

Управление данными

**Лекция 4. Подъязык данных,
основанный на реляционной
алгебре. Реляционное исчисление**

Соответствие операторов

Реляционная алгебра	Проектируемый язык
$R \cup S$	R UNION S
$R \cap S$	R INTERSECT S
$R - S$	R MINUS S
$R * S$	R TIMES S
Селекция $R = \sigma F (R_1)$	SELECT R WHERE F
$R \bowtie S$	JOIN R AND S
$R \div S$	DIVIDE R BY S
Проекция $R = \pi_{a_1, \dots, a_n}(R_1)$	PROJECT R OVER A_1, A_2, \dots, A_n

Операции соединения и деления нуждаются в пояснениях.

Поскольку операция соединения может выполняться по одному из 6 условий $=$, $<$, $>$, \leq , \geq , \neq , то это условие будем указывать сразу после указания операндов. Например, соединение при условии "больше" отношения R по атрибуту X с отношением S по атрибуту Y будет найдено по команде:

JOIN R AND S WHERE $X > Y$

и представляет множество кортежей t таких, что t есть конкатенация кортежа r , принадлежащего R и кортежа s , принадлежащего S , где $x > y$ (x есть X -компонент r , а y есть Y -компонент s).

Естественное соединение находится по команде:

JOIN R AND S OVER X.

OVER X является сокращением записи WHERE $X = X$, т.е. считается, что атрибут X содержится и в R и в S.

Операция DIVIDE определяется как операция между бинарным отношением (делимое) и унарным отношением (делитель), в результате которой получается унарное отношение (частное).

Пусть делимое R имеет атрибуты X и Y и пусть делитель S имеет атрибут Z .

Пусть также Y и Z определены на одном и том же домене.

Тогда операция

DIVIDE R BY S OVER Y AND Z

дает частное, определенное на том же домене, что и X ; значение x появляется в частном тогда и только тогда, когда пара $\langle x, y \rangle$ появляется в R для всех значений Z содержащихся в S .

Пример.

Рассмотрим базу данных, состоящую из 3-х отношений:

S (ПОСТАВЩИКИ)

S#	ПОСТИМЯ	СТАТУС	ГОРОД
S1	Семёнов	20	Москва
S2	Дубов	10	Одесса
S3	Бугаев	30	Одесса
S4	Казаков	20	Москва
S5	Адамов	30	Алушта

P (ДЕТАЛИ)

P#	ДЕТИМЯ	ЦВЕТ	ВЕС	ГОРОД
P1	Гайка	Красный	12	Москва
P2	Болт	Зелёный	17	Одесса
P3	Винт	Синий	17	Севастополь
P4	Винт	Красный	14	Москва
P5	Кулачок	Синий	12	Одесса
P6	Шестерня	Красный	19	Москва

SP(ПОСТАВКИ)

S#	P#	Колич
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Реляционная схема

DOMAIN S# CHARACTER (5)

DOMAIN ПОСТИМЯ CHARACTER (20)

DOMAIN СТАТУС NUMERIC (3)

DOMAIN ГОРОД CHARACTER (15)

DOMAIN P# CHARACTER (6)

DOMAIN ДЕТИМЯ CHARACTER (20)

DOMAIN ЦВЕТ CHARACTER (6)

DOMAIN ВЕС NUMERIC (4)

DOMAIN КОЛИЧ NUMERIC (5)

RELATION S (S#, ПОСТИМЯ, СТАТУС, ГОРОД) KEY (S#)

RELATION P(P#, ДЕТИМЯ, ЦВЕТ, ВЕС, ГОРОД) KEY (P#)

RELATION SP(S#, P#, КОЛИЧ) KEY (S#, P#)

Запрос

Выдать номера поставщиков, поставляющих деталь P2:

```
SELECT SP WHERE P# = "P2" GIVING TEMP  
PROJECT TEMP OVER S# GIVING RESULT
```

Две операции записаны как два предложения для наглядности.

Можно иначе

```
PROJECT (SELECT SP WHERE P# = "P2") OVER S# GIVING  
RESULT
```

Следовательно, выражения реляционной алгебры могут быть вложенными. Эта вложенность может достигать любой глубины.

Запрос

Выдать номера поставщиков, поставляющих красные детали:

```
SELECT P WHERE COLOR = "RED" GIVING TEMP1  
JOIN TEMP1 AND SP OVER P# GIVING TEMP2  
PROJECT TEMP2 OVER S# GIVING RESULT
```

С вложенностью:

```
PROJECT (JOIN (SELECT P WHERE COLOR = "RED" AND  
SP OVER P#) OVER S# GIVING RESULT
```

ПРИМЕРЫ операций запоминания.

Для того, чтобы включить новый кортеж в отношение, можно воспользоваться операцией UNION.

Например добавить деталь Р7 (имя 'ШАЙБА', цвет 'СЕРЫЙ', вес 2, город 'АФИНЫ'), в отношение Р можно поступить так:

$R \cup \{ 'P7', 'ШАЙБА', 'СЕРЫЙ', '2', 'АФИНЫ' \}$ GIVING R.

