

# SCC-240 Banco de Dados

Profa. Elaine Parros Machado de Sousa

## Álgebra Relacional - Parte 2

# Exemplo

- Ex: “*Listar as disciplinas em que os alunos de computação se matricularam*”

■ Aluno = {Nome, Idade, Curso}

■ Matricula = {NomeA, Disciplina, Nota}

$\pi_{(\text{Disciplina})}(\sigma_{(\text{Curso} = \text{"computação"})}(\sigma_{(\text{Nome} = \text{NomeA})}(\text{Matricula X Aluno})))$

# Exemplo

$\pi_{(\text{Disciplina})}(\sigma_{(\text{curso} = \text{"computação"})}(\sigma_{(\text{Nome} = \text{NomeA})}(\text{Matricula X Aluno})))$

- Mais eficiente  $\Rightarrow$  gravar somente as tuplas que atendem ao critério de seleção aplicado ao Produto Cartesiano
- Algoritmos eficientes para a combinação de operações
  - ex: Seleção + Produto Cartesiano

# Operações Relacionais Binárias


# Operações Relacionais Binárias

- Junção
- Divisão

# Junção

- **JUNÇÃO (“Join<sub>R</sub>”)**  *(condição da junção)* **S**

$\pi_{(Disciplina)}(\sigma_{(Curso = \text{"computação"})}(\sigma_{(Nome = NomeA)}(Matricula \times$   
 $Aluno)))$

  
 $(Matricula \times Aluno)$   
 $(NomeA = Nome)$

**Como representar a mesma consulta,  
usando junção, de maneira ainda mais  
eficiente?**

# Junção

■ **JUNÇÃO ("Join")**  $R \bowtie_{(condição\ da\ junção)} S$

$\pi_{(Disciplina)} (Matricula \bowtie_{(\sigma_{(Curso = "computação")}} Aluno))$   
(NomeA = Nome)

## Em SQL

```
select Disciplina
from Aluno, Matricula
where Nome = NomeA
      and Curso = 'computacao'
```

# Junção

- **JUNÇÃO ("Join")** -R ☒ *(condição da junção)* S

```
select Disciplina from Aluno, Matricula  
where Nome = NomeA  
and Curso = 'computacao'
```

## Usando operador JOIN

```
select Disciplina  
from Aluno join Matricula on Nome = NomeA  
where Curso = 'computacao'
```

# Junção

## ■ $R \bowtie$ (condição da junção) $S$

### ■ condição da Junção:

■  $\langle \text{condição} \rangle$  **AND**  $\langle \text{condição} \rangle$  **AND** ...  $\langle \text{condição} \rangle$

### ■ $\langle \text{condição} \rangle$ : comparação entre atributos, ou conjunto de atributos:

#### ■ $\text{Atrib}_R \theta \text{Atrib}_S$

- $\text{Atrib}_R$  - atributo da relação R
- $\text{Atrib}_S$  - atributo da relação S
- $\text{Atrib}_R$  e  $\text{Atrib}_S$  são **atributos de junção** - mesmo domínio
- $\theta$  - operador de comparação válido no domínio



# Junção

- $Q \leftarrow R \bowtie_{(condição\ de\ junção)} S$ 
  - resultado: Q tem uma tupla para cada combinação das tuplas de R com as tuplas de S que satisfaz a condição de junção
    - $|Q| \leq |R \times S|$
  - tuplas com valores nulos para atributos de junção **não estão** no resultado
  - $(\text{grau de } Q) = (\text{grau de } R) + (\text{grau de } S)$

# Junção

- Operação importante em bases de dados relacionais
  - usada para combinar tuplas (relacionadas) de diferentes relações em uma única tupla
  - permite processamento de relacionamentos entre relações
  - muito usada com relações vinculadas por chave estrangeira

# Tipos de Junção

- Junções Internas (*inner joins*)
  - junção theta
  - equi-junção
  - junção natural
- Junções externas (*outer joins*)
  - *left outer join*
  - *right outer join*
  - *full outer join*

# Junções Internas


- **Junção Theta ( $\theta$ -join)**
  - **$\theta$  é qualquer operador** válido no domínio dos atributos de junção
  - atributos de junção das duas relações aparecem na relação resultado
  - variação mais genérica

