



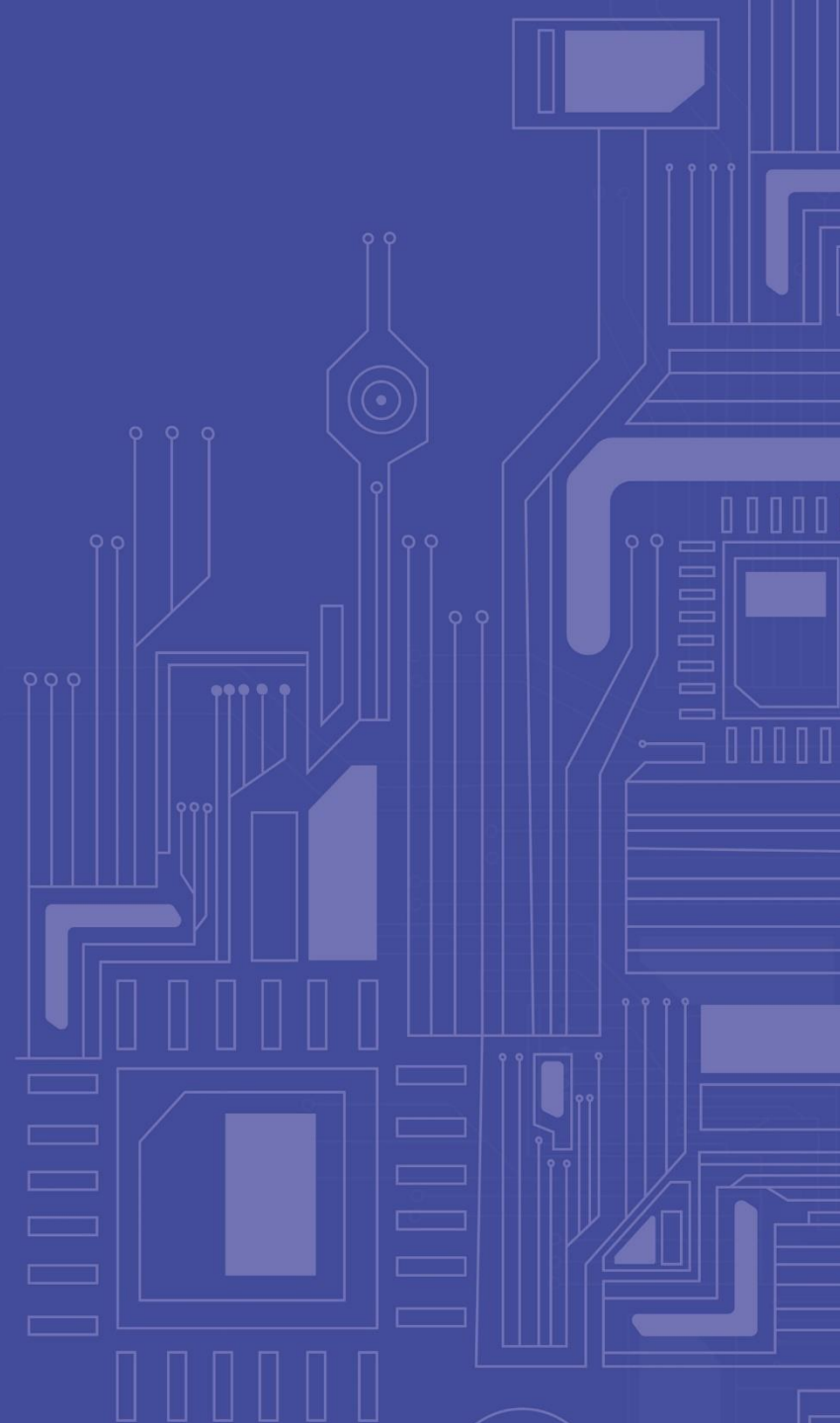
МИНОБРНАУКИ
РОССИИ



Передовые
инженерные
школы

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

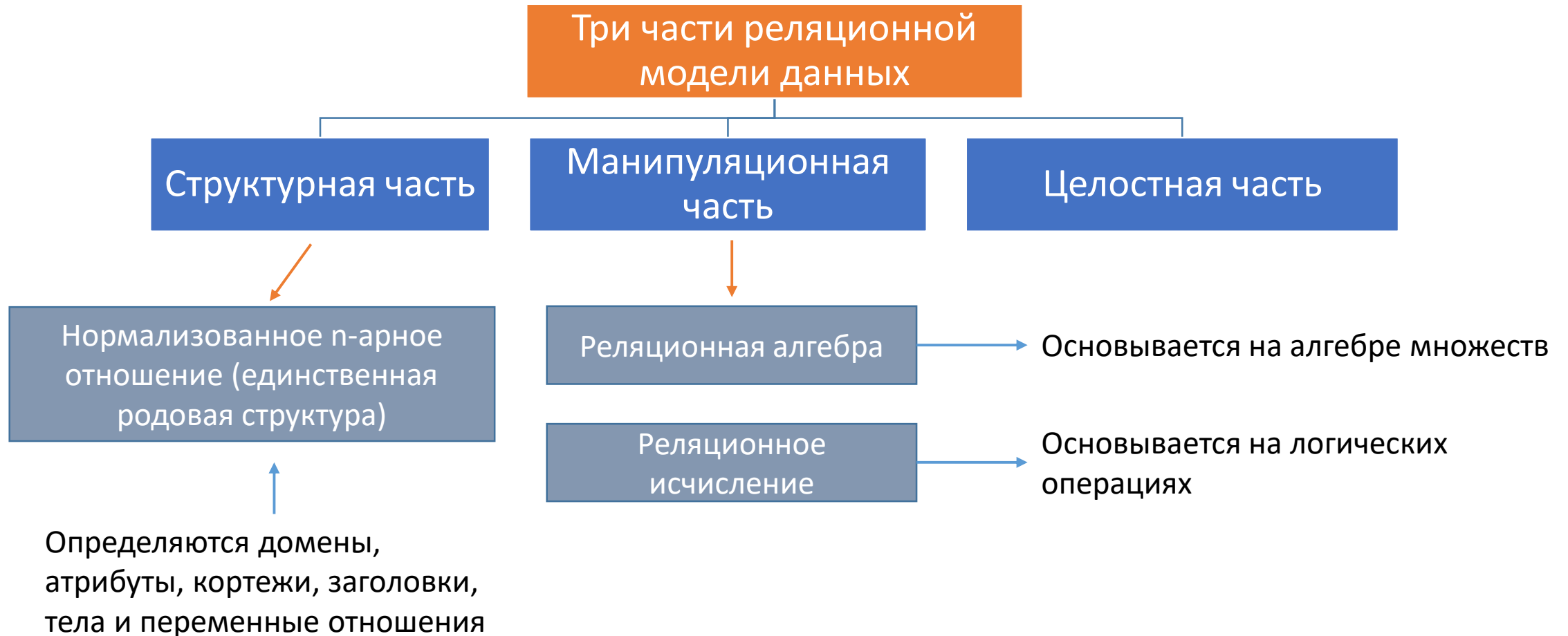
Лекция 3



- Описание реляционной модели данных
 - Алгебра Кодда
 - Специальные реляционные операции
 - Алгебра А Дейта и Дарвена
 - Реляционное исчисление

