

# Álgebra Relacional

Prof. Daniel Callegari

Material de base: Profa. Karin Becker



# Álgebra Relacional

- Modelo Relacional (Codd 70)
  - modelo de dados relacional
  - álgebra relacional
  - regras de projeto (normalização)
- Álgebra Relacional
  - extensão da álgebra de conjuntos
  - toma uma ou duas relações (tabelas) como entrada e produz uma relação resultado
  - unária (1 entrada)
  - binária (2 entradas)
- fundamento conceitual da linguagem SQL (padrão para SGBDs relacionais)

# Álgebra Relacional

- Operadores
  - Unários
    - seleção ( $\sigma$ ) – sigma
    - projeção ( $\pi$ ) – pi
  - Binários
    - interseção ( $\cap$ ), união ( $\cup$ ), diferença ( $-$ )
    - produto cartesiano ( $\times$ )
    - junção
    - divisão ( $\div$ )
- Esquema da Relação Resultante
  - Igual ao da entrada: **seleção, intersecção, união, diferença**
  - Novo: **projeção, produto cartesiano, junção, divisão**

# Seleção

- Definição
  - operador unário
  - esquema da relação resultado = relação entrada
  - seleciona tuplas da relação argumento que satisfazem uma dada condição
- Representação
$$\sigma_{\langle \text{condição} \rangle} (\langle \text{relação} \rangle)$$
- Observações
  - condições são construídas com o uso de atributos, constantes, operadores de comparação, operadores lógicos e parênteses (precedência)

# Seleção

- Exemplos

## Empregado

C o d e	N o m e	S a l a r i o	C o d d
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

$\sigma_{\text{salario} > 100}$  (EMPREGADO)

## Relação Resultado

Code	Nome	Salario	Codd
e2	Jose	200	d1
e4	Maria	500	d1

# Seleção

- Exemplos

$\sigma_{\text{salario} > 100 \vee \text{codd} = \text{'d1'}} (\text{EMPREGADO})$

$\sigma_{\text{salario} > 100 \wedge \text{codd} = \text{'d1'}} (\text{EMPREGADO})$

$\sigma_{\neg(\text{salario} > 100 \wedge \text{codd} = \text{'d1'})} (\text{EMPREGADO})$

# Projeção

- Definição
  - operador unário
  - esquema da relação resultado  $\neq$  relação entrada
  - retorna da relação argumento somente os atributos especificados
  - elimina tuplas duplicadas, se necessário
- Representação
$$\pi \langle \text{lista-atributos} \rangle (\langle \text{relação} \rangle)$$
- Observações
  - os nomes dos atributos originais podem ser alterados através de renomeação

# Projeção

- Exemplos

Empregado

Code	Nome	Salario	Codd
e1	Joao	100	d1
e2	Jose	200	d1
e3	Pedro	100	d2
e4	Maria	500	d1

$\pi_{\text{nome, salario}}$  (EMPREGADO)

Relação Resultado

Nome	Salario
Joao	100
Jose	200
Pedro	100
Maria	500



# Projeção

- Exemplos

Empregado

Code	Nome	Salario	Codd
e1	Joao	100	d1
e2	Jose	200	d1
e3	Pedro	100	d2
e4	Maria	500	d1

$\pi_{\text{salario}}$  (EMPREGADO)

Relação Resultado

Salario
100
200
500



sem tuplas  
duplicadas

# Produto Cartesiano

- Definição
  - operador binário
  - esquema da relação resultado  $\neq$  relação entrada
  - produto cartesiano da álgebra de conjuntos
  - retorna uma relação que combina cada tupla da relação-1 com cada tupla da relação-2
    - colunas: relação-1 + relação-2
    - tuplas: relação-1 x relação-2
- Representação

$\langle \text{relação-1} \rangle \mathbf{X} \langle \text{relação-2} \rangle$

# Produto Cartesiano

- Exemplos

## Empregado

C o d e	N o m e	S a l a r i o	C o d d
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

