

# 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:

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

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

# Exemplo

---

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

$\{ t.PNOME, t.SNOME \mid EMPREGADO(t) \text{ AND } t.SALARIO > 5000 \}$

- ❑ 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 ( $\mid$ ).

# Expressões e Fórmulas

---

- Uma expressão geral do CRT é da forma:

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

- Onde:

- $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$  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.
  - $t_i.A \text{ op } t_j.B$ , onde  $\text{op} \in \{=, <, \leq, >, \geq, \neq\}$
  - $t_i.A \text{ op } c$  ou  $c \text{ op } t_j.B$ , 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
  - $F_1 \text{ AND } F_2$ ,  $F_1 \text{ OR } F_2$ ,  $\text{NOT}(F_1)$  e  $\text{NOT}(F_2)$  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

---

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

# Exemplo de Projeção

---

- ❑ Recupere o nome e o endereço de todos os empregados.

- Em Álgebra Relacional:

- ❑  $\pi_{\text{SNOME, PNO ME, SALÁRIO}}(\text{EMPREGADO})$

- Em CRT:

- ❑  $\{ \text{ t.PNO ME, t.SNO ME, t.ENDERE CO } \mid \text{ EMPREGADO}(t) \}$

# Exemplo de Seleção

---

- ❑ Recupere todos os empregados do sexo feminino.

- Em Álgebra Relacional:

- ❑  $\sigma_{\text{SEXO}='F'}(\text{EMPREGADO})$

- Em CRT:

- ❑  $\{ t \mid \text{EMPREGADO}(t) \text{ AND } t.\text{SEXO}='F' \}$

# Exemplo de Join

---

- ❑ Recupere o nome e o endereço de todos os empregados que trabalham para o departamento 'Pesquisa'.

- Em Álgebra Relacional

- ❑  $DEP \leftarrow \sigma_{DNOME = 'Pesquisa'}(DEPARTAMENTO)$
- ❑  $EMPDEP \leftarrow (DEP \bowtie_{DNÚMERO = NDEP} EMPREGADO)$
- ❑  $RESULT \leftarrow \pi_{PNOME, SNOME, ENDEREÇO}(EMPDEP)$

- Em CRT

- ❑  $\{ t.PNOME, t.SNOME, t.ENDEREÇO \mid EMPREGADO(t) \text{ AND } (\exists d) ( DEPARTAMENTO(d) \text{ AND } d.DNOME='Pesquisa' \text{ AND } d.DNUMERO=t.DNO ) \}$

# 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  
( $\exists$  d) ( DEPARTAMENTO(d) AND  
d.DNUMERO = p.DNUM AND  
d.GERNSS = m.NSS ) }

# Exemplo de Duplo Join

---

- Se exemplo anterior, trocamos 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  
( $\exists p, w$ ) ( 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  
(( $\exists$  e, w)( EMPREGADO(e) AND TRABALHA-EM(w) AND  
w.PNO=p.PNUMERO AND  
e.SNOME='Smith' AND e.NSS=w.ENSS )

OR

- ( $\exists$  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.NSSUPER = s.NSS }

# Exemplo de Divisão

- Liste o nome de todos os empregados que trabalham em todos os projetos:

- Em Álgebra Relacional

- $\text{PROJNSS}(\text{PNRO}, \text{NSS}) \leftarrow \pi_{\text{PNRO}, \text{NSSEMP}} (\text{TRABALHA\_EM})$
- $\text{PROJS}(\text{PNO}) \leftarrow \pi_{\text{PNUMERO}} (\text{PROJETO})$
- $\text{EMP} \leftarrow \text{PROJNSS} \div \text{PROJS}$
- $\text{RESULTADO} \leftarrow \pi_{\text{PNOME}} (\text{EMP} * \text{EMPREGADO})$

- Em CRT

- $\{ e.\text{PNOME} \mid \text{EMPREGADO}(e) \text{ AND } (\forall p)(\text{PROJETO}(p) \Rightarrow (\exists w)(\text{TRABALHA-EM}(w) \text{ AND } w.\text{PNO}=p.\text{PNUMERO} \text{ AND } w.\text{ENSS}=e.\text{NSS} ) ) \}$

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

- ❑ Liste o nome dos empregados que não tenham dependentes:
  - $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } \text{NOT } (\exists d)( \text{DEPENDENTE}(d) \text{ AND } e.NSS = d.ENSS ) \}$
  - $\text{NOT } (\exists t)( \text{NOT } F(t) ) \equiv (\forall t)( F(t) )$
  - $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } (\forall d)( \text{NOT } ( \text{DEPENDENTE}(d) \text{ AND } e.NSS = d.ENSS ) ) \}$
  - $\text{NOT } (F1 \text{ AND } F2) \equiv \text{NOT } F1 \text{ OR } \text{NOT } F2$
  - $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } (\forall d)( \text{NOT } \text{DEPENDENTE}(d) \text{ OR } e.NSS \neq d.ENSS ) \}$
  - $\text{NOT } F1 \text{ OR } F2 \equiv F1 \Rightarrow F2$
  - $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } (\forall d)( \text{DEPENDENTE}(d) \Rightarrow e.NSS \neq d.ENSS ) \}$

