

Теория баз данных

Лекция 3. Реляционная алгебра и реляционное исчисление

Е. П. Моргунов

Сибирский федеральный университет

г. Красноярск

Институт космических и информационных технологий

emorgunov@mail.ru

3.1. Введение

- Важной составной частью реляционной модели является манипуляционный механизм, т. е. языки запросов, позволяющие извлекать и обновлять данные.
- Это реляционная алгебра и реляционное исчисление предложенные Э. Коддом в 1971 г.
- Неформально можно определить реляционную алгебру как высокоуровневый процедурный язык: она может использоваться для того, чтобы указать СУБД, как сформировать новое отношение из одного или более отношений в базе данных.
- Аналогично реляционное исчисление можно описать как непроцедурный язык: оно может использоваться для того, чтобы сформулировать определение нового отношение в терминах одного или более отношений, существующих в базе данных.
- Реляционная алгебра и реляционное исчисление формально эквивалентны друг другу, т. е. для каждого выражения реляционной алгебры существует эквивалентное выражение реляционного исчисления (и наоборот).

3.1. Введение (продолжение)

- Реляционная алгебра и реляционное исчисление являются формальными языками, они не являются дружественными к пользователю.
- Они имеют значение поскольку:
 - Иллюстрируют базовые операции, требующиеся от любого языка манипулирования данными (DML)
 - Выступают в роли стандарта для сравнения других реляционных языков
- Реляционное исчисление используется для оценки мощности реляционных языков. Язык, который позволяет получить любое отношение, которое можно получить с помощью реляционного исчисления, называется **реляционно-полным**.
- Реальные языки зачастую не только являются реляционно-полными, но и имеют большую выразительность, чем реляционная алгебра и реляционное исчисление, за счет наличия дополнительных операций, таких как вычисления, суммирование, упорядочивание.

3.2. Реляционная алгебра

- Реляционная алгебра — это коллекция *операций*, которые принимают отношения в качестве операндов и возвращают отношение в качестве результата.
- Результат выполнения одной операции может использоваться в качестве входных данных (операнда) другой операции. Поэтому возможно использование вложенных операций. Это свойство называется **замкнутостью (closure)**.
- Алгебра, определенная Э. Коддом, включала две группы по 4 операции:
- Традиционные операции с множествами — объединение (Union), пересечение (Intersection), разность (Set difference) и декартово произведение (Cartesian product)
- Специальные реляционные операции — сокращение (Restriction), или выборка (Selection), проекция (Projection), соединение (Join) и деление (Division)
- Эти операции можно разделить на базовые (выборка, проекция, декартово произведение, объединение и разность) и дополнительные (соединение, пересечение и деление).
- Операции выборки и проекции являются унарными (работают с одним отношением), остальные операции — бинарные (работают с двумя отношениями).
- Отметим, что все эти операции предназначены *только для чтения* (т. е. они «читают», но не обновляют свои операнды).

3.2.1. Модельная база данных

Таблица «Студенты» (students)

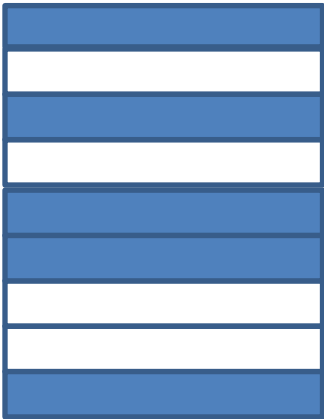
Номер зачетной книжки (record_book)	Ф. И. О. (name)	Серия паспорта (psp_ser)	Номер паспорта (psp_num)
55500	Иванов Иван Петрович	0402	645327
55800	Климов Андрей Иванович	0402	673211
55865	Новиков Николай Юрьевич	0202	554390

Таблица «Успеваемость» (progress)

Номер зачетной книжки (record_book)	Предмет (discipline)	Учебный год (acad_year)	Семестр (term)	Оценка (mark)
55500	Физика	2017/2018	1	5
55500	Математика	2017/2018	1	4
55800	Физика	2017/2018	1	4
55800	Физика	2017/2018	2	5

3.2.2. Выборка (Selection), или сокращение (Restriction)

- Эта операция выполняется на одном отношении R и определяет отношение, которое содержит только те кортежи исходного отношения R, которые удовлетворяют заданному условию (предикату).
- К. Дейт использует такое обозначение: $R \text{ WHERE } p$
- Некоторые другие авторы — такое: $\sigma_{\text{predicate}}(R)$
- Выражение p — это **предикат**, или **условие сокращения**



Примеры.