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:

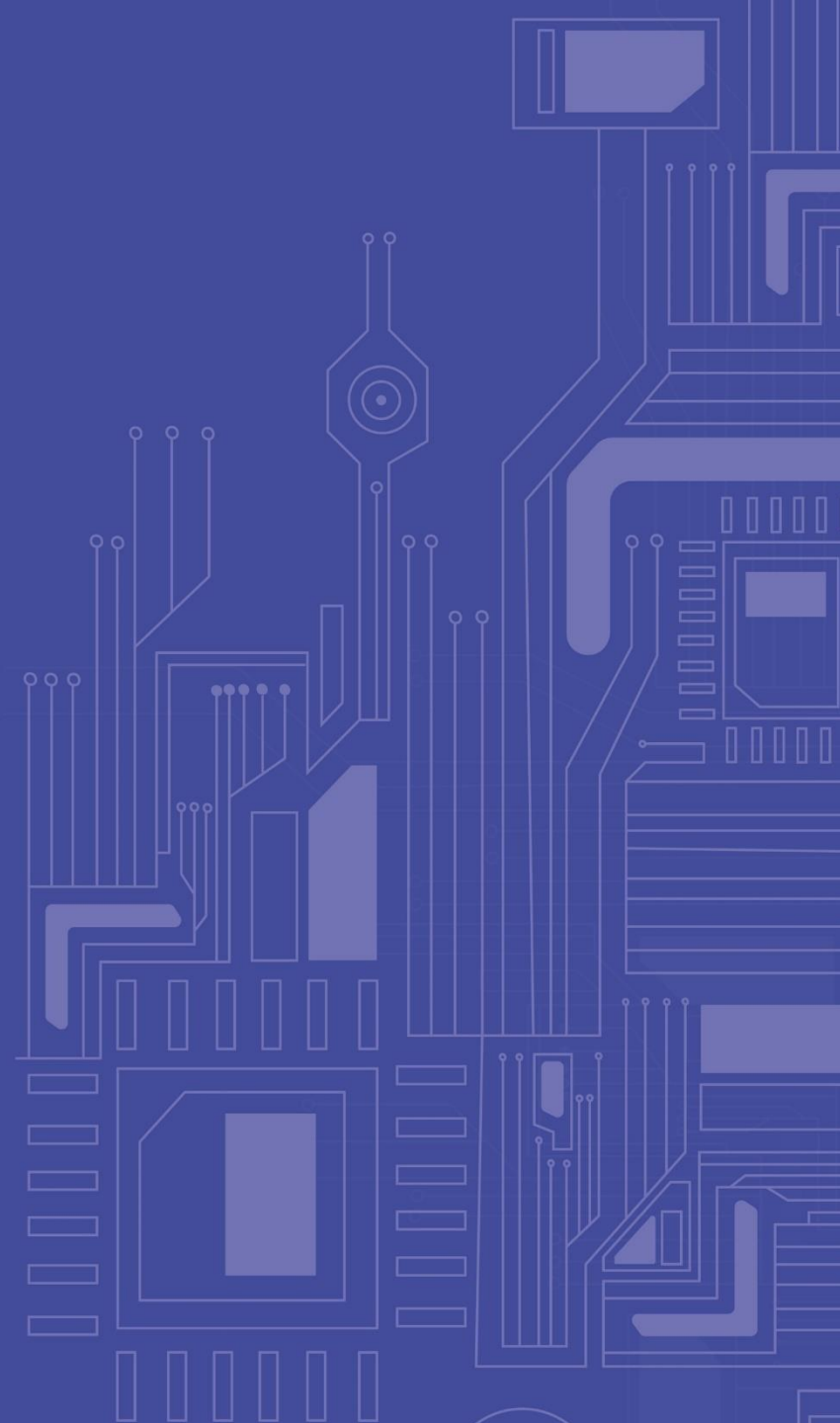
- Удалить все связанные записи (удалили отдел, удалить из базы всех служащих этого отдела)
- Всем связанным записям поставить внешний ключ NULL
- Запретить удаление кортежа, если на него есть ссылки (сначала удалить все ссылки, а потом сам кортеж)

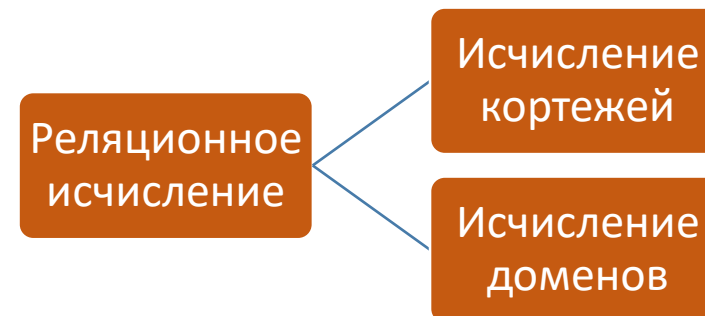
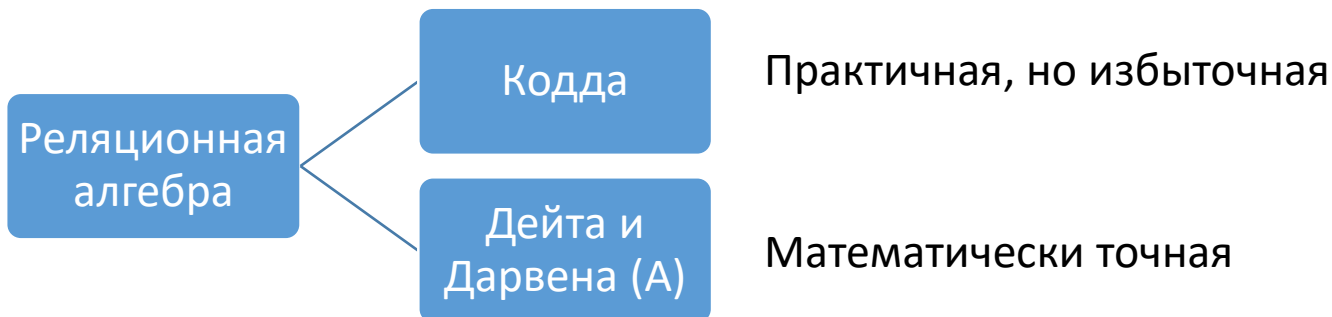
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315



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.

