

# Лекция 8

**Базисные средства манипулирования РД**

# Базисные средства манипулирования реляционными данными

---

Определены два базовых механизма манипулирования РД:

- 1) *реляционная алгебра* (алгебра отношений);
- 2) *реляционное исчисление* (исчисление отношений).

**Реляционная алгебра** (далее – РА) основана на теории множеств.

**Реляционное исчисление** (далее – РИ) базируется на математической логике, точнее, на исчислении предикатов I-го порядка

Оба механизма обладают одним важным свойством: они **замкнуты относительно понятия *отношение***. Это означает, что выражения РА и формулы РИ определяются над *отношениями* РБД и результатом вычисления также являются *отношения*.

В результате любое выражение и любая формула могут интерпретироваться как отношение, что позволяет использовать их в других выражениях и формулах.

# Базисные средства манипулирования реляционными данными

---

Реляционная алгебра и реляционное исчисление обладают большой выразительной мощностью: очень сложные запросы к БД могут быть выражены с помощью одного выражения РА или одной формулы РИ.

Механизмы РА и РИ эквивалентны, т.е. для любого допустимого выражения РА можно построить эквивалентную (по результату) формулу РИ, и наоборот.

На основе реляционной алгебры и реляционного исчисления строятся языки баз данных.

Конкретный язык манипулирования РБД называется *реляционно-полным*, если любой запрос, представленный с помощью одного выражения РА или одной формулы РИ, может быть выражен с помощью одного оператора этого языка.

# Базисные средства манипулирования реляционными данными

---

Так как механизмы РА и РИ эквивалентны, то для проверки степени реляционности языка РБД можно использовать любой из них.

Для чего же нужны два эквивалентных формализма? Дело в том, что РА и РИ отличаются уровнем процедурности.

Выражения РА строятся на основе алгебраических операций и имеют процедурную интерпретацию.

Для формулы РИ однозначная интерпретация отсутствует. Формула только устанавливает условия, которым должны удовлетворять кортежи результирующего отношения, поэтому язык РИ является более декларативным, чем язык РА.

# Базисные средства манипулирования реляционными данными

---

РА говорит, **что и как делать** (аналогично процедурным языкам программирования таким, как Паскаль или Си), описывает последовательность действий.

РИ говорит, **какой нужен результат**, но то, как его получить, оставляет машине.

Обычно языки БД основываются на смеси РА и РИ. Здесь учитывается удобство для пользователя – как ему удобней и привычней представлять свои запросы.

# Реляционная алгебра

---

Основная идея РА состоит в том, что поскольку отношения являются множествами, то средства манипулирования ими могут базироваться на традиционных теоретико-множественных операциях, дополненных специальными операциями, специфичными для РБД.

Расширенный вариант РА, предложенный Коддом, включает 8 основных алгебраических операций и 2 дополнительные. Весь набор включает два класса – теоретико-множественные операции и специальные реляционные операции.

# Реляционная алгебра

---

## 1. Теоретико-множественные операции

В состав теоретико-множественных операций входят следующие:

- 1) объединение отношений,
- 2) пересечение отношений,
- 3) взятие разности отношений,
- 4) прямое произведение отношений.

Теоретико-множественные операции обладают свойством ассоциативности (все) и коммутативности (все, кроме разности).

# Реляционная алгебра

---

## *1. Объединение отношений*

При объединении двух отношений получается новое отношение, включающее кортежи, входящие хотя бы в одно из отношений-операндов.

## *2. Пересечение отношений*

Результат пересечения двух отношений – это отношение, включающее все кортежи, входящие в оба отношения-операнда одновременно.



# Реляционная алгебра

---

## **3. Разность отношений**

---

Результат этой операции – отношение, включающее все кортежи, входящие в отношение-первый операнд, такие что ни один из них не входит в отношение, являющееся вторым операндом.