# Exemplo: Junção- $\theta$

Aluno = {Nome, Idade, Curso}      Matricula = {NomeA, Disciplina, Nota}

{<Zeca, 25, computação>, <Zico, 18, eletrônica>, <Juca, 21, odontologia>, <Tuca, 18, computação> }      {<Zeca, SCC-125, 8.5>, <Zico, SCC-148, 5.2>, <Juca, SCC-125, 6.0>, <Juca, SCC-148, 7.0> }

**Aluno**  $\bowtie$  **Matricula**  
(Nome = NomeA)



**{Nome, Idade, Curso, NomeA, Disciplina, Nota}**  
{<Zeca, 25, computação, Zeca, SCC-125, 8.5>, <Zico, 18, eletrônica, Zico, SCC-148, 5.2>, <Juca, 21, odontologia, Juca, SCC-125, 6.0>, <Juca, 21, odontologia, Juca, SCC-148, 7.0> }

# Junções Internas

## ■ **Equi-Junção** (*Equi-join*)

- $\theta$  é um operador de **igualdade**
- os atributos de junção das duas relações aparecem na relação resultado

Duas maneiras de simbolizar *equi-join*:


$$\begin{array}{c} \boxtimes \\ R \quad \boxtimes \text{AtribR} = \text{AtribS} \quad S \quad \text{ou} \\ R \quad (\text{AtribR}, \text{AtribS}) \quad S \end{array}$$

# Exemplo: Equi-Junção

Empregado = {Nome, Depto}    Depto = {NomeD, Codigo}

{<Zeca, D1>,                      {<Vendas, D1>,  
<Zico, D2>,  
<Juca, **null**>}

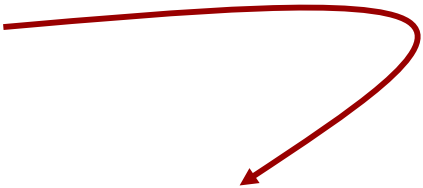
<Pessoal, D2>,  
<Juridico, D3>}



Empregado  Depto  
(Deppto, Codigo)

{Nome, **Deppto**, NomeD,  
**Codigo**}

{<Zeca, D1,    Vendas,    D1>,  
  <Zico,    D2,    Pessoal,    D2>}



# Junções Internas

## ■ Junção Natural - $R * S$

- semelhante à Equi-Junção
- apenas os atributos de junção de uma das relações aparece na relação resultado
  - requer que os atributos de junção tenham **nomes iguais** nas duas relações, ou sejam renomeados




# Exemplo: Junção Natural

[illegible]

Empregado \*  $\rho_{(\text{NomeD}, \text{Depto})}$

# Departamento

{Nome, **Depto**, NomeD}  
 {<Zeca, D1, Vendas>,   
 <Zico, D2, Pessoal>}

# Junções Internas

- Resumindo... 3 tipos de Junção Interna (*inner joins*)

- Junção -  $\theta \rightarrow \boxtimes R$  (condição de junção) **S**

- Equi-Junção  $\rightarrow \bowtie R$  (Atrib\_R, Atrib\_S) **S**

- Junção Natural  $\rightarrow R * S$

**Em  
SQL?**

# Junções Externas

- Junções externas (*outer joins*)
  - *Left Outer Join*
  - *Right Outer Join*
  - *Full Outer Join*

# Junções Externas

- ***Left Outer Join***  $\bowtie$  – R (condição de junção) **S**

- resultado:

- tuplas que atendem à condição de junção

+

- tuplas de R que não têm correspondentes em S

# Exemplo: *Left Join*

Aluno = {Nome, Idade, Curso} {<Zeca, 25, computação>, <Zico, 18, eletrônica>, <Juca, 21, odontologia>, <Tuca, 18, computação> }

Matricula = {NomeA, Disciplina, Nota} {<Zeca, SCC-125, 8.5>, <Zico, SCC-148, 5.2>, <Juca, SCC-125, 6.0>, <Juca, SCC-148, 7.0> }

“Selecionar as informações de todos os alunos e, para os que estão matriculados, os códigos e notas das disciplinas que cursam.”

