

Álgebra Relacional e SQL

operações de interesse

Banco de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Álgebra Relacional

- Maneira teórica de se manipular o banco de dados relacional
- Linguagem de consulta procedural
 - usuários especificam os dados necessários e como obtê-los
- Consiste de um conjunto de operações
 - entrada: uma ou duas relações
 - saída: uma nova relação resultado

Operações

- Fundamentