Álgebra relacional

BCD29008 – Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

INSTITUTO FEDERAL
Santa Catarina
Campus São José

Licenciamento

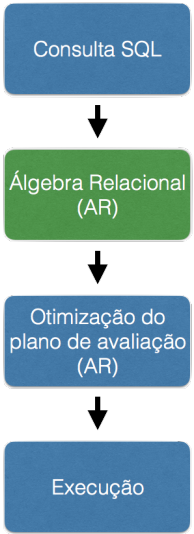
Slides licenciados sob Creative Commons “Atribuição 4.0 Internacional”

Linguagens formais de consulta relacional

Linguagem de consulta relacional

Permite ao usuário solicitar informações do banco de dados e geralmente está em um nível mais alto de abstração do que as tradicionais linguagens de programação

Arquitetura de SGBD



1. Usuário escreve a consulta
2. Traduzida para uma expressão de álgebra relacional
3. Encontra uma expressão equivalente que seja mais eficiente
4. Executa as operações do plano otimizado

Álgebra relacional

Nos permite traduzir consultas SQL para expressões mais precisas e otimizadas

Álgebra relacional

Conjunto de operações aplicado sobre um ou mais conjuntos e que **produz um novo conjunto como resultado**

- Semelhante com operações algébricas normais, como adição, subtração ou multiplicação

- **Operações fundamentais**
  - σ Seleção
  - Π Projeção
  - $\times$ Produto cartesiano
  - $\cup$ União
  - $-$ Diferença

- **Operações derivadas**
  - $\cap$ Interseção
  - $\bowtie$ Junções
  - ρ Atribuição

Seleção – σ (sigma)

- Retorna todas as tuplas que satisfaçam uma condição
  - Condição c pode ser: $=, <, \leq, \geq, <>$
  - Pode-se também fazer uso de operadores lógicos AND ($\wedge$), OR ($\vee$) e NOT ($\neg$)
- **Notação:** $\sigma_c(R)$
- Exemplos

```
Funcionario(id, nome, nomeDept, salario)
```

  - Obter todas as colunas de todas as tuplas cujo valor na coluna `salario` seja maior que `1.000`
    - $\sigma_{salario>1000}(Funcionario)$
  - Obter todas as colunas de todas as tuplas cujo valor na coluna `salario` seja maior que `1.000` e `nomeDept` igual a `vendas`
    - $\sigma_{salario>1000 \ \wedge \ nomeDept='vendas'}(Funcionario)$

Projeção – Π (pi)

- Seleciona um conjunto de colunas de uma relação e elimina as demais
- **Notação:** $\Pi_{C_1,\ldots,C_n}(R)$
- Exemplo

```
Funcionario(id, nome, nomeDept, salario)
```

  - Obter uma listagem contendo as colunas id, nome e salário de todos os funcionários
  - $\Pi_{id,nome,salario}(Funcionario)$

Composição das operações relacionais

```
Funcionario(id, nome, nomeDept, salario)
```

- Operações podem ser compostas e uma **expressão de álgebra relacional**
- Ex: Nome e salário de todos os funcionários com salário maior que `1.000`
  - $\Pi_{nome,salario}(\sigma_{salario>1000}(Funcionario))$ ou $\sigma_{salario>1000}(\Pi_{nome,salario}(Funcionario))$

Composição das operações relacionais

```
Funcionario(id, nome, nomeDept, salario)
```

- Operações podem ser compostas e uma **expressão de álgebra relacional**
- Ex: Nome e salário de todos os funcionários com salário maior que `1.000`
  - $\Pi_{nome,salario}(\sigma_{salario>1000}(Funcionario))$ ou $\sigma_{salario>1000}(\Pi_{nome,salario}(Funcionario))$

- $R1 \leftarrow \sigma_{salario>1000}(Funcionario)$
- $R2 \leftarrow \Pi_{nome,salario}(R1)$

Exercícios

- Acesse https://bcd29008.github.io/relax e carregue o gist `34bb7c2574120aa2bf461e9b8d679d1e`

```
Funcionario(id, nome, nomeDept, salario, anoNasc, mesNasc, diaNasc)
```

1. Selecione todos os funcionários do departamento de “TI”
2. Selecione todos os funcionários que nasceram em 2000 e que trabalham no departamento de “P&D”
3. Selecione o nome de todos os funcionários e o nome do departamento onde cada um trabalha
4. Selecione o nome todos os funcionários que fazem aniversário em “maio”

Exercícios

- Acesse https://bcd29008.github.io/relax e carregue o gist `34bb7c2574120aa2bf461e9b8d679d1e`

```
Funcionario(id, nome, nomeDept, salario, anoNasc, mesNasc, diaNasc)
```

1. Selecione todos os funcionários do departamento de "TI"
   - $\sigma_{nomeDept='TI'}(Funcionario)$
2. Selecione todos os funcionários que nasceram em 2000 e que trabalham no departamento de "P&D"
   - $\sigma_{nomeDept='P\&D' \wedge\ ano=2000}(Funcionario)$
3. Selecione o nome de todos os funcionários e o nome do departamento onde cada um trabalha
   - $\Pi_{nome,nomeDept}(Funcionario)$
4. Selecione o nome todos os funcionários que fazem aniversário em "maio"
   - $\Pi_{nome}(\sigma_{mesNasc=5}(Funcionario))$

Atribuição – ρ (rô)

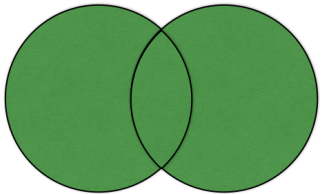
- Usada para alterar o nome da relação, das colunas ou de ambos
- **Notação:** $\rho_{B_1,\ldots,B_n}(R)$
- Exemplos:

