Резниченко Валерий Анатольевич
Организация баз данных и знаний

Лекция 7. Реляционное исчисление

Национальный авиационный университет
Факультет компьютерных наук
Кафедра инженерии программного обеспечения

СОДЕРЖАНИЕ

- Реляционные исчисления
- Исчисление Кодда
- Язык ALPHA
- Эквивалентность и полнота
- Примеры

Теория множеств и логика

Теория множеств (отношений)	Исчисление предикатов	Связь
Множество M x – переменная	P(x); P – предикатный символ	$x \in M \Leftrightarrow P(x) = t$
$M\ \theta\ N$ $\theta = \{\cup, \cap, \square, -\}$	$P(x)\ \theta\ Q(y)$ $\theta = \{\vee, \wedge, \neg, \rightarrow\}$	$x \in M\ \theta\ N \Leftrightarrow P(x)\ \theta\ Q(y) = t$

Теория множеств ⟺ Исчисление предикатов

↓ ↓

Реляционная алгебра ⟺ Реляционное исчисление

Пример связи ТМ и ИП: $T(A) = R(A) \square S(A) = \{t \mid t \in R \vee t \in S\}$

Реляционное исчисление

- Подмножество формул исчисление предикатов
- Формальное описание того, ЧТО следует получить из базы данных. Например:

  ЕЯ - "Выдать факультеты с фондом финансирования > 10000"

  РИ - **{t | (FAC(t) & t.fund > 10000}**

- Имеет средства языка запросов, но не обладает возможностями манипулирования данными (как и реляционная алгебра)

Два варианта реляционного исчисления:

- **Кортежное реляционное исчисление** (TRC) – переменные представляют строки отношений
- **Доменное реляционное исчисление** (DRC) - переменные представляют домены атрибутов отношений

Кортежное реляционное исчисление (TRC)

Запрос (в простейшем случае) имеет вид **{t | (F(t)}**

- t - кортежная переменная;
- F(t) - формула, в которой присутствует кортежная переменная t

Ответ: Множество всех таких кортежей t, для которых формула F(t) принимает значение true.

Формула: Рекурсивно определяется через атомарные формулы и построением более сложных формул с помощью логических связок и кванторов.

SQL: Является формальной основной языка SQL.

TRC - Базовые составляющие языка

- **Константы:** 7, 'john', 3.14159 и т.д.
- **Кортежные переменные:** t1, t2,... – интерпретируются кортежами отношений.
- **Срезы кортежных переменных:** t.a,… a - имя атрибута, значение которого входит в состав кортежа.
- **Предикатные символы:** P, P1,...,Q, Q1,… Интерпретируются отношениями
- **Операторы (предикаты) сравнения :** = {=, <, <=, >, >=, <>} и т.д.
- **Логические связки:** ∨ (или), ∧(&) (и), ¬ (не), → (влечет)
- **Кванторы:** ∃(существует), ∀(для любого)

TRC - Правильно определенные формулы

Атомарные формулы:

- **P(t)** - **P** - предикатный символ,
  - **t** - кортежная переменная.
    Если **P** интерпретируется отношением **R**, то **P(t)** означает **t ∈ R**
- **t.a θ t.b** - **t.a** и **t.b** - срезы
- **t.a θ c** - **t.a** - срез, **c** - константа

Правильно построенные формулы (ппф):

- Атомарные формулы являются ппф;
- Если **F** ппф, то ппф также являются **¬F** и **(F)**
- Если **F** и **G** ппф, то ппф также являются **F ∨ G, F ∧ G, F → G**
- Если F ппф со свободной переменной t, то ппф также являются **∃tF(t)** и **∀tF(t).**

TRC - Свободные и связанные переменные. Запросы.

Говорят, что наличие кванторов **∃tF(t)** и **∀tF(t).** связывает переменную t в формуле F. Переменная, которая не связана, называется ***свободной***.

Запрос - в общем случае это выражение вида:

{(t1,t2,...) | (F(t1,t2,...)}

где - t1,t2,... - кортежные переменные или их срезы;

- F(t1,t2,...) - формула, в которой присутствуют свободные кортежные переменные t1,t2,...