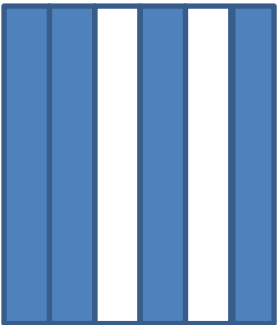
Students WHERE record_book = 55500

Students WHERE record_book = 55500 OR record_book = 55800

Progress WHERE acad_year = '2017/2018' AND term = 1

3.2.3. Проекция (Projection)

- Эта операция выполняется на одном отношении R , которое имеет атрибуты X, Y, \dots, Z (и, возможно, другие атрибуты) и определяет отношение, которое содержит вертикальное подмножество исходного отношения R , извлекая значения указанных атрибутов и отбрасывая дубликаты полученных кортежей. В таком случае **проекция** отношения R по атрибутам X, Y, \dots, Z определяется с помощью следующего выражения (К. Дейт): $\mathbf{R} \{ X, Y, \dots, Z \}$.
- Его заголовок формируется из заголовка отношения R путем удаления всех атрибутов, не указанных в множестве $\{ X, Y, \dots, Z \}$.
- Некоторые авторы используют обозначение: $\mathbf{\Pi}_{a_1, \dots, a_n}(R)$

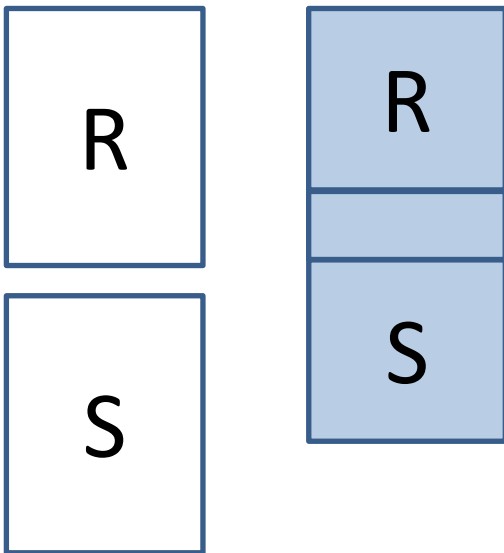


Пример.
 $\text{Progress} \{ \text{acad_year}, \text{term} \}$

Учебный год (acad_year)	Семестр (term)
2017/2018	1
2017/2018	2

3.2.4. Объединение (Union)

- Объединением отношений R и S является отношение того же типа с телом, которое состоит из всех кортежей, присутствующих в R или в S , или в обоих отношениях. При этом кортежи-дубликаты устраняются.
- Отношения R и S должны принадлежать к **одному типу**. Они должны быть совместимыми по объединению (union compatible), т. е. иметь одинаковые заголовки.
- К. Дейт использует такое обозначение: $R \text{ UNION } S$
- Некоторые другие авторы — такое: $R \cup S$



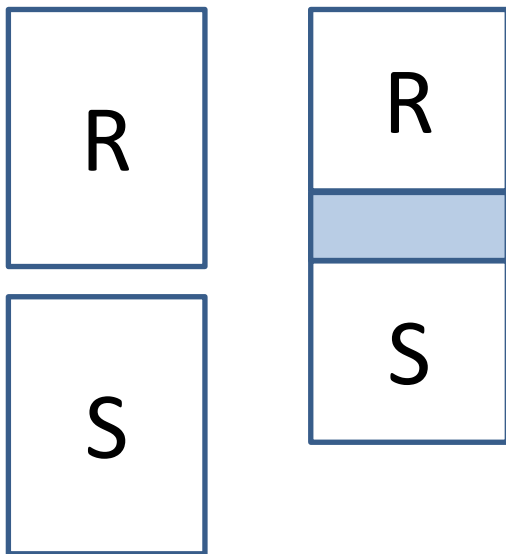
Пример.

