

Теоретические языки запросов в реляционных базах данных

- 1. Реляционная алгебра**
- 2. Реляционное исчисление**

Введение

В реляционных СУБД для выполнения операций над отношениями используется две группы языков, имеющие в качестве своей математической основы теоретические языки запросов, предложенные Э. Коддом:

- *реляционная алгебра;*
- *реляционное исчисление.*

Языки реляционной алгебры являются *процедурными*. Операторы этой алгебры состоят из *операндов*, в роли которых выступают отношения, и *реляционных операций*. Результатом реляционной операции является *отношение*.

Языки исчислений, в отличие от реляционной алгебры, являются *непроцедурными* (описательными или декларативными) и позволяют выражать запросы с помощью *предикатов первого порядка*. Запрос к **БД**, выполненный с использованием подобного языка, содержит лишь информацию о желаемом результате, а не как его получить.

Реляционные таблицы для примеров

В дальнейшем при рассмотрении языков реляционной алгебры и реляционного исчисления для демонстрационных примеров будем использовать БД, включающую в себя следующие три таблицы:

1. Таблица **S** (поставщики деталей).
2. Таблица **P** (характеристики деталей).
3. Таблица **SP** (поставки).

Таблица S

П#	Имя поставщика	Статус	Город_П
S1	Дышекова А.	20	Москва
S2	Кудинов А.	10	Киев
S3	Оленева М.	30	Киев
S4	Романова Е.	20	Москва
S5	Шахбазян Т.	30	Минск

Таблица Р

Д#	Название	Тип	Вес	Город_Д
Р1	Гайка	Каленый	12	Москва
Р2	Болт	Мягкий	17	Киев
Р3	Винт	Твердый	17	Ростов
Р4	Винт	Каленый	14	Москва
Р5	Палец	Твердый	12	Киев
Р6	Шпилька	Каленый	19	Москва

П#	Д#	Количество
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

Слева таблица SP

Первичными ключами этих таблиц являются соответственно:

П# - код поставщика,

Д# - код детали,

(П#, Д#) – составной ключ.

Каждый из атрибутов П# и Д# таблицы SP в отдельности является внешним ключом к таблице S и P соответственно.

1. Реляционная алгебра

Вариант реляционной алгебры, предложенный Э. Коддом, можно разделить на две группы: 1) базовые теоретико-множественные операции – *объединение, пересечение, разность и декартово произведение*; 2) специальные операции - *проекция, селекция, деление и соединение*.

Перечисленный набор из 8 операций является избыточным. Минимально необходимый набор составляет 5 операций – объединение, вычитание, декартово произведение, проекция и выборка. Три оставшиеся операции можно определить через указанные выше 5 операций, например, *соединение – это проекция выборки декартова произведения.*

С другой стороны указанных **8** операций недостаточно для построения реальной СУБД на принципах реляционной алгебры. Требуются расширения, включающие операции *переименования атрибутов, образования новых вычисляемых атрибутов, вычисления итоговых функций* и др. Эти дополнительные операции были введены **К. ДЖ. Дейтом**.

Операции реляционной алгебры могут выполняться над одним отношением – **унарные операции**, - например, проекция; над двумя отношениями – **бинарные операции**, например, объединение. При выполнении бинарных операций участвующие в операциях отношения должны быть **совместимы по структуре**, что означает совместимость имен атрибутов и типов соответствующих доменов. Частным случаем совместимости является идентичность (совпадение).

1.1. Объединение

Объединением двух совместимых отношений **R1** и **R2** с одинаковыми заголовками называется отношение **R**, содержащее все элементы исходных отношений (с исключением повторений кортежей).
Обозначение: **$R = R1 \text{ UNION } R2$**

Пример 1

Пусть отношением **R1** будет множество поставщиков из г. Москвы, а **R2** – множество поставщиков, которые поставляют деталь **P2**. Тогда отношение **R** представляет собой поставщиков, находящихся в г. Москве, выпускающих деталь **P2**, либо тех и других:

<i>П#</i>	<i>Имя поставщика</i>	<i>Статус</i>	<i>Город_П</i>
S1	Дышекова А.	20	Москва
S2	Кудинов А.	10	Киев
S3	Оленева М.	30	Киев
S4	Романова Е.	20	Москва

1.2. Вычитание

Разность двух совместимых отношений $R1$ и $R2$ с *одинаковыми заголовками* есть отношение R , состоящее из множества кортежей отношения $R1$, не принадлежащих отношению $R2$.