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

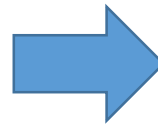
ОСОБЕННОСТИ ОПЕРАЦИЙ (2/4)



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

СПЕЦИАЛЬНЫЕ РЕЛЯЦИОННЫЕ ОПЕРАЦИИ



Операция ограничения WHERE требует наличия двух операндов: ограничиваемого отношения и простого условия ограничения. Простое условие ограничения может иметь:

- вид $(a \text{ comp-ор } b)$, где a и b – имена атрибутов ограничиваемого отношения; атрибуты a и b должны быть определены на одном и том же домене, для значений базового типа данных которого поддерживается операция сравнения *comp-ор*, или на базовых типах данных, над значениями которых можно выполнять эту операцию сравнения;
- или вид $(a \text{ comp-ор } \text{const})$, где a – имя атрибута ограничиваемого отношения, а *const* – литерально заданная константа; атрибут a должен быть определен на домене или базовом типе, для значений которого поддерживается операция сравнения *comp-ор*.

Операцией сравнения *comp-ор* могут быть «=», «≠», «>», «≥», «<», «≤».

Для **обозначения вызова операции ограничения** будем использовать конструкцию $A \text{ WHERE comp}$, где A – ограничиваемое отношение, а *comp* – простое условие сравнения. Пусть *comp1* и *comp2* – два простых условия ограничения. Тогда по определению:

- $A \text{ WHERE } (comp1 \text{ AND } comp2)$ равносильно $(A \text{ WHERE } comp1) \text{ INTERSECT } (A \text{ WHERE } comp2)$;
- $A \text{ WHERE } (comp1 \text{ OR } comp2)$ равносильно $(A \text{ WHERE } comp1) \text{ UNION } (A \text{ WHERE } comp2)$;
- $A \text{ WHERE NOT } comp1$ равносильно $A \text{ MINUS } (A \text{ WHERE } comp1)$.

Операция взятия проекции также требует наличия двух операндов – проецируемого отношения А и подмножества множества имен атрибутов, входящих в заголовок отношения А.

Результатом проекции отношения А на множество атрибутов { a_1, a_2, \dots, a_n } (PROJECT А { a_1, a_2, \dots, a_n }) является отношение с заголовком, определяемым множеством атрибутов { a_1, a_2, \dots, a_n }, и с телом, состоящим из кортежей вида $\langle a_1:v_1, a_2:v_2, \dots, a_n:v_n \rangle$ таких, что в отношении А имеется кортеж, атрибут a_1 которого имеет значение v_1 , атрибут a_2 имеет значение v_2 , ..., атрибут a_n имеет значение v_n . Тем самым, при выполнении операции проекции выделяется «вертикальная» вырезка отношения-операнда с естественным уничтожением потенциально возникающих кортежей-дубликатов.

ПРИМЕРЫ

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

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2938	Иванова	22000.00	315

СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 **WHERE** (СЛУ_ЗАРП 200 > 20000.00 AND (СЛУ_ОТД_НОМ = 310 OR СЛУ_ОТД_НОМ = 315))

СЛУ_ОТД_НОМЕР
310
313
315

PROJECT
СЛУЖАЩИЕ_В_ПРОЕКТЕ_1
{СЛУ_ОТД_НОМ}

Уничтожение дублей требует дорогостоящей (по числу операций) внешней сортировки.

Результатом **операции соединения** **A JOIN B WHERE *сomp*** совместимых по взятию расширенного декартова произведения отношений A и B является отношение, получаемое путем выполнения операции ограничения по условию *сomp* расширенного декартова произведения отношений A и B.

A JOIN B WHERE *сomp* равносильно **(A TIMES B) WHERE *сomp***

Операцию соединения с условием *сomp* разумно считать операцией **действительного** соединения, если оно имеет вид (или может быть преобразовано к виду) *сomp1 AND (a *сomp*-or b)*, где *a* и *b* – имена атрибутов **разных отношений-операндов**.

СЛУЖАЩИЕ			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

ПРОЕКТЫ		
ПРО_НОМ	ПРОЕКТ_РУК	ПРО_ЗАРП
1	Иванов	22400.00
2	Иваненко	22400.00

СЛУЖАЩИЕ JOIN ПРОЕКТЫ WHERE
(СЛУ_ЗАРП > ПРО_ЗАРП)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРО_НОМ1	ПРОЕКТ_РУК	ПРО_ЗАРП
2935	Петров	29600.00	1	1	Иванов	22400.00
2935	Петров	29600.00	2	2	Иваненко	22400.00

Действительное соединение реально уменьшает мощность промежуточного декартова произведения отношений-операндов в отличие от простого условия лишь на атрибут одного операнда.

ЧАСТНЫЕ СЛУЧАИ СОЕДИНЕНИЯ

EQUIJOIN

Операция соединения называется *операцией эквисоединения*, если условие соединения имеет вид ($a = b$), где a и b – атрибуты разных операндов соединения. Этот случай важен потому, что он чаще всего встречается на практике, и для него существуют наиболее эффективные алгоритмы реализации.

NATURAL JOIN

Операция естественного соединения применяется к паре отношений A и B , обладающих (возможно, составным) общим атрибутом c (т. е. атрибутом с одним и тем же именем и определенным на одном и том же домене). Пусть AB обозначает объединение заголовков отношений A и B . Тогда *естественное соединение* A и B – это спроецированный на AB результат эквисоединения A и B по условию $A.c = B.c$.

Результат операции `СЛУЖАЩИЕ JOIN (ПРОЕКТЫ RENAME (ПРО_НОМ, ПРО_НОМ1)) WHERE (СЛУ_ЗАРП = ПРО_ЗАРП)`

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРО_НОМ1	ПРОЕКТ_РУК	ПРО_ЗАРП
2934	Иванов	22400.00	1	1	Иванов	22400.00
2934	Иванов	22400.00	2	2	Иваненко	22400.00

Результат операции `СЛУЖАЩИЕ NATURAL JOIN ПРОЕКТЫ`

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ	ПРОЕКТ_РУК	ПРО_ЗАРП
2934	Иванов	22400.00	1	Иванов	22400.00
2935	Петров	29600.00	1	Иванов	22400.00
2936	Сидоров	18000.00	1	Иванов	22400.00
2937	Федоров	20000.00	1	Иванов	22400.00
2938	Иванова	22000.00	1	Иванов	22400.00
2934	Иванов	22400.00	2	Иваненко	22400.00
2935	Петров	29600.00	2	Иваненко	22400.00
2939	Сидоренко	18000.00	2	Иваненко	22400.00
2940	Федоренко	20000.00	2	Иваненко	22400.00
2941	Иваненко	22000.00	2	Иваненко	22400.00

Естественное соединение часто используется для восстановления после декомпозиции в 1NF сложных структур.

