

# Управление данными

## Лекция 2 РЕЛЯЦИОННАЯ АЛГЕБРА

# Формальные языки манипулирования данными для реляционной модели

Техника формулирования запросов является наиболее важным аспектом языка манипулирования данными.

Остальные функции реляционного языка манипулирования данными достаточно просты и связаны с включением, удалением и обновлением кортежей.

Языки запросов для реляционной модели разбиваются на два класса:

- 1) Алгебраические языки, позволяющие выражать запросы средствами специализированных операторов, применяемых к отношениям.
- 2) Языки исчисления предикатов, где запросы описывают требуемое множество кортежей путем спецификации предиката, которому должны удовлетворять эти кортежи.

Языки, основанные на исчислении предикатов, в свою очередь, делятся на два класса:

- 1) Реляционное исчисление с переменными кортежами
- 2) Реляционное исчисление с переменными на доменах

В зависимости от того, являются информационными объектами кортежи или элементы домена некоторого атрибута.

Таким образом, существует три различных вида языков запросов к реляционным базам данных.

Каждый из трех указанных абстрактных языков запросов эквивалентен по своей выразительности двум другим.

**Реляционная алгебра** — это коллекция операций, которые принимают отношения в качестве операндов и возвращают отношение в качестве результата.

Первая версия этой алгебры была определена Э. Коддом в 1970 годах.

Базировалась алгебра на теоретико-множественных отношениях.

Эта "оригинальная" алгебра включала восемь операций, которые подразделялись на описанные ниже две группы с четырьмя операциями каждая.

1. Традиционные операции с множествами — объединение, пересечение, разность и декартово произведение (их операндами являются именно отношения, а не произвольные множества).

2. Специальные реляционные операции, такие как сокращение (известное также под названием выборки), проекция, соединение и деление.

**Реляционная алгебра** - это язык операций, выполняемых над отношениями - таблицами реляционной базы данных.

Операции реляционной алгебры позволяют на основе одного или нескольких отношений создавать другое отношение без изменения самих исходных отношений.

Полученное другое отношение обычно не записывается в базу данных, а существует в результате выполнения SQL-запроса - массиве, создаваемом функциями для работы с базами данных в языках программирования.

Для каждой операции реляционной алгебры будет дана её реализация в виде запросов на языке SQL( *structured query language* — «язык структурированных запросов»).

Основы реляционной алгебры - математическая теория множеств и операции над множествами.

**Приоритеты выполнения операций** реляционной алгебры (в порядке убывания пунктов списка, а в одном пункте - операции с равными приоритетами):

1. селекция, проекция
2. декартово произведение, соединение, пересечение, деление
3. объединение, разность.

Для простоты изложения материала, будем обозначать отношения буквами  $R$ ,  $R_1$ , и т.д.

Атрибуты:  $A_1$ ,  $A_2$ , и т. д.

## Операция выборки

Операция выборки работает с одним отношением и определяет результирующее отношение  $R$ , которое содержит только те кортежи (или строки, или записи), отношения, которые удовлетворяют заданному условию (предикату  $P$ ).

Таким образом, операция выборки - унарная операция - и записывается следующим образом:

$$R = \sigma_P(R_1)$$

где  $P$  - предикат (логическое условие).

# Запрос SQL

SELECT \*

from R3

WHERE A3 > 'd0'

Теперь посмотрим, что получится в результате выполнения этой операции реляционной алгебры и соответствующего ей запроса SQL.

В таблице ниже дано одно отношение, с которым работает эта операция.



## Пример

R3			
A1	A2	A3	A4
2	RH	Y1	MS
4	HF	A3	MR
7	AA	Y1	MS

Просматриваем столбец A3 и устанавливаем, что предикату  $A3 > 'd0'$  удовлетворяют записи в первой и третьей строках исходного отношения (так как номер буквы у в алфавите больше номера буквы d).

В результате получаем следующее новое отношение, в котором две строки:

R			
A1	A2	A3	A4
2	RH	Y1	MS
7	AA	Y1	MS

# Операция проекции

Операция проекции  $R = \pi_{a_1, \dots, a_n}(R_1)$  работает, как и операция выборки, только с одним отношением и определяет новое отношение  $R$ , в котором есть лишь те атрибуты (столбцы), которые заданы в операции, и их значения.