Обозначение: **$R1 \text{ MINUS } R2$** .

В предыдущем примере

$$R1 \text{ MINUS } R2 = \emptyset.$$

1.3. Пересечение

Пересечением двух совместимых отношений **R1** и **R2** с *одинаковыми заголовками* называется отношение **R**, включающее в себя кортежи, принадлежащие одновременно отношениям **R1** и **R2**.

Обозначение: **$R = R1 \text{ INTERSECT } R2$** .

В примере 1 **$R1 \text{ INTERSECT } R2 = R1$** .

1.4. Декартово произведение

Декартовым произведением отношения $R1$ с длиной кортежей $k1$ и отношения $R2$ с длиной кортежей $k2$, которые не имеют одинаковых имен атрибутов, есть отношение R с длиной кортежей $k1+k2$, заголовок которого представляет собой сцепление заголовков отношений $R1$ и $R2$, и состоящее из таких кортежей, что первые $k1$ элементов принадлежат отношению $R1$, а последние $k2$ кортежей принадлежат отношению $R2$.

Обозначение: **$R=R1 \text{ TIMES } R2$** .

Замечание 1

В реляционной алгебре используется расширенный вариант декартова произведения двух отношений, при котором элементы кортежей двух исходных отношений сливаются, что означает удаление лишних скобок, то есть вместо упорядоченной пары $((a, b), (c, d))$ пишут (a, b, c, d) . Здесь $(a, b) \in R1$, $(c, d) \in R2$.

Пример 2

Пусть отношение $R1$ представляет собой множество номеров всех текущих поставщиков $\{S1, S2, S3, S4, S5\}$, а отношение $R2$ — множество номеров всех деталей $\{P1, P2, P3, P4, P5, P6\}$. Результатом операции $R=R1 \text{ TIMES } R2$ является множество всех пар типа «поставщик-деталь», то есть

$$R=\{(S1, P1), (S1, P2), \dots, (S5, P6)\}.$$

Таким образом, отношение R будет состоять из 30 кортежей по 9 полей в каждом.

Замечание 2

Напрямую операция декартова произведения на практике не используется, однако, она является составной частью операции *соединения*. Поэтому данная операция важна в *концептуальном* плане. Кроме того, операция декартова произведения используется в языке запросов **SQL**.

1.5. Выборка (ограничение)

Выборка отношения **R** по формуле **f** представляет собой новое отношение с таким же заголовком и телом, состоящим из тех и только тех кортежей, которые удовлетворяют истинности логического выражения, задаваемого формулой **f**.

При этом для записи формулы **f** используются имена атрибутов, константы, логические операции **and**, **or**, **not**, операции сравнения и скобки.

Обозначение: **R WHERE f**

Пример 3

- Запрос Р WHERE Вес<14 дает отношение

Д#	Название	Тип	Вес	Город-Д
Р1	Гайка	Каленый	12	Москва
Р5	Палец	Твердый	12	Киев

Пример 4

Для операции выборки используют также обозначение

SELECT (имя таблицы: условие выбора)

Например, по запросу

SELECT (SP: П# = 'S1' and Д# = 'P1')

получим следующее отношение:

П#	Д#	Количество
S1	P1	300

1.6. Проекция (PROJECT)

Проекция отношения A на *различные* атрибуты $X, Y, \dots, Z$, где множество $\{X, Y, \dots, Z\}$ является подмножеством полного списка атрибутов заголовка отношения A , представляет собой отношение с заголовком $X, Y, \dots, Z$ и телом, содержащим кортежи отношения A , за исключением повторяющихся кортежей.