### **Замечание.**

При выполнении операций объединения, пересечения и разности отношений отношения-операнды должны быть совместимы по этим операциям, т.е. должны обладать одинаковыми заголовками.

Более точно: в заголовках обоих отношений содержится один и тот же набор имен атрибутов и одноименные атрибуты определены на одном и том же домене. Только в этом случае результатом этих трех операций является отношение с корректно определенным заголовком, совпадающим с заголовком каждого из отношений-операндов.

Если два отношения "почти" совместимы, т.е. совместимы во всем, кроме имен некоторых атрибутов, то перед выполнением операций эти отношения можно сделать совместимыми путем применения операции переименования (см. дальше).

# Реляционная алгебра

---

## 4. Прямое произведение

---

Напомним, что в теории множеств прямое произведение двух множеств  $A$  и  $B$  ( $A \times B$ ) есть множество  $C = \{ \langle a, b \rangle \mid a \in A, b \in B \}$ . Согласно этому определению если  $A$  и  $B$  – отношения, то их прямое произведение дает множество пар кортежей, которое уже не является отношением.

Поэтому в РА используется специальная форма прямого произведения – *расширенное прямое произведение отношений*. Результатом этой операции являются не пары кортежей, а конкатенации всех кортежей из 1-го операнда со всеми кортежами из 2-го операнда.

Заметим, что мощность результирующего отношения очень велика и к тому же оно не очень осмысленно, поэтому на практике эта операция самостоятельно не применяется.

# Реляционная алгебра

---

Каждое отношение представляется не только набором кортежей (*телом*), но и своей схемой (*заголовком*).

В случае прямого произведения отношение должно иметь *корректно сформированный заголовок*, а для этого, отношения-операнды должны быть совместимы по взятию расширенного прямого произведения, т.е. множества имен атрибутов этих отношений не должны пересекаться. (Иначе произойдет потеря элементов множества, т.к. в нем все элементы должны быть различимы.)

Любые два отношения могут быть сделаны совместимыми по взятию расширенного прямого произведения путем применения операции переименования к одному из этих отношений.

## 2. Специальные реляционные операции

К специальным реляционным операциям относятся:

- 1) ограничение отношений,
- 2) проекция отношений,
- 3) соединение отношений,
- 4) деление отношений.

# Реляционная алгебра

---

## *I. Ограничение отношений*

---

Операция ограничения отношений (ОО) выполняется над одним операндом-отношением и включает простое условие ограничения

*A WHERE comr,*

где *A* – ограничиваемое отношение,

*comr* – простое условие ограничения, имеющее вид:

$a \Theta b$ , где  $a$  и  $b$  – имена атрибутов,  $\Theta$  – операция сравнения ( $>, <, =, \#, \dots$ ),

либо

$a \Theta const$ , где  $a$  – имя атрибута, а  $const$  – константа.

Результатом выполнения операции ОО является отношение, заголовок которого совпадает с заголовком отношения-операнда, а в тело входят те кортежи отношения-операнда, для которых значением условия ограничения является *true* (истина).

Приведем пример операции ограничения отношений:

*НЕСОВЕРШЕННОЛЕТНИЙ =*

*СТУДЕНТ WHERE СТУДЕНТ.*

*ВОЗРАСТ < 18*

---

# Реляционная алгебра

---

Пусть  $comp1$  и  $comp2$  – два простых условия ограничения. Тогда по определению