## Departamento

Codd	Nomed
d1	RH
d2	Tesouraria

## EMPREGADO X DEPARTAMENTO

### Relação Resultado

Code	Nome	Salario	EMPREGADO. Codd	DEPARTAMENTO. Codd	Nomed
e1	Joao	100	d1	d1	RH
e1	Joao	100	d1	d2	Tesouraria
e2	Jose	200	d1	d1	RH
e2	Jose	200	d1	d2	Tesouraria
e3	Pedro	100	d2	d1	RH
e3	Pedro	100	d2	d2	Tesouraria
e4	Maria	500	d1	d1	RH
e4	Maria	500	d1	d2	Tesouraria

# Junção

- Definição
  - operador binário
  - esquema da relação resultado  $\neq$  relação entrada
  - adaptação do produto cartesiano (PC):
    - PC seguido de seleção sobre condição de junção
  - condição de junção
    - comparação de atributos vindos das duas relações
- Representação
$$\sigma \langle \text{condição} \rangle (\langle \text{relação-1} \rangle \times \langle \text{relação-2} \rangle)$$

# Junção

- Exemplos

## Empregado

C o d e	N o m e	S a l a r i o	C o d d
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

## Departamento

Codd	Nomed
d1	RH
d2	Tesouraria

σ EMPREGADO.codd = DEPARTAMENTO.codd (EMPREGADO X DEPARTAMENTO)

## Relação Resultado

Code	Nome	Salario	EMPREGADO. Codd	DEPARTAMENTO. Codd	Nomed
e1	Joao	100	d1	d1	RH
e2	Jose	200	d1	d1	RH
e3	Pedro	100	d2	d2	Tesouraria
e4	Maria	500	d1	d1	RH

# Junção

- Uso Clássico
  - junção por igualdade relacionando chave estrangeira de <relação-1> com chave primária de <relação-2> (ou vice-versa)
  - contudo, pode-se juntar por qualquer atributo, **independente** de definições de chaves primárias e estrangeira

# Junção

- Exemplos

- o nome do empregado e do departamento em que trabalha, para empregados que ganham mais que \$100
- Alternativa 1

$\pi_{\text{nome, nome}} ($

$\sigma_{\text{EMPREGADO.codd} = \text{DEPARTAMENTO.codd} \wedge \text{salario} > 100}$   
(EMPREGADO X DEPARTAMENTO) )

- Alternativa 2 (mais eficiente)

$\pi_{\text{nome, nome}} ($

$\sigma_{\text{EMPREGADO.codd} = \text{DEPARTAMENTO.codd}}$   
( $\sigma_{\text{salario} > 100}$  (EMPREGADO) X DEPARTAMENTO) )

# Junção Natural

- Definição

- adaptação da junção por igualdade

- elimina as colunas duplicadas (colunas de junção)
    - baseia-se em colunas com nomes iguais (conceitualmente, domínios iguais)
    - notação mais sintética para caso mais comum (junção por igualdade PK=FK)
    - cuidado
      - com efeitos inesperados
      - com colunas equivalentes mas sem nomes iguais

