

Лекция 8

**Особенности теоретико-множественных
операций реляционной алгебры
(дополнительные операторы,
продолжение)**

Реляционное исчисление

Операция обобщения SUMMARIZE

Пусть A — исходное отношение с заголовком (A_1, A_2, \dots, A_n) , а отношение B — проекция отношения A на атрибуты A_1, A_2, \dots, A_k .

Результатом операции обобщения SUMMARIZE отношения A является отношение R , заголовок которого $(A_1, A_2, \dots, A_k, Z)$, получается объединением заголовка (A_1, A_2, \dots, A_k) отношения B и добавляемого атрибута Z . Тело отношения R состоит из кортежей t вида $(a_1, a_2, \dots, a_k, z)$, где $(a_1, a_2, \dots, a_k) \in B$. Значение z нового атрибута Z , есть результат вычисления значения выражения $\langle expr \rangle$ по всем кортежам отношения A , которые имеют те же значения для атрибутов A_1, A_2, \dots, A_k , что и кортеж t .

Синтаксис

SUMMARIZE A PER B ADD $\langle expr \rangle$ AS Z

Здесь A — исходное отношение, B — проекция отношения A , $\langle expr \rangle$ — допустимое скалярное выражение, Z — имя добавляемого атрибута.

Атрибута Z не должно быть в списке атрибутов отношения A и в выражении $\langle expr \rangle$.

Пример 5.1. SUMMARIZE SP PER SP [P#] ADD SUM (QTY) AS TOTQTY

SP			P#	TOTQTY
S#	P#	QTY		
S ₁	P ₁	300	P ₁	600
S ₁	P ₂	200	P ₂	1000
S ₁	P ₃	400	P ₃	400
S ₁	P ₄	200	P ₄	500
S ₁	P ₅	100	P ₅	500
S ₁	P ₆	100	P ₆	100
S ₂	P ₁	300		
S ₂	P ₂	400		
S ₃	P ₂	200		
S ₄	P ₂	200		
S ₄	P ₄	300		
S ₄	P ₅	400		

Таблица 5.1.
Пример выполнения
операции SUMMARIZE

SUMMARIZE < рел. выражение > PER < рел. выражение > ADD < список обобщений >

Список обобщений :

< тип обобщения > [(< скалярное выражение >)] AS < имя атрибута >

Пример 5.2.SUMMARIZE (P JOIN SP) PER [CITY] ADD COUNT AS NSP

P					SP		
P#	PNAME	COLOR	WEIGHT	CITY	S#	P#	QTY
P ₁	Nut	Red	12.0	London	S ₁	P ₁	300
P ₂	Bolt	Green	17.0	Paris	S ₁	P ₂	200
P ₃	Screw	Blue	17.0	Rome	S ₁	P ₃	400
P ₄	Screw	Red	14.0	London	S ₁	P ₄	200
P ₅	Cam	Blue	12.0	Paris	S ₁	P ₅	100
P ₆	Cog	Red	19.0	London	S ₁	P ₆	100
					S ₂	P ₁	300
					S ₂	P ₂	400
					S ₃	P ₂	200
					S ₄	P ₂	200
					S ₄	P ₄	300
					S ₄	P ₅	400

CITY	NSP
London	5
Paris	6
Rome	1

Таблица 5.2. (а)

Таблица 5.2. (б)

EXTEND S ADD COUNT ((SP RENAME S# AS X) WHERE X = S#) AS NP (лекция 4)

SUMMARIZE SP PER [P#]

ADD SUM (QTY) AS TOTQTY, AVG (QTY) AS AVGQTY

SP		
S#	P#	QTY
S ₁	P ₁	300
S ₁	P ₂	200
S ₁	P ₃	400
S ₁	P ₄	200
S ₁	P ₅	100
S ₁	P ₆	100
S ₂	P ₁	300
S ₂	P ₂	400
S ₃	P ₂	200
S ₄	P ₂	200
S ₄	P ₄	300

P#	TOTQTY	AVGQTY
P ₁	600	300
P ₂	1000	250
P ₃	400	400
P ₄	500	250
P ₅	100	100
P ₆	100	100

Таблица 5.3.. Пример выполнения операции SUMMARIZE

SUMMARIZE SP PER SP [] ADD SUM (QTY) AS GRANDTOTAL

Группирование производится для отношения не имеющего атрибутов [].

