Banco de Dados

Cálculo Relacional de Tuplas

João Eduardo Ferreira
Osvaldo Kotaro Takai
Marcelo Finger

Introdução

- O Cálculo Relacional de Tuplas (CRT) é uma alternativa à Álgebra Relacional (AR).
- A AR é procedimental, o CRT é declarativa:
  - O CRT permite descrever um conjunto de respostas sem explicitar como elas serão computadas.
- O CRT influenciou fortemente as linguagens de consulta comerciais, tais como a SQL.
- Uma linguagem de consulta L é considerada **relacionalmente completa** se L expressar qualquer consulta que possa ser realizada em CRT.

Introdução

- A consulta em CRT tem a forma:

$$\{ t \mid P(t) \}$$

  - { t | P(t) } representa o conjunto de todas as tuplas t, tal que o predicado P é verdadeiro para t.
  - t é uma variável de tuplas.
  - P é uma expressão condicional.
  - t.A ou t[A] denota o valor do atributo A da tupla t.

Exemplo

- Exemplo de uma consulta em CRT:
  - Obter todos os empregados cujo salário é acima de 50 mil:

    $\{ t \mid \text{EMPREGADO}(t) \text{ AND } t.\text{SALARIO} > 5000 \}$
  - EMPREGADO(t) é o mesmo que $t \in$ EMPREGADO.
  - A consulta acima resulta em uma relação que contém todas tuplas t da relação EMPREGADO, que satisfaça a condição t.SALARIO > 5000.

Exemplo

- Para recuperar apenas os atributos PNOME e SNOME dos empregados cujo salário é acima de 50 mil, escrevemos:

  { t.PNOME, t.SNOME | EMPREGADO(t) AND t.SALARIO > 5000 }

- No CRT especificamos primeiro os atributos desejados (t.PNOME e t.SNOME), da tupla selecionada t.
- Depois, estabelecemos a condição para selecionar uma tupla após a barra (|).

Expressões e Fórmulas

- Uma expressão geral do CRT é da forma:

  $\{ t_1.A_j, t_2,A_k, ..., t_n.A_m \mid P(t_1, t_2, ..., t_n, t_{n+1}, t_{n+2}, ..., t_{n+m}) \}$

- Onde:
  - $t_1, t_2, ..., t_n, t_{n+1}, t_{n+2}, ..., t_{n+m}$ são variáveis de tuplas.
  - A_i é um atributo correspondente à tupla associada.
  - P refere-se a uma **condição** ou **fórmula**.

Expressões e Fórmulas

- Uma fórmula é feita de átomos que podem ser:
  - $R(t_i)$, onde R é a relação e t_i é uma variável de tupla.
  - $t_i.A$ **op** $t_j.B$, onde **op** $\in \{=, <, \leq, >, \geq, \neq\}$
  - $t_i.A$ **op** c ou c **op** $t_j.B$, onde c é um valor constante.

Expressões e Fórmulas

- Cada átomo resulta em valor TRUE ou FALSE.
- Para átomos da forma R(t), se t $\in$ R, então é TRUE, senão é FALSO.
- Uma fórmula pode ser composta por um ou mais átomos conectados pelos operadores lógicos AND, OR e NOT.
- A implicação também pode ser usada ($\Rightarrow$):
  - $X \Rightarrow Y \equiv (\text{NOT } X) \text{ OR } Y$
- A dupla implicação também pode ser usada ($\Leftrightarrow$):
  - $X \Leftrightarrow Y \equiv (X \Rightarrow Y) \text{ AND } (Y \Rightarrow X)$

Quantificadores Universais e Existenciais

- Uma fórmula pode possuir quantificadores:
  - $\forall$ - Quantificador Universal
    - Para todo
    - Qualquer que seja
  - $\exists$ - Quantificador Existencial
    - Existe ao menos um.

- t_1 e t_2, nas cláusulas $\forall t_1$ ou $\exists t_2$, são variáveis de tupla vinculadas.
- Se t não for vinculada, então será livre.

Definição Geral e Recursiva de Expressões e Fórmulas

- Todo átomo é uma fórmula.
- Se F_1 e F_2 são fórmulas, então
  - F_1 AND F_2, F_1 OR F_2, NOT(F_1) e NOT(F_2) são fórmulas.
- Se F é fórmula, então $(\exists t)(F(t))$, também será.
  - $(\exists t)(F(t))$ será TRUE se F for TRUE para pelo menos uma tupla t.
- Se F é fórmula, então $(\forall t)(F(t))$, também será.
  - $(\forall t)(F(t))$ será TRUE se F for TRUE para todas as tuplas t no universo.

