

Linguagens de Consulta

- Linguagem de consulta: é uma linguagem em que um usuário requisita informações do banco de dados (Silberchatz; Korth; Sudarshan, 2006)
 - Normalmente estão em um nível mais alto do que uma linguagem de programação padrão
- Categorias:
 - Não procedural. O usuário descreve as informações que deseja obter, sem indicar procedimentos específicos a serem executados (Cálculo Relacional)
 - Procedural. O usuário indica ao sistema a sequência de operações a serem executadas no banco de dados para obter o resultado desejado (Álgebra Relacional)

Cálculo Relacional de Tupla

- Não procedural
- É escrita uma expressão declarativa que expressa um requisito de recuperação (O que será recuperado)
 - Não é dito **como** deve ser feita esta recuperação
 - O Cálculo relacional de tupla está baseado na especificação de um número de variáveis de tupla
 - Cada variável de tupla está relacionada a uma relação
 - O valor de uma variável de tupla pode ser o de qualquer tupla de uma relação

Cálculo Relacional de Tupla

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

Onde:

t é uma variável de tupla;

$COND(t)$ é uma expressão condicional envolvendo t

Exemplo

$$\{t \mid Funcionario(t) \text{ AND } t.salário > 10.000,00\}$$

Recupera todas linhas da tabela funcionário que atendem a condição

Álgebra Relacional

- Na matemática: conjunto de objetos e um conjunto de operações sobre estes objetos.
 - Aritmética conjunto de números e de operações sobre números (soma, subtração...)
 - Álgebra (elementar) são usadas variáveis no lugar dos números
- Na álgebra relacional os objetos são relações (tabelas).

Álgebra Relacional

- Álgebra relacional: conjunto de operações aplicadas sobre relações que produzem outras relações
 - É procedural (digo como, isto é, em que ordem as operações devem ser executadas)
 - Não é implementada diretamente em nenhum SGBD

Álgebra Relacional

- Álgebra Relacional
 - Entendimento sobre as operações sobre um banco relacional
 - Compreender o processo de otimização de consultas

Álgebra Relacional: Operações Unárias

- Seleção
 - Subconjunto de tuplas de uma relação que satisfaçam uma condição (linhas de uma tabela)
- Projeção
 - Seleciona atributos de uma relação (colunas de uma tabela)

Álgebra Relacional: Operações Unárias

- Seleção

- Resultado é um subconjunto de tuplas que satisfaça uma condição
- Representada por σ (sigma)
- $\sigma \langle \text{condição} \rangle (R)$
 - *R pode ser uma expressão de álgebra relacional*
 - *Podem ser usados conectivos: \wedge, \vee*
- Exemplo:
 $\sigma \text{ salário} > 10.000,00 (\text{Funcionário})$
Lista todos os funcionários com salário maior que 10.000,00

Em SQL: `Select * from funcionário where salário > 10000,00`

Álgebra Relacional: Operações Unárias

- Projeção
 - Resultado é um subconjunto de atributos
 - Representada por π (π)
 - $\pi \langle \text{lista de atributos} \rangle (R)$
 - *R pode ser uma expressão de álgebra relacional*
 - Exemplo:
 $\pi \text{ nomfunc (Funcionário)}$
Lista todos os nomes dos funcionários

Em SQL: `Select nomfunc from funcionário`

Álgebra Relacional: Combinando

π nomfunc (σ salário > 10.000,00 (Funcionário))

Em SQL: select nomfunc

from funcionário

where salário > 10000,00;

Produto Cartesiano

- Produto Cartesiano (*Cross Product*)
 - Resultado é a combinação de atributos e tuplas de duas relações
 - Representado por: \times
 $R \times S$
 - *Relação resultante = (atributos de R + atributos de S)*
 - *Cada tupla de R combinada com cada tupla de S*

Produto Cartesiano

Departamento

coddepto	Nomdepto
1	d1
2	d2
3	d3

Funcionário

codfunc	nomfunc	coddepto	Salario
1	João	2	R\$ 1.000,00
2	Maria	1	R\$ 3.000,00
3	Jose	2	R\$ 4.000,00
4	Ana	NULL	R\$ 3.000,00

Departamento × Funcionário

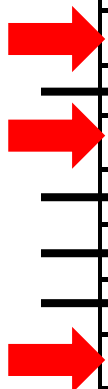
*Em SQL: select * from departamento, funcionario;*

Produto Cartesiano

Codfunc	Nomfunc	Coddepto	Salario	coddepto	nomdepto
1	Joao	2	R\$1.000,00	1	d1
1	Joao	2	R\$1.000,00	2	d2
1	Joao	2	R\$1.000,00	3	d3
2	Maria	1	R\$3.000,00	1	d1
2	Maria	1	R\$3.000,00	2	d2
2	Maria	1	R\$3.000,00	3	d3
3	Jose	2	R\$4.000,00	1	d1
3	Jose	2	R\$4.000,00	2	d2
3	Jose	2	R\$4.000,00	3	d3
4	Ana		R\$3.000,00	1	d1
4	Ana		R\$3.000,00	2	d2
4	Ana		R\$3.000,00	3	d3

Departamento × Funcionário

Produto Cartesiano



Codfunc	Nomfunc	Coddepto	Salario	coddepto	nomdepto
1	Joao	2	R\$1.000,00	1	d1
1	Joao	2	R\$1.000,00	2	d2
1	Joao	2	R\$1.000,00	3	d3
2	Maria	1	R\$3.000,00	1	d1
2	Maria	1	R\$3.000,00	2	d2
2	Maria	1	R\$3.000,00	3	d3
3	Jose	2	R\$4.000,00	1	d1
3	Jose	2	R\$4.000,00	2	d2
3	Jose	2	R\$4.000,00	3	d3
4	Ana		R\$3.000,00	1	d1
4	Ana		R\$3.000,00	2	d2
4	Ana		R\$3.000,00	3	d3

σ Departamento.codepto = Funcionário.codepto
(Departamento \times Funcionário)

Em SQL: `select * from departamento, funcionario`
`where departamento.coddepto = funcionario.coddepto`

Junção

- Combina as tuplas de duas relações *quando estas* satisfazem uma determinada condição
- Representado por: \bowtie

$\langle \text{tabela1} \rangle \bowtie \langle \text{condição de junção} \rangle \langle \text{tabela2} \rangle$

Funcionário \bowtie $\langle \text{funcionário.coddepto} = \text{departamento.coddepto} \rangle$ Departamento

```
select * from
    funcionario inner join departamento
    on
    funcionario.coddepto = departamento.coddepto;
```

Referências

- ELMASRI, R.; NAVATHE, S. B. . Sistemas de Banco de Dados. 6. ed. São Paulo: Pearson Education, 2010.