

Введение в системы управления базами данных

А.Ю.Пушников

Данное учебное пособие было опубликовано в 1999 г. в двух частях издательством Башкирского государственного университета. Выходные данные: УДК 519.6, ББК 32.973-018.2 Пушников А.Ю. Введение в системы управления базами данных. Часть 1. Реляционная модель данных: Учебное пособие/Изд-е Башкирского ун-та. - Уфа, 1999. - 108 с. - ISBN 5-7477-0350-1. В электронной версии - полный текст книги

Web-страница: <http://www.citforum.ru/database/dblearn/index.shtml>

Реляционная алгебра

Обзор реляционной алгебры

Третья часть реляционной модели, манипуляционная часть, утверждает, что доступ к реляционным данным осуществляется при помощи реляционной алгебры или эквивалентного ему реляционного исчисления.

В реализациях конкретных реляционных СУБД сейчас не используется в чистом виде ни реляционная алгебра, ни реляционное исчисление. Фактическим стандартом доступа к реляционным данным стал язык SQL (Structured Query Language). Язык SQL представляет собой смесь операторов реляционной алгебры и выражений реляционного исчисления, использующий синтаксис, близкий к фразам английского языка и расширенный дополнительными возможностями, отсутствующими в реляционной алгебре и реляционном исчислении. Вообще, язык доступа к данным называется **реляционно полным**, если он по выразительной силе не уступает реляционной алгебре (или, что то же самое, реляционному исчислению), т.е. любой оператор реляционной алгебры может быть выражен средствами этого языка. Именно таким и является язык SQL.

В данной главе будут рассмотрены основы реляционной алгебры.

Замкнутость реляционной алгебры

Реляционная алгебра представляет собой набор операторов, использующих отношения в качестве аргументов, и возвращающие отношения в качестве результата. Таким образом, реляционный оператор f выглядит как функция с отношениями в качестве аргументов:

$$R = f(R_1, R_2, \dots, R_n)$$

Реляционная алгебра является замкнутой, т.к. в качестве аргументов в реляционные операторы можно подставлять другие реляционные операторы, подходящие по типу:

$$R = f(f_1(R_{11}, R_{12}, \dots), f_2(R_{21}, R_{22}, \dots), \dots)$$

Таким образом, в реляционных выражениях можно использовать вложенные выражения сколь угодно сложной структуры.

Каждое отношение обязано иметь уникальное имя в пределах базы данных. Имя отношения, полученного в результате выполнения реляционной операции, определяется в

левой части равенства. Однако можно не требовать наличия имен от отношений, полученных в результате реляционных выражений, если эти отношения подставляются в качестве аргументов в другие реляционные выражения. Такие отношения будем называть **неименованными отношениями**. Неименованные отношения реально не существуют в базе данных, а только вычисляются в момент вычисления значения реляционного оператора.

Традиционно, вслед за Коддом [43], определяют восемь реляционных операторов, объединенных в две группы.

Теоретико-множественные операторы:

- Объединение
- Пересечение
- Вычитание
- Декартово произведение

Специальные реляционные операторы:

- Выборка
- Проекция
- Соединение
- Деление

Не все они являются независимыми, т.е. некоторые из этих операторов могут быть выражены через другие реляционные операторы.

Отношения, совместимые по типу

Некоторые реляционные операторы (например, объединение) требуют, чтобы отношения имели одинаковые заголовки. Действительно, отношения состоят из заголовка и тела. Операция объединения двух отношений есть просто объединение двух множеств кортежей, взятых из тел соответствующих отношений. Но будет ли результат отношением? Во-первых, если исходные отношения имеют разное количество атрибутов, то, очевидно, что множество, являющееся объединением таких разнотипных кортежей, нельзя представить в виде отношения. Во-вторых, пусть даже отношения имеют одинаковое количество атрибутов, но атрибуты имеют различные наименования. Как тогда определить заголовок отношения, полученного в результате объединения множеств кортежей? В-третьих, пусть отношения имеют одинаковое количество атрибутов, атрибуты имеют одинаковые наименования, но определены на различных доменах. Тогда снова объединение кортежей не будет образовывать отношение.

Определение 1. Будем называть отношения **совместимыми по типу**, если они имеют идентичные заголовки, а именно,

- Отношения имеют *одно и то же множество имен атрибутов*, т.е. для любого атрибута в одном отношении найдется атрибут с таким же наименованием в другом отношении,
- Атрибуты с одинаковыми именами *определены на одних и тех же доменах*.