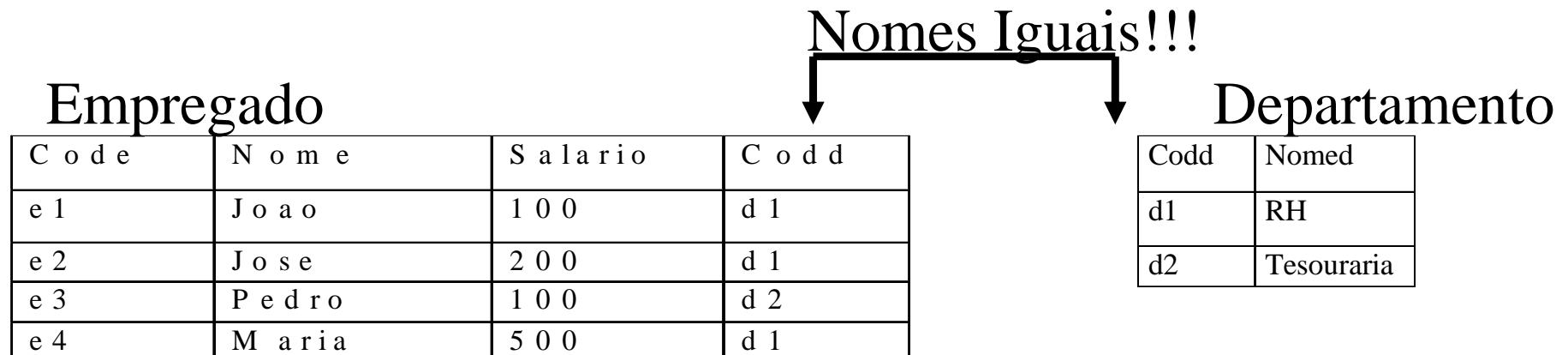
- Representação

<relação-1> | X | <relação-2>



# Junção Natural

- Exemplo (1)



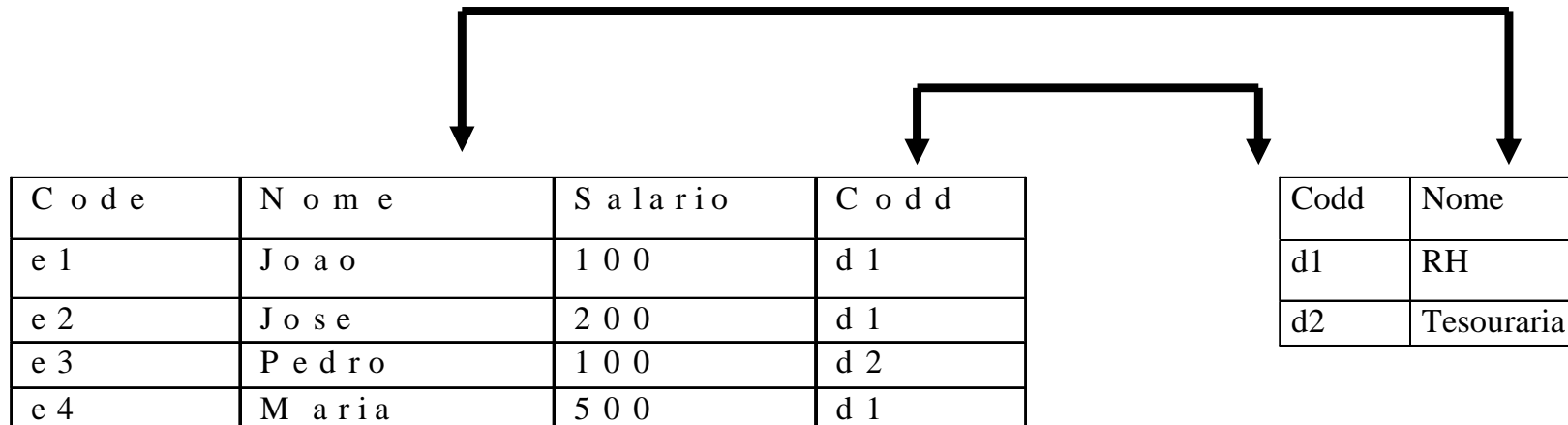
EMPREGADO | X | DEPARTAMENTO

## Relação Resultado

Code	Nome	Salario	Codd	Nomed
e1	Joao	100	d1	RH
e2	Jose	200	d1	RH
e3	Pedro	100	d2	Tesouraria
e4	Maria	500	d1	RH

# Junção Natural

- Exemplo 2: Resultados inesperados !!!



EMPREGADO | X | DEPARTAMENTO

## Relação Resultado

Code	Nome	Salario	Codd

“ os dados dos empregados e respectivos departamentos, para todo empregado que tiver nome igual ao de onde trabalha”

# Junção Natural

- Exemplo 3:
  - atributos com nomes diferentes???
  - Renomear !!!

Empregado

C o d e	N o m e	S a l a r i o	D e p
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

Nomes Diferentes!!!