(1)  $A \text{ WHERE } comp1 \text{ AND } comp2$

эквивалентно

$(A \text{ WHERE } comp1) \cap (A \text{ WHERE } comp2);$

(2)  $A \text{ WHERE } comp1 \text{ OR } comp2$

эквивалентно

$(A \text{ WHERE } comp1) \cup (A \text{ WHERE } comp2);$

(3)  $A \text{ WHERE NOT } comp$

эквивалентно

$A \setminus (A \text{ WHERE } comp).$

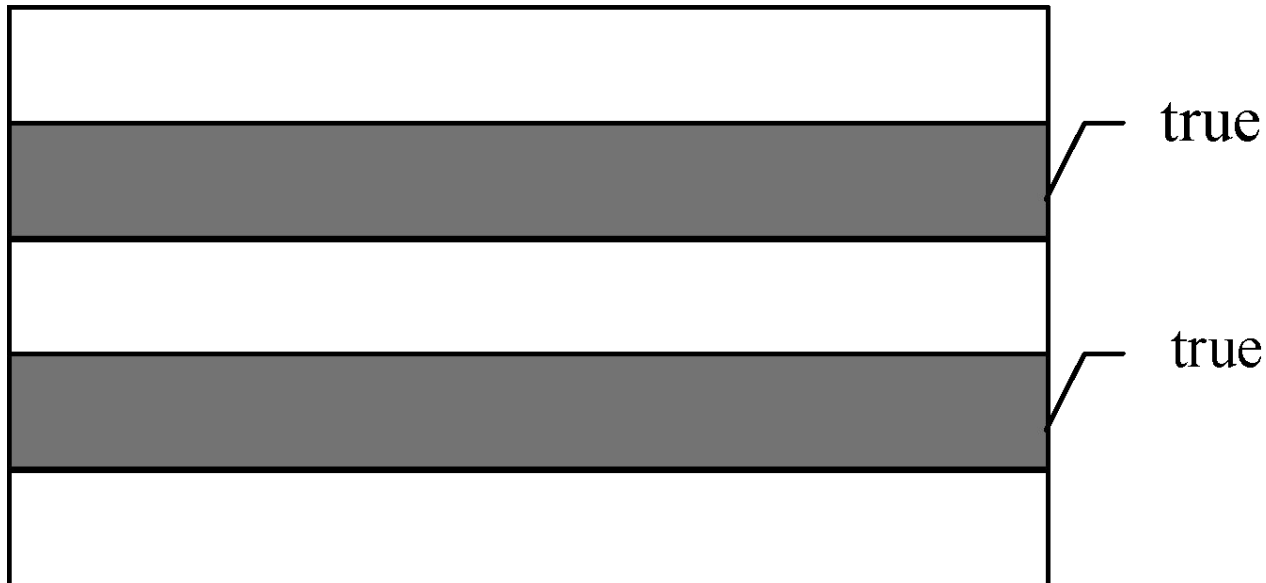
Благодаря этому, можно формулировать операции  $\circ\circ$ , в которых условием ограничения является любое булево выражение, составленное из простых условий с использованием логических связок *AND*, *OR*, *NOT* и скобок.

# Реляционная алгебра

---

На интуитивном уровне операцию ограничения можно представить как взятие некоторой горизонтальной вырезки отношения-операнда

*A WHERE comp*



# Реляционная алгебра

---

## 2. Взятие проекции

---

Операция взятия проекции требует наличия двух операндов: проецируемого отношения  $A$  и списка имен атрибутов, входящих в его заголовок.

Результатом проекции отношения  $A$  по списку атрибутов  $(a_1, a_2, \dots, a_m)$  является новое отношение  $B$  с заголовком, определяемым множеством атрибутов  $(a_1, a_2, \dots, a_m)$ , и телом, состоящим из кортежей вида  $\langle a_1 : v_1, a_2 : v_2, \dots, a_m : v_m \rangle$ , таких что в отношении  $A$  имеется кортеж, атрибут  $a_1$  которого имеет значение  $v_1, a_2 - v_2, \dots, a_m - v_m$ .

Тем самым при выполнении операции проекции делается вертикальная вырезка отношения-операнда с естественным уничтожением возникающих кортежей-дубликатов.