отношение SP		
S#	P#	QTY
S ₁	P ₁	300
S ₁	P ₂	200
S ₁	P ₃	400
S ₁	P ₄	200
S ₁	P ₅	100
S ₁	P ₆	100
S ₂	P ₁	300
S ₂	P ₂	400
S ₃	P ₂	200
S ₄	P ₂	200
S ₄	P ₄	300
S ₄	P ₅	400

GRANDTOTAL
3100

Все кортежи отношения SP по пустому множеству атрибутов ("0-кортежу") будут иметь "одинаковые значения".

Все кортежи отношения образуют единственную группу, для которой будет построен один кортеж.

Обобщающая функция вычисляется только один раз для всего отношения SP в целом

Таблица 5.4.

Транзитивное замыкание TCLOSE

Пусть A — произвольное отношение с атрибутами X и Y , принадлежащими одному типу T .

Транзитивным замыканием отношения A ($TCLOSE A$) называется отношение A^+ , заголовок которого совпадает с заголовком отношения A .

Тело отношения A^+ состоит из кортежей $\{ X:\mathbf{x}, Y:\mathbf{y} \}$ таких, что кортеж принадлежит отношению A^+ тогда и только тогда, когда он принадлежит отношению A или существует последовательность значений z_1, z_2, \dots, z_n (типа T), такая, что все кортежи $\{ X:\mathbf{x}, Y:z_1 \}, \{ X:z_1, Y:z_2 \}, \dots, \{ X:z_n, Y:\mathbf{y} \}$ принадлежат отношению A .

Тело отношения A всегда содержится в теле отношения A^+ как подмножество.

Операция группирования

SP GROUP (P#, QTY) AS PQ

SP		
S#	P#	QTY
S ₁	P ₁	300
S ₁	P ₂	200
S ₁	P ₃	400
S ₁	P ₄	200
S ₁	P ₅	100
S ₁	P ₆	100
S ₂	P ₁	300
S ₂	P ₂	400
S ₃	P ₂	200
S ₄	P ₂	200
S ₄	P ₄	300
S ₄	P ₅	400

SPQ		
S#	PQ	
S ₁	P#	QTY
	P ₁	300
	P ₂	200
	P ₃	400
	P ₄	200
	P ₅	100
S ₂	P#	QTY
	P ₁	300
S ₃	P#	QTY
	P ₂	200
S ₄	P#	QTY
	P ₂	200
	P ₄	300
	P ₅	400

"сгруппировать отношение SP по атрибуту S# [атрибут S# является единственным атрибутом отношения SP, не упомянутым в GROUP].

Каждый кортеж отношения SP заменяется кортежем x , в котором компоненты P# и QTY "упакованы" в один компонент y .

$$x = (s, y); y = (p, q);$$

Таблица 5.5.
Группирование отношения SP по атрибуту S#

Операция разгруппирования.

Выражение $SPQ \text{ UNGROUP } PQ$ возвращает отношение SP .

Результатом операции $UNGROUP$ является отношение SP , заголовок которого $\{S\#, P, QTY\}$ состоит из атрибутов $P\#$ и QTY (полученных из атрибута PQ), и из всех остальных атрибутов отношения SPQ (атрибут $S\#$).

Тело отношения SP состоит из кортежей t , каждый кортеж t содержит соответствующее значение s атрибута $S\#$, и значения p и q атрибутов $P\#$ и QTY ($t = (s, p, q)$).