Transformações

- $F_1 \Rightarrow F_2 \equiv$ NOT F_1 OR F_2
- F_1 AND $F_2 \equiv$ NOT(NOT F_1 OR NOT F_2)
- $(\forall\ t)$ (F (t)) $\equiv$ NOT $(\exists\ t)$ (NOT F (t))
- $(\exists\ t)$ (F (t)) $\equiv$ NOT $(\forall\ t)$ (NOT F (t))
- $(\forall\ t)$ (F_1 (t) AND F_2(t)) $\equiv$ NOT $(\exists\ t)$ (NOT (F_1 (t)) OR NOT (F_2 (t)))
- $(\forall\ t)$ (F_1 (t) OR F_2(t)) $\equiv$ NOT $(\exists\ t)$ (NOT (F_1 (t)) AND NOT (F_2 (t)))
- $(\exists\ t)$ (F_1 (t) AND F_2(t)) $\equiv$ NOT $(\forall\ t)$ (NOT (F_1 (t)) OR NOT (F_2 (t)))
- $(\exists\ t)$ (F_1 (t) OR F_2(t)) $\equiv$ NOT $(\forall\ t)$ (NOT (F_1 (t)) AND NOT (F_2 (t)))

9. $(\forall\ t)$ (F (t)) $\Rightarrow$ $(\exists\ t)$ (F (t))

- NOT $(\exists\ t)$ (F (t)) $\Rightarrow$ NOT $(\forall\ t)$ (F (t))

Ver página 253 do livro de Elmasri & Navathe Terceira Ed.

Exemplo de Projeção

- Recupere o nome e o endereço de todos os empregados.
  - Em Álgebra Relacional:
    - $\pi_{\text{SNOME, PNOME, SALÁRIO}}$ (EMPREGADO)
  - Em CRT:
    - { t.PNOME, t.SNOME, t.ENDERECO | EMPREGADO(t) }

Exemplo de Seleção

- Recupere todos os empregados do sexo feminino.
  - Em Álgebra Relacional:
    - $\sigma_{\text{sexo='F'}}$ (EMPREGADO)
  - Em CRT:
    - { t | EMPREGADO(t) AND t.SEXO='F') }

Exemplo de Join

- Recupere o nome e o endereço de todos os empregados que trabalham para o departamento ‘Pesquisa’.
  - Em Álgebra Relacional
    - DEP $\leftarrow \sigma$ DNOME = 'Pesquisa' (DEPARTAMENTO)
    - EMPDEP$\leftarrow$(DEP $\bowtie_{\text{DNÚMERO = NDEP}}$ EMPREGADO)
    - RESULT $\leftarrow \pi_{\text{PNOME, SNOME, ENDEREÇO}}$ (EMPDEP)
  - Em CRT
    - { t.PNOME, t.SNOME, t.ENDERECO | EMPREGADO(t) AND
      ($\exists$d) (DEPARTAMENTO(d) AND
      d.DNOME=‘Pesquisa’ AND d.DNUMERO=t.DNO) }

Exemplo de Duplo Join

- Para todos os projetos localizado em Houston, liste o número do projeto, o número do departamento que o controla e o nome do seu gerente:
  - { p.PNUMERO, p.DNUM, m.PNOME |
    PROJETO(p) AND EMPREGADO(m) AND
    p.PLOCALIZACAO = ‘Houston’ AND
    (∃ d) (DEPARTAMENTO(d) AND
    d.DNUMERO = p.DNUM AND
    d.GERNSS = m.NSS) }

Exemplo de Duplo Join

- Se exemplo anterior, trocarmos p.DNUM por d.DNUMERO na saída da consulta, podemos eliminar o quantificador existencial:
  - { p.PNUMERO, d.DNUMERO, m.PNOME |
    PROJETO(p) AND
    EMPREGADO(m) AND
    DEPARTAMENTO(d) AND
    p.PLOCALIZACAO = ‘Houston’ AND
    d.DNUMERO = p.DNUM AND
    d.GERNSS = m.NSS }

Outro Exemplo de Duplo Join

- Liste o nome dos empregados que trabalham em **algum** projeto controlado pelo departamento 5:
  - { e.PNOME, e.SNOME |
    EMPREGADO(e) AND
    (∃ p, w) (PROJETO(p) AND TRABALHA-EM(w) AND
    p.DNUM = 5 AND
    w.ENSS = e.NSS AND
    p.PNUMERO = w.PNO) }