# Реляционная алгебра

---

Отношение

*СЛУЖАЩИЙ* (*Номер\_Сл, Имя\_Сл, Номер\_Отд, Зарплата, Адрес*)

проецируется в новое отношение

*СЛУЖ* (*Номер\_Сл, Имя\_Сл, Номер\_Отд, Зарплата*)

оператором

$СЛУЖ = \Pi СЛУЖАЩИЙ (Номер\_Сл, Имя\_Сл, Номер\_Отд, Зарплата),$

где  $\Pi$  – операция проекции.

Аналогично, отношение

*ОТДЕЛ* (*Номер\_Отд, Адрес*)

можно получить с помощью оператора

$ОТДЕЛ = \Pi СЛУЖАЩИЙ (Номер\_Отд, Адрес).$

# СЛУЖАЩИЙ

Номер Сл	Имя Сл	Номер Отд	Зарплата	Адрес
5021	Иванов	721	1200	Москва
5022	Петров	721	1200	Москва
5023	Сидоров	750	1500	Париж
5024	Осипов	750	1400	Париж
5025	Попов	721	1300	Москва

ОТДЕЛ

Номер Отд	Адрес
721	Москва
750	Париж

СЛУЖ

Номер Сл	Имя Сл	Номер Отд	Зарплата
5021	Иванов	721	1200
5022	Петров	721	1200
5023	Сидоров	750	1500
5024	Осипов	750	1400
5025	Попов	721	1300

# Реляционная алгебра

---

## 3. Соединение отношений

---

Эта операция имеет три операнда: два из них – соединяемые отношения, третий – простое условие.

Эта операция имеет еще одно название – *соединение отношений по условию*:

$A, B \text{ WHERE } \text{comp}$ ,

где  $A, B$  – соединяемые отношения,

$\text{comp}$  – простое условие вида  $a \Theta b$  или  $a \Theta \text{const}$ , где  $a, b$  – имена атрибутов соответственно из  $A$  и  $B$ , а  $\text{const}$  – константа.

По определению результатом операции соединения является отношение, получаемое путем выполнения операции ограничения по условию  $\text{comp}$  прямого произведения отношений  $A$  и  $B$ .

# Реляционная алгебра

---

Частными случаями соединений являются эквисоединение и естественное соединение.

*Эквисоединение* – это соединение, где условие соединения имеет вид  $(a = b)$ , где  $a$  и  $b$  – атрибуты из разных операндов.  
(Этот случай имеет важное значение, ибо часто встречается на практике и имеет эффективные алгоритмы реализации.)

Операция *естественное соединение* применяется к паре отношений  $A$  и  $B$ , обладающих общим, возможно составным, атрибутом  $c$  (атрибут  $c$  имеет одно и то же имя и определен на одном и том же домене).

# Реляционная алгебра

Для иллюстрации естественного соединения вернемся к примеру, иллюстрирующему проекцию отношения. Только теперь будет решаться обратная задача: из отношений *СЛУЖ* и *ОТДЕЛ* получить новое отношение *СЛУЖАЩИЙ*, являющееся их естественным соединением. При этом общим атрибутом будет являться *Номер\_Отд.*

*СЛУЖАЩИЙ*

Номер Сл	Имя Сл	Номер Отд	Зарплата	Адрес
5021	Иванов	721	1200	Москва
5022	Петров	721	1200	Москва
5023	Сидоров	750	1500	Париж
5024	Осипов	750	1400	Париж
5025	Попов	721	1300	Москва

*ОТДЕЛ*

Номер Отд	Адрес
721	Москва
750	Париж

*СЛУЖ*

Номер Сл	Имя Сл	Номер Отд	Зарплата
5021	Иванов	721	1200
5022	Петров	721	1200
5023	Сидоров	750	1500
5024	Осипов	750	1400
5025	Попов	721	1300

# Реляционная алгебра

Если вспомнить введенное ранее определение *внешнего ключа*, то понятно, что основной смысл операции естественного соединения – это обеспечение возможности восстановления сложной сущности, декомпозированной (разбитой) по причине требования нормализации отношения.

В данном примере атрибут *Номер\_Отд* играет роль внешнего ключа и служит для восстановления сущности *ОТДЕЛ*.