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

## □ Para analisar o último caso:

- $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } (\forall d)(DEPENDENTE(d) \Rightarrow e.NSS \neq d.ENSS) \}$

## □ Suponha que EMPREGADO e DEPENDENTE fossem as seguintes relações:

EMPREGAD	Nom	NS
O	e	S
	a1	b1
	a2	b2
	a3	b3
	a4	b4

DEPENDENT	ENS	NOM
E	S	E
	b1	c1
	b3	c3

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

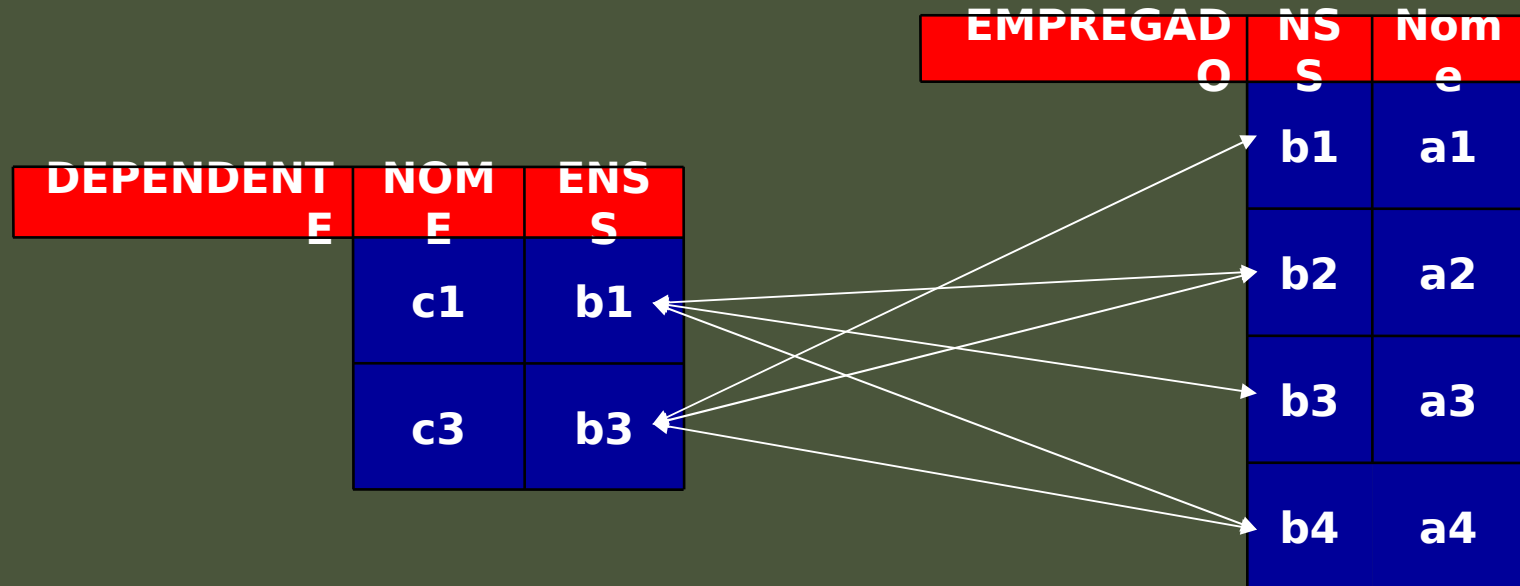
- Sabemos que para  $(\forall d)(\text{DEPENDENTE}(d) \Rightarrow e.\text{NSS} \Leftrightarrow d.\text{ENSS})$  seja verdade, basta que  $e.\text{NSS} \Leftrightarrow d.\text{ENSS}$  seja verdade, pois  $\text{DEPENDENTE}(d)$  sempre será verdade para todo  $d$ :

Esses casos  
nunca irão  
ocorrer

P	q	$P \Rightarrow Q$
DEPENDENTE(d)	$e.\text{NSS} \Leftrightarrow d.\text{ENSS}$	Q
V	V	V
V	F	F
F	V	V
F	F	V