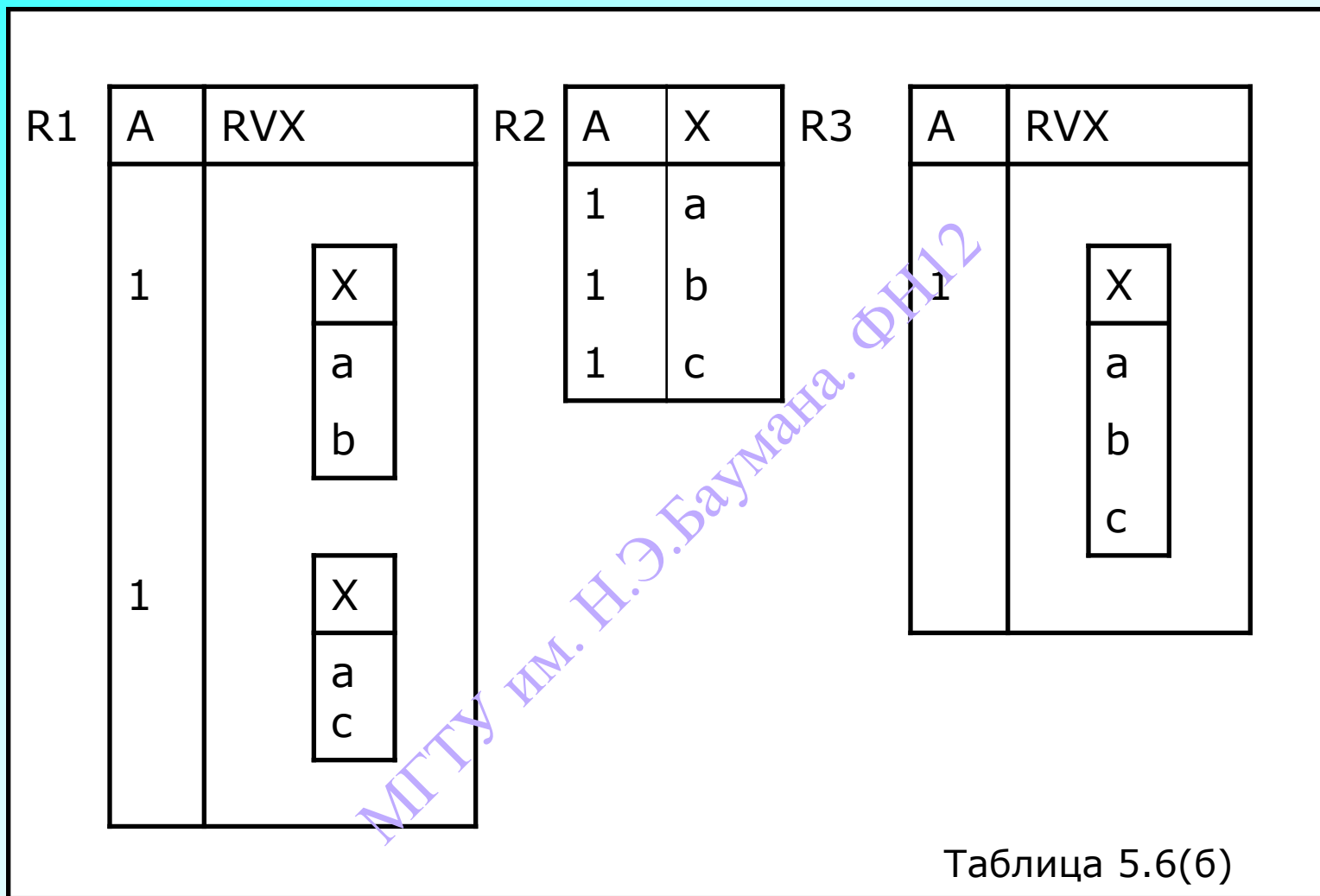
Каждый кортеж SPQ $x = (s, y)$, где $y = (pq)$ заменяется множеством кортежей.