*СЛУЖАЩИЙ*

Номер Сл	Имя Сл	Номер Отд	Зарплата	Адрес
5021	Иванов	721	1200	Москва
5022	Петров	721	1200	Москва
5023	Сидоров	750	1500	Париж
5024	Осипов	750	1400	Париж
5025	Попов	721	1300	Москва

*ОТДЕЛ*

Номер Отд	Адрес
721	Москва
750	Париж

*СЛУЖ*

Номер Сл	Имя Сл	Номер Отд	Зарплата
5021	Иванов	721	1200
5022	Петров	721	1200
5023	Сидоров	750	1500
5024	Осипов	750	1400
5025	Попов	721	1300

# Реляционная алгебра

---

## 4. Деление отношений

---

Пусть даны два отношения:

$A$  с заголовком  $\{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$  и

$B$  с заголовком  $\{b_1, b_2, \dots, b_m\}$ .

Будем считать, что атрибуты  $b_i$  отношения  $A$  и атрибуты  $b_i$  отношения  $B$  не только обладают одним и тем же именем, но и определены на одном и том же домене.

Назовем множество атрибутов  $\{a_1, a_2, \dots, a_n\}$  составным атрибутом  $a$ , а множество  $\{b_1, b_2, \dots, b_m\}$  составным атрибутом  $b$ . Тогда можно говорить о реляционном делении бинарного отношения  $A(a, b)$  на унарное отношение  $B(b)$ .

Результатом деления  $A$  на  $B$  является унарное отношение  $C(a)$ , состоящее из кортежей  $\langle v \rangle$ , таких что в  $A$  имеются кортежи  $\langle v, w \rangle$ , такие что для каждого  $w$  в  $B$  есть кортеж  $\langle w \rangle$ .

# Реляционная алгебра

---

## Пример.

Пусть есть два отношения *СОТРУДНИКИ* (*Имя, Номер\_Отд*) и *ИМЕНА* (*Имя*), причем унарное отношение *ИМЕНА* содержит фамилии всех сотрудников организации, имеющих ученую степень.

Тогда в результате деления отношения *СОТРУДНИКИ* на отношение *ИМЕНА* получим унарное отношение, содержащее номера отделов, в которых работают сотрудники, чьи фамилии включены в отношение *ИМЕНА*.



## 3. Дополнительные операции

*Операция переименования* выполняется над отношением-операндом и имеет результатом новое отношение, тело которого совпадает с телом операнда, но в заголовке имена атрибутов изменены.

*Операция присваивания* позволяет сохранить результат вычисления реляционного выражения в существующем отношении базы данных.

# Реляционное исчисление

---

## 1. Основные понятия

Основными понятиями реляционного исчисления являются *переменная* и *правильно построенная формула* (далее – ППФ).

В зависимости от области определения *переменной* (далее – ООП) различают *исчисление кортежей* и *исчисление доменов* и соответственно кортежные и доменные переменные.

В первом случае ООП переменной являются кортежи некоторого отношения, во втором – домены.

Для определения кортежной переменной используется оператор

*RANGE*, например,

*RANGE COTP IS СОТРУДНИКИ*,

где *COTP* – переменная, *СОТРУДНИКИ* – отношение, на котором определена переменная *COTP*.

В любой момент переменная *COTP* представляет некоторый кортеж отношения *СОТРУДНИКИ*.

# Реляционное исчисление

---

При использовании кортежной переменной в формулах можно ссылаться на значения атрибутов переменной, например, *СОТР.ИМЯ*.

Правильно построенная формула служит для выражения условий, накладываемых на кортежные переменные.

При построении ППФ используются простые сравнения, логические связки (*NOT, AND, OR, IF...THEN*), а также кванторы существования (*EXISTS*) и всеобщности (*FORALL*).

# Реляционное исчисление

---

Приведем правила построения ППФ.