Некоторые отношения не являются совместимыми по типу, но становятся таковыми после некоторого переименования атрибутов. Для того чтобы такие отношения можно было

использовать в реляционных операторах, вводится вспомогательный **оператор переименования атрибутов**.

Оператор переименования атрибутов

Оператор переименования атрибутов имеет следующий синтаксис:

$$R \text{ RENAME } Atr_1, Atr_2, \dots \text{ AS } NewAtr_1, NewAtr_2, \dots$$

где

R - отношение,

$Atr_1, Atr_2, \dots$ - исходные имена атрибутов,

$NewAtr_1, NewAtr_2, \dots$ - новые имена атрибутов.

В результате применения оператора переименования атрибутов получаем новое отношение, с измененными именами атрибутов.

Пример 1.

Следующий оператор возвращает неименованное отношение, в котором атрибут $City_Num$ переименован в $CityId$:

$$City \text{ RENAME } City_Num \text{ AS } CityId$$

Теоретико-множественные операторы

Объединение

Определение 2. **Объединением** двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из кортежей, принадлежащих или A , или B , или обоим отношениям.

Синтаксис операции объединения:

$A \text{ UNION } B$

Замечание. Объединение, как и любое отношение, не может содержать одинаковых кортежей. Поэтому, если некоторый кортеж входит и в отношение A , и отношение B , то в объединение он входит один раз.

Пример 2. Пусть даны два отношения A и B с информацией о сотрудниках:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 1 Отношение А

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 2 Отношение В

Объединение отношений A и B будет иметь вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000
2	Пушников	2500
4	Сидоров	3000

Таблица 3 Отношение A UNION B

Замечание. Как видно из приведенного примера, потенциальные ключи, которые были в отношениях A и B не наследуются объединением этих отношений. Поэтому, в объединении отношений A и B атрибут "Табельный номер" может содержать дубликаты значений. Если бы это было не так, и ключи наследовались бы, то это противоречило бы понятию объединения как "объединение множеств". Конечно, объединение отношений A и B имеет, как и любое отношение, потенциальный ключ, например, состоящий из всех атрибутов.

Пересечение

Определение 3. Пересечением двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям A и B .

Синтаксис операции пересечения:

$A \text{ INTERSECT } B$

Пример 3. Для тех же отношений A и B , что и в предыдущем примере пересечение имеет вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000

Таблица 4 Отношение $A \text{ INTERSECT } B$

Замечание. Казалось бы, что в отличие от операции объединения, потенциальные ключи могли бы наследоваться пересечением отношений. Однако это не так. Вообще, *никакие реляционные операторы не передают результирующему отношению никаких данных о потенциальных ключах*. В качестве причины этого можно было бы привести тривиальное соображение, что так получается более просто и симметрично - все операторы устроены одинаково. На самом деле причина более глубока, и заключается в том, что потенциальный ключ - семантическое понятие, отражающее различимость объектов предметной области. Наличие потенциальных ключей *не выводится* из структуры отношения, а явно задается для каждого отношения, исходя из его смысла. Реляционные же операторы являются *формальными* операциями над отношениями и выполняются одинаково, независимо от смысла данных, содержащихся в отношениях. Поэтому, реляционные операторы ничего не могут "знать" о смысле данных. Трактовка результата реляционных операций - дело пользователя.

Вычитание

Определение 4. Вычитанием двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из кортежей, принадлежащих отношению A и не принадлежащих отношению B .

Синтаксис операции вычитания:

$A \text{ MINUS } B$

Пример 4. Для тех же отношений A и B , что и в предыдущем примере вычитание имеет вид:

Табельный номер	Фамилия	Зарплата
2	Петров	2000
3	Сидоров	3000

Таблица 5 Отношение A MINUS B

Декартово произведение

Определение 5. Декартовым произведением двух отношений $A(A_1, A_2, \dots, A_n)$ и $B(B_1, B_2, \dots, B_m)$ называется отношение, заголовок которого является **сцеплением заголовков** отношений A и B :

$$(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m),$$

а тело состоит из кортежей, являющихся **сцеплением кортежей** отношений A и B :

$$(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m),$$

таких, что $(a_1, a_2, \dots, a_n) \in A$, $(b_1, b_2, \dots, b_m) \in B$.

