940 \rangle, \langle \text{слу\_имя}, \text{имена}, \text{Кузнецов} \rangle, \langle \text{слу\_зарп}, \text{размеры\_выплат}, 35.000 \rangle, \langle \text{слу\_отд\_номер}, \text{номера\_отделов}, 320 \rangle\}$ .

# ТАБЛИЦА ЗНАЧЕНИЙ ЛОГИЧЕСКИХ ОПЕРАЦИЙ



AND	true	false	OR	true	false	NOT	true	false
true	true	false	true	true	true		false	true
false	false	false	false	true	false			