- 1) Простое сравнение есть ППФ.
- 2) Если  $F$  и  $Q$  – формулы, то  $NOT F$ ,  $(F AND Q)$ ,  $(F OR Q)$ ,  $(IF F THEN Q)$  – тоже формулы.
- 3) Если  $F$  – формула, а  $x$  – свободная переменная в  $F$ , то  $EXISTS x (F)$  и  $FORALL x (F)$  – формулы.
- 4) Других формул нет.

О переменных говорят, что они либо *свободные*, либо *связанные*. Переменные, входящие в ППФ без кванторов, являются свободными. Только свободные переменные могут входить в результирующее отношение.

Переменные под квантором являются связанными. Это означает, что такая переменная связана с данной ППФ, т.е. не видна за ее пределами. При вычислении значения такой ППФ используется не одно значение связанной переменной, а **вся** ее область определения.

# Реляционное исчисление

---

Пример: пусть  $COTP1$  и  $COTP2$  – кортежные переменные, определенные на отношении  $COTРУДНИКИ$ , тогда ППФ  $EXISTS\ COTP2\ (COTP1.Compr\_Zarp > COTP2.Compr\_Zarp)$  для текущего кортежа переменной  $COTP1$  принимает значение *true* в том и только том случае, если во всем отношении  $COTРУДНИКИ$  найдется кортеж такой, что значение его атрибута  $Compr\_Zarp$  удовлетворяет внутреннему условию сравнения.

Формула

$FORALL\ COTP2\ (COTP1.Compr\_Zarp > COTP2.Compr\_Zarp)$  для текущего кортежа переменной  $COTP1$  принимает значение *true* в том и только том случае, если для всех кортежей отношения  $COTРУДНИКИ$  (связанных с переменной  $COTP2$ ) значение атрибута  $Compr\_Zarp$  удовлетворяет внутреннему условию сравнения.

# Реляционное исчисление

---

На самом деле принято говорить не о свободных и связанных переменных, а о свободных и связанных вхождениях переменных.

Так, если переменная  $x$  является связанной в  $F$ , то во всех других ППФ она может быть как свободной, так и связанной, но в любом случае не имеет никакого отношения к вхождению  $x$  в  $F$ .

Рассмотрим следующую формулу:

*EXISTS COTP2*

$(COTP1.Compr\_Отд\_Ном = COTP2.Compr\_Отд\_Ном)$

*AND*

*FORALL COTP2*

$(COTP1.Compr\_Зарп > COTP2.Compr\_Зарп).$

Здесь мы имеем два связанных вхождения переменной  $COTP2$  с совершенно разным смыслом.

## 2. Целевые списки и выражения

Правильно построенные формулы обеспечивает только средства формулирования условий выборки кортежей из отношений БД. Для определения набора имен атрибутов результирующего отношения служат *целевые списки (target lists)*.

Целевой список строится из *целевых элементов* следующего вида:

$x.attr$ , где  $x$  – имя свободной переменной из ППФ, а  $attr$  – имя атрибута отношения, на котором определена  $x$ ;

$x$ , что эквивалентно наличию подписка  $x.attr_1 \dots x.attr_n$ , где  $attr_1 \dots attr_n$  включает все имена атрибутов определяющего отношения;

$new\_name = x.attr$ , где  $new\_name$  – новое имя соответствующего атрибута результирующего отношения.

Последний элемент требуется, когда в ППФ используется несколько свободных переменных с одинаковой областью определения.

# Реляционное исчисление

---

Выражение реляционного исчисления имеет вид

*target\_list WHERE ППФ.*

Значением такого выражения является отношение, тело которого определяется *ППФ*, а набор атрибутов и их имена – целевым списком *target\_list*.

Пример: пусть мы имеем БД со схемой

*СОТРУДНИКИ (Сотр\_Ном, Сотр\_Имя, Сотр\_Зарп, Отд\_Ном)*

*ОТДЕЛЫ (Отд\_Ном, Отд\_Кол, Отд\_Нач).*

Требуется выдать имена и номера сотрудников, являющихся начальниками отделов с количеством сотрудников больше 50.