Синтаксис операции декартового произведения:

A TIMES B

Замечание. Мощность произведения A TIMES B равна произведению мощностей отношений A и B , т.к. каждый кортеж отношения A соединяется с каждым кортежем отношения B .

Замечание. Если в отношения A и B имеются атрибуты с одинаковыми наименованиями, то перед выполнением операции декартового произведения такие атрибуты необходимо переименовать.

Замечание. Перемножать можно любые два отношения, совместимость по типу при этом не требуется.

Пример 5. Пусть даны два отношения A и B с информацией о поставщиках и деталях:

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Таблица 6 Отношение А (Поставщики)

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Таблица 7 Отношение В (Детали)

Декартово произведение отношений A и B будет иметь вид:

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали
1	Иванов	1	Болт
1	Иванов	2	Гайка
1	Иванов	3	Винт
2	Петров	1	Болт
2	Петров	2	Гайка
2	Петров	3	Винт
3	Сидоров	1	Болт
3	Сидоров	2	Гайка
3	Сидоров	3	Винт

Таблица 8 Отношение A TIMES B

Замечание. Сама по себе операция декартового произведения не очень важна, т.к. она не дает никакой новой информации, по сравнению с исходными отношениями. Для реальных запросов эта операция почти никогда не используется. Однако операция декартового произведения важна для выполнения специальных реляционных операций, о которых речь пойдет ниже.

Специальные реляционные операторы

Выборка (ограничение, селекция)

Определение 6. Выборкой (ограничением, селекцией) на отношении A с условием C называется отношение с тем же заголовком, что и у отношения A , и телом, состоящем из кортежей, значения атрибутов которых при подстановке в условие C дают значение ИСТИНА. C представляет собой логическое выражение, в которое могут входить атрибуты отношения A и (или) скалярные выражения.

В простейшем случае условие C имеет вид $X \Theta Y$, где Θ - один из операторов сравнения ($=, \neq, <, \leq, >, \geq$ и т.д.), а X и Y - атрибуты отношения A или скалярные значения. Такие выборки называются Θ -выборки (тэта-выборки) или Θ -ограничения, Θ -селекции.

Синтаксис операции выборки:

$A \text{ WHERE } c$,

или

$A \text{ WHERE } X \Theta Y$

Пример 6. Пусть дано отношение A с информацией о сотрудниках:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 9 Отношение A

Результат выборки $A \text{ WHERE Зарплата} < 3000$ будет иметь вид:

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000

Таблица 10 Отношение $A \text{ WHERE Зарплата} < 3000$

Смысл операции выборки очевиден - выбрать кортежи отношения, удовлетворяющие некоторому условию. Таким образом, операция выборки дает "*горизонтальный срез*" отношения по некоторому условию.

Проекция

Определение 7. Проекцией отношения A по атрибутам $X, Y, \dots, Z$, где каждый из атрибутов принадлежит отношению A , называется отношение с заголовком $(X, Y, \dots, Z)$ и телом, содержащим множество кортежей вида $(x, y, \dots, z)$, таких, для которых в отношении A найдутся кортежи со значением атрибута X равным x , значением атрибута Y равным y , ..., значением атрибута Z равным z .

Синтаксис операции проекции:

$A[X, Y, \dots, Z]$

Замечание. Операция проекции дает "*вертикальный срез*" отношения, в котором удалены все возникшие при таком срезе дубликаты кортежей.

Пример 7. Пусть дано отношение A с информацией о поставщиках, включающих наименование и месторасположение:

Номер поставщика	Наименование поставщика	Город поставщика
1	Иванов	Уфа
2	Петров	Москва
3	Сидоров	Москва
4	Сидоров	Челябинск

Таблица 11 Отношение A (Поставщики)

Проекция $\pi_{A[Город поставщика]}$ будет иметь вид:

Город поставщика
Уфа
Москва
Челябинск

Таблица 12 Отношение A[Город поставщика]

Соединение

Операция соединения отношений, наряду с операциями выборки и проекции, является одной из наиболее важных реляционных операций.

Обычно рассматривается несколько разновидностей операции соединения:

- Общая операция соединения
- $\bowtie$ -соединение (тэта-соединение)
- Экви-соединение
- Естественное соединение

Наиболее важным из этих частных случаев является операция естественного соединения. Все разновидности соединения являются частными случаями общей операции соединения.