## Запрос SQL

```
SELECT DISTINCT A4, A3  
from R3
```

Пусть вновь дано то же отношение R3:

R3			
A1	A2	A3	A4
2	RH	Y1	MS
4	HF	A3	MR
7	AA	Y1	MS

Из исходного отношения выбираем только столбцы A4 и A3 и видим, что строки со значениями - первая и третья - идентичны.

Исключаем дубликат (за это отвечает ключевое слово DISTINCT в SQL-запросе, которое говорит, что нужно выбрать только уникальные записи) и получаем следующее новое отношение, в котором два атрибута и две строки (записи):

R	
A4	A3
MS	Y1
MR	A3

## Операция объединения

Результатом объединения двух множеств (отношений)  $A$  и  $B$  ( $A \cup B$ ) будет такое множество (отношение)  $C$ , которое включает в себя те и только те элементы, которые есть или во множестве  $A$  или во множестве  $B$ .

Говоря упрощённо, все элементы множества  $A$  и множества  $B$ , за исключением дубликатов, образующихся за счёт того, что некоторые элементы есть и в первом, и во втором множестве.

## Запрос SQL

```
SELECT A1, A2, A3  
from R1  
UNION  
SELECT A1, A2, A3  
from R2
```

Теперь посмотрим, что получится в результате выполнения этой операции реляционной алгебры и соответствующего ей запроса SQL.

Даны два отношения, так как операция объединения - бинарная операция:

R1				R2		
A1	A2	A3		A1	A2	A3
Z7	aa	w11		X8	pp	k21
B7	hh	h15		Q2	ee	h15
X8	pp	w11		X8	pp	w11

Объединяем строки первого и второго отношения и видим, что третья строка, которая является третьей и в первом, и во втором отношении - идентичны, поэтому её включаем в новое отношение только один раз.



<b>R</b>		
<b>A1</b>	<b>A2</b>	<b>A3</b>
Z7	aa	w11
B7	hh	h15
X8	pp	w11
X8	pp	k21
Q2	ee	h15

**Важно следующее:** операция объединения может быть выполнена только тогда, когда два отношения обладают одинаковым числом и названиями атрибутов (столбцов), или, говоря формально, совместимы по объединению.

## Операция пересечения

Результатом пересечения двух множеств (отношений)  $A$  и  $B$  ( $A \cap B$ ) будет такое множество (отношение)  $C$ , которое включает в себя те и только те элементы, которые есть и во множестве  $A$ , и во множестве  $B$ .

### Запрос SQL

```
SELECT A1, A2, A3  
from R1  
INTERSECT  
SELECT A1, A2, A3  
from R2
```

В некоторых диалектах SQL отсутствует ключевое слово  
INTERSECT.

Поэтому, например, в MySQL и других, операция  
пересечения множеств может реализована с применением  
предиката EXISTS.

Даны два отношения. Выполним запрос

R1				R2		
A1	A2	A3		A1	A2	A3
Z7	aa	w11		X8	pp	k21
B7	hh	h15		Q2	ee	h15
X8	pp	w11		X8	pp	w11

Просматриваем все записи в двух отношениях, и обнаруживаем, что и в первом, и во втором отношении есть одна строка - та, которая является третьей и в первом, и во втором отношении. Получаем новое отношение:

R		
A1	A2	A3
X8	pp	w11

## Операция разности

Разность двух отношений  $R_1$  и  $R_2$  ( $R_1 - R_2$ ) состоит из кортежей (или записей, или строк), которые имеются в отношении  $R_1$ , но отсутствуют в отношении  $R_2$ .

Отношения  $R_1$  и  $R_2$  должны быть совместимы по объединению.

### Запрос SQL

```
SELECT A1, A2, A3  
from R2  
EXCEPT  
SELECT A1, A2, A3  
from R1
```

Установим, что получится в результате выполнения этой операции реляционной алгебры и соответствующего ей запроса SQL. Вновь даны два отношения R1 и R2:

R1				R2		
A1	A2	A3		A1	A2	A3
Z7	aa	w11		X8	pp	k21
B7	hh	h15		Q2	ee	h15
X8	pp	w11		X8	pp	w11

Из отношения R2 исключаем строку, которая есть также в отношении R1 - третью - и получаем новое отношение:

R		
A1	A2	A3
X8	pp	k21
Q2	ee	h15

В некоторых диалектах SQL отсутствует ключевое слово EXCEPT. Поэтому, например, в MySQL и других, операция пересечения множеств может реализована с применением предиката NOT EXISTS.