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

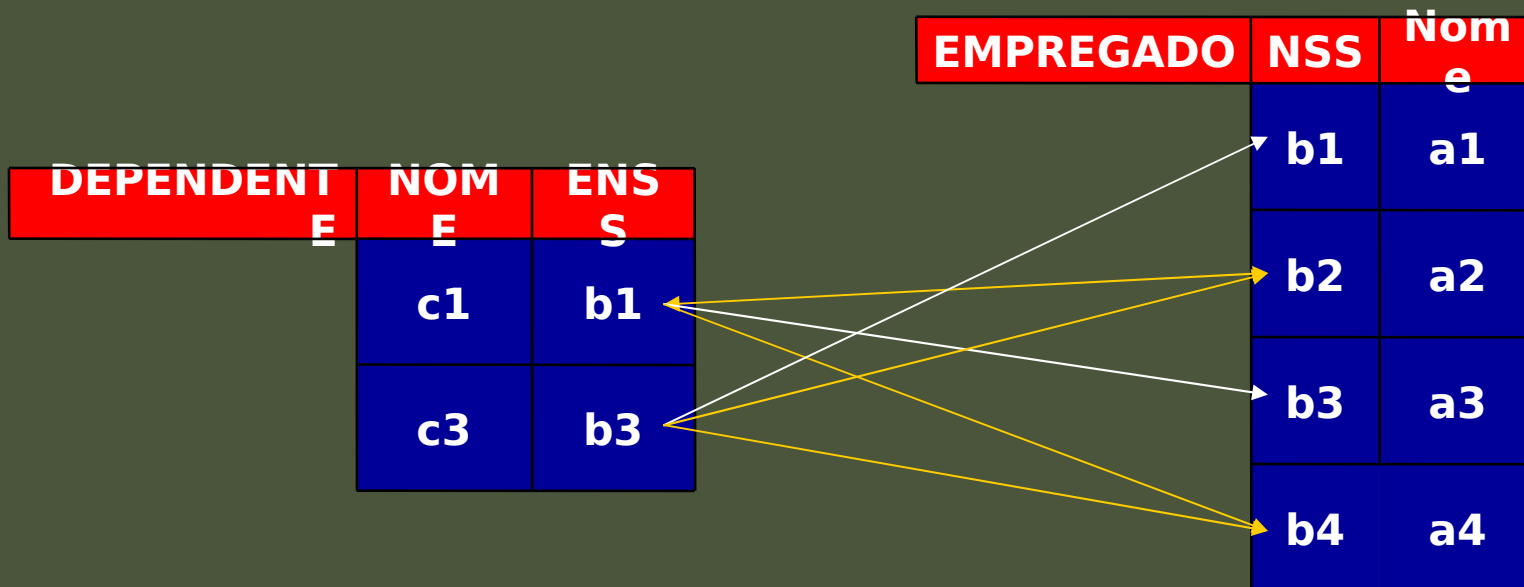
- Agora, basta saber se  $e.NSS \leftrightarrow d.ENSS$  é verdade **para todo d** de  $DEPENDENTE(d)$ .
- Assim, vamos analisar para cada dependente d, os possíveis resultados de  $e.NSS \leftrightarrow d.ENSS$ :





# 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 $(\forall t)(F(t))$

---

- Portanto, o resultado da consulta:
  - $\{ e.PNOME \mid EMPREGADO(e) \text{ AND } (\forall d)( DEPENDENTE(d) \Rightarrow e.NSS \neq d.ENSS ) \}$
- Será a relação que contém os nomes das tuplas de empregados, e, os quais não possuem dependentes:

<b>Nom</b>
<b>e</b>
<b>a2</b>
<b>a4</b>

# Um outro Exemplo de Divisão

---

- Encontrar os nomes de empregados que trabalham em todos os projetos controlados pelo departamento 5:
  - $\{ e.PNOME, e.SNOME \mid EMPREGADO(e) \text{ AND } (\forall x) ( ( PROJETO(x) \text{ AND } x.DNUM=5 ) \Rightarrow (\exists w) (TRABALHA-EM(w) \text{ AND } w.ENSS=e.NSS \text{ AND } x.PNUMERO=w.PNO) ) );$
- Aqui, o lado esquerdo da implicação ( $\Rightarrow$ ) restringe os projetos,  $x$ , do departamento 5.

# Um outro Exemplo de Divisão

- Assim, PROJETO( x ) AND x.DNUM=5 é:

O nosso Universo {

PNOME	<u>PNÚMER</u>	PLOCALIZAÇÃO	DNUM
ProdutoX	1	Bellaire	5
ProdutoY	2	Sugarland	5
ProdutoZ	3	Houston	5
Automação	10	Stafford	4
Reorganização	20	Houston	1
Beneficiário	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:

$(\exists w) (\text{TRABALHA-EM}(w) \text{ AND } w.\text{ENSS}=e.\text{NSS} \text{ AND } x.\text{PNUMERO}=w.\text{PNO})$

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.

$(\exists w) (\text{TRABALHA-EM}(w) \text{ AND } w.\text{ENSS}=e.\text{NSS} \text{ AND } x.\text{PNUMERO}=w.\text{PNO})$

PROJDEP5( x )	
PNOME	<u>PNUMERO</u>
Produto X	1
Produto Y	2
Produto Z	3

TRABALHA-EM ( w )	
<u>PNRO</u>	<u>NSSEMP</u>
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 )		
<u>NSS</u>	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 ( $\exists$ )  
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

$$\{t \mid \text{NOT } (R(t))\}$$

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/>).