Аналогично для удаления кортежа из отношения мы можем использовать операцию MINUS.

Например, удалить поставщика S1 можно следующим путем:

$S \text{ MINUS } \{ 'S1', 'СЕМЕНОВ', 20, 'МОСКВА' \}$
GIVING S

Для удаления всех кортежей отношения SP для поставщика S1 можно записать:

SP MINUS {'S1',?,?} GIVING SP

Здесь знак ? обозначает любое значение, поэтому любой кортеж, содержащий 'S1' в позиции первого атрибута должен быть удален из отношения SP.

Операция обновления **UPDATE** может быть реализована как последовательное выполнение MINUS и UNION, а можно и включить в язык операцию UPDATE.

Реляционное исчисление

Базовыми понятиями реляционного исчисления являются:

1. Понятие переменной с определенной для нее ОДЗ.
2. Понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы.

В зависимости от определения переменной различаются:

- исчисление кортежей (областями определения переменных является кортеж некоторого отношения)
- исчисление доменов (областью определения являются домены, на которых определены атрибуты отношений БД, то есть допустимым значением переменной является значение домена)

Для начала поясним, в чём отличие реляционной алгебры от реляционного исчисления.

Пусть есть база данных, которая состоит из отношений

СЛУЖАЩИЕ {СЛУ_НОМ, СЛУ_ИМЯ, СЛУ_ЗАРП, ПРО_НОМ} и

ПРОЕКТЫ {ПРО_НОМ, ПРОЕКТ_РУК, ПРО_ЗАРП}

(в отношении ПРОЕКТЫ атрибут ПРОЕКТ_РУК содержит имена служащих, являющихся руководителями проектов, а атрибут ПРО_ЗАРП – среднее значение зарплаты, получаемой участниками проекта).

Допустим, мы хотим узнать имена и номера служащих, которые являются руководителями проектов со средней заработной платой, превышающей 18000 руб.

Формулировкой такого запроса с использованием реляционной алгебры, может быть, следующее алгебраическое выражение:

(СЛУЖАЩИЕ JOIN ПРОЕКТЫ WHERE (СЛУ_ИМЯ = ПРОЕКТ_РУК AND ПРО_ЗАРП > 18000.00)) ПРОЕКТ (СЛУ_ИМЯ, СЛУ_НОМ)

Это выражение можно было бы прочесть, например, следующим образом:

- выполнить эквисоединение отношений СЛУЖАЩИЕ и ПРОЕКТЫ по условию СЛУ_ИМЯ = ПРОЕКТ_РУК;
- ограничить полученное отношение по условию ПРО_ЗАРП > 18000.00;
- спроецировать результат предыдущей операции на атрибут СЛУ_ИМЯ, СЛУ_НОМ.

Формулировка этого же запроса с использованием реляционного исчисления, может дать два определения переменных:

RANGE СЛУЖАЩИЙ IS СЛУЖАЩИЕ и

RANGE ПРОЕКТ IS ПРОЕКТЫ

и выражение

СЛУЖАЩИЙ.СЛУ_ИМЯ, СЛУЖАЩИЙ.СЛУ_НОМ WHERE EXISTS
(СЛУЖАЩИЙ.СЛУ_ИМЯ = ПРОЕКТ.ПРОЕКТ_РУК AND
ПРОЕКТ.ПРО_ЗАРП > 18000.00).

Это выражение можно было бы прочесть, например, следующим образом:

-выдать значения СЛУ_ИМЯ и СЛУ_НОМ для каждого кортежа служащих такого, что существует кортеж проектов со значением ПРОЕКТ_РУК, совпадающим со значением СЛУ_НОМ этого кортежа служащих, и значением ПРО_ЗАРП, большим 18000.00.

Во второй формулировке мы указали лишь характеристики результирующего отношения, но ничего не сказали о способе его формирования.

В этом случае система сама должна решить, какие операции и в каком порядке нужно выполнить над отношениями СЛУЖАЩИЕ и ПРОЕКТЫ.

Таким образом, можно сказать, что алгебраическая формулировка является процедурной, т. е. задающей последовательность действий для выполнения запроса, а логическая (исчисление) – описательной (или декларативной), поскольку она всего лишь описывает свойства желаемого результата.

С точки зрения результата эти два механизма эквивалентны, и существуют не слишком сложные правила преобразования одного формализма в другой.

Исчисление кортежей

Для определения кортежной переменной используется оператор RANGE. Например, для того чтобы определить переменную СЛУЖАЩИЙ, областью определения которой является отношение СЛУЖАЩИЕ, нужно употребить конструкцию

RANGE СЛУЖАЩИЙ IS СЛУЖАЩИЕ

Из этого определения следует, что в любой момент времени переменная СЛУЖАЩИЙ представляет некоторый кортеж отношения СЛУЖАЩИЕ.

При использовании кортежных переменных в формулах можно ссылаться на значение атрибута переменной (это аналогично тому, как, например, при программировании на языке С можно сослаться на значение поля структурной переменной).