ДЕЛЕНИЕ ОТНОШЕНИЙ

Пусть заданы 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 атрибут b_i отношения A и атрибут b_i отношения B ($i = 1, 2, \dots, m$) не только обладают одним и тем же именем, но и определены на одном и том же домене. Назовем множество атрибутов $\{a_j\}$ составным атрибутом a , а множество атрибутов $\{b_j\}$ – составным атрибутом b . После этого будем говорить о реляционном делении «бинарного» отношения $A\{a, b\}$ на унарное отношение $B\{b\}$.

Результатом **деления** A на B (A **DIVIDE BY** B) является «унарное» отношение $C\{a\}$, тело которого состоит из кортежей v таких, что в теле отношения A содержатся кортежи $v \text{ UNION } w$ такие, что множество $\{w\}$ включает тело отношения B.

СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

Отношение НОМЕРА_ПРОЕКТОВ

ПРО_НОМ
1
2

Результат операции СЛУЖАЩИЕ DIVIDE BY НОМЕРА_ПРОЕКТОВ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00

АЛГЕБРА А ДЕЙТА И ДАРВЕНА



Сравнение алгебры Кодда и алгебры А Дейта и Дарвена

Алгебра Кодда

Опирается на теорию множеств.

Базовые операции:

- переименование атрибутов,
- объединение,
- пересечение,
- взятие разности,
- декартово произведение,
- проекция,
- ограничение

Соединение можно выразить через другие операции.
Проще определяются алгебраические черты SQL.

Алгебра А Дейта и Дарвена

Опирается на логические операции.

Базовые операции:

- реляционного отрицания (дополнения),
- реляционной конъюнкции (или дизъюнкции),
- проекции (удаления атрибута).

Через базовые операции выражаются:

- переименования атрибутов,
- соединения общего вида,
- взятия разности отношений.

Менее практичная. Более теоретизированная.

Обозначения

r – отношение,
 A – имя атрибута отношения r ,
 T – имя соответствующего типа (т. е. типа или домена атрибута A),
 v – значение типа T .

Описание операции состоит из:

1. Формальная спецификация ограничений,
2. Формальная спецификация заголовка результата,
3. Пояснения.

- заголовком H_r отношения r называется множество атрибутов, т.е. упорядоченных пар вида $\langle A, T \rangle$. По определению никакие два атрибута в этом множестве не могут содержать одно и то же имя атрибута A ;
- кортеж t_r соответствующий заголовку H_r – это множество упорядоченных триплетов вида $\langle A, T, v \rangle$, по одному такому триплету для каждого атрибута в H_r ;
- тело B_r отношения r – это множество кортежей t_r . Заметим, что (в общем случае) могут существовать такие кортежи t_r которые соответствуют H_r но не входят в B_r .

Поскольку некоторые базовые операции Алгебры А имеют названия обычных логических операций, чтобы избежать путаницы, имена реляционных операций берутся в угловые скобки: $\langle \text{NOT} \rangle$, $\langle \text{AND} \rangle$, $\langle \text{OR} \rangle$ и т. д. В исходный базовый набор операций входят операции реляционного дополнения $\langle \text{NOT} \rangle$, удаления атрибута $\langle \text{REMOVE} \rangle$, переименования атрибута $\langle \text{RENAME} \rangle$, реляционной конъюнкции $\langle \text{AND} \rangle$ и реляционной дизъюнкции $\langle \text{OR} \rangle$.

Операция реляционного дополнения

Пусть s обозначает результат операции $\langle \text{NOT} \rangle r$. Тогда:

- $H_s = H_r$ (заголовок результата совпадает с заголовком операнда);
- $B_s = \{t_s : \text{exists } t_r (t_r \notin B_r \text{ and } t_s = t_r) \}$ (в тело результата входят все кортежи, соответствующие заголовку и не входящие в тело операнда).

Операция $\langle \text{NOT} \rangle$ производит дополнение s заданного отношения r . Заголовком s является заголовок r . Тело s включает все кортежи, соответствующие этому заголовку и не входящие в тело r .

Почему дополнение?
Результат дополняет тело отношения до полного множества возможных кортежей, определенных на заголовке, определенном на доменах атрибутов.

$r =$

дни недели	чет-нечет
понедельник	1
вторник	0
четверг	0
суббота	1
вторник	1
четверг	1
пятница	1

$\langle \text{NOT} \rangle r =$

дни недели	чет-нечет
среда	1
пятница	0
воскресенье	1
понедельник	0
среда	0
суббота	0
воскресенье	0

Операция удаления атрибута

Пусть s обозначает результат операции $r \text{ <REMOVE> } A$. Для обеспечения возможности выполнения операции требуется, чтобы существовал некоторый тип (или домен) T такой, что $\langle A, T \rangle \in H_r$ (т. е. в состав заголовка отношения r должен входить атрибут A). Тогда:

- $H_s = H_r \text{ minus } \{\langle A, T \rangle\}$, т. е. заголовок результата получается из заголовка операнда изъятием атрибута A ;
- $B_s = \{ts : \text{exists } tr \text{ exists } v (tr \in B_r \text{ and } v \in T \text{ and } \langle A, T, v \rangle \in tr \text{ and } ts = tr \text{ minus } \{\langle A, T, v \rangle\})\}$, т. е. в тело результата входят все кортежи операнда, из которых удалено значение атрибута A .

Операция переименования атрибута

Пусть s обозначает результат операции $r \text{ <RENAME> } (A, B)$. Для обеспечения возможности выполнения операции требуется, чтобы существовал некоторый тип T , такой, что $\langle A, T \rangle \in H_r$, и чтобы не существовал такой тип T , что $\langle B, T \rangle \in H_r$. (Другими словами, в схеме отношения r должен присутствовать атрибут A и не должен присутствовать атрибут B .) Тогда:

- $H_s = (H_r \text{ minus } \{\langle A, T \rangle\}) \text{ union } \{\langle B, T \rangle\}$, т. е. в схеме результата B заменяет A ;
- $B_s = \{ts : \text{exists } tr \text{ exists } v (tr \in B_r \text{ and } v \in T \text{ and } \langle A, T, v \rangle \in tr \text{ and } ts = (tr \text{ minus } \{\langle A, T, v \rangle\}) \text{ union } \{\langle B, T, v \rangle\})\}$, т. е. в кортежах тела результата имя значений атрибута A меняется на B .