Общая операция соединения

Определение 8. Соединением отношений A и B по условию c называется отношение $(A \text{ TIMES } B) \text{ WHERE } c$

c представляет собой логическое выражение, в которое могут входить атрибуты отношений A и B и (или) скалярные выражения.

Таким образом, операция соединения есть результат последовательного применения операций декартового произведения и выборки. Если в отношениях A и B имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать.

Тэта-соединение

Определение 9. Пусть отношение A содержит атрибут X , отношение B содержит атрибут Y , а Θ - один из операторов сравнения ($=, \neq, <, \leq, >, \geq$ и т.д.). Тогда Θ -соединением отношения A по атрибуту X с отношением B по атрибуту Y называют отношение

$(A \text{ TIMES } B) \text{ WHERE } X \Theta Y$

Это частный случай операции общего соединения.

Иногда, для операции Θ -соединения применяют следующий, более короткий синтаксис:

$A[X \Theta Y]B$

Пример 8. Рассмотрим некоторую компанию, в которой хранятся данные о поставщиках и поставляемых деталях. Пусть поставщикам и деталям присвоен некий статус. Пусть бизнес компании организован таким образом, что поставщики имеют право поставлять только те детали, статус которых не выше статуса поставщика (смысл этого может быть в том, что хороший поставщик с высоким статусом может поставлять больше разновидностей деталей, а плохой поставщик с низким статусом может поставлять только ограниченный список деталей, важность которых (статус детали) не очень высока).

Номер поставщика	Наименование поставщика	X (Статус поставщика)
1	Иванов	4
2	Петров	1
3	Сидоров	2

Таблица 13 Отношение А (Поставщики)

Номер детали	Наименование детали	Y (Статус детали)
1	Болт	3
2	Гайка	2
3	Винт	1

Таблица 14 Отношение В (Детали)

Ответ на вопрос "какие поставщики имеют право поставлять какие детали?" дает Θ -соединение $A[X \geq Y]B$:

Номер поставщика	Наименование поставщика	Х (Статус поставщика)	Номер детали	Наименование детали	У (Статус детали)
1	Иванов	4	1	Болт	3
1	Иванов	4	2	Гайка	2
1	Иванов	4	3	Винт	1
2	Петров	1	3	Винт	1
3	Сидоров	2	2	Гайка	2
3	Сидоров	2	3	Винт	1

Таблица 15 Отношение "Какие поставщики поставляют какие детали"

Экви-соединение

Наиболее важным частным случаем $\bowtie$ -соединения является случай, когда $\bowtie$ есть просто равенство.

Синтаксис *экви-соединения*:

$$A[X=Y]B$$

Пример 9. Пусть имеются отношения P , D и PD , хранящие информацию о поставщиках, деталях и поставках соответственно (для удобства введем краткие наименования атрибутов):

Номер поставщика	Наименование поставщика
<i>PNUM</i>	<i>PNAME</i>
1	Иванов
2	Петров
3	Сидоров

Таблица 16 Отношение P (Поставщики)

Номер детали	Наименование детали
<i>DNUM</i>	<i>DNAME</i>
1	Болт
2	Гайка
3	Винт

Таблица 17 Отношение D (Детали)

<i>Номер поставщика</i>	<i>Номер детали</i>	<i>Поставляемое количество</i>
<i>PNUM</i>	<i>DNUM</i>	<i>VOLUME</i>
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	1	1000

Таблица 18 Отношение PD (Поставки)

Ответ на вопрос, какие детали поставляются поставщиками, дает экви-соединение $P[PNUM = PNUM]PD$. На самом деле, т.к. в отношениях имеются одинаковые атрибуты, то требуется сначала переименовать атрибуты, а потом выполнить экви-соединение. Запись становится более громоздкой:

$$(P \text{ RENAME } PNUM \text{ AS } PNUM1)[PNUM1 = PNUM2](PD \text{ RENAME } PNUM \text{ AS } PNUM2)$$

Обычно, такой сложной формой записи не пользуются. Но как бы то ни было, в результате имеем отношение:

Номер поставщика	Наименование поставщика	Номер поставщика	Номер детали	Поставляемое количество
PNUM1	PNAME	PNUM2	DNUM	VOLUME
1	Иванов	1	1	100
1	Иванов	1	2	200
1	Иванов	1	3	300
2	Петров	2	1	150
2	Петров	2	2	250
3	Сидоров	3	1	1000

Таблица 19 Отношение "Какие детали поставляются какими поставщиками"

Недостатком экви-соединения является то, что если соединение происходит по атрибутам с одинаковыми наименованиями (а так чаще всего и происходит!), то в результирующем отношении появляется два атрибута с одинаковыми значениями. В нашем примере атрибуты PNUM1 и PNUM2 содержат дублирующие данные. Избавиться от этого недостатка можно, взяв проекцию по всем атрибутам, кроме одного из дублирующих. Именно так действует естественное соединение.