Компоненты y кортежа x , значение $S\#$ которых равно s , "разворачиваются" в отдельные компоненты p и q (значения атрибутов $P\#$ и QTY).

SPQ		
S#	PQ	
S ₁	P#	QTY
	P ₁	300
	P ₂	200
	P ₃	400
	P ₄	200
	P ₅	100
	P ₆	100
S ₂	P#	QTY
	P ₁	300
	P ₂	400
S ₃	P#	QTY
	P ₂	200
S ₄	P#	QTY
	P ₂	200
	P ₄	300
	P ₅	400

Таблица 5.5 (а).

Обратимость операций группирования и разгруппирования



Операции разгруппирования и группирования необязательно являются обратимыми, решающим моментом здесь является ФЗ

Операция разгруппирования отношения R (относительно атрибута-отношения RA) обратима тогда и только тогда, когда :

1. Ни один кортеж отношения R не принимает в качестве значения атрибута RA пустое отношение.
2. Атрибут RA функционально зависит от комбинации всех остальных атрибутов отношения R , т.е. должен быть потенциальный ключ, не содержащий атрибут RA .

Группирующие функции в SQL.

```
SELECT Groupe, count(*) AS NGr FROM  
student GROUP BY Groupe ORDER BY NGr;
```

Groupe	NGr
Cotr11	21
Inf11	23
Inf12	20

```
SELECT count(*) AS NGr FROM student;
```

NGr
64

```
SELECT MAX(age) FROM employee;
```

```
SELECT MIN (absence) AS NAbs FROM student;
```

```
SELECT SUM(salary) FROM employee;
```

```
SELECT AVG(age) FROM employee;
```

```
SELECT COUNT(*) FROM employee;
```

```
SELECT COUNT(*) FROM employee WHERE Job = 'programmer';
```

Реляционное исчисление

Исчисление кортежей

Исчисление доменов

МГТУ им. Н.Э.Баумана. ФН12

Реляционное исчисление

Реляционное исчисление это специального применения исчисления предикатов в реляционных БД,

Реляционное исчисление основано на **исчислением предикатов** (логике 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 логике первого порядка кванторы можно навешивать только на переменные).

В зависимости от того, что является областью определения переменной, различают *исчисление **кортежей*** и *исчисление **доменов***.

Реляционное исчисление

В исчислении кортежей областями определения переменных являются тела отношений БД, т. е. допустимым значением каждой переменной является кортеж тела некоторого отношения.

В исчислении доменов областями определения переменных являются домены, на которых определены атрибуты отношений БД, т. е. допустимым значением каждой переменной является значение некоторого домена

Понятие реляционного исчисления было предложено Коддом.

*Кодд представил язык, основанный непосредственно на реляционном исчислении и названный "подъязык данных ALPHA". Сам язык ALPHA никогда не был реализован, однако этот язык оказал заметное влияние на построение языка **QUEL**, который был реализован и некоторое время конкурировал с языком SQL.*

Справка. (praedicatum — сказанное)

Предикатом $P(x_1, x_2, \dots, x_n)$ называется функция $P: M^n \rightarrow \{0, 1\}$, где M произвольное множество. (Обобщение $P: (M_1 \times M_2 \times \dots \times M_n) \rightarrow \{0, 1\}$.)

Каждому предикату $P(x_1, x_2, \dots, x_n)$ можно поставить в соответствие n -местное отношение ρ и наоборот:

$$(a_1, a_2, \dots, a_n) \in \rho \Leftrightarrow P_\rho(a_1, a_2, \dots, a_n) = 1.$$

(n -местным предикатом можно называть предложение, содержащее переменные, которое превращается в высказывание при подстановке вместо этих переменных любых конкретных значений.)

Предикат — один из элементов логики первого и высших порядков, начиная с логики второго порядка, в формулах можно ставить кванторы по предикатам.