Операция реляционной конъюнкции

Пусть s обозначает результат операции $r_1 \text{ <AND> } r_2$. Для обеспечения возможности выполнения операции требуется, чтобы если $\langle A, T1 \rangle \in r_1$ и $\langle A, T2 \rangle \in r_2$, то $T1=T2$. (Другими словами, если в двух отношениях-операндах имеются одноименные атрибуты, то они должны быть определены на одном и том же типе (домене).) Тогда:

- $H_s = H_{r_1} \text{ union } H_{r_2}$, т. е. заголовок результата получается путем объединения заголовков отношений-операндов, как в операциях TIMES и JOIN
- $B_s = \{ ts : \text{exists } tr_1 \text{ exists } tr_2 ((tr_1 \in B_{r_1} \text{ and } tr_2 \in B_{r_2}) \text{ and } ts = tr_1 \text{ union } tr_2) \}$; обратите внимание на то, что кортеж результата определяется как *объединение кортежей операндов*; поэтому:
 - если схемы отношений-операндов имеют непустое пересечение, то операция <AND> работает как естественное соединение;
 - если пересечение схем операндов пусто, то <AND> работает как расширенное декартово произведение;
 - если схемы отношений полностью совпадают, то результатом операции является пересечение двух отношений-операндов.

Операция <AND> является реляционной конъюнкцией, в некоторых случаях выдающей в результате отношение r_s , ранее называвшееся естественным соединением двух заданных отношений r_1 и r_2 . Заголовок r_s является объединением заголовков r_1 и r_2 .

Операция реляционной дизъюнкции

Пусть s обозначает результат операции $r_1 \langle OR \rangle r_2$. Для обеспечения возможности выполнения операции требуется, чтобы если $\langle A, t_1 \rangle \in r_1$ и $\langle A, t_2 \rangle \in r_2$, то должно быть $t_1 = t_2$ (одноименные атрибуты должны быть определены на одном и том же типе). Тогда:

- $H_s = H_{r_1} \cup H_{r_2}$ (из схемы результата удаляются атрибуты-дубликаты);
- $B_s = \{ t_s : \text{exists } t_{r_1} \text{ exists } t_{r_2} ((t_{r_1} \in B_{r_1} \text{ or } t_{r_2} \in B_{r_2}) \text{ and } t_s = t_{r_1} \cup t_{r_2}) \}$; очевидно, что при этом:
 - если у операндов нет общих атрибутов, то в тело результирующего отношения входят все такие кортежи t_s , которые являются объединением кортежей t_{r_1} и t_{r_2} , соответствующих заголовкам отношений-операндов, и хотя бы один из этих кортежей принадлежит телу одного из операндов;
 - если у операндов имеются общие атрибуты, то в тело результирующего отношения входят все такие кортежи t_s , которые являются объединением кортежей t_{r_1} и t_{r_2} , соответствующих заголовкам отношений-операндов, если хотя бы один из этих кортежей принадлежит телу одного из операндов, и значения общих атрибутов t_{r_1} и t_{r_2} совпадают;
 - если же схемы отношений-операндов совпадают, то тело отношения-результата является объединением тел операндов.

Операция $\langle OR \rangle$ является реляционной дизъюнкцией и обобщением того, что ранее называлось объединением.

Алгебра А является полной, т. е. на основе введенных операций выражаются все операции алгебры Кодда.

Операция взятия разности. Если отношения r_1 и r_2 совместимы по объединению, то

$$r_1 \text{ MINUS } r_2 = r_1 \text{ <AND> <NOT> } r_2.$$

Операция ограничения (WHERE) выводится путем наложения ограничения на операнд, а потом применением операции <AND> к получившемуся отношению.

Чтобы получить результат **соединения общего вида** произвольных отношений А и В, нужно:

- выполнить над одним из отношений одну или несколько операций <RENAME>, чтобы избавиться от общих имен атрибутов;
- выполнить над полученными отношениями операцию <AND>, производящую расширенное декартово произведение;
- и для полученного отношения выполнить одну или несколько операций <AND> с отношениями-константами, чтобы должным образом ограничить его.

Операция реляционного деления. $r_1 \text{ DIVIDE BY } r_2$ совпадает с результатом выражения $(r_1 \text{ PROJECT } A) \text{ MINUS } (((r_2 \text{ TIMES } (r_1 \text{ PROJECT } A)) \text{ MINUS } r_1) \text{ PROJECT } A)$ в терминах операций реляционной алгебры Кодда или $(r_1 \text{ <REMOVE> } B) \text{ <AND> <NOT> } (((r_2 \text{ <AND> } (r_1 \text{ <REMOVE> } B)) \text{ <AND> <NOT> } r_1) \text{ <REMOVE> } B)$ в терминах операций Алгебры А.

ИЗБЫТОЧНОСТЬ АЛГЕБРЫ А (1/2)



Избыточность – это когда одни операции можно выразить через другие. То есть операций с избытком.

Булевские функции: отрицание, конъюнкция, дизъюнкция.

NOT	1	0
	0	1

AND	1	0
1	1	0
0	0	0

OR	1	0
1	1	1
0	1	0

Этот набор булевских функций избыточен. Существуют функции «стрелка Пирса» и «штрих Шеффера», через **каждую** из которых можно выразить эти три булевских.

$sh(A, A) \equiv NOT A;$
 $sh(NOT A, NOT B) \equiv A OR B$
 $NOT sh(A, B) \equiv A AND B.$

$pi(A, A) \equiv NOT A;$
 $pi(NOT A, NOT B) \equiv A AND B$
 $NOT pi(A, B) \equiv A OR B.$

Реляционные
аналоги

$\langle sh \rangle (r1, r2) \equiv \langle NOT \rangle r1 \langle OR \rangle \langle NOT \rangle r2$

$\langle pi \rangle (r1, r2) \equiv \langle NOT \rangle r1 \langle AND \rangle \langle NOT \rangle r2$

Можно свести набор операций Алгебры А к трем операциям: $\langle sh \rangle$ (или $\langle pi \rangle$), $\langle RENAME \rangle$ и $\langle REMOVE \rangle$.

ИЗБЫТОЧНОСТЬ АЛГЕБРЫ А (2/2)



Операция <RENAME> тоже избыточна.

Пусть мы хотим в отношении r заменить имя атрибута A на $A1$.

Возьмем отношение $r1$ такое, которое состоит из кортежей $\{A1, T, v\}$, где T и v соответствуют кортежам $\{A, T, v\}$ исходного отношения, отличаясь только названием атрибута.

Тогда операция r <RENAME> ($A, A1$) эквивалентна по результату выражению $(r$ <AND> $r1$) <REMOVE> A .



Базисом Алгебры A являются операции реляционного отрицания (дополнения), реляционной конъюнкции (или дизъюнкции) и проекции (удаления атрибута). Реляционные аналоги логических операций определяются в терминах отношений на основе обычных теоретико-множественных операций и позволяют выражать напрямую операции пересечения, декартова произведения, естественного соединения и объединения отношений. Путем комбинирования базовых операций выражаются операции переименования атрибутов, соединения общего вида, взятия разности отношений. Алгебра A позволяет лучше осознать логические основы реляционной модели, хотя, безусловно, является в меньшей степени ориентированной на практическое применение, чем алгебра Кодда.