Aluno ⊗ Matricula  
(Nome = NomeA)

**{Nome, Idade, Curso, NomeA, Disciplina, Nota}**

{<Zeca, 25, computação, Zeca, SCC-125, 8.5>, <Zico, 18, eletrônica, Zico, SCC-148, 5.2>, <Juca, 21, odontologia, Juca, SCC-125, 6.0>, <Juca, 21, odontologia, Juca, SCC-148, 7.0>, <Tuca, 18, computação, null, null, null> }

junção interna

junção externa

# Exemplo: *Left Join*

Aluno = {Nome, Idade, Curso} {<Zeca, 25, computação>, <Zico, 18, eletrônica>, <Juca, 21, odontologia>, <Tuca, 18, computação> }

Matricula = {NomeA, Disciplina, Nota} {<Zeca, SCC-125, 8.5>, <Zico, SCC-148, 5.2>, <Juca, SCC-125, 6.0>, <Juca, SCC-148, 7.0> }

“Selecionar as informações de todos os alunos e, para os que estão matriculados, os códigos e notas das disciplinas que cursam.”

**{Nome, Idade, Curso, NomeA, Disciplina, Nota}**

{<Zeca, 25, computação, Zeca, SCC-125, 8.5>, <Zico, 18, eletrônica, Zico, SCC-148, 5.2>, <Juca, 21, odontologia, Juca, SCC-125, 6.0>, <Juca, 21, odontologia, Juca, SCC-148, 7.0>, <Tuca, 18, computação, null, null, null> }

Aluno ⊗ Matricula  
(Nome = NomeA)

**Em SQL?**

# Junções Externas

- **Right Outer Join** -  $R \bowtie_{\text{(condição de junção)}} S$ 
  - resultado:
    - tuplas que atendem à condição de junção
    - +
    - tuplas de S que não têm correspondentes em R

# Exemplo: *Right Join*

Empregado = {Nome, Depto}      Departamento = {NomeD, Codigo}  
                  {<Zeca, D1>,                    {<Vendas, D1>,  
                  <Zico, D2>,  
                  <Juca, null>}

{<Pessoal, D2>,  
                  <Juridico,  
                  D3>}

“Selecionar as informações de todos os departamentos e, se houver, dos empregados que trabalham neles.”

Empregado ~~×~~  
Departamento (Deppto, Codigo)

{Nome, Depto, NomeD, Código}  
{<Zeca, D1, Vendas, D1>,  
  <Zico, D2, Pessoal, D2>,  
  <null, null, Juridico, D3>}

**Em SQL?**



# Junções Externas

- **Full Outer Join**  $\exists \bowtie R$  (condição de junção) **S**

- resultado:

- tuplas que atendem à condição de junção

+

- tuplas de R que não têm correspondentes em S

+

- tuplas de S que não têm correspondentes em R

## Exemplo: *Full Join*

[illegible]

Empleado ☒ (Departamento Codigo)

```
{Nome, Depto, NomeD, Código}
{<Zeca, D1, Vendas, D1>,
 <Zico, D2, Pessoal, D2>,
 <Juca, null, null, null>,
 <null, null, Juridico,
 D3>}
```

# Em SQL?

# Junções Externas em SQL

- *Left Outer Join*

```
SELECT <atributos> FROM tabela1 T1  
      LEFT [OUTER] JOIN tabela2 T2 ON T1.trib1 = T2.trib2
```

- *Right Outer Join*

```
SELECT <atributos> FROM tabela1 T1  
      RIGHT [OUTER] JOIN tabela2 T2 ON T1.trib1 = T2.trib2
```

- *Full Outer Join*

```
SELECT <atributos> FROM tabela1 T1  
      FULL [OUTER] JOIN tabela2 T2 ON T1.trib1 = T2.trib2
```

# Operações Relacionais Binárias

- Junção

- Divisão



# DIVISÃO

Exemplo:

Matricula = {NomeA, Disciplina,  
Nota} — {<Zeca, SCC-125, 8.5>,  
          <Zeca, SCC-148, 8.0>,  
          <Zeca, SCC-180, 7.5>,  
          <Zico, SCC-148, 5.2>,  
          <Juca, SCC-125, 6.0>,  
          <Juca, SCC-148, 7.0>}

Aulas = {NomeP, Disciplina}  
          {<João, SCC-125>,  
          <João, SCC-148>,  
          <Eva, SCC-180>}


“Quais alunos  
cursam **todas** as  
disciplinas  
ministradas pelo  
Prof. João? “

## Exemplo (cont.)

Selecionar as disciplinas ministradas por João

Aulas = {NomeP, Disciplina}  
{<João, SCC-125>,  
  <João, SCC-148>,  
  <Eva, SCC-180>}