Exemplo de União

- Listar os nomes de projetos em que o empregado de sobrenome Smith trabalhe ou que sejam controlados por algum departamento gerenciado pelo empregado de sobrenome Smith:
  - { p.PNUMERO | PROJETO(p) AND
    ((∃ e, w)(EMPREGADO(e) AND TRABALHA-EM(w) AND
    w.PNO=p.PNUMERO AND
    e.SNOME='Smith' AND e.NSS=w.ENSS)

OR

(∃ m, d)(EMPREGADO(m) AND DEPARTAMENTO(d) AND
p.DNUM=d.DNUMERO AND
d.GERNSS=m.NSS AND
m.SNOME='Smith‘)) }

Exemplo de Join de uma Relação com ela mesma

- Listar o nome de cada empregado e o nome do seu supervisor imediato:
  - { e.PNOME, s.PNOME |
    EMPREGADO(e) AND
    EMPREGADO(s) AND
    e.NSSSUPER = s.NSS }

Exemplo de Divisão

- Liste o nome de todos os empregados que trabalham em todos os projetos:
  - Em Álgebra Relacional
    - PROJNSS(PNRO, NSS) $\leftarrow \pi_{\text{PNRO, NSSEMP}}$ (TRABALHA_EM)
    - PROJS(PNO)$\leftarrow \pi_{\text{PNUMERO}}$ (PROJETO)
    - EMP $\leftarrow$ PROJNSS $\div$ PROJS
    - RESULTADO $\leftarrow \pi_{\text{PNOME}}$ (EMP * EMPREGADO)
  - Em CRT
    - { e.PNOME | EMPREGADO(e) AND
(∀ p)(PROJETO(p) ⇒
(∃ w)(TRABALHA-EM(w) AND
w.PNO=p.PNUMERO AND
w.ENSS=e.NSS)) }

Um melhor exemplo mais did

Entendendo o $(\forall t)(F(t))$

- Liste o nome dos empregados que não tenham dependentes:
  - { e.PNOME | EMPREGADO(e) AND
    NOT (∃ d)(DEPENDENTE(d) AND e.NSS = d.ENSS) }
  - Ou **NOT (∃ t)(NOT F (t)) ≡ (∀ t) (F (t))**
  - { e.PNOME | EMPREGADO(e) AND
    (∀ d)(NOT (DEPENDENTE(d) AND e.NSS = d.ENSS)) }
  - Ou **NOT (F1 AND F2) ≡ NOT F1 OR NOT F2**
  - { e.PNOME | EMPREGADO(e) AND
    (∀ d)(NOT DEPENDENTE(d) OR e.NSS <> d.ENSS) }
  - Ou **NOT F1 OR F2 ≡ F1 ⇒ F2**
  - { e.PNOME | EMPREGADO(e) AND
    (∀ d)(DEPENDENTE(d) ⇒ e.NSS <> d.ENSS) }

Entendendo o (∀t)(F(t))

- Para analisar o último caso:
  - { e.PNOME | EMPREGADO(e) AND
    (∀ d)(DEPENDENTE(d) ⇒ e.NSS <> d.ENSS) }

- Suponha que EMPREGADO e DEPENDENTE fossem as seguintes relações:

EMPREGADO	Nome	NSS
	a1	b1
	a2	b2
	a3	b3
	a4	b4

DEPENDENTE	ENSS	NOME
	b1	c1
	b3	c3

Entendendo o (∀t)(F(t))