РЕЛЯЦИОННОЕ ИСЧИСЛЕНИЕ



РЕЛЯЦИОННАЯ АЛГЕБРА

VS

РЕЛЯЦИОННОЕ ИСЧИСЛЕНИЕ

ЗАПРОС



ПОСЛЕДОВАЕЛЬНОСТЬ
ДЕЙСТВИЙ



Процедурная
формулировка

ПРОГРАММА



ПРОЧТЕНИЕ ФОРМУЛЫ



Декларативная
формулировка

Предикат – это символ, который представляет свойство или отношение. Пример: $P(a)$, P – предикат, применимый к индивидуальной константе a . Предикат первого порядка принимает в качестве переменной только одиночные переменные или константы.

Реляционное исчисление является прикладной ветвью формального механизма исчисления предикатов первого порядка. В основе исчисления лежит понятие переменной с определенной для нее областью допустимых значений и понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы.

В качестве синтаксиса будем использовать нечто похожее на QUEL, используемый в СУБД Ingres.

ИСЧИСЛЕНИЕ КОРТЕЖЕЙ (1/3)



RANGE СЛУЖАЩИЙ IS СЛУЖАЩИЕ

- Определение кортежной переменной. СЛУЖАЩИЙ.СЛУ_ИМЯ – значение атрибута.

Правильно построенная формула (Well-Formed Formula, WFF) служит для выражения условий, накладываемых на кортежные переменные.

Простые условия

Основой WFF являются простые условия, представляющие собой операции сравнения скалярных значений (значений атрибутов переменных или литерально заданных констант). Например, конструкции

СЛУЖАЩИЙ.СЛУ_НОМ = 2934 И

СЛУЖАЩИЙ.СЛУ_НОМ = ПРОЕКТ.ПРОЕКТ_РУК

По определению, простое сравнение является WFF, а WFF, заключенная в круглые скобки, представляет собой простое сравнение.

Обозначим:

comp – простое сравнение
form - WFF

WFF:

NOT **form**
comp AND **form**
comp OR **form**
IF **comp** THEN **form**

WFF выдает либо *true* либо *false*. Результатом WFF будет отношение, для которого *true*.

Кванторы EXISTS и FORALL

EXISTS $var (form)$ (WFF)
FORALL $var (form)$ var – какое-то отношение.

EXISTS равносильно «СУЩЕСТВУЕТ кортеж, для которого $form$ принимает значение *true*».
FORALL равносильно «ЛЮБОЙ кортеж, для которого $form$ принимает значение *true*».

Свободные и связанные переменные

По определению, все переменные, входящие в WFF, при построении которой не использовались кванторы, являются **свободными**. Фактически, это означает, что если для какого-то набора значений свободных кортежных переменных при вычислении WFF получено значение *true*, то эти значения кортежных переменных могут входить в результирующее отношение.

Если же имя переменной использовано сразу после квантора при построении WFF вида EXISTS $var (form)$ или FORALL $var (form)$, то в этой WFF и во всех WFF, построенных с ее участием, var является **связанной переменной**. Это означает, что такая переменная не видна за пределами минимальной WFF, связавшей эту переменную. При вычислении значения такой WFF используется не одно значение связанной переменной, а вся область ее определения.

Каждая новая связанная переменная – плюс один цикл при выполнении запроса.

Целевые списки

Целевой список строится из целевых элементов, каждый из которых может иметь следующий вид:

- `var.attr`, где `var` – имя свободной переменной соответствующей WFF, а `attr` – имя атрибута отношения, на котором определена переменная `var`;
- `var`, что эквивалентно наличию подписка `var.attr1, var.attr2, ..., var.attrn`, где `{attr1, attr2, ..., attrn}` включает имена всех атрибутов определяющего отношения;
- `new_name = var.attr`; `new_name` – новое имя соответствующего атрибута результирующего отношения.

Последний вариант нужен для решения конфликтов, когда требуется переименование.

Выражение реляционного исчисления кортежей имеет вид:

`target_list WHERE WFF`

ПРИМЕР СЛУЖАЩИЕ DIVIDE BY НОМЕРА_ПРОЕКТОВ

```
СЛУ1, СЛУ2 RANGE IS СЛУЖАЩИЕ
НОМЕР_ПРОЕКТА RANGE IS НОМЕРА_ПРОЕКТОВ
СЛУ1.СЛУ_НОМЕР, СЛУ1.СЛУ_ИМЯ, СЛУ1.СЛУ_ЗАРП
WHERE FORALL НОМЕР_ПРОЕКТА EXISTS СЛУ2
  (СЛУ1.СЛУ_НОМЕР = СЛУ2.СЛУ_НОМЕР AND
   СЛУ1.ПРО_НОМ = НОМЕРА_ПРОЕКТОВ.ПРО_НОМ)
```



СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00

В исчислении доменов областью определения переменных являются домены.
Главное понятие исчисления доменов - условия членства.

Условия членства

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

$R(a_{i_1} : v_{i_1}, a_{i_2} : v_{i_2}, \dots, a_{i_m} : v_{i_m}) (m \leq n)$,

где v_{ij} – это либо литерально задаваемая константа, либо имя доменной переменной.

Условие членства принимает значение true в том и только в том случае, если в отношении R существует кортеж, содержащий указанные значения указанных атрибутов.