$S \leftarrow \pi_{\{Disciplina\}}(\sigma_{(NomeP = "João")}(Aulas))$



$S = \{Disciplina\}$   
 $\{<SCC-125>,$   
   $<SCC-148>\}$

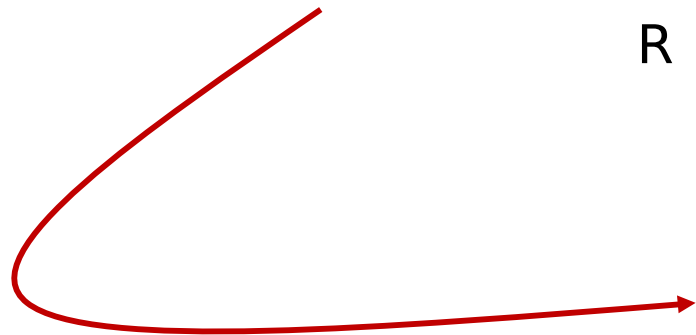
## Exemplo (cont.)

Selecionar as disciplinas cursadas por cada aluno

Matricula = {NomeA, Disciplina, Nota}

{<Zeca, SCC-125, 8.5>,  
<Zeca, SCC-148, 8.0>,  
<Zeca, SCC-180, 7.5>,  
<Zico, SCC-148, 5.2>,  
<Juca, SCC-125, 6.0>,  
<Juca, SCC-148, 7.0>}

$R \leftarrow \pi_{\{\text{NomeA}, \text{Disciplina}\}}(\text{Matricula})$



$R = \{\text{NomeA}, \text{Disciplina}\}$   
{<Zeca, SCC-125>,  
<Zeca, SCC-148>,  
<Zeca, SCC-180>,  
<Zico, SCC-148>,  
<Juca, SCC-125>,  
<Juca, SCC-148>}

## Exemplo (cont.)

### 3. Aplicar operação de divisão

$R \leftarrow \pi_{\{\text{NomeA}, \text{Disciplina}\}}(\text{Matricula})$

$R = \{\text{NomeA}, \text{Disciplina}\}$

$\{ \langle \text{Zeca}, \text{SCC-125} \rangle, \langle \text{Zeca}, \text{SCC-148} \rangle, \langle \text{Zeca}, \text{SCC-180} \rangle, \dots \}$

$S \leftarrow \pi_{\{\text{Disciplina}\}}(\sigma_{(\text{NomeP} = \text{"João"})}(\text{Aulas}))$

$S = \{\text{Disciplina}\}$

$\{ \langle \text{SCC-125} \rangle, \langle \text{SCC-148} \rangle, \dots \}$

**$T \leftarrow R \div S$**



# Exemplo (cont.)

$$(\pi_{\{\text{NomeA}, \text{Disciplina}\}}(\text{Matricula})) \div (\pi_{\{\text{Disciplina}\}}(\sigma_{(\text{NomeP} = \text{"João"})}(\text{Aulas})))$$

**R = {NomeA, Disciplina}**

{<Zeca, SCC-125>,   
<Zeca, SCC-148>,   
<Zeca, SCC-180>,   
<Zico, SCC-148>,   
<Juca, SCC-125>,   
<Juca, SCC-148>}

**S = {Disciplina}**

{<SCC-125>,   
<SCC-148>}

**T = {NomeA}**

{<Zeca>,   
<Juca>}

# Divisão

## ■ Operação de Divisão - $R \div S$

- $T \leftarrow R \div S$
- S é uma relação cujos atributos (B) são um subconjunto dos atributos (A) da relação R
  - $T(C) \leftarrow R(A) \div S(B)$ , com:
    - $B \subseteq A$
    - $C = A - B$
- uma tupla t pertence ao resultado  $T(C)$  se existirem tuplas  $t_R$  em R tal que  $t_R[C] = t$ , e com  $t_R[B] = t_S$  para toda tupla  $t_S$  em S

# Divisão

- **Exercício:** pesquise como fazer uma divisão em SQL
  - Dica: livro [Elmasri&Navathe]

# Leitura recomendada

- R. Elmasri, S. Navathe:  
*Fundamentals of Database Systems* – 4th Edition
  - capítulo 6