ВНИМАНИЕ: Переменные t1,t2,..., расположенные слева от символа '|', должны быть единственными свободными переменными в формуле F. Это значит, что если F содержит другие переменные, то они должны быть связанными с помощью кванторов.

Пример БД для запросов в РИ

FAC	FACULTY (FNo, Name, Dean, Bld, Fund)
DEP	DEPARTMENT (DNo, FNo, Name, Head, Bld, Fund)
TCH	TEACHER(TNo, DNo, Name, Post, Tel, Salary, Comm)
GRP	GROUP(GNo, DNo, Course, Num, Quantity, CurNo)
SBJ	SUBJECT(SNo, Name)
ROM	ROOM (RNo, Num, Building, Seats)
LEC	LECTURE (TNo, GNo, SNo, RNo, Type, Day, Week)

Предикатные символы

Отношения БД

TRC - Примеры проекции, селекции и соединения

1) Проекция

Запрос. Вывести названия факультетов и имена их деканов

{(f.Name, f.Dean)|FAC(f)}

2) Селекция и проекция

Запрос. Вывести имена и должности преподавателей, имеющих зарплату > 1000 и надбавку > 500

{(t.Name, t.Post) | TCH(t) & t.Salary > 1000 & t.Comm > 500}

3) Соединение, селекция и проекция

Запрос. Вывести имена факультетов и их кафедр, расположенных (факультетов) в корпусе 5

{(f.Name,d.Name) | FAC(f) & DEP(d) & f.FNo = d.FNo & f.Bld = 5}

TRC - Примеры кванторов существования

Запрос. Вывести имена факультетов корпуса 5 и их кафедр

{(f.Name,d.Name) | DEP(d) & FAC(f) & f.FNo = d.FNo & f.Bld = 5}

Запрос. Вывести имена кафедр факультетов корпуса 5

{d.Name | DEP(d) & ∃f(FAC(f) & f.FNo = d.FNo & f.Buld = 5)}

Запрос. Вывести имена преподавателей-профессоров, имеющих лекции в группах первого курса

{t.Name | TCH(t) & t.Post='professor' &
∃l(LEC(l) & l.TNo = t.Tno &
∃g(GRP(g) & l.GNo = g.GNo & g.Course = 1))}

Вывести имена профессоров,
для которых существуют такие лекции,
для которых (лекций) существуют группы первого курса

GRP **GROUP(GNo, Course...)** **TCH** **TEACHER (TNo, Name, Post...)**

LEC **LECTURE(GNo,TNo...)**

TRC - Примеры кванторов общности

Запрос. Вывести номера преподавателей, читающих лекции во всех группах

$$\{l.TNo \mid LEC(l) \;\&\; \forall g(GRP(g) \rightarrow g.GNo = l.GNo)\}$$

Запрос. Вывести номера преподавателей, которые читают лекции во всех группах первого курса

$$\{l.TNo \mid LEC(l) \;\&\; \forall g((GRP(g) \;\&\; g.Course = 1) \rightarrow g.GNo = l.GNo)\}$$

Запрос. Вывести имена преподавателей, которые читают лекции во всех группах первого курса

$$\{t.Name \mid TCH(t) \;\&\; \exists l(LEC(l) \;\&\; l.TNo = t.TNo \;\&\; \forall g((GRP(g) \;\&\; g.Course = 1) \rightarrow g.GNo = l.GNo)\}$$

Вывести имена преподавателей,

для которых существуют такие лекции,

которые читаются всем группам 1 курса

TRC - Безопасные формулы (запросы)

Можно формулировать запросы, содержащие ппф, но не имеющие интерпретации в БД. Или, другими словами, имеющие бесконечное к-во ответов (дающие в результате бесконечные отношения). Примеры:

$\{t \mid t.Post = 'professor'\}$
$\{t \mid \neg TCH(t)\}$
$\{(t,d) \mid TCH(t) \vee DEP(d)\}$

Запрос называется ***безопасным***, если он интерпретируется конечным отношением.

Для построения безопасных запросов:

- все переменные в формуле должны быть ограниченными;
- все логические связки должны быть ограниченными;
- все кванторы должны быть ограниченными.