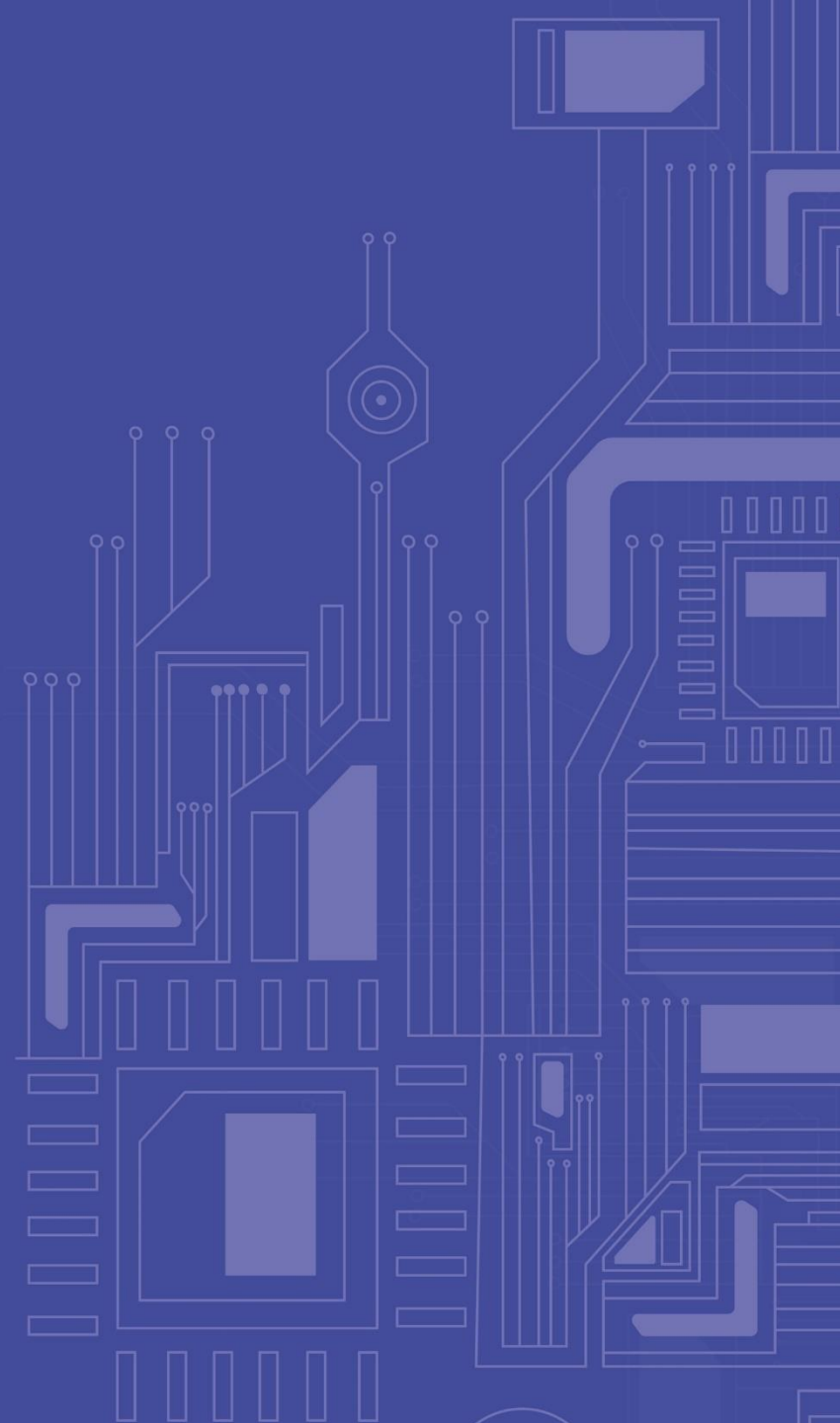
Если v_{ij} – константа, то на атрибут a_{ij} накладывается жесткое условие, не зависящее от текущих значений доменных переменных; если же v_{ij} – имя доменной переменной, то условие членства может принимать разные значения при разных значениях этой переменной.

СЛУЖАЩИЕ (СЛУ_НОМ:2934, СЛУ_ИМЯ:'Иванов',
СЛУ_ЗАРП:22400.00, ПРО_НОМ:ПРО_НОМ)



<2934, 'Иванов', 22400.00, 1>
<2934, 'Иванов', 22400.00, 2>

Во всех остальных отношениях формулы и выражения исчисления доменов выглядят похожими на формулы и выражения исчисления кортежей. В частности, формулы могут включать кванторы, и различаются свободные и связанные вхождения доменных переменных.



СЕМИНАР



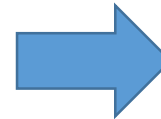
ПРИМЕР ОПЕРАЦИИ УДАЛЕНИЯ



Результат операции СЛУЖАЩИЕ <REMOVE> ПРО_НОМ

СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2



СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00
2936	Сидоров	18000.00
2937	Федоров	20000.00
2938	Иванова	22000.00
2939	Сидоренко	18000.00
2940	Федоренко	20000.00
2941	Иваненко	22000.00

ПРИМЕР ОПЕРАЦИИ КОНЪЮНКЦИИ (<AND>)



СЛУЖАЩИЕ_В_ПРОЕКТЕ_1

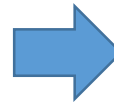
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310
2936	Сидоров	18000.00	313
2937	Федоров	20000.00	310
2938	Иванова	22000.00	315

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

ПРОЕКТЫ

ПРО_НОМ	ПРОЕКТ_РУК
1	Иванов
2	Иваненко



Результат операции СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 <AND> ПРОЕКТЫ

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

Результат операции СЛУЖАЩИЕ_В_ПРОЕКТЕ_1 <AND>
СЛУЖАЩИЕ_В_ПРОЕКТЕ_2

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	СЛУ_ОТД_НОМЕР
2934	Иванов	22000.00	310
2935	Петров	30000.00	310

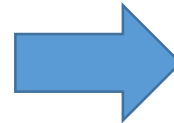
ПРИМЕР ОПЕРАЦИИ ДИЗЪЮНКЦИИ (<OR>) 1



Операция <OR> является реляционной дизъюнкцией и обобщением того, что ранее называлось объединением. Заголовок s есть объединение заголовков r_1 и r_2 . Тело s состоит из всех кортежей, соответствующих заголовку s и являющихся надмножеством *либо* некоторого кортежа из тела r_1 , *либо* некоторого кортежа из тела r_2 .

Предположим, у нас имеются отношения ПРОЕКТЫ_1 {ПРОЕКТ_НАЗВ, ПРОЕКТ_РУК} и НОМЕРА_ПРОЕКТОВ {ПРО_НОМ}. Предположим также, что домен атрибута ПРОЕКТ_НАЗВ включает значения ПРОЕКТ_1, ПРОЕКТ_2, ПРОЕКТ_3, домен атрибута ПРОЕКТ_РУК ограничен значениями Иванов, Иваненко, а доменом атрибута ПРО_НОМ является множество {1, 2, 3}.

ПРОЕКТЫ_1		НОМЕРА_ПРОЕКТОВ
ПРОЕКТ_НАЗВ	ПРОЕКТ_РУК	ПРО_НОМ
ПРОЕКТ 1	Иванов	1
ПРОЕКТ 2	Иваненко	2



Результат операции ПРОЕКТЫ <OR> НОМЕРА_ПРОЕКТОВ

ПРОЕКТ_НАЗВ	ПРОЕКТ_РУК	ПРО_НОМ
ПРОЕКТ 1	Иванов	1
ПРОЕКТ 2	Иванов	1
ПРОЕКТ 3	Иванов	1
ПРОЕКТ 1	Иваненко	1
ПРОЕКТ 2	Иваненко	1
ПРОЕКТ 3	Иваненко	1
ПРОЕКТ 1	Иванов	2
ПРОЕКТ 2	Иванов	2
ПРОЕКТ 3	Иванов	2
ПРОЕКТ 1	Иваненко	2
ПРОЕКТ 2	Иваненко	2
ПРОЕКТ 3	Иваненко	2
ПРОЕКТ 1	Иванов	3
ПРОЕКТ 2	Иваненко	3

ПРИМЕР ОПЕРАЦИИ ДИЗЪЮНКЦИИ (<OR>) 2



Алгебра А

R1

сезон	чет
осень	1
зима	0

R2

сезон	ок
осень	+
зима	-

R1 <OR> R2

сезон	чет	ок
осень	1	+
осень	1	-
зима	0	+
зима	0	-
осень	0	+
зима	1	-

Если у операндов имеются общие атрибуты, то в тело результирующего отношения входят все такие кортежи t которые являются объединением кортежей t соответствующих заголовкам отношений-операндов, если хотя бы один из этих кортежей принадлежит телу одного из операндов, и значения общих атрибутов tr_1 и tr_2 совпадают;