Предикат $P(x_1, x_2, \dots, x_n)$ называют **тождественно-истинным**, если $P(x_1, x_2, \dots, x_n) = 1$ на любом наборе значений переменных.

Предикат $P(x_1, x_2, \dots, x_n)$ называют **тождественно-ложным**, если $P(x_1, x_2, \dots, x_n) = 0$ на любом наборе значений переменных.

Предикат называют **выполнимым (опровержимым)**, если хотя бы на одном наборе аргументов он принимает значение 1 (0).

К предикатам применимы все операции булевой алгебры.

Реляционное исчисление

Исчисление кортежей

Кортежная переменная

RANGEVAR < *имя кортежной переменной* >

RANGES OVER < *список реляционных выражений* > ;

Реляционное исчисление

Примеры .

RANGEVAR SX RANGES OVER S ;

RANGEVAR SY RANGES OVER S ;

RANGEVAR SPX RANGES OVER SP ;

RANGEVAR SPY RANGES OVER SP ;

RANGEVAR PX RANGES OVER P ;

RANGEVAR SU RANGES OVER

(SX WHERE SX.CITY = 'London') ,

(SX WHERE EXISTS SPX

(SPX.S# = SX.S# AND SPX.P# = P₁)) ;

Правильно построенные формулы

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

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

STUDENT . StudentID = 1234

S.S# = SP.S#

Понятие *правильно построенной формулы* введем индуктивно.

- 1. Базис индукции. Правильно построенной формулой (ППФ)** считают любую константу и любое скалярное значение атрибута.
- 2. Индуктивный переход.**
 - Если Φ_1, Φ_2 – **ППФ**, а comp – операция сравнение, то выражение $(\Phi_1 \text{ comp } \Phi_2)$ – есть **ППФ**, которая является **простым сравнением**.
 - Если $\text{Comp_}\Phi$ простое сравнение, то $\text{NOT Comp_}\Phi$, $\text{Comp_}\Phi \text{ AND } \Phi_1$, $\text{Comp_}\Phi \text{ OR } \Phi_1$ и $\text{IF Comp_}\Phi \text{ THEN } \Phi_1$ являются **ППФ**.
 - Если Φ_1 – это **ППФ**, в которой участвует переменная V то $\text{EXISTS } V (\Phi_1)$ и $\text{FORALL } V(\Phi_1)$ являются **ППФ**, где EXISTS – квантор существования(\exists), FORALL квантор всеобщности(\forall).
- 1. Замыкание.** Никаких других **ППФ**, кроме определенных выше, не существует.

Пример 5.6.

Правильно построенной является следующая формула:

IF STUDENT.Name = 'Гришин Г. Г.' THEN

(STUDENT.StudentID >= 1235 AND STUDENT.Groupe='У-11')

StudentID	Name	Groupe
1232	Алексеев А. А.	И-12
1233	Борисов Б. Б.	И-12
1234	Васькин В.В.	И-12
1235	Гришин Г. Г	У-11

Таблица 5.7.

Кванторы. Свободные и связанные переменные кортежей

EXISTS - квантор существования и *FORALL* — квантор всеобщности.

EXISTS $V (p (V))$ равносильна $\text{false OR } p (t_1) \text{ OR } \dots \text{ OR } p (t_m)$
Если $\{ t_i \} = \emptyset$, то результатом будет значение *ложь*.

FORALL $V (p (V))$ равносильна $\text{true AND } p (t_1) \text{ AND } \dots \text{ AND } p (t_m)$
Если $\{ t_i \} = \emptyset$, то результатом будет значение *истина*

Все переменные, входящие в *ППФ*, при построении которой не использовались кванторы, являются **свободными**.

Если имя переменной использовано сразу после квантора при построении *ППФ* вида *EXISTS* $V(\Phi)$ или *FORALL* $V(\Phi)$, то в этой и во всех *ППФ*, построенных с ее участием, V является **связанной переменной**.