Departamento

C o d d	N o m e d
d1	RH
d2	Tesouraria

$\pi_{\text{code, nome, salario, dep as codd}}(\text{EMPREGADO}) \mid \times \mid \text{DEPARTAMENTO}$

Relação Resultado

Code	Nome	Salario	Codd	Nomed
e1	Joao	100	d1	RH
e2	Jose	200	d1	RH
e3	Pedro	100	d2	Tesouraria
e4	Maria	500	d1	RH

# União

- Definição
  - operador binário
  - esquema da relação resultado = relação entrada
  - união da álgebra dos conjuntos
    - tuplas de relação-1 ou de relação-2
- Representação
$$\langle \text{relação-1} \rangle \cup \langle \text{relação-2} \rangle$$
- Observações
  - $\langle \text{relação-1} \rangle$  e  $\langle \text{relação-2} \rangle$  devem ter esquemas compatíveis

# Compatibilidade de Esquemas

- Duas relações são compatíveis sse:
  - estão definidas sobre os mesmos domínios (i.e. têm as mesmas colunas)
  - os domínios (colunas) estão nas mesmas posições
- Observações
  - usar projeção e renomeação para compatibilizar relações, quando estas não forem iguais

# União : exemplo(1)

Empregados\_Ativos

C o d e	N o m e	S a l a r i o	C o d d
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

Empregados\_Cedidos

C o d e	N o m e	S a l a r i o	C o d d
e 1 0	P e d r o	1 0 0 0	d 2
e 2 0	C a r l o s	2 0 0 0	d 1
e 3 0	M a u r a	1 0 0 0	d 1

EMPREGADOS\_ATIVOS U EMPREGADOS\_CEDIDOS

Tabela Resultado

Code	Nome	Salario	Codd
e1	Joao	100	d1
e2	Jose	200	d1
e3	Pedro	100	d2
e4	Maria	500	d1
e10	Pedro	1000	d2
e20	Carlos	2000	d1
e30	Maura	1000	d1

# União : exemplo(2)

Empregados

C o d e	N o m e	S a l a r i o	C o d d
e 1	J o a o	1 0 0	d 1
e 2	J o s e	2 0 0	d 1
e 3	P e d r o	1 0 0	d 2
e 4	M a r i a	5 0 0	d 1

Voluntários

Cod	Codd	Nome
e10	d1	Pedro
e20	d2	Carlos
e30	d1	Maura

$\pi_{\text{cod as code, nome, codd}}$  (VOLUNTARIOS)  $\cup$

$\pi_{\text{code, nome, codd}}$  ( EMPREGADOS)

Tabela Resultado

Code	Nome	Codd
e1	Joao	d1
e2	Jose	d1
e3	Pedro	d2
e4	Maria	d1
e10	Pedro	d1
e20	Carlos	d2
e30	Maura	d1

# Intersecção

- Definição

- operador binário
- esquema da relação resultado = relação entrada
- intersecção da álgebra dos conjuntos
  - tuplas de relação-1 que também estão em relação-2

- Representação

$$\langle \text{relação-1} \rangle \cap \langle \text{relação-2} \rangle$$

- Observações

- $\langle \text{relação-1} \rangle$  e  $\langle \text{relação-2} \rangle$  devem ter esquemas compatíveis



# Diferença

- Definição
  - operador binário
  - esquema da relação resultado = relação entrada
  - diferença da álgebra dos conjuntos
    - tuplas de relação-1 que não estão em relação-2
- Representação
$$\langle \text{relação-1} \rangle - \langle \text{relação-2} \rangle$$
- Observações
  - $\langle \text{relação-1} \rangle$  e  $\langle \text{relação-2} \rangle$  devem ter esquemas compatíveis

# Álgebra: Otimização

- Álgebra é uma teoria de manipulação de relações
  - demonstrar viabilidade de manipular dados independentemente de representação física (ordem, índices, etc)
  - não uma linguagem de SGBD completa: fundamenta SQL
- SQL é não procedural
  - SGBD transforma SQL em operações menores da álgebra, e define um plano otimizado de consultas, combinado com o uso de índices
  - otimizador algébrico
    - antecipa operações que reduzem volumes de dados
    - seleção, projeção, diferença e intersecção