Естественное соединение

Определение 10. Пусть даны отношения $A(A_1, A_2, \dots, A_n, X_1, X_2, \dots, X_p)$ и $B(X_1, X_2, \dots, X_p, B_1, B_2, \dots, B_m)$, имеющие одинаковые атрибуты $X_1, X_2, \dots, X_p$ (т.е. атрибуты с одинаковыми именами и определенные на одинаковых доменах).

Тогда **естественным соединением** отношений A и B называется отношение с заголовком $(A_1, A_2, \dots, A_n, X_1, X_2, \dots, X_p, B_1, B_2, \dots, B_m)$ и телом, содержащим множество кортежей $(a_1, a_2, \dots, a_n, x_1, x_2, \dots, x_p, b_1, b_2, \dots, b_m)$, таких, что $(a_1, a_2, \dots, a_n, x_1, x_2, \dots, x_p) \in A$ и $(x_1, x_2, \dots, x_p, b_1, b_2, \dots, b_m) \in B$.

Естественное соединение настолько важно, что для него используют специальный синтаксис:

$A \text{ JOIN } B$

Замечание. В синтаксисе естественного соединения не указываются, по каким атрибутам производится соединение. Естественное соединение производится *по всем* одинаковым атрибутам.

Замечание. Естественное соединение эквивалентно следующей последовательности реляционных операций:

1. Переименовать одинаковые атрибуты в отношениях
2. Выполнить декартово произведение отношений
3. Выполнить выборку по совпадающим значениям атрибутов, имевших одинаковые имена
4. Выполнить проекцию, удалив повторяющиеся атрибуты
5. Переименовать атрибуты, вернув им первоначальные имена

Замечание. Можно выполнять последовательное естественное соединение нескольких отношений. Нетрудно проверить, что естественное соединение (как, впрочем, и соединение общего вида) обладает свойством **ассоциативности**, т.е.

$$(A \text{ JOIN } B) \text{ JOIN } C = A \text{ JOIN } (B \text{ JOIN } C)$$

поэтому такие соединения можно записывать, опуская скобки:

$A \text{ JOIN } B \text{ JOIN } C$

Пример 10. В предыдущем примере ответ на вопрос "какие детали поставляются поставщиками", более просто записывается в виде естественного соединения трех отношений $P \text{ JOIN } PD \text{ JOIN } D$ (для удобства просмотра порядок атрибутов изменен, это является допустимым по свойствам отношений):

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Наименование детали DNAME	Поставляемое количество VOLUME
1	Иванов	1	Болт	100
1	Иванов	2	Гайка	200
1	Иванов	3	Винт	300
2	Петров	1	Болт	150
2	Петров	2	Гайка	250
3	Сидоров	1	Болт	1000

Таблица 20 Отношение P JOIN PD JOIN D

Деление

Определение 11. Пусть даны отношения $A(X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m)$ и $B(Y_1, Y_2, \dots, Y_m)$, причем атрибуты $Y_1, Y_2, \dots, Y_m$ - общие для двух отношений. *Делением отношений* A на B называется отношение с заголовком $(X_1, X_2, \dots, X_n)$ и телом, содержащим множество кортежей $(x_1, x_2, \dots, x_n)$, таких, что для *всех* кортежей $(y_1, y_2, \dots, y_m) \in B$ в отношении A найдется кортеж $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$.

Отношение A выступает в роли *делимого*, отношение B выступает в роли *делителя*. Деление отношений аналогично делению чисел с остатком.

Синтаксис операции деления:

A DIVIDE BY B

Замечание. Типичные запросы, реализуемые с помощью операции деления, обычно в своей формулировке имеют слово "все" - "какие поставщики поставляют *все* детали?".

Пример 11. В примере с поставщиками, деталями и поставками ответим на вопрос, "какие поставщики поставляют *все* детали?".