`Students1 UNION Students2`

(предположим, что оба эти отношения имеют тот же тип, что и отношение `Students`)

3.2.5. Пересечение (Intersection)

- Пересечением отношений R и S является отношение того же типа с телом, которое состоит из всех кортежей, присутствующих одновременно в R и в S .
- Отношения R и S должны принадлежать к **одному типу**. Они должны быть совместимыми по объединению (*union compatible*), т. е. иметь одинаковые заголовки.
- К. Дейт использует такое обозначение: $R \text{ INTERSECT } S$
- Некоторые другие авторы — такое: $R \cap S$



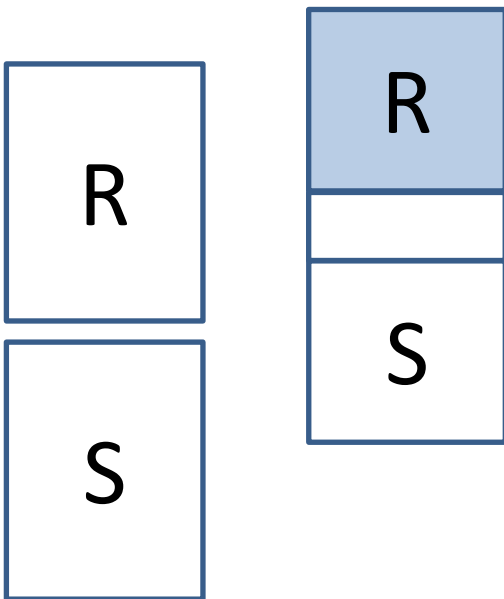
Пример.

`Students1 INTERSECT Students2`

(предположим, что оба эти отношения имеют тот же тип, что и отношение `Students`)

3.2.6. Разность (Set difference)

- Разностью отношений R и S (в указанном порядке) является отношение того же типа с телом, которое состоит из всех кортежей, присутствующих в R , но не присутствующих в S .
- Отношения R и S должны принадлежать к **одному типу**. Они должны быть совместимыми по объединению (union compatible), т. е. иметь одинаковые заголовки.
- Важно: оператор MINUS характеризуется направленностью (некоммутативностью), так же, как вычитание в обычной арифметике.



- К. Дейт использует такое обозначение:
 R MINUS S
- Некоторые другие авторы — такое:
 $R - S$

Пример.

Students1 MINUS Students2

(предположим, что оба эти отношения имеют тот же тип, что и отношение Students)

3.2.7. Декартово произведение (Cartesian product)

- Декартово произведение отношений R и S , не имеющих общих атрибутов – это отношение, заголовок которого представляет собой (теоретико-множественное) объединение заголовков отношений R и S , а тело состоит из всех кортежей t , таких, что t является (теоретико-множественным) объединением кортежа, принадлежащего к отношению R , и кортежа, принадлежащего к отношению S . Следует отметить, что кардинальность результата равна произведению кардинальностей входных отношений, R и S , а степень результата — сумме степеней входных отношений.
- Если необходимо сформировать декартово произведение двух отношений, имеющих общие имена атрибутов, то следует вначале воспользоваться оператором RENAME, чтобы переименовать атрибуты должным образом.
- К. Дейт использует такое обозначение: **R TIMES S**
- Некоторые другие авторы — такое: **R × S**

3.2.7. Декартово произведение (Cartesian product) (пример)

Отношение R

Last_name	First_name
Иванов	Петр
Петров	Иван

Отношение S

Discipline	Exam_date
Web-программирование	12.01.2018
Базы данных	15.01.2018
Язык C	21.01.2018

Отношение R × S

Last_name	First_name	Discipline	Exam_date
Иванов	Петр	Web-программирование	12.01.2018
Иванов	Петр	Базы данных	15.01.2018
Иванов	Петр	Язык C	21.01.2018
Петров	Иван	Web-программирование	12.01.2018
Петров	Иван	Базы данных	15.01.2018
Петров	Иван	Язык C	21.01.2018

3.2.8. Соединение (Join)