- Примеры**
- Простые сравнения
 - $SX.S\# = S_1$
 - $SX.S\# = SPX.S\#$
 - $SPX.P\# \neq PX.P\#$
 - Логические выражения из простых сравнений
 - $PX.WEIGHT < 15.5 \text{ OR } PX.CITY = 'Rome'$
 - $\text{NOT} (SX.CITY = 'London')$
 - Формулы с кванторами
 - $\text{EXISTS } SPX (SPX.S\# = SX.S\# \text{ AND } SPX.P\# = P_2)$
 - $\text{FORALL } PX (PX.COLOR = Red)$

Квантор существования EXISTS.

EXISTS SPX (SPX.S# = SX.S# AND SPX.P# = P₂)

В текущем значении SP существует кортеж SPX такой, что значение атрибута S# равно значению атрибута SX. S#, а значение атрибута P# – 'P₂'.

Пример 5.7.

Отношение V(A, B, C) [A, B, C : INTEGER.] содержащее следующие кортежи: (1, 2, 3) (1, 2, 4) (1, 3, 4)

EXISTS V (V.C > 1) : true

EXISTS V (V.B > 3) : false

EXISTS V (V.A > 1 OR V.C = 4) : true

Квантор всеобщности FORALL.

FORALL PX (PX.COLOR = Red) *В текущем значении P для всех кортежей PX значение атрибута COLOR равно 'Red'.*

Пример 5.8.

FORALL V (V.A > 1) : false

FORALL V (V.B > 1) : true

FORALL V (V.A = 1 AND V.C > 2) : true

FORALL V (p) ≡ NOT EXISTS V (NOT p)

Формула, в которой все переменные связаны, называется **закрытой формулой** (фактически она является *высказыванием*). **Открытая формула** — это формула, которая не является закрытой, т.е. такая формула, которая содержит по крайней мере одну ссылку на свободную переменную.

Результирующие списки и выражения реляционного исчисления

ППФ обеспечивают средства формулировки условия выборки из отношений БД. Чтобы можно было использовать исчисление для реальной работы с БД, требуется еще один компонент, который определяет набор и имена атрибутов результирующего отношения. Этот компонент называется **результирующим (целевым) списком (*target list*)**.

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

1. $var.attr$, где var – имя *свободной переменной* соответствующей ППФ, а $attr$ – имя атрибута отношения, на котором определена переменная var ;
2. var , эквивалентно наличию подписка $var.attr1, var.attr2, \dots, var.attrn$, где $\{attr1, attr2, \dots, attrn\}$ включает имена всех атрибутов определяющего отношения;
3. $new_name = var.attr$; new_name – новое имя соответствующего атрибута результирующего отношения

Применение результирующего *списка* к области истинности ППФ эквивалентно выполнению алгебраической операции проекции.

Выражением реляционного исчисления кортежей называется конструкция вида `target_list WHERE ППФ`. Значением *выражения* является отношение, тело которого определяется *ППФ*, а множество атрибутов и их имена – результирующим *списком*.

Все ссылки на переменные кортежей в параметре `target_list` должны быть **свободными** в пределах значения этого параметра.

Ссылка на кортежную переменную в предложении `WHERE` *может быть свободной* только в случае, если ссылка на эту же *переменную* (*обязательно свободная*) присутствует в соответствующем значении параметра `target_list`.

"Получить номера поставщиков, находящихся в Лондоне».

```
SX.S# WHERE SX.CITY = 'London'
```

«Получить имена поставщиков детали с номером P_2 »

```
SX.SNAME WHERE EXISTS SPX ( SPX.S# = SX.S# AND SPX.P# = P2 )
```

Все ссылки на переменную SX являются свободными, все ссылки на переменную SPX (в предложении WHERE) являются связанными, на них нет ссылок в target_list.