Это пример на дизъюнкцию, когда у операндов есть общая часть

осень	0 -	не может быть, потому что нет кортежа <осень, 0> или <осень, ->
зима	1 +	не может быть, потому что нет кортежа <зима,1> или <зима, +>

ИНТЕРПРЕТАЦИЯ ОГРАНИЧЕНИЯ WHERE (1/2)



Результат операции СЛУЖАЩИЕ_1 WHERE СЛУ_ЗАРП = 20000.00

СЛУЖАЩИЕ_1

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	РУК_НОМ
2934	Иванов	22000.00	2934
2935	Петров	30000.00	2934
2936	Сидоров	18000.00	2934
2937	Федоров	20000.00	2934
2938	Иванова	22000.00	2941
2939	Сидоренко	18000.00	2941
2940	Федоренко	20000.00	2941
2941	Иваненко	22000.00	2941

1 Создаем отношение

ЗАРП_20000

СЛУ_ЗАРП
20000.00



2

СЛУЖАЩИЕ_1 <AND> ЗАРП_20000

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	РУК_НОМ
2937	Федоров	20000.00	2934
2940	Федоренко	20000.00	2941

ИНТЕРПРЕТАЦИЯ ОГРАНИЧЕНИЯ WHERE (2/2)



Результат операции СЛУЖАЩИЕ_1 WHERE СЛУ_НОМЕР = РУК_НОМ

СЛУЖАЩИЕ_1

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	РУК_НОМ
2934	Иванов	22000.00	2934
2935	Петров	30000.00	2934
2936	Сидоров	18000.00	2934
2937	Федоров	20000.00	2934
2938	Иванова	22000.00	2941
2939	Сидоренко	18000.00	2941
2940	Федоренко	20000.00	2941
2941	Иваненко	22000.00	2941



(((СЛУЖАЩИЕ_1 <REMOVE> СЛУ_НОМЕР) <REMOVE> СЛУ_ИМЯ)
<REMOVE> СЛУ_ЗАРП) <RENAME> (РУК_НОМ, СЛУ_НОМЕР)

СЛУ_НОМЕР
2934
2941

СЛУЖАЩИЕ_1 <AND> (((СЛУЖАЩИЕ_1 <REMOVE> СЛУ_НОМЕР)
<REMOVE> СЛУ_ИМЯ) <REMOVE> СЛУ_ЗАРП) <RENAME>
(РУК_НОМ, СЛУ_НОМЕР))

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	РУК_НОМ
2934	Иванов	22000.00	2934
2941	Иваненко	22000.00	2941

ИНТЕРПРЕТАЦИЯ ДЕЛЕНИЯ DIVIDE BY (1/3)



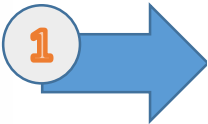
Результат операции СЛУЖАЩИЕ **DIVIDE BY** НОМЕРА_ПРОЕКТОВ

СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

НОМЕРА_ПРОЕКТОВ

ПРО_НОМ
1
2



СЛУЖАЩИЕ **PROJECT** (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00
2936	Сидоров	18000.00
2937	Федоров	20000.00
2938	Иванова	22000.00
2939	Сидоренко	18000.00
2940	Федоренко	20000.00
2941	Иваненко	22000.00

ИНТЕРПРЕТАЦИЯ ДЕЛЕНИЯ DIVIDE BY (2/3)



СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00
2936	Сидоров	18000.00
2937	Федоров	20000.00
2938	Иванова	22000.00
2939	Сидоренко	18000.00
2940	Федоренко	20000.00
2941	Иваненко	22000.00



(СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП))

TIMES НОМБРА ПРОЕКТОВ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	1
2935	Петров	29600.00	2
2936	Сидоров	18000.00	1
2936	Сидоров	18000.00	2
2937	Федоров	20000.00	1
2937	Федоров	20000.00	2
2938	Иванова	22000.00	1
2938	Иванова	22000.00	2
2939	Сидоренко	18000.00	1
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	1
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	1
2941	Иваненко	22000.00	2

ИНТЕРПРЕТАЦИЯ ДЕЛЕНИЯ DIVIDE BY (3/3)



((СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП))
TIMES НОМБРА_ПРОЕКТОВ)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	1
2935	Петров	29600.00	2
2936	Сидоров	18000.00	1
2936	Сидоров	18000.00	2
2937	Федоров	20000.00	1
2937	Федоров	20000.00	2
2938	Иванова	22000.00	1
2938	Иванова	22000.00	2
2939	Сидоренко	18000.00	1
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	1
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	1
2941	Иваненко	22000.00	2



((СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП))
TIMES НОМБРА_ПРОЕКТОВ) MINUS СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2936	Сидоров	18000.00	2
2937	Федоров	20000.00	2
2938	Иванова	22000.00	2
2939	Сидоренко	18000.00	1
2940	Федоренко	20000.00	1
2941	Иваненко	22000.00	1



((СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП))
MINUS (((СЛУЖАЩИЕ PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП)) TIMES
НОМБРА_ПРОЕКТОВ) MINUS СЛУЖАЩИЕ)
PROJECT (СЛУ_НОМЕР, СЛУ_ИМЯ, СЛУ_ЗАРП))

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП
2934	Иванов	22400.00
2935	Петров	29600.00

ПРИМЕР WFF



СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

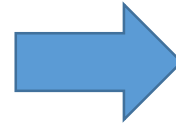
ПРОЕКТЫ

ПРО_НОМ	ПРОЕКТ_РУК
1	Иванов
2	Иваненко

НОМЕРА ПРОЕКТОВ

ПРО_НОМ
1
2

IF СЛУЖАЩИЙ.СЛУ_ИМЯ = 'Иванов'
THEN (СЛУЖАЩИЙ.СЛУ_ЗАРП >= 22400.00 AND СЛУЖАЩИЙ.ПРО_НОМ = 1)



СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

Формула неверна только для записи про Иванова на 2м проекте, остальные записи подходят.

ПРИМЕР ДЛЯ КВАНТОРОВ EXISTS И FORALL



Пусть СЛУ1 и СЛУ2 представляют собой две кортежные переменные, определенные на отношении СЛУЖАЩИЕ.

(а) Область истинности WFF EXISTS СЛУ2 (СЛУ1.СЛУ_ЗАРП > СЛУ2.СЛУ_ЗАРП)			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

(б) Область истинности WFF FORALL СЛУ2 (СЛУ1.СЛУ_ЗАРП > СЛУ2.СЛУ_ЗАРП)			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2935	Петров	29600.00	1
2935	Петров	29600.00	2