Обозначение: $[X, Y, \dots, Z]$.

Дополнительные варианты записи операции проекции

- Отсутствие списка атрибутов подразумевает указание всех атрибутов (операция тождественной проекции).
- Выражение вида $A[]$ означает пустую проекцию, результатом которой является пустое отношение.
- Операция проекции может применяться и к результату выборки.

Замечание 3

Если операцию выборки можно представить как исключение ненужных **кортежей**, то операцию создания проекций можно представить как исключение ненужных **атрибутов**.

1.7. Деление

Операция деления определена для двух отношений $R1$ и $R2$ при следующих условиях: отношение $R1$ должно иметь минимум два простых или составных атрибута A и B , а отношение $R2$ должно содержать атрибут B , при этом атрибуты A и B должны быть определены на одном и том же домене.

Результатом деления отношения $R1$ на отношение $R2$ является отношение R с заголовком A и телом, состоящим из кортежей r таких, что в отношении $R1$ имеются кортежи (r, s) , причем множество значений s включает множество значений атрибута B в отношении $R2$.

Обозначение: **$R = R1 \text{ DIVIDEBY } R2$** .

На практике операция деления нужна для того, чтобы мы могли выполнять кванторные операции $\forall x$ и $\exists x$.

Пример 5

Пусть $R1$ - проекция $SP[П\#, Д\#]$, а $R2$ – отношение с заголовком $Д\#$ и телом $\{P2, P4\}$.

$Д\#$

$P2$

$P4$

$П\#$

$Д\#$

$S1$

$P1$

$S1$

$P2$

$S1$

$P3$

$S1$

$P4$

$S1$

$P5$

$S1$

$P6$

$S2$

$P1$

$S2$

$P2$

$S3$

$P2$

$S4$

$P2$

$S4$

$P4$

$S4$

$P5$

Результатом деления отношения $R1$ на отношение $R2$ будет отношение с заголовком $\Pi\#$ и телом $\{S1, S4\}$.

Операция деления является *обратной* к операции декартова произведения: если отношение является декартовым произведением двух отношений $R1$ и $R2$, то мы можем получить таблицу $R1$, разделив декартово произведение на $R2$, то есть

$$(R1 \text{ TIMES } R2) \text{ DIVIDEBY } R2 = R1.$$

Иногда используют и такое обозначение: $(R1 * R2) / R2 = R1$, где декартово произведение обозначено символом $*$, а деление – символом $/$.

1.8. Соединение

Операция соединения используется для связывания данных между отношениями. Это, возможно, наиболее важная функция любого языка БД.

В общем случае соединением отношения $R1$ по атрибуту A с отношением $R2$ по атрибуту B (отношения $R1$ и $R2$ не имеют общих атрибутов) является результат выполнения операции **$(R1 \text{ TIMES } R2) \text{ WHERE } A \cdot B$** , где $\cdot$ - логическое выражение над атрибутами A и B , определенными на одном и том же домене.

У операции соединения есть несколько вариантов:

- *естественное соединение* (обозначение: **JOIN (R1, R2)**);
- *тета-соединение*;
- *эквисоединение* (обозначение: **EQUIJOIN (R1, R2)**) - частный случай тета-соединения;
- *внешнее соединение* (обозначение: **OUTERJOIN (R1, R2)**).

1.8. 1. Естественное соединение

Естественное соединение (NATURAL JOIN) – это операция, связывающая отношения, когда общие атрибуты имеют равные значения. Общее определение естественного соединения таково: пусть отношения $R1$ и $R2$ имеют общие атрибуты $C1, C2, \dots, Cn$. Тогда операция $JOIN(R1, R2)$ выполняется за следующие три шага:

- 1) Берется декартово произведение отношений $R1$ и $R2$. В результате получается отношение R , содержащее по два столбца на каждый атрибут $C1, C2, \dots, Cn$.
- 2) Из отношения R исключаются все кортежи кроме тех, в которых значения атрибутов $C1, C2, \dots, Cn$ из отношения $R1$ равны соответственно значениям этих кортежей в отношении $R2$.
- 3) Операцией проектирования исключается одна копия кортежей $C1, C2, \dots, Cn$.