- Соединение равносильно выполнению операции выборки (selection) над результатом декартова произведения: $R \text{ JOIN } S = (R \text{ TIMES } S) \text{ WHERE } p$
- Имеются следующие виды соединений:
 - θ -соединение
 - Эквисоединение (частный случай θ -соединения)
 - Естественное соединение
 - Внешнее соединение
 - Полусоединение
- **θ -соединение** определяет отношение, которое содержит кортежи из декартова произведения отношений R и S , удовлетворяющие предикату p . Предикат p имеет форму $R.a_i \theta S.b_j$, где θ может являться одним из операторов сравнения ($<, \leq, >, \geq, =, \neq$), а $R.a_i$ и $S.b_j$ – атрибуты отношений R и S соответственно.
- **Эквисоединение** – это частный случай θ -соединения. Он имеет место, когда в качестве оператора сравнения используется только проверка на равенство ($=$).
- **Естественное соединение** – это частный случай эквисоединения. Он имеет место, когда отношения R и S соединяются по всем общим атрибутам. Один экземпляр каждого из общих атрибутов исключается из результирующего отношения.

3.2.8. Соединение (Join) (продолжение)

Естественное соединение

Таблицы «Студенты» (students) и «Успеваемость» (progress)

Общий атрибут – record_book

Students JOIN Progress эквивалентно следующему выражению:

```
( Students TIMES ( Progress RENAME record_book AS p_record_book )  
  WHERE record_book = p_record_book )
```

Степень результата естественного соединения равна сумме степеней отношений R и S минус число общих атрибутов этих отношений.

Номер зачетной книжки	Ф. И. О.	Серия паспорта	Номер паспорта	Предмет	Учебный год	Семестр	Оценка
55500	Иванов Иван Петрович	0402	645327	Физика	2017/2018	1	5
55500	Иванов Иван Петрович	0402	645327	Математика	2017/2018	1	4
55800	Климов Андрей Иванович	0402	673211	Физика	2017/2018	1	4
55800	Климов Андрей Иванович	0402	673211	Физика	2017/2018	2	5

3.2.8. Соединение (Join) (продолжение)

Внешнее соединение

Таблицы «Студенты» (students) и «Успеваемость» (progress)

Общий атрибут – record_book

Левое внешнее соединение отношений R и S – это соединение, в котором кортежи из R, которые не имеют соответствующих значений в общих атрибутах отношения S, также включаются в результирующее отношение. Отсутствующие значения атрибутов отношения S заменяются на NULL.

Номер зачетной книжки	Ф. И. О.	Серия паспорта	Номер паспорта	Предмет	Учебный год	Семестр	Оценка
55500	Иванов Иван Петрович	0402	645327	Физика	2017/2018	1	5
55500	Иванов Иван Петрович	0402	645327	Математика	2017/2018	1	4
55800	Климов Андрей Иванович	0402	673211	Физика	2017/2018	1	4
55800	Климов Андрей Иванович	0402	673211	Физика	2017/2018	2	5
55865	Новиков Николай Юрьевич	0202	554390				

Правое внешнее соединение определяется аналогично. **Полное внешнее соединение** сохраняет в результирующем отношении кортежи из обоих исходных отношений, заменяя отсутствующие значения на NULL.

3.2.8. Соединение (Join) (продолжение)

- **Полусоединение** определяет отношение, которое содержит кортежи из отношения R, которые участвуют в соединении R и S, удовлетворяющие предикату p.
- R SEMIJOIN S
- Полусоединение выполняет соединение двух отношений, а затем выполняет проекцию по атрибутам первого операнда.
- Таблицы «Студенты» (students) и «Успеваемость» (progress)
Общий атрибут – record_book
(Students TIMES (Progress RENAME record_book AS p_record_book)
WHERE record_book = p_record_book AND discipline = 'Математика')
{record_book, name, psp_ser, psp_name}