В качестве делимого возьмем проекцию $X = PD[PNUM, DNUM]$, содержащую номера поставщиков и номера поставляемых ими деталей:

Номер поставщика	Номер детали
PNUM	DNUM
1	1
1	2
1	3
2	1
2	2
3	1

Таблица 21 Проекция $X=PD[PNUM,DNUM]$

В качестве делителя возьмем проекцию $Y = D[DNUM]$, содержащую список номеров *всех* деталей (не обязательно поставляемых кем-либо):

Номер детали
DNUM
1
2
3

Таблица 22 Проекция $Y=D[DNUM]$

Деление $X \text{ DIVIDEBY } Y$ дает список номеров поставщиков, поставляющих *все* детали:

Номер поставщика
PNUM
1

Таблица 23 Отношение $X \text{ DIVIDEBY } Y$

Оказалось, что только поставщик с номером 1 поставляет все детали.

Примеры использования реляционных операторов

Пример 12. Получить имена поставщиков, поставляющих деталь номер 2.

Решение:

$((DP \text{ JOIN } P) \text{ WHERE } DNUM = 2)[PNAME]$

Пример 13. Получить имена поставщиков, поставляющих по крайней мере одну гайку.

Решение:

$((D \text{ WHERE } DNAME = \text{Гайка}) \text{ JOIN } DP) \text{ JOIN } P[PNAME]$

Ответ на этот запрос можно получить и иначе:

$((D \text{ JOIN } DP) \text{ JOIN } P) \text{ WHERE } DNAME = \text{Гайка}[PNAME]$

Пример 14. Получить имена поставщиков, поставляющих все детали.

Решение:

$((DP[PNUM, DNUM] \text{ DEVIDE BY } D[DNUM]) \text{ JOIN } P)[PNAME]$

Пример 15. Получить имена поставщиков, не поставляющих деталь номер 2.

Решение:

$((P[PNUM] \text{ MINUS } (P \text{ JOIN } DP \text{ WHERE } DNUM = 2)[PNUM]) \text{ JOIN } P)[PNAME]$

Ответ на этот запрос можно получить и пошагово:

$T_1 = P[PNUM]$ - получить список номеров всех поставщиков

$T_2 = P \text{ JOIN } DP$ - соединить данные о поставщиках и поставках

$T_3 = T_2 \text{ WHERE } DNUM = 2$ - в данных о поставщиках и поставках оставить только данные о поставках детали номер 2.

$T_4 = T_3[PNUM]$ - получить список номеров поставщиков, поставляющих деталь номер 2.

$T_5 = T_1 \text{ MINUS } T_4$ - получить список номеров поставщиков, не поставляющих деталь номер 2.

$T_6 = T_5 \text{ JOIN } P$ - соединить список номеров поставщиков, не поставляющих деталь номер 2 с данными о поставщиках (получатся полные данные о поставщиках, не поставляющих деталь номер 2).

$T_7 = T_6[PNAME]$ - искомый ответ (имена поставщиков, не поставляющих деталь номер 2).

Зависимые реляционные операторы

Как было сказано в начале главы, не все операторы реляционной алгебры являются независимыми - некоторые из них выражаются через другие реляционные операторы.

Оператор соединения

Оператор соединения определяется через операторы декартового произведения и выборки. Для оператора естественного соединения добавляется оператор проекции.

Оператор пересечения

Оператор пересечения выражается через вычитание следующим образом:

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

Оператор деления

Оператор деления выражается через операторы вычитания, декартового произведения и проекции следующим образом:

$$A \text{ DEVIDEBY } B = A[X] \text{ MINUS } ((A[X] \text{ TIMES } B) \text{ MINUS } A[X])$$

Таким образом показано, что операторы *соединения*, *пересечения* и *деления* можно выразить через другие реляционные операторы, т.е. эти операторы не являются примитивными.

Примитивные реляционные операторы

Оставшиеся реляционные операторы (*объединение*, *вычитание*, *декартово произведение*, *выборка*, *проекция*) являются **примитивными операторами** - их нельзя выразить друг через друга.

Оператор декартового произведения

Оператор декартового произведения - это единственный оператор, *увеличивающий количество атрибутов*, поэтому его нельзя выразить через объединение, вычитание, выборку, проекцию.

Оператор проекции

Оператор проекции - единственный оператор, *уменьшающий количество атрибутов*, поэтому его нельзя выразить через объединение, вычитание, декартово произведение, выборку.