1.8. 2. Тета-соединение и эквисоединение

Тета-соединение – операция, связывающая отношения, когда значения из определенных атрибутов находятся в определенном соотношении. Общий вид тета-соединения следующий:

JOIN ($R_1, R_2: X \cdot Y$), где R_1, R_2 – соединяемые отношения, а $\cdot$ – одно из бинарных отношений: $=, \neq, <, >, \leq, \geq, X$ и Y – атрибуты отношений R_1 и R_2 .

Тета-соединение, основанное на отношении **равенства** определенных атрибутов, называется **эквисоединением**.

Замечание 4

Тета-соединение, в отличие от естественного соединения, не включает удаления одного или более атрибутов на последнем шаге 3). Так, если отношение $R1$ содержит k атрибутов, а отношение $R2$ – m атрибутов, то тета-соединение будет состоять из $k+m$ атрибутов.

1.8. 3. Внешнее соединение

Внешнее соединение является **расширением** естественного соединения, включает все кортежи из обоих отношений и гарантирует, что каждый кортеж из обоих исходных отношений будет представлен в отношении соединения хотя бы один раз. Внешнее соединение выполняется в два этапа. Сначала выполняется естественное соединение, затем, если какой-либо кортеж одного из исходных отношений не подходит ни к какому кортежу второго отношения, он включается в отношение соединения, а все дополнительные кортежи заполняются пустыми значениями (**NULL**).

Пример 6 (эквисоединение)

Найдем соединение отношений S и P по общему атрибуту, характеризующе-му город (в отношении S – это город_П, а в отношении P – город_Д). Поскольку условие операции требует одинаковости имен атрибутов, по которым выполняется соединение, то применяется операция переименования атрибутов - RENAME:

```
(S RENAME Город_П as Город) JOIN  
(P RENAME Город_Д as Город).
```

Результатом эквисоединения будет следующее отношение

П#	Имя	Ст.	Город	Д#	Название	Тип	Вес
S1	ДЫШЕКОВА	20	Москва	P1	Гайка	Каленый	12
S1	ДЫШЕКОВА	20	Москва	P4	Винт	Каленый	14
S1	ДЫШЕКОВА	20	Москва	P6	Шпилька	Каленый	19
S2	КУДИНОВ	10	Киев	P2	Болт	Мягкий	17
S2	КУДИНОВ	10	Киев	P5	Палец	Твердый	12
S3	ОЛЕНЕВА	30	Киев	P2	Болт	Мягкий	17
S3	ОЛЕНЕВА	30	Киев	P5	Палец	Твердый	12
S4	РОМАНОВА	20	Москва	P1	Гайка	Каленый	12
S4	РОМАНОВА	20	Москва	P4	Винт	Каленый	14
S4	РОМАНОВА	20	Москва	P5	Шпилька	Каленый	19

Обзор дополнительных операций реляционной алгебры, предложенных Дейтом

1Д. *Операция переименования атрибутов:*

RENAME <имя отношения> <старое имя
атрибута> **AS** <новое имя атрибута>

(см. пример 6)

2Д. *Операция расширения*, заключающаяся в добавлении нового атрибута в данное отношение, значения которого получаются путем некоторых скалярных вычислений:

EXTEND <имя отношения> **ADD** <выражение>
AS <имя нового атрибута>

В операции расширения имя нового атрибута не должно повторять имен имеющихся в данном отношении атрибутов и его значения вычисляются по правилам, заданным в <выражении>; имя нового атрибута не должно входить в <выражение>.

Помимо обычных алгебраических операций и операций сравнения в <выражении> можно использовать различные функции, называемые *итоговыми*, такими как **COUNT** (количество), **SUM** (сумма), **AVG** (среднее), **MAX** (максимальное), **MIN** (минимальное), например:

EXTEND (P **JOIN** SP) **ADD** (Вес * количество)
AS общий_вес.