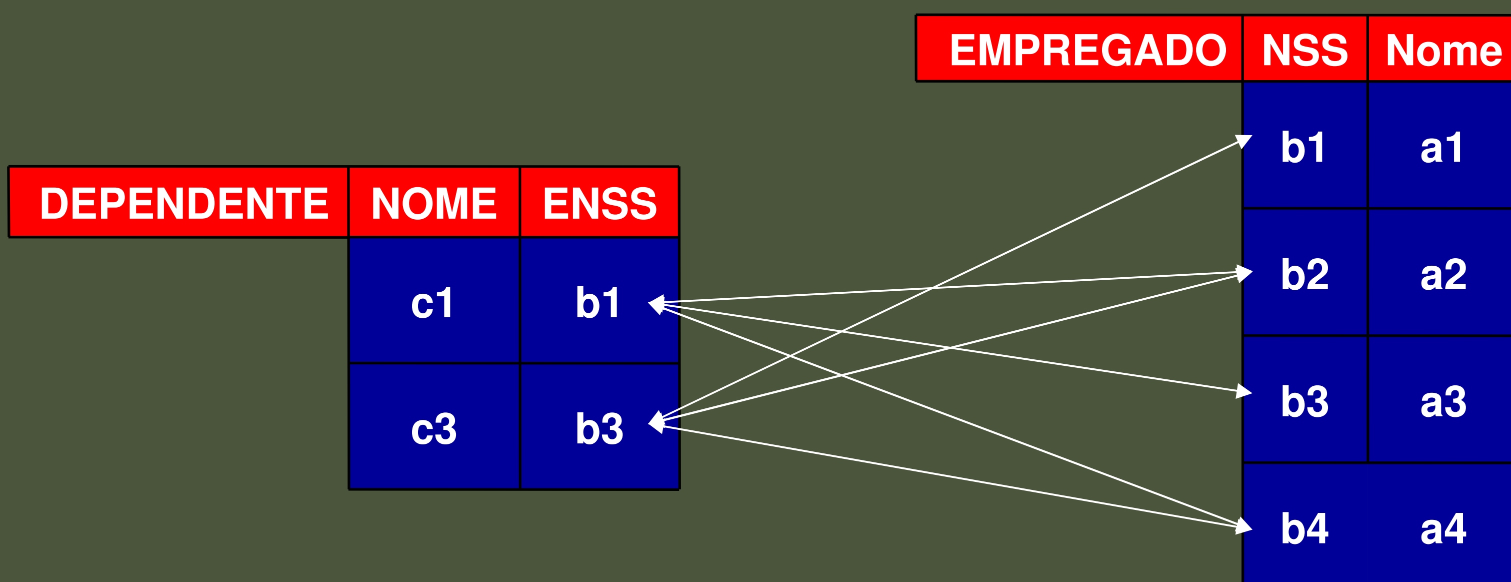
- Sabemos que para (∀ d)(DEPENDENTE(d) ⇒ e.NSS <> d.ENSS) seja verdade, basta que e.NSS <> d.ENSS seja verdade, pois DEPENDENTE(d) sempre será verdade para todo d:

P DEPENDENTE(d)	q e.NSS <> d.ENSS	P ⇒ Q
V	V	V
V	F	F
F	V	V
F	F	V

Esses casos nunca irão ocorrer

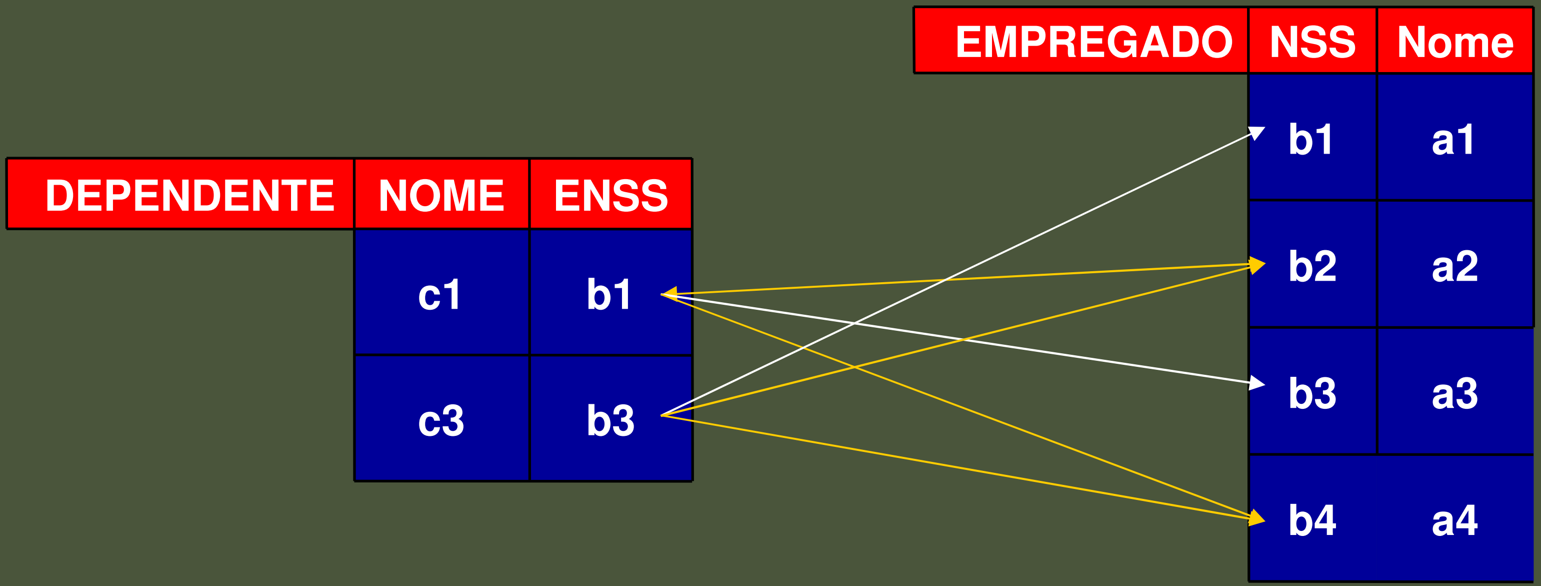
Entendendo o (∀t)(F(t))