Например, для того, чтобы сослаться на значение атрибута СЛУ_ИМЯ переменной СЛУЖАЩИЙ, нужно употребить конструкцию СЛУЖАЩИЙ.СЛУ_ИМЯ.

Правильно построенные формулы

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

являются простыми условиями.

Первое условие принимает значение true в том и только в том случае, когда значение атрибута СЛУ_НОМ кортежной переменной СЛУЖАЩИЙ равно 2934.

Второе условие принимает значение true в том и только в том случае, когда значения трибутов СЛУ_НОМ и ПРОЕКТ_РУК переменных СЛУЖАЩИЙ и ПРОЕКТ совпадают.

Более сложные варианты WFF строятся с помощью логических связок NOT, AND, OR и IF ... THEN) с учетом обычных приоритетов операций (NOT > AND > OR) и возможности расстановки скобок.

Так, если form – WFF, а comp – простое сравнение, то NOT form, comp AND form, comp OR form и IF comp THEN form являются WFF.

Для отношений Служащие и Проекты

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

Правильно построенной является следующая формула:

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

Эта формула будет принимать значение **true** для следующих значений кортежной переменной СЛУЖАЩИЙ:

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

Кванторы

При построении WFF допускается использование кванторов существования (EXISTS) и всеобщности (FORALL).

Если $form$ – это WFF, в которой участвует переменная var , то конструкции $EXISTS\ var\ (form)$ и $FORALL\ var\ (form)$ представляют собой WFF.

По определению, формула $EXISTS\ var\ (form)$ принимает значение true в том и только в том случае, если в области определения переменной var найдется хотя бы одно значение (кортеж), для которого WFF $form$ принимает значение true.

Формула $FORALL\ var\ (form)$ принимает значение true, если для всех значений переменной var из ее области определения WFF $form$ принимает значение true.

Свободные и связанные переменные

Переменные, входящие в WFF, могут быть свободными или связанными.

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.

Целевые списки и выражения реляционного исчисления

Целевой список (target list). 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 $var.attr_1, var.attr_2, \dots, var.attr_n$, где $\{attr_1, attr_2, \dots, attr_n\}$ включает имена всех атрибутов определяющего отношения;
- $new_name = var.attr$; new_name — новое имя соответствующего атрибута результирующего отношения.

Выражением реляционного исчисления кортежей называется конструкция вида **target_list WHERE WFF**.

Значением выражения является отношение, тело которого определяется WFF, а множество атрибутов и их имена — целевым списком.

Исчисление доменов

В исчислении доменов областью определения переменных являются не отношения, а домены.

Применительно к базе данных СЛУЖАЩИЕ-ПРОЕКТЫ можно говорить, например, о доменных переменных ИМЯ (значения – допустимые имена) или НОМ_СЛУ (значения – допустимые номера служащих).

Условия членства

Основным формальным отличием исчисления доменов от исчисления кортежей является наличие дополнительного множества предикатов, позволяющих выражать так называемые *условия членства*.

Если R – это n -арное отношение с атрибутами a_1, a_2, \dots, a_n , то условие членства имеет вид $R(a_{i1} : v_{i1}, a_{i2} : v_{i2}, \dots, a_{im} : v_{im})$ ($m \leq n$), 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.

Пример. Для простоты будем считать, что мы определили доменные переменные, имена которых совпадают с именами атрибутов отношения СЛУЖАЩИЕ, а в случае, когда требуется несколько доменных переменных, определенных на одном домене, мы будем добавлять в конце имени цифры. WFF исчисления доменов

СЛУЖАЩИЕ (СЛУ_НОМ:2934, СЛУ_ИМЯ:'Иванов', СЛУ_ЗАРП:22400.00, ПРО_НОМ:1) примет значение true в том и только в том случае, когда в теле отношения СЛУЖАЩИЕ содержится кортеж <2934, 'Иванов', 22400.00, 1>.

Соответствующие значения доменных переменных образуют область истинности этой WFF.

С другой стороны, WFF

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

будет принимать значение true для всех комбинаций явно заданных значений и допустимых значений переменной ПРО_НОМ, которые соответствуют кортежам, входящим в тело отношения СЛУЖАЩИЕ.

Выражения исчисления доменов

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.

Для примера выражения исчисления доменов сформулируем с использованием исчисления доменов запрос «Выдать номера и имена служащих, не получающих минимальную заработную плату»:

СЛУ_НОМ, СЛУ_ИМЯ WHERE EXISTS СЛУ_ЗАРП1

(СЛУЖАЩИЕ (СЛУ_ЗАРП1) AND

СЛУЖАЩИЕ (СЛУ_НОМ, СЛУ_ИМЯ, СЛУ_ЗАРП) AND

СЛУ_ЗАРП > СЛУ_ЗАРП1)

Реляционное исчисление доменов является основой большинства языков запросов, основанных на использовании форм.

В частности, на этом исчислении базировался известный язык Query-by-Example (QBE), который был первым (и наиболее интересным) языком в семействе языков, основанных на табличных формах.

СПАСИБО ЗА ВНИМАНИЕ!