3Д. Операция подведения итогов

Синтаксис этой операции имеет вид:

SUMMARIZE <имя исходного отношения> **BY**
(<список атрибутов>) **ADD** <выражение> **AS**
 <имя нового атрибута> ,

где <имя исходного отношения> может задаваться заключенным в круглые скобки выражением реляционной алгебры; <список атрибутов> представляет собой разделенные запятыми имена атрибутов исходного отношения; <выражение> - скалярное выражение реляционной алгебры.

Результатом операции подведения итогов является отношение R с заголовком, состоящим из атрибутов списка, расширенного новым атрибутом. Для получения отношения R сначала выполняется операция проекции исходного отношения на атрибуты $A_1, A_2, \dots, A_n$, после чего каждый кортеж проекции расширяется новым $(n+1)$ -ым атрибутом.

Пример 6

Пусть требуется вычислить число поставок по каждому из поставщиков. Это можно осуществить с помощью операции подведения итогов:

SUMMARIZE SP **BY**
(П#) **ADD COUNT AS**
Кол-ство_поставок
Получим отношение

П#	Кол-во_поставок
S1	6
S2	2
S3	1
S4	3

В приведенной операции функция COUNT определяет количество кортежей в каждой из групп исходного отношения.

4Д. Операция присваивания

Синтаксис операции:

**<выражение-цель>:= <выражение-
источник>**,

где оба выражения задают совместимые по структуре отношения. Типичный случай выражений: в левой части – имя отношения, а в правой – некоторое выражение реляционной алгебры.

5Д. Операция вставки

Синтаксис операции:

INSERT <выражение-источник> **INTO**
 <выражение-цель> ,

где оба выражения должны быть совместимы по структуре. Результатом этой операции является вставка полученных кортежей в отношение, заданное в <выражение-цель>, например:

INSERT (S **WHERE** Город_П = 'Москва')
 INTO Temp

6Д. Операция обновления

Синтаксис операции:

UPDATE <выражение-цель><список элементов> ,

где <список элементов> представляет собой последовательность разделенных запятыми операций присваивания вида <атрибут>:= <скалярное выражение>, например:

UPDATE WHERE Тип:='Каленый',
Город:='Киев'

7Д. Операция удаления

8Д. Операции реляционного сравнения

Синтаксис операции 7Д:

DELETE <выражение-
цель > ,

где <выражение-цель>
представляет собой реляционное выражение,
описывающее удаляемые кортежи.

К операциям
8Д

относятся:

=, **≠** (или **<**
>), **<**, **≤** (или
< =), **>**, **≥**
(или **> =**).

Замечание 5

При записи выражений в реляционной алгебре следует придерживаться следующих правил:

- должен быть определен приоритет выполнения операций. Для изменения порядка выполнения операций в выражениях, как обычно, используются круглые скобки;

- выражения можно заменять другими выражениями, получающимися с помощью тождественных преобразований;
- составляя выражение, нужно обеспечивать совместимость участвующих в операциях отношений. При необходимости изменения заголовков отношений следует выполнять переименование атрибутов.

2. Реляционное исчисление

Принципиальное отличие между реляционной алгеброй и реляционным исчислением состоит в том, что в первом случае процесс получения искомого результата описывается явным образом путем указания набора операций, которые надо выполнить, а во втором – указываются свойства искомого отношения без конкретизации процедуры его получения.

Понятие реляционного исчисления, как языка работы с БД, было впервые предложено **Э. Коддом**. Им же был разработан язык **ALPHA** – прототип программно реализованного языка **QUEL**, который некоторое время конкурировал с языком **SQL**.

Существует два варианта реляционного исчисления: ***исчисление кортежей*** и ***исчисление доменов***.