TRC - Ограниченные переменные

Кортежная переменная ***ограничена***, если она принадлежит предикату, который интерпретируется отношением БД, или если она ограничена другой переменной или константой.

Переменная t ограниченна, если:

- она появляется в предикате P(t), где P интерпретируется отношением БД;
- она появляется в формуле t.a1 = c1 & ... & t.an = cn, где a1,..., an - все атрибуты корежа t, а c1, ..., cn - константы;
- она появляется в формуле t = s, где s – ограниченная переменная.

Примеры:

P(t) - переменная t ограничена предикатом P
P(t) & t = x - переменная x ограничена ограниченной переменной t
t.Name='john' & t.Post='prof' & t.Salary = '1200'

Ограниченные логические связки

Если две формулы F и G имеют ограниченные переменные, то:

- F ∨ G будет являться формулой с ограниченными переменными, если F и G имеют одинаковые свободные переменные;
- F & G всегда является формулой с ограниченными переменной не зависимо от того, совпадают ли в этих формулах переменные;
- F & ¬G является формулой с ограниченной переменной, если в этих формулах совпадают свободные переменные;
- F → G никогда не является формулой с ограниченной переменной.

Примеры:

P(t) ∨ Q(t) – формула с ограниченной переменной
P(t) ∨ Q(q) – переменные в результирующей формуле не ограничены
P(t) & Q(q) – формула с ограниченной переменной
¬Q(t) – не является формулой с ограниченной переменной
P(t) & ¬Q(q) – не является формулой с ограниченной переменной
P(t) & ¬Q(t) – формула с ограниченной переменной

Ограниченные кванторы

Если F(t) - ограниченная формула а G(t) – произвольная формула, то формула $\exists t(F(t) \& G(t))$ называется формулой с ***ограниченным квантором существования.*** Сама формула интерпретируется ограниченным отношением.

Если F(t) - ограниченная формула, а G(t) – произвольная формула, то формула $\forall t(F(t) \rightarrow G(t))$ называется формулой с ***ограниченным квантором общности.*** Сама формула интерпретируется ограниченным отношением

Примеры:

$\exists x(x.Fund < 5)$ – квантор существования не ограничен

$\exists x(P(x) \& x.Fund < 5)$ – квантор существования ограничен

$\forall x(P(x)) \rightarrow Q(x, y))$ – квантор общности ограничен

Доменное реляционное исчисление (DRC)

Запрос имеет вид **{x1,x2,...,xn | (F(x1,x2,...,xn)}**

- x1,x2,...,xn - доменные переменные;
- F(x1,x2,...,xn) - формула, в которой единственными свободными переменными являются x1,x2,...,xn

Ответ. Множество всех таких кортежей <c1,c2,...,cn>, для которых формула F принимает значение true.

Формула. Рекурсивно определяется через атомарные формулы и построением более сложных формул с помощью логических связок и кванторов **точно так же, как и в TRC**.

QBE. Является формальной основной языка QBE.