Оператор выборки

Оператор выборки - единственный оператор, позволяющий проводить сравнения по атрибутам отношения, поэтому его нельзя выразить через объединение, вычитание, декартово произведение, проекцию.

Операторы объединения и вычитания

Доказательство примитивности операторов объединения и вычитания более сложны, и мы их здесь не приводим.

Запросы, невыразимые средствами реляционной алгебры

Несмотря на мощь языка реляционной алгебры, имеется ряд типов запросов, которые принципиально нельзя выразить только при помощи операторов реляционной алгебры. Это вовсе не означает, что ответы на эти запросы нельзя получить вообще. Просто, для получения ответов на подобные запросы приходится применять процедурные расширения реляционных языков.

Плохая нормализация отношений

Данный пример взят из книги Гилуа М.М. [6, стр.43].

Пример 16. Пусть имеется отношение ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ с набором атрибутов (Наименование вещества, Водород, Гелий, ..., 105_элемент). Значением атрибута "Вещество" являются наименования химических веществ, значениями остальных атрибутов - процентный состав соответствующих элементов в этом веществе. Такое отношение могло бы иметь, к примеру, следующий вид:

Наименование вещества	Водород	Гелий	...	105 элемент
Дезоксирибону-клеиновая кислота	5	3	...	0.01
Бензин	50	0	...	0
...	...	...	...	...

Таблица 24 Отношение ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ

Рассмотрим запрос "Найти все химические элементы, содержание которых в каком-либо из веществ превышает заданный процент (скажем, 90)".

С алгоритмической точки зрения этот запрос выполняется элементарно - просматриваются все столбцы таблицы, если в столбце присутствует хотя бы одно значение, большее 90, то запоминается заголовок этого столбца. Набор наименований запомненных столбцов и является ответом на запрос.

Формально невозможно выразить этот запрос в рамках реляционной алгебры, т.к. ответом на этот запрос должен быть *список атрибутов* отношений, удовлетворяющих определенному условию. В реляционной алгебре нет операторов, манипулирующих с наименованиями атрибутов.

На самом деле, этот пример показывает, что таблица плохо нормализована (нормализация отношений рассматривается в гл.6 и 7). В таблице есть набор однотипных атрибутов ("Водород", "Гелий" и т.д. в количестве 105 столбцов).

Правильнее разбить это отношение на три различных отношения:

1. ВЕЩЕСТВО(НОМ_ВЕЩЕСТВА, ВЕЩЕСТВО),
2. ЭЛЕМЕНТЫ(НОМ_ЭЛЕМЕНТА, ЭЛЕМЕНТ),
3. ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ(НОМ_ВЕЩЕСТВА, НОМ_ЭЛЕМЕНТА, ПРОЦЕНТ).

НОМ_ВЕЩЕСТВА	ВЕЩЕСТВО
1	Дезоксирибонуклеиновая кислота
2	Бензин

Таблица 25 Отношение ВЕЩЕСТВО

НОМ_ЭЛЕМЕНТА	ЭЛЕМЕНТ
1	Водород
2	Гелий
...	...
105	...

Таблица 26 Отношение ЭЛЕМЕНТЫ

НОМ_ВЕЩЕСТВА	НОМ_ЭЛЕМЕНТА	ПРОЦЕНТ
1	1	5
1	2	3
1	105	0.01
2	1	50

Таблица 27 Отношение ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ

Для отношений, нормализованных таким образом, исходный запрос реализуется следующей последовательностью операторов:

1. $R1(\text{НОМЕР_ВЕЩЕСТВА}, \text{НОМ_ЭЛЕМЕНТА}, \text{ПРОЦЕНТ}) = \text{ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ}[\text{ПРОЦЕНТ} > 90]$. (Выборка из отношения).
2. $R2(\text{НОМ_ЭЛЕМЕНТА}) = R1[\text{НОМ_ЭЛЕМЕНТА}]$. (Проекция отношения).
3. $R3(\text{НОМ_ЭЛЕМЕНТА}, \text{ЭЛЕМЕНТ}) = R2[\text{НОМ_ЭЛЕМЕНТА} = \text{НОМ_ЭЛЕМЕНТА}]\text{ЭЛЕМЕНТЫ}$. (Естественное соединение)
4. $\text{ОТВЕТ}(\text{ЭЛЕМЕНТ}) = R3[\text{ЭЛЕМЕНТ}]$. (Проекция таблицы).