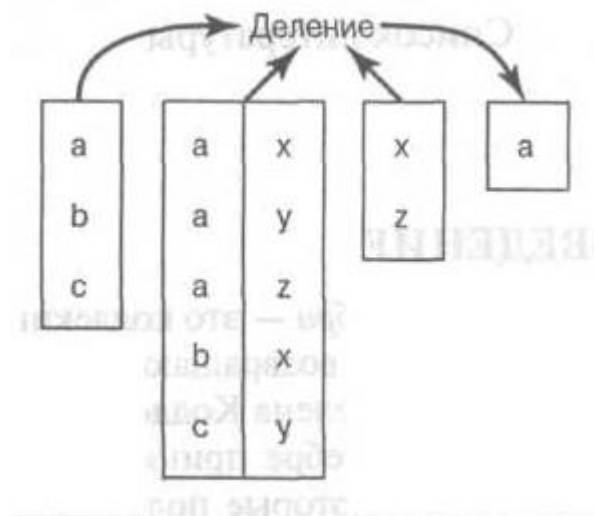
Номер зачетной книжки	Ф. И. О.	Серия паспорта	Номер паспорта
55500	Иванов Иван Петрович	0402	645327

3.2.9. Деление (Division)

- Предположим, что отношения R и S , соответственно, имеют следующие атрибуты: X_1, X_2, \dots, X_m и Y_1, Y_2, \dots, Y_n . Здесь ни один из атрибутов X_i ($i = 1, 2, \dots, m$) не имеет одинакового имени с любым из атрибутов Y_j ($j = 1, 2, \dots, n$). Пусть отношение T имеет следующие атрибуты: X_1, X_2, \dots, X_m и Y_1, Y_2, \dots, Y_n . Это означает, что T имеет заголовок, представляющий собой (теоретико-множественное) объединение заголовков отношений R и S . Будем рассматривать множества $\{X_1, X_2, \dots, X_m\}$ и $\{Y_1, Y_2, \dots, Y_n\}$, соответственно, как составные атрибуты X и Y . В таком случае операция деления R на S по T (где R — делимое, S — делитель, а T — посредник) может быть представлена с помощью следующего выражения:

R DIVIDEBY S PER T

Результат представляет собой отношение с заголовком $\{X\}$ и телом, состоящим из таких кортежей $\{X\ x\}$, присутствующих в R , что кортеж $\{X\ x, Y\ y\}$ присутствует в T для всех кортежей $\{Y\ y\}$, присутствующих в S . Иными словами, данный результат состоит из тех значений X , присутствующих в R , для которых соответствующие значения Y в T включают все значения Y из S .



3.2.10. Дополнительные операции

- Назначение операции **расширения** EXTEND состоит в поддержке вычислительных возможностей. Если быть точнее, то операция EXTEND принимает одно отношение и возвращает другое, идентичное заданному, если не считать того, что оно включает дополнительный атрибут, значения которого получены путем вычисления некоторого специального вычислимого выражения. Например, можно записать следующее выражение.
EXTEND P ADD (WEIGHT * 454) AS GMWT
- Данное выражение (следует подчеркнуть, что это — выражение, а не команда или оператор и поэтому может вкладываться в другие выражения) приводит к получению отношения с таким же заголовком, как P, не считая того, что оно содержит дополнительный атрибут с именем GMWT.
- Каждый кортеж этого отношения совпадает с соответствующим кортежем отношения P, за исключением того, что он дополнительно содержит значение веса в граммах GMWT, вычисленного с помощью заданного арифметического выражения WEIGHT * 454

P#	PNAME	COLOR	WEIGHT	CITY	GMWT
P1	Nut	Red	12.0	London	5448.0
P2	Bolt	Green	17.0	Paris	7718.0
P3	Screw	Blue	17.0	Oslo	7718.0
P4	Screw	Red	14.0	London	6356.0
P5	Cam	Blue	12.0	Paris	5448.0
P6	Cog	Red	19.0	London	8626.0

3.2.10. Дополнительные операции (продолжение)

- Операция **агрегирования** выполняет аналогичную функцию для так называемых *вертикальных вычислений*, или «вычислений, осуществляемых в пределах атрибута». Например, в результате вычисления следующего выражения

```
SUMMARIZE SP PER P { P# } ADD SUM ( QTY ) AS TOTQTY
```