$\rho_S(R)$	Alterando o nome de uma relação
$\rho_{B_1,\ldots,B_n}(R)$	Alterando o nome de colunas
$\rho_{S(B_1,\ldots,B_n)}(R)$	Alterando o nome da relação e das colunas

```
Funcionario(id, nome, nomeDept)
```

- $\rho_{codigo \leftarrow id, nome \leftarrow nome, departamento \leftarrow nomeDept}(Funcionario)$

União – $\cup$

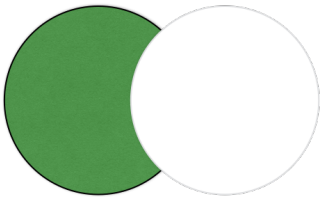


- Selecionar nome dos funcionários que trabalham no departamento de "TI" ou que supervisionam um funcionário que trabalha no departamento de "TI"

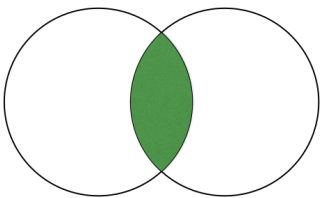
```
Funcionario(id,nome,nomeDept,nomeChefe)
```

- $\Pi_{nome}(\sigma_{nomeDept='TI'}(Funcionario)) \cup$
  $\rho_{nome}(\Pi_{nomeChefe}(\sigma_{nomeDept='TI'}(Funcionario)))$

Diferença e Interseção



- Diferença
- $R_1 - R_2$



- Interseção
- $R_1 \cap R_2 = R_1 - (R_2 - R_1)$

Produto cartesiano – $\times$

- Relaciona todas as tuplas de R_1 com todas as tuplas em R_2
- **Notação:** $R_1 \times R_2$

```
Funcionario(id, nome, sobrenome, salario, dId)
Departamento(dId, dNome, dFilial)
```

- $\Pi_{nome,sobrenome}(Funcionario) \times \Pi_{dNome}(Departamento)$

Gist para carregar no https://bcd29008.github.io/relax

af7a5fe2be634d725c3167c40c0c3022

Exemplo: Encontre o maior salário da empresa

1. Criar uma relação temporária com todos os salários menores que o maior
   - $\Pi_{Funcionario.salario}(\sigma_{Funcionario.salario<d.salario}(Funcionario \times \rho_d(Funcionario)))$
   - O uso da operação ρ foi necessária para não haver ambiguidade nos nomes das colunas das duas relações
2. Fazer a diferença de conjuntos entre a relação Funcionário e a recém criada no item anterior
   - $\Pi_{salario}(Funcionario) - \Pi_{Funcionario.salario}(\sigma_{Funcionario.salario<d.salario}(Funcionario \times \rho_d(Funcionario)))$

Junção natural – $\bowtie$

natural join

- Normalmente uma consulta que envolve um produto cartesiano inclui uma operação de seleção no resultado do produto cartesiano
- A junção natural é uma operação binária que combina as operações de produto cartesiano e seleção
  - Força a igualdade entre as colunas que aparecem em ambas as relações
  - Remove as colunas duplicadas
- Exemplo: Liste o nome de cada funcionário, bem como o nome do departamento onde trabalha: $\Pi_{nome,dNome}(Funcionario \bowtie Departamento)$

```
Funcionario(id, nome, sobrenome, dId)
Departamento(dId, dNome, dFilial)
```

Junção teta – θ

- Uma junção natural que possui um predicado θ, de forma que θ pode ser qualquer condição aceita em uma operação de seleção
  - $R_1 \bowtie_\theta R_2 = \sigma_\theta(R_1 \times R_2)$
- Pode ser usado quando as relações não possuírem uma coluna com nome em comum
- Se for usado o operador $=$, então essa junção também é chamada de *equijunção*
- Exemplo
  - $Funcionario \bowtie_{nomeDepto=dNome} Departamento$

Resumo

Símbolo	Nome	Exemplo	Resultado
σ	Seleção	$\sigma_{salario>500}(funcionario)$	Todas tuplas que satisfaçam o predicado
Π	Projeção	$\Pi_{nome,salario}(funcionario)$	Colunas nome e salário de todas as tuplas
$\times$	Produto cartesiano	$professor \times curso$	Todas tuplas independente de terem os mesmos valores nas colunas de mesmo nome
$\bowtie$	Junção	$professor \bowtie curso$	Todas tuplas que possuem o mesmo valor para colunas de mesmo nome

RelaX - relational algebra calculator

https://bcd29008.github.io/relax

- No menu lateral a esquerda clique em **Select DB** e depois informe um dos valores abaixo no campo "*Load dataset stored in a gist*" e clique em *Load*
- Funcionário e Departamento – `af7a5fe2be634d725c3167c40c0c3022`

```
Funcionario(id, nome, sobrenome, salario, dId)
Departamento(dId, dNome, dFilial)
```

- Funcionário, Empresa e Gerente – `858bd98309ee0bf936f020bfd2a5cf55`

```
Funcionario(nomeFunc, endereco, cidade)
Empresa(nomeEmp, cidade)
Trabalha(nomeFunc, nomeEmp, salario)
Gerencia(nomeFunc, nomeGerente)
```

Aulas baseadas em


Henry F.; Sudarshan Silberschatz, Abraham; Korth.
Sistemas de banco de dados.
6a. Edição - Editora Campus, 2012


Navathe, S.
Sistemas de banco de dados
4a. Edição - Editora Person Addison, 2005


Ré, C.
CS145 Introduction to databases – Stanford University