

SCC-240 Banco de Dados

Profa. Elaine Parros Machado de Sousa

Estagiários PAE:
Pedro Bugatti
Robson Cordeiro

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}

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

Exemplo

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

- 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") – $R \bowtie_{(condição\ da\ junção)} S$

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

\downarrow

$(Matricula \bowtie_{(NomeA = Nome)} Aluno)$

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

Junção

- JUNÇÃO ("Join") – $R \bowtie_{(condição\ da\ junção)} S$

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

Em SQL?

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

Junção

▪ JUNÇÃO ("Join") – $R \bowtie_{(condição\ da\ junção)} S$

```
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:
 - $\langle condição \rangle$ AND $\langle condição \rangle$ AND ... $\langle condição \rangle$
- $\langle condição \rangle$ - comparação entre atributos, ou conjunto de atributos:
 - **Atrib_R θ Atrib_S**
 - Atrib_R - atributo da relação R
 - Atrib_S - atributo da relação S
 - Atrib_R e Atrib_S são **atributos de junção** - mesmo domínio
 - θ - operador de comparação válido no domínio desses atributos

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
 - (grau de Q) = (grau de R) + (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 (θ -join)**
 - **θ é qualquer operador** válido no domínio dos atributos de junção
 - atributos de junção aparecem ambos na relação resultado
 - variação mais genérica

Exemplo: Junção-θ

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

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

Aluno ⋈ Matricula
 (Nome = NomeA)

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

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

Junções Internas

- **Equi-Junção (Equi-join)**
 - **θ** é um operador de **igualdade**
 - os atributos de junção aparecem ambos na relação resultado

Duas maneiras de simbolizar *equi-join*:

$R \bowtie_{(AtribR = AtribS)} S$ ou $R \bowtie_{(AtribR, AtribS)} S$

Exemplo: Equi-Junção

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

{<Zeca, D1>, {<Vendas, D1>, }
 {<Zico, D2>, {<Pessoal, D2>, }
 {<Juca, D1> } {<Juridico, D3> }

Empregado ⋈ Depto
 (Depto, Codigo)

{Nome, Depto, NomeD, Codigo}

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

Junções Internas

- **Junção Natural - R*S**
 - semelhante à Equi-Junção
 - apenas um dos atributos de junção aparece na relação resultado
 - requer que os atributos de junção tenham **nomes iguais** nas duas relações, ou sejam renomeados

Exemplo: Junção Natural

Empregado = {NomeA, Depto} Departamento = {NomeD, Codigo}

{<Zeca, D1>, {<Vendas, D1>, }
 {<Zico, D2>, {<Pessoal, D2>, }
 {<Juca, D1> } {<Juridico, D3> }

Empregado *_p(NomeD, Depto) Departamento

{NomeA, Depto, NomeD}

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

Junções Internas

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

- Junção - θ → $R \bowtie_{(condição\ de\ junção)} S$
- Equi-Junção → $R \bowtie_{(Atrib_R, Atrib_S)} S$
- Junção Natural → **R*S**

Em SQL?

Junções Externas

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

Junções Externas

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

Exemplo: Left Join

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

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

"Selecionar as informações dos alunos e, para os que estão matriculados, os códigos e notas das disciplinas que cursam."

Aluno \bowtie Matricula
(Nome = NomeA)

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

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

Exemplo: Left Join

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

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

"Selecionar as informações dos alunos e, para os que estão matriculados, os códigos e notas das disciplinas que cursam."

Aluno \bowtie Matricula
(Nome = NomeA)

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

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

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 = {NomeA, Depto} Departamento = {NomeD, Código}

{<Zeca, D1>, {<Vendas, D1>, }
 {<Zico, D2>, {<Pessoal, D2>, }
 {<Juca, D1>, {<Juridico, D3>, }
 {<Ana, null >, }

"Selecionar as informações dos departamentos e, se houver, dos empregados que trabalham neles."

Empregado \bowtie Departamento
(Depto = Código)

{NomeA, Depto, NomeD, Código}

{<Zeca, D1, Vendas, D1>, }
 {<Zico, D2, Pessoal, D2>, }
 {<Juca, D1, Vendas, D1>, }
 {<null, null, Juridico, D3>, }

Em SQL?

Junções Externas

■ **Full Outer Join** – R \bowtie (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**

Empregado = {NomeA, Depto} Departamento = {NomeD, Código}

{<Zeca, D1>, <Zico, D2>, <Juca, D1>, <Ana, null>}

{<Vendas, D1>, <Pessoal, D2>, <Juridico, D3>}

Empregado \bowtie Departamento
(Depto = Codigo)

{NomeA, Depto, NomeD, Código}

{<Zeca, D1, Vendas, D1>, <Zico, D2, Pessoal, D2>, <Juca, D1, Vendas, D1>, <Ana, null, null, null>, <null, null, Juridico, D3>}

Em SQL?

Operações Relacionais Binárias

■ Junção

■ Divisão



DIVISÃO

Exemplo:

Matricula = {NomeA, Disciplina, Nota}

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

Aulas = {NomeP, Disciplina}

{<João, SCE-125>, <João, SCE-148>, <Eva, SCE-180>}

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

Exemplo (cont.)

① Selecionar as disciplinas ministradas por João

Aulas = {NomeP, Disciplina}

{<João, SCE-125>, <João, SCE-148>, <Eva, SCE-180>}

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

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

Exemplo (cont.)

② Selecionar as disciplinas cursadas por cada aluno

Matricula = {NomeA, Disciplina, Nota}

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

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

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

Exemplo (cont.)

3 Aplicar operação de divisão

$R \leftarrow \pi_{\{NomeA, Disciplina\}}(Matricula)$ $R = \{NomeA, Disciplina\}$
 $\{<Zeca, SCE-125>, <Zeca, SCE-148>, <Zeca, SCE-180>, <Zico, SCE-148>, <Juca, SCE-125>, <Juca, SCE-148>\}$

$S \leftarrow \pi_{\{Disciplina\}}(\sigma_{(NomeP = "João")}(Aulas))$ $S = \{Disciplina\}$
 $\{<SCE-125>, <SCE-148>\}$

$T \leftarrow R \div S$

Exemplo (cont.)

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

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

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

$T = \{NomeA\}$
 $\{<Zeca>, <Juca>\}$

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 a $T(C)$ 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

Exercício

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

trabalha = {empregado, companhia, salário}

companhia = {nome, CNPJ}

filial = {CNPJ, cidade}

gerente = {Empregado, Gerente}



Exercício

- **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.