## Операция декартова произведения

Операция декартова произведения ( $R_1 \times R_2$ ) определяет новое отношение R, которое является результатом конкатенации каждого кортежа отношения R1 с каждым кортежем отношения R2.

### **Запрос SQL**

SELECT \*

from R3, R4

Установим, что получится в результате выполнения этой операции реляционной алгебры и соответствующего ей запроса SQL. Даны два отношения R3 и R4:

R3					R4	
A1	A2	A3	A4		A5	A6
3	hh	yl	ms		3	hh
4	pp	al	sr		4	pp
1	rr	yl	ms			



В новом отношении должны присутствовать все атрибуты (столбцы) двух отношений. Сначала первая строка отношения R3 сцепляется с каждой из двух строк отношения R4, затем вторая строка отношения R3, затем третья.

В результате должно получиться  $3 \times 2 = 6$  кортежей (строк).

Получаем такое новое отношение:

R					
A1	A2	A3	A4	A5	A6
3	hh	yl	ms	3	hh
3	hh	yl	ms	4	pp
4	pp	a1	sr	3	hh
4	pp	a1	sr	4	pp
1	rr	yl	ms	3	hh
1	rr	yl	ms	4	pp

## Операция деления

Результатом операции деления ( $R = R_1 \div R_2$ ) является набор кортежей (строк) отношения R1, которые соответствуют комбинации всех кортежей отношения R2.

Для этого нужно, чтобы в отношении R2 была часть атрибутов (можно и один), которые есть в отношении R1.

В результирующем отношении присутствуют только те атрибуты отношения R1, которых нет в отношении R2.

### **Запрос SQL**

```
SELECT DISTINCT A1, A4
from R5
WHERE
NOT EXIST (SELECT *   from R6
            WHERE NOT EXIST
            R6.A2 = R5.A2 AND
            R6.A3 = R5.A3)
```

Получим результат выполнения этой операции  
реляционной алгебры и соответствующего ей запроса  
SQL.

Даны два отношения R5 и R6:

R5					R6	
A1	A2	A3	A4		A2	A3
2	S3	4	sun		R4	8
3	X8	7	kab		X8	7
3	R4	8	kab			

Комбинации всех кортежей отношения R6  
соответствуют вторая и третья строки отношения R5.

Но после исключения атрибутов (столбцов) A2 и A3  
эти строки становятся идентичными.

Поэтому в новом отношении присутствует эта строка  
один раз. Новое отношение:

R	
A1	A4
3	kab

## Операция тета-соединения

В результате этой операции получается отношение, которое содержит кортежи из декартова произведения отношений R1 и R2 удовлетворяющие предикату P.

Значением предиката P может быть один из операторов сравнения ( $<$ ,  $<=$ ,  $>$ ,  $>=$ ,  $=$  или  $\neq$ ).

### **Запрос SQL**

```
SELECT *
```

```
from R3, R4
```

```
WHERE A1 $\geq$ A5
```

Посмотрим, что получится в результате выполнения этой операции реляционной алгебры и соответствующего ей запроса SQL.

Даны два отношения R3 и R4:

R3					R4	
A1	A2	A3	A4		A5	A6
3	hh	yl	ms		3	hh
4	pp	a1	sr		4	pp
1	rr	yl	ms			

Выполняем операцию декартового произведения.

R					
A1	A2	A3	A4	A5	A6
3	hh	yl	ms	3	hh
3	hh	yl	ms	4	pp
4	pp	a1	sr	3	hh
4	pp	a1	sr	4	pp
1	rr	yl	ms	3	hh
1	rr	yl	ms	4	pp

Видим, что условию предиката R удовлетворяют два кортежа декартового произведения - первый (так как  $3 \geq 3$ ) и четвёртый. Получаем следующее новое отношение:

R					
A1	A2	A3	A4	A5	A6
3	hh	yl	ms	3	hh
4	pp	a1	sr	4	pp

**СПАСИБО ЗА ВНИМАНИЕ!**