# Exercícios

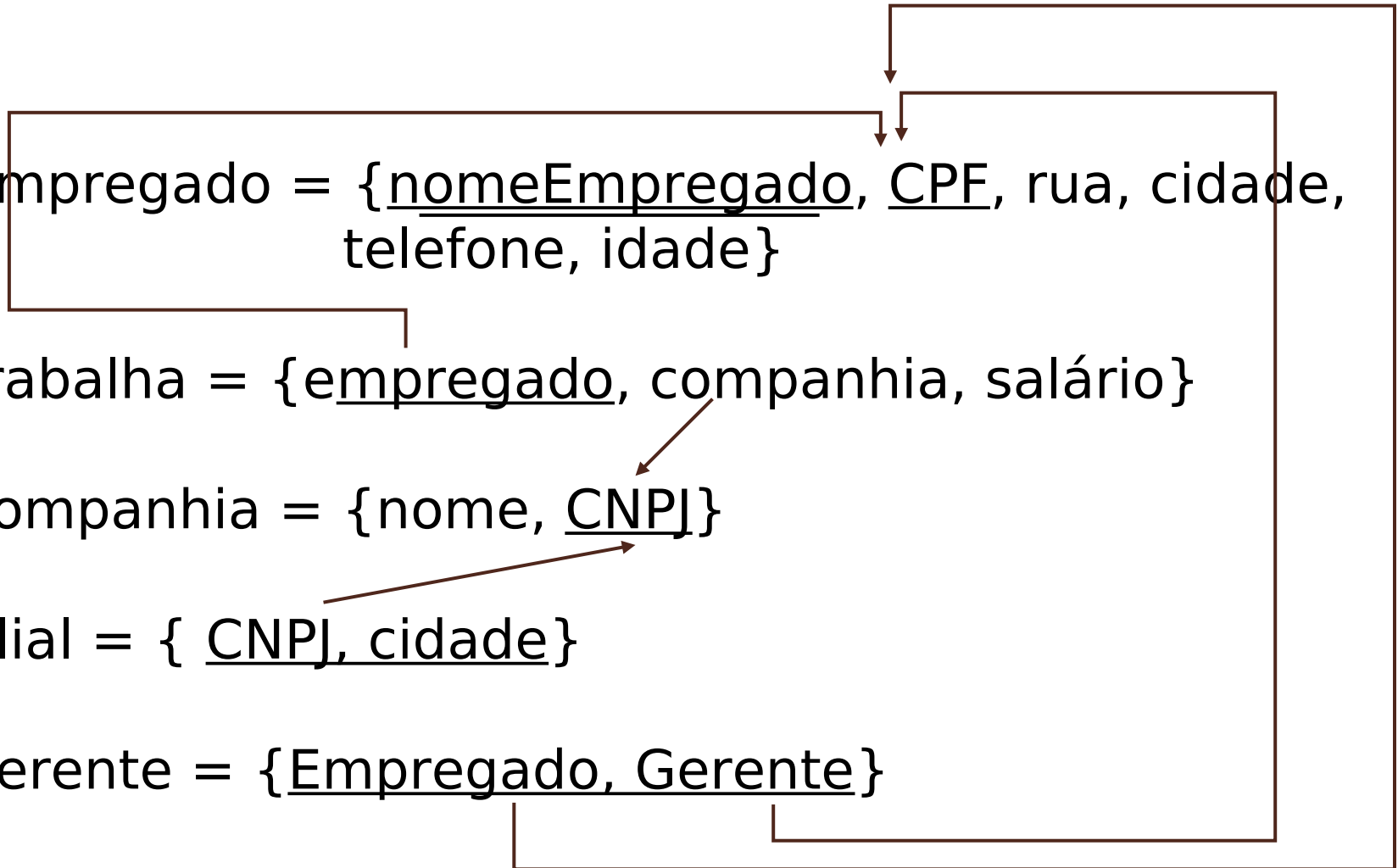
empregado = {nomeEmpregado, CPF, rua, cidade, telefone, idade}

trabalha = {empregado, companhia, salário}

companhia = {nome, CNPJ}

filial = {CNPJ, cidade}

gerente = {Empregado, Gerente}



# Exercícios

- **Q1:** Liste nome e cidade de todos os empregados da IBM que ganham mais de dez mil dólares por mês
- **Q2:** Liste os nomes de todos os empregados que não trabalham para a IBM.
- **Q3:** Liste os nomes de todos os empregados que moram numa cidade onde há filial da companhia em que trabalham.
- **Q4:** Liste o CNPJ de todas as companhias com filiais em **todas** as cidades onde haja unidades da IBM
- **Q5:** Liste nome e CPF de todos os empregados e, para os que tiverem trabalhando, liste o CNPJ e o nome da companhia em que trabalham.