формируется отношение с атрибутами P# и TOTQTY, в котором имеется по одному кортежу для каждого значения P# в проекции отношения P по атрибуту P#, содержащему это значение P# и соответствующее общее количество. Иными словами, отношение SP концептуально подразделяется на группы, или множества кортежей (где имеется по одной группе для каждого номера детали в отношении P), после чего каждая такая группа используется для выработки одного кортежа в составе общего результата

- Вообще говоря, значение следующей **операции агрегирования**
SUMMARIZE R PER S ADD summary AS Z
представляет собой отношение

P#	TOTQTY
P1	600
P2	1000
P3	400
P4	500
P5	500
P6	100

3.3. Реляционное исчисление

- Реляционное исчисление получило название от той части символической логики, которая называется *исчислением предикатов*. В контексте баз данных оно существует в двух формах: в форме предложенного Э. Коддом *реляционного исчисления кортежей* и в форме предложенного Лакруа и Пиро *реляционного исчисления доменов*.
- В логике первого порядка (или теории исчисления предикатов) под *предикатом* подразумевается истинностная функция с параметрами. После подстановки значений вместо параметров функция становится выражением, называемым *суждением*, которое может быть истинным или ложным.
- Например, предложения «Иван Петров является сотрудником данной организации» и «Иван Петров имеет более высокую зарплату, чем Петр Иванов» являются суждениями, поскольку можно определить их истинность или ложность. В первом случае функция «является сотрудником данной организации» имеет один параметр («Иван Петров»), а во втором случае функция «имеет более высокую зарплату, чем» имеет два параметра («Иван Петров» и «Петр Иванов»).

3.3. Реляционное исчисление (продолжение)

- Если предикат содержит переменную, например в виде « x является сотрудником этой организации», то у этой переменной должна быть соответствующая *область определения*. При подстановке вместо переменной x одних значений из ее области определения данное суждение может оказаться истинным, а при подстановке других — ложным.
- Если P — предикат, то множество всех значений переменной x , при которых P становится истинным, можно символически записать следующим образом:
$$\{x \mid P(x)\}$$
- Предикаты могут соединяться с помощью логических операций \wedge (AND), \vee (OR) и \sim (NOT) с образованием составных предикатов.

3.3. Реляционное исчисление (продолжение)

- Задача: Выбрать номера поставщиков и названия городов, в которых находятся поставщики детали с номером P2
- Реляционная алгебра:
 - Выполнить соединение отношений поставщиков S и поставок SP по атрибуту S#.
 - С помощью операции сокращения выделить из результатов этого соединения кортежи, которые относятся к детали с номером P2.
 - Сформировать проекцию результатов этой операции сокращения по атрибутам s# и CITY.
- Реляционное исчисление:
 - Получить атрибуты s# и CITY для таких поставщиков, для которых в отношении SP существует запись о поставке с тем же значением атрибута s# и со значением атрибута P#, равным P2.
- В реляционном исчислении просто указывается, в чем заключается проблема, тогда как в реляционной алгебре задается *процедура решения* этой проблемы. На самом деле *реляционная алгебра и реляционное исчисление логически эквивалентны.*

3.3.1. Реляционное исчисление кортежей

- В реляционном исчислении кортежей задача состоит в нахождении таких кортежей, для которых предикат является истинным. Это исчисление основано на *переменных кортежа*. Переменными кортежа являются такие переменные, областью определения которых служит указанное отношение.
- Например, для указания отношения Staff в качестве области определения переменной кортежа S используется следующая форма записи: Staff (S)
- Запрос «найти множество всех кортежей S, для которых F(S) является истинным» можно записать следующим образом: { S | F(S) }
- Здесь предикат F называется *формулой* (в математической логике, *правильно построенной формулой* — Well-Formed Formula, или сокращенно WFF).
- Например, запрос «Создайте список всех менеджеров, зарплата которых превышает 25000 рублей» можно записать следующим образом:
{ S.fName, S.lname | Staff(S) ∧ S.position = 'Manager' ∧ S.salary > 25000 }

3.3.1. Реляционное исчисление кортежей (продолжение)