- Agora, basta saber se e.NSS <> d.ENSS é verdade **para todo d** de DEPENDENTE(d).
- Assim, vamos analisar para cada dependente d, os possíveis resultados de e.NSS <> d.ENSS:



Entendendo o $(\forall t)(F(t))$

- Note que as únicas associações que valem **para todos** os dependentes são as tuplas de EMPREGADO apontados pelas setas amarelas:



Entendendo o (∀t)(F(t))

- Portanto, o resultado da consulta:
  - { e.PNOME | EMPREGADO(e) AND
    (∀ d)(DEPENDENTE(d) ⇒ e.NSS <> d.ENSS) }
- Será a relação que contém os nomes das tuplas de empregados, e, os quais não possuem dependentes:

Nome
a2
a4

Um outro Exemplo de Divisão

- Encontrar os nomes de empregados que trabalham em todos os projetos controlados pelo departamento 5:
  - { e.PNOME, e.SNOME | EMPREGADO(e) AND

    (∀ x) ((PROJETO(x) AND x.DNUM=5) ⇒

    (∃ w) (TRABALHA-EM(w) AND

    w.ENSS=e.NSS AND

    x.PNUMERO=w.PNO)) };
- Aqui, o lado esquerdo da implicação (⇒) restringe os projetos, x, do departamento 5.

Um outro Exemplo de Divisão

- Assim, PROJETO(x) AND x.DNUM=5 é:

O nosso Universo

PNOME	PNÚMERO	PLOCALIZAÇÃO	DNUM
ProdutoX	1	Bellaire	5
ProdutoY	2	Sugarland	5
ProdutoZ	3	Houston	5
Automação	10	Stafford	4
Reorganização	20	Houston	1
Beneficiamento	30	Stafford	4

- Apenas para facilitar o nosso entendimento, vamos chamar essa relação, com o universo de tuplas válidas, de PROJDEP5.

Um outro Exemplo de Divisão

- Agora, o lado direito a implicação:

  (∃ w) (TRABALHA-EM(w) AND w.ENSS=e.NSS AND x.PNUMERO=w.PNO)

  pode ser analisada para cada tupla x de PROJDEP5 em relação às tuplas de TRABALHA-EM e EMPREGADO.
- Novamente, para facilitar, eliminamos atributos desnecessários para a consulta.

(∃ w) (TRABALHA-EM(w) AND w.ENSS=e.NSS AND x.PNUMERO=w.PNO)

PROJDEP5(x)

PNOME	PNÚMERO
ProdutoX	1
ProdutoY	2
ProdutoZ	3

TRABALHA-EM (w)

PNRO	NSSEMP
1	123456789
2	123456789
3	666884444
1	453453453
2	453453453
2	333445555
3	333445555
10	333445555
20	333445555
30	999887777
10	999887777
10	987987987
30	987987987
30	987654321
20	987654321
20	888775555
3	123456789

EMPREGADO (e)

NSS	PNOME	SNOME
123456789	John	Smith
333445555	Franklin	Wong
999887777	Alícia	Zelaya
987654321	Jennifer	Wallace
666884444	Ramesh	Narayan
453453453	Joyce	English
987987987	Ahmad	Jabbar
888665555	James	Borg

Logo, existe (∃) um w que satisfaz a todos os projetos do departamento 5:

- **O John Smith.**

Expressões Seguras

- Uma expressão em CRT pode gerar uma infinidade de relações.
  - Por exemplo, a expressão

    {t | NOT (R(t))}

  pode gerar uma infinidade de tuplas que não estão em R.

- Assim, quando escrever uma consulta em CRT, verifique ela é **segura**.
  - Uma expressão segura no CRT é uma expressão que garante a produção de um número finito de tuplas como resultado.

Questões

- Estude os exemplos de consulta em Cálculo Relacional de Tuplas da pág. 64 da apostila.
- Refaça as consultas de álgebra relacional utilizando, agora, cálculo relacional.

Sugestão: Utilize o WinRDBI para validar as consultas (http://www.eas.asu.edu/~winrdbi/).