Выражение реляционного исчисления кортежей, результат которого совпадает с результатом операции ***Student JOIN Curator _Group***

RANGEVAR Studen1 RANGES OVER Student

RANGEVAR Curator1 RANGES OVER Curator _Group

Student1. StudentID, Student1. Name, Student1. Groupe,

Curator1. Curator1

WHERE (Student1. Groupe = Curator1. Groupe))

<i>Student</i>		
StudentID	Name	Group
1232	Алексеев А. А.	И-12
1233	Борисов Б. Б.	И-12
1234	Васькин В.В.	И-12
1235	Гришин Г. Г.	У-11
1237	Иванов И.В.	И-13
1240	Степанов А. С.	К-14

<i>Curator_Group</i>	
Group	Curator
И-12	Семенова И.А.
У-11	Иванова Е.А.
И-13	Демин А.Д
К-14	Лосев А.Д.

<i>Student JOIN Curator_Group</i>			
StudentID	Name	Group	Curator
1232	Алексеев А. А.	И-12	Семенова И.А.
1233	Борисов Б. Б.	И-12	Семенова И.А.
1234	Васькин В.В.	И-12	Семенова И.А.
1235	Гришин Г. Г.	У-11	Иванова Е.А.
1237	Иванов И.В.	И-13	Демин А.Д
1240	Степанов А. С.	К-14	Лосев А.Д

Таблица 5.8. (*Student JOIN Curator_Group*)

Примеры

Использование реляционного исчисления кортежей для формулирования запросов

1. Определить номера поставщиков из Парижа со статусом, большим 20

(SX.S#, SX.STATUS) WHERE SX.CITY = 'Paris' AND SX.STATUS > 20

2. Найти все такие пары номеров поставщиков, в которых два поставщика находятся в одном городе

(SX.S# AS SA, SY.S# AS SB) WHERE SX.CITY = SY.CITY AND
SX.S# < SY.S#

Спецификации AS в результирующем списке используются для именованя атрибутов результата. Следовательно, эти имена недоступны для использования в фразе WHERE, и потому второе сравнение в фразе WHERE записано как SX.S# < SY.S#, а не в виде SA < SB.

3. Определить все атрибуты поставщиков детали с номером 'P₂'

`SX WHERE EXISTS SPX (SPX.S#=SX.S# AND SPX.P#=P2)`

Имя кортежной переменной в результирующем списке записано без атрибутов. Этот пример является сокращенной записью следующего выражения.

`(SX.S#, SX.NAME, SX.STATUS, SX.CITY)
WHERE EXISTS SPX (SPX.S#=SX.S# AND SPX.P#=P2)`

4. Определить имена поставщиков по крайней мере одной красной детали

`SX.SNAME`

`WHERE EXISTS SPX (SX.S#=SPX.S# AND
EXISTS PX (PX.P#=SPX.P# AND PX.COLOR=Red))`

5. Найти имена поставщиков по крайней мере одной детали, поставляемой поставщиком с номером 'S₂'

SX.SNAME

WHERE EXISTS SPX (EXISTS SPY (SX.S#=SPX.S# AND
SPX.P#=SPY.P# AND
SPY.S#=S₂))

6. Выбрать имена поставщиков всех типов деталей

SX.SNAME WHERE FORALL PX

(EXISTS SPX (SPX.S# = SX.S# AND SPX.P# = PX.P#))

SX.SNAME WHERE NOT EXISTS PX

(NOT EXISTS SPX (SPX.S# = SX.S# AND SPX.P# = PX.P#))

7. Определить имена поставщиков, которые не поставляют деталь с номером 'P₂'.

SX.SNAME WHERE NOT EXISTS SPX

(SPX.S# = SX.S# AND SPX.P# = P₂)

8. Получить номера деталей, которые либо весят более 16 фунтов, либо поставляются поставщиком с номером 'S₂', либо и то, и другое

PU.P# WHERE PX.WEIGHT > 16.0 OR EXISTS SPX(SPX.P#=PX.P# AND SPX.S#=S₂)