На языке SQL такой запрос реализуется одной командой:

```

SELECT ЭЛЕМЕНТЫ.ЭЛЕМЕНТ
FROM ЭЛЕМЕНТЫ, ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ
WHERE
    ЭЛЕМЕНТЫ.НОМ_ЭЛЕМЕНТА=ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ.НОМ_ЭЛЕМЕНТА
    AND ХИМИЧЕСКИЙ_СОСТАВ_ВЕЩЕСТВ.ПРОЦЕНТ>90;

```

Невыразимость транзитивного замыкания реляционными операторами

Следующий пример иллюстрирует класс запросов, невыразимых средствами реляционной алгебры или реляционного исчисления по причине невыразимости средствами реляционной алгебры транзитивного замыкания отношений (см. гл. 1).

Пример 17. Рассмотрим отношение, описывающее сотрудников некоего предприятия. Отношение содержит данные о табельном номере сотрудника, фамилии, должности и табельном номере руководителя сотрудника – СОТРУДНИКИ (ТАБ_НОМ, ФАМИЛИЯ, ДОЛЖНОСТЬ, ТАБ_НОМ_РУК):

ТАБ_НОМ	ФАМИЛИЯ	ДОЛЖНОСТЬ	ТАБ_НОМ_РУК
1	Иванов	Директор	1
2	Петров	Глав.бухгалтер	1
3	Сидоров	Бухгалтер	2
4	Васильев	Начальник цеха	1
5	Сухов	Мастер	4
6	Шарипов	Рабочий	5
...	...	...	...

Таблица 28 Отношение СОТРУДНИКИ

Рассмотрим запрос "Перечислить всех руководителей (прямых и непрямых) данного сотрудника".

Ответом на запрос может быть получен при помощи понятия транзитивного замыкания. Однако транзитивное замыкание не может быть выражено операторами реляционной алгебры.

Кросс-таблицы

Одной из задач, связанных с представлением табличных данных является построение так называемых кросс-таблиц.

Пусть имеется отношение с тремя атрибутами и потенциальным ключом, включающим первые два атрибута. Примером такого отношения могут быть данные с объемами продаж различных товаров за некоторые промежутки времени:

Товар	Месяц	Количество
Компьютеры	Январь	100
Принтеры	Январь	200
Сканеры	Январь	300
Компьютеры	Февраль	150
Принтеры	Февраль	250
Сканеры	Февраль	350
...	...	...

Таблица 29 Данные о продажах

Требуется представить эти данные в виде таблицы, по строкам которой идут наименования товаров, по столбцам - месяцы, а в ячейках содержатся объемы продаж. Это и будет кросс-таблицей:

Товар	Январь	Февраль	...
Компьютеры	100	150	...
Принтеры	200	250	...
Сканеры	300	350	...

Таблица 30 Кросс-таблица

Построение кросс-таблицы средствами реляционной алгебры невозможно, т.к. для этого требуется превратить данные в ячейках таблицы в наименования новых столбцов таблицы.

Выводы

Доступ к реляционным данным возможен при помощи операторов реляционной алгебры. Реляционная алгебра представляет собой набор операторов, использующих отношения в качестве аргументов, и возвращающие отношения в качестве результата. Реляционная алгебра замкнута таким образом, что результаты одних реляционных выражений можно использовать в других выражениях.

Традиционно определяют восемь реляционных операторов, объединенных в две группы.

Теоретико-множественные операторы: *объединение, пересечение, вычитание, декартово произведение.*

Специальные реляционные операторы: *выборка, проекция, соединение, деление.*

Для выполнения некоторых реляционных операторов требуется, чтобы отношения были *совместимы по типу.*

Не все операторы реляционной алгебры являются независимыми - некоторые из них выражаются через другие реляционные операторы. Операторы *соединения, пересечения и деления* можно выразить через другие реляционные операторы, т.е. эти операторы не являются примитивными. Оставшиеся реляционные операторы (*объединение, вычитание, декартово произведение, выборка, проекция*) являются **примитивными операторами** - их нельзя выразить друг через друга.

Имеется несколько типов запросов, которые нельзя выразить средствами реляционной алгебры. К ним относятся запросы, требующие дать в ответе список атрибутов, удовлетворяющих определенным условиям, построение транзитивного замыкания отношений, построение кросс-таблиц. Для получения ответов на подобные запросы приходится использовать процедурные расширения реляционных языков.