- Для указания количества экземпляров, к которым должен быть применен предикат, в формулах могут использоваться два типа *кванторов*.
- **Квантор существования** \exists («существует») используется в формуле, которая должна быть истинной хотя бы для одного экземпляра, например:
 $\text{Staff}(S) \wedge (\exists B) (\text{Branch}(B) \wedge (B.\text{branchNo} = S.\text{branchNo}) \wedge B.\text{city} = \text{'London'})$
- Это выражение означает, что в отношении Branch существует кортеж, который имеет такое же значение атрибута branchNo, что и значение атрибута branchNo в текущем кортеже S из отношения Staff, а атрибут city из кортежа B имеет значение 'London'.
- **Квантор всеобщности** \forall , или так называемый символ «для всех», используется в выражениях, которые относятся ко всем экземплярам, например:
 $(\forall B) (B.\text{city} \neq \text{'Paris'})$
- Это выражение означает, что ни в одном кортеже отношения Branch значение атрибута city не равно 'Paris'.
- Переменные кортежа называются *свободными переменными*, если они не квалифицируются кванторами \forall или \exists ; в противном случае они называются *связанными переменными*.

3.3.2. Реляционное исчисление доменов

- В реляционном исчислении кортежей используются переменные, областью определения которых являются кортежи в отношении. С другой стороны, в реляционном **исчислении доменов** также используются переменные, но их значения берутся из области определения **атрибутов**, а не из кортежей отношения. Любое выражение в реляционном исчислении доменов имеет следующую общую форму: $\{ d_1, d_2, \dots, d_n \mid F(d_1, d_2, \dots, d_m) \}$ $m \geq n$
- Здесь d_1, d_2, \dots, d_n — переменные области определения (домена), а $F(d_1, d_2, \dots, d_m)$ — формула.
- Найдите имена всех менеджеров, зарплата которых превышает 25000 рублей:
 $\{ fN, IN \mid (\exists sN, posn, sex, DOB, sal, bN) (Staff(sN, fN, IN, posn, sex, DOB, sal, bN) \wedge posn = 'Manager' \wedge sal = 25000) \}$

Литература

1. Гарсиа-Молина, Г. Системы баз данных. Полный курс : пер. с англ. / Гектор Гарсиа-Молина, Джеффри Ульман, Дженнифер Уидом. – М. : Вильямс, 2003. – 1088 с.
2. Грофф, Дж. SQL. Полное руководство : пер. с англ. / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. – 3-е изд. – М. : Вильямс, 2015. – 960 с.
3. **Дейт, К. Дж. Введение в системы баз данных : пер. с англ. / Крис Дж. Дейт. – 8-е изд. – М. : Вильямс, 2005. – 1328 с.**
4. **Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика : пер. с англ. / Томас Коннолли, Каролин Бегг. – 3-е изд. – М. : Вильямс, 2003. – 1436 с.**
5. Кузнецов, С. Д. Основы баз данных : учеб. пособие / С. Д. Кузнецов. – 2-е изд., испр. – М. : Интернет-Университет Информационных Технологий ; БИНОМ. Лаборатория знаний, 2007. – 484 с.
6. Лузанов, П. PostgreSQL для начинающих / П. Лузанов, Е. Рогов, И. Лёвшин ; Postgres Professional. – М., 2017. – 146 с.
7. Моргунов, Е. П. Язык SQL. Базовый курс : учеб.-практ. пособие. / Е. П. Моргунов ; под ред. Е. В. Рогова, П. В. Лузанова ; Postgres Professional. – М., 2017. – 257 с.
8. PostgreSQL [Электронный ресурс] : официальный сайт / The PostgreSQL Global Development Group. – <http://www.postgresql.org>.
9. Postgres Professional [Электронный ресурс] : российский производитель СУБД Postgres Pro : официальный сайт / Postgres Professional. – <http://postgrespro.ru>.

Задание

Для выполнения практических заданий необходимо использовать книгу:

Моргунов, Е. П. Язык SQL. Базовый курс : учеб.-практ. пособие / Под ред. Е. В. Рогова, П. В. Лузанова ; Postgres Professional. – М., 2017. – 257 с.

<https://postgrespro.ru/education/books/sqlprimer>

1. Изучить материал главы 5. Запросы к базе данных выполнять с помощью утилиты psql, описанной в главе 2, параграф 2.2.