2.1. Исчисление кортежей

В исчислении кортежей, как и в процедурных языках программирования, сначала нужно описать используемые переменные, а затем записывать некоторые выражения. Описательную часть исчисления можно представить в виде:

RANGE OF <переменная> **IS** <список> ,

где **RANGE**, **OF**, **IS** – ключевые слова,

<переменная> - идентификатор переменной кортежа (области значений), а <список> - последовательность одного или более элементов, разделенных запятыми, то есть конструкция вида

$$x_1 [, x_2 [, ..., x_n] ...].$$

Вся конструкция **RANGE** указывает идентификатор переменной и область ее допустимых значений. Список элементов содержит элементы, каждый из которых является либо отношением, либо выражением над отношением.

Все элементы списка должны быть совместимы по типу, то есть соответствующие элементам отношения должны иметь идентичные заголовки. Область допустимых значений <переменной> образуется путем объединения значений всех элементов списка. Так, запись **RANGE OF T IS X1, X2** означает, что область определения переменной **T** включает в себя все значения из отношения, которое является объединением отношений **X1** и **X2**.

Пример 1

Варианты описаний:

- RANGE OF SX IS S;
- RANGE OF SPX IS SP;
- RANGE OF SY IS (SX) WHERE SX.

Город_П = 'Москва', (SX) WHERE
EXISTS SPX (SPX. П# = X. П# AND SPX.
Д# = 'P1').

Переменные **SX** и **SPX** в первых двух описаниях определены соответственно на отношениях **S** и **SP**. В третьем описании переменная **SY** может принимать значения из множества кортежей отношения **S** для поставщиков из г. Москвы или поставщиков деталей **P1**. Служебное слово **EXISTS** означает *существование* хотя бы одного кортежа, удовлетворяющего условию, записанному в последних круглых скобках.

Пример 2

Записать выражение на языке исчисления кортежей, соответствующее запросу «Получить имена поставщиков, которые поставляют все детали».

SX. Город_П **WHERE FORALL PX (EXISTS**
SPX (SPX.П# = SX.П# AND SPX. Д# = SX. Д#))
или (второй вариант этого же запроса):

SX. Город_П **WHERE NOT EXISTS PX (NOT**
EXISTS SPX (SPX. П# = SX.П# AND SPX. Д#
= SX. Д#))

Замечание

В исчисление кортежей можно добавить вычислительные функции из реляционной алгебры: **COUNT**, **SUM**, **AVG**, **MAX**, **MIN**. Тогда это исчисление будет обладать вычислительной полнотой.

2.2. Исчисление доменов

Исчисление доменов было предложено **Лакроиксом** и **Пиротте**, которые на его основе разработали язык **ILL**. Другими языками, основанными на исчислении доменов, являются языки **FQL**, **DEDUCE**, а также **QBE**.

В исчислении доменов основой любого запроса выступают переменные доменов. ***Переменная домена – это скалярная переменная, значения которой охватывают элементы некоторого домена.***

Отличительной особенностью исчисления доменов от исчисления кортежей является то, что в нем поддерживается дополнительная форма условия, называемого условием принадлежности. В общем виде это условие записывается так:

$R(A_1:\eta_1, A_2:\eta_2, \dots)$,

где A_i – атрибут отношения R , а η_i – переменная домена или литерал.

Проверяемое условие будет истинным тогда и только тогда, когда существует кортеж в отношении R , имеющий атрибуты A_i , равные заданным в выражении соответствующим значениям η_i .

Примеры выражений исчисления доменов

- $(SX) \text{ WHERE } S (\Pi\# : SX)$ – множество всех номеров поставщиков отношения S ;
- $(SX) \text{ WHERE } S (\Pi\# : SX, \text{Город_П: 'Москва'})$ – множество поставщиков из Москвы;
- $\text{NAMEX WHERE EXISTS } SX (S(\Pi\# : SX, \text{Имя: NAMEX}) \text{ AND FORALL } PX (\text{IF } P (\text{Д\# : } PX) \text{ THEN } SP (\Pi\# : SX, \text{Д\# : } PX))))$.

Третье выражение соответствует запросу на получение имен поставщиков, производящих все детали.

В примерах **SX** – домен атрибута П#, **PX** – домен атрибута Д#, **NAMEX** – домен атрибута Имя (переменные объявляются заранее с помощью оператора **RANGE**).