# Реляционное исчисление

Пример: пусть мы имеем БД со схемой

*СОТРУДНИКИ* (*Сотр\_Ном*, *Сотр\_Имя*, *Сотр\_Зарп*, *Отд\_Ном*)

*ОТДЕЛЫ* (*Отд\_Ном*, *Отд\_Кол*, *Отд\_Нач*).

Требуется выдать имена и номера сотрудников, являющихся начальниками отделов с количеством сотрудников больше 50.

В терминах реляционной алгебры потребовалось бы выполнить следующую последовательность операций:

- 1) выполнить соединение отношений *СОТРУДНИКИ* и *ОТДЕЛЫ* по условию  $Сотр\_Ном = Отд\_Нач$ ;
- 2) ограничить полученное отношение по условию  $Отд\_Кол > 50$ ;
- 3) спроецировать результат операции на атрибуты *Сотр\_Ном* и *Сотр\_Имя*.

# Реляционное исчисление

---

Пример: пусть мы имеем БД со схемой

*СОТРУДНИКИ (Сотр\_Ном, Сотр\_Имя, Сотр\_Зарп, Отд\_Ном)*

*ОТДЕЛЫ (Отд\_Ном, Отд\_Кол, Отд\_Нач).*

Требуется выдать имена и номера сотрудников, являющихся начальниками отделов с количеством сотрудников больше 50.

В терминах реляционного исчисления этот же запрос выглядит следующим образом:

```
SELECT СОТР.Сотр_Ном, СОТР.Сотр_Имя WHERE  
  EXISTS ОТД (ОТД.Отд_Нач = СОТР.Сотр_Ном  
             AND ОТД.Отд_Кол > 50),
```

где *RANGE СОТР IS СОТРУДНИКИ*, а *RANGE ОТД IS ОТДЕЛЫ*.

## 3. Особенности исчисления доменов

В исчислении доменов областью определения переменной являются не отношения, а домены. Например, переменные *ИМЯ* и *НОМ\_СОТР* могут быть определены на доменах соответственно атрибутов *Сотр\_Имя* и *Сотр\_Ном*.

Основным формальным отличием исчисления доменов от исчисления кортежей является наличие в ППФ дополнительных предикатов, позволяющих выражать условия членства. В остальном формулы и выражения исчисления доменов похожи на формулы и выражения исчисления кортежей.

# Реляционное исчисление

---

Если  $R$  – это  $n$ -арное отношение с атрибутами  $a_1, a_2, \dots, a_n$ , то условие членства имеет вид

$$R ( a_1 : v_1, a_2 : v_2, \dots, a_m : v_m ) (m \leq n),$$

где  $v_i$  - константа или доменная переменная.

Условие членства истинно, если в отношении  $R$  существует кортеж, содержащий указанные значения атрибутов.

Исчисление доменов проиллюстрируем на *примере* следующего запроса: *Выдать номера и имена сотрудников, получающих зарплату больше минимальной.*

# Реляционное исчисление

---

Для простоты будем считать, что мы определили доменные переменные, имена которых совпадают с именами атрибутов отношения *СОТРУДНИКИ*; если требуется несколько доменных переменных, определенных на одном и том же домене, мы будем добавлять в конце их имени цифры. Тогда запрос будет иметь вид:

```
SELECT Сотр_Ном, Сотр_Имя WHERE  
    EXISTS Сотр_Зарп1 ( СОТРУДНИКИ (Сотр_Зарп1)  
        AND СОТРУДНИКИ (Сотр_Ном, Сотр_Имя, Сотр_Зарп)  
        AND Сотр_Зарп > Сотр_Зарп1).
```

Заметим, что на исчислении доменов базировался известный язык Query-by-Example.