Примеры запросов в DRC

1) Проекция

Запрос. Вывести названия факультетов и имена их деканов

$$\{(y, z) \mid \exists x \exists u \exists v FAC(x,y,z,u,v)\}$$

2) Селекция и проекция

Запрос. Вывести названия факультетов корпуса 5 и их деканов

$$\{(y, z) \mid \exists x \exists u \exists v FAC(x,y,z,u,v) \ \&\ u = 5\}$$

3) Соединение, селекция и проекция

Запрос. Вывести имена факультетов корпуса 5 и имена их кафедр

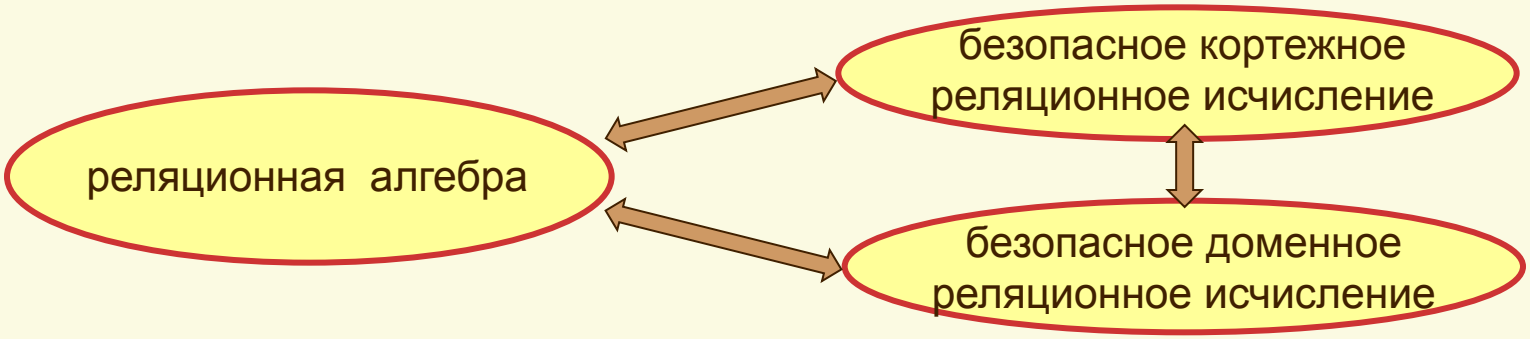
$$\{(y,n) \mid \exists x \exists f(\exists z \exists u \exists v(FAC(x,y,z,u,v) \ \&\ u=5) \ \&\ \exists d \exists h \exists b \exists u DEP(d,f,n,h,b,u) \ \&\ x=f)\}$$

FAC (FNo,Name,Dean,Bld,Fund) **DEP(DNo,FNo,Name,Head,Bld,Fund)**

x y z u v **d f n h b u**

Эквивалентность RA, TRC, DRC и реляционная полнота.

Утверждение: Реляционная алгебра, безопасное кортежное реляционное исчисление и безопасное доменное реляционное исчисление являются эквивалентными.



Тезис (о реляционной полноте): Язык реляционной модели является реляционно полным, если он обладает селективными возможностями реляционной алгебры.

Утверждение: Язык SQL является реляционное полным.

Реляционное исчисление Кодда (CRC)

- Является подмножеством исчисления предикатов (1-го порядка)
- Является кортежно-ориентированным
- Решает проблему различия между ппф и безопасными формулами
- Запросы являются безопасными
- Является эквивалентным реляционное алгебре; значит обладает реляционной полнотой

CRC – основные составляющие

- **Константы:** 7, 'john', 3.14159 и т.д.
- **Кортежные (строковые) переменные:** t1, t2,... – интерпретируются кортежами отношений.
- **Срезы кортежных переменных:** t.[i],…
  i – компонента кортежной переменной.
- **Предикатные символы:** P, P1,...,Q, Q1,…
  Интерпретируются отношениями
- **Операторы (предикаты) сравнения :**
  = {=, <, <=, >, >=, <>} и т.д.
- **Логические связки:** ∨ (или), ∧(&) (и), ¬ (не), → (влечет)
- **Кванторы:** ∃(существует), ∀(для любого)

CRC - правильно определенные формулы

- Термы:
  - **P.t** – терм значений: P - предикат, t - кортежная переменная.
    Если P интерпретируется отношением R, то
    P.t означает t ∈ R
  - **t[i] θ s[j], t[i] = c** - термы соединений

    Примеры: $t_1[3] = t_2[1]$; $t_4[7] = 15$
- Формулы, правильно определенные над строковой переменной:
  - Все входящие в формулу предикатные символы интерпретируются совместимыми по объединению отношениями.
  - В формулу входят только термы значений с единственной переменной.
  - Термы значений соединяются связками ∨, &, ¬. Причем, связке ¬ непосредственно предшествует связка &.

  Примеры формул, правильно определенных над переменной t.

  $P_1.t \vee P_2.t \vee (P_3.t \,\&\, P_4.t)$; $(P_1.t \vee P_2.t) \,\&\, \neg P_3.t$.

Формула, правильно определенная на кванторах

Пусть F – формула, правильно определенная над строковой переменной t, а G – формула, содержащая t в качестве свободной переменной, но не содержащая термов значений вида P.t. Тогда формулы:

$\exists t(F \,\&\, G)$; $\forall t(F \rightarrow G)$

называются ***правильно определенными на кванторах***.

По сути, это частный случай ограниченных кванторов существования и общности. Здесь формула F указывает ту область, в пределах которой изменяется переменная, используемая в кванторах.

Будем записывать эти формулы следующим образом:

$\exists F(G)$ и $\forall F(G)$

Примеры:

$\exists t((P.t \vee Q.t) \,\&\, \neg S.t) \,\&\, t[2] = \text{‘john’} \,\&\, t[5] = 1000)$

$\forall t((P.t \vee Q.t) \,\&\, \neg S.t) \rightarrow (t[2] = \text{‘john’} \,\&\, t[5] = 1000))$

Формула с областью определения

Формула W называется ***формулой с областью определения***, если она имеет вид:

W = U1 & U2 &…& Un & V, n ≥ 1

и обладает следующими свойствами:

- Каждая формула Ui является правильно определенной над кортежной переменной ti.
- Формула V либо пуста, либо обладает следующими свойствами:
  - Если в ней содержатся кванторы, то они правильно определены.
  - Каждая свободная переменная из V является одной из переменных формул U1, U2,…, Un
  - В V нет термов значений, содержащих свободные переменные.

Примеры: **P.r & Q.s & R.t** - декартово произведение

P.r & (r[3] = b) - селекция

P.r & Q.s & (r [2] = s[1]) - соединение

Альфа-выражения

Выражение (t1, t2,…, tk) : W называется ***простым альфа-выражением***, если выполнены следующие условия:

- W – формула с областью определения.
- t1, t2,…, tk – последовательность кортежных переменных и срезов. Эти переменные должны быть свободными в W.

Произвольное ***альфа-выражение*** определяется рекурсивно следующим образом:

- Каждое простое альфа-выражение является альфа-выражением.
- Пусть t = (t1, t2,…, tk). Если t : W1 и t : W2 – альфа-выражения, то выражения t : W1 ∨ W2, t : W1 & W2 и t : W1 & ¬W2 являются альфа-выражениями.

Язык ALPHA

Упрощенный синтаксис:

RANGE <relation name> <variable name> [SOME | ALL]

…

GET <workspace> (<target list>) : <wff>

range – указание имени кортежной переменной **<variable name>**, которая принимает значения из отношения **<relation name>.**

SOME | ALL – указывают, следует ли интерпретировать переменную с квантором существования или общности.

workspace – идентификатор, который именует временное рабочее отношение, содержащее результат выполнения команды Get.

target list – целевой список кортежных переменных и их срезов, указывающих столбцы, которые проецируются в результирующее отношение.

wff – правильно построенная формула кортежного реляционного исчисления

Примеры запросов в ALPHA и CRC

Запрос. Вывести названия факультетов и их деканов

CRC: **{f[2], f[3] | FAC.f}**
ALPHA: **RANGE FACULTY f**
GET W(f.Name, f.Dean)

Запрос. Вывести имя декана факультета CSF

CRC: **{f[3] | FAC.f & f[2] = 'CSF'}**
ALPHA: **RANGE FACULTY f**
GET W(f.Dean) : f.Name = 'CSF'

Запрос. Вывести названия кафедр факультета CSF

CRC: **{d[3] | DEP.d & ∃f(FAC.f & f[2] = 'CSF' & f[1] = d[2])}**
ALPHA: **RANGE FACULTY f SOME**
RANGE DEPARTMENT d
GET W(d.Name) = f.Name = 'CSF' & f.FNo = d.DNo

Заключение

- Реляционное исчисление является декларативным (не процедурным) языком, в котором запросы формулируются в терминах того, ЧТО требуется, а не КАК вычислить требуемые значения.
- (Безопасное) реляционное исчисление и реляционная алгебра имеют одинаковую выразительную силу, и на их основании формулируется тезис о реляционной полноте.
- Имеется возможность формулировать правильные, но не эффективные запросы. В связи с этим необходимы оптимизаторы.
- Имеются языки непосредственно реализующие реляционное исчисление (ALPHA, QUEL), а также базирующиеся на кортежном исчислении (SQL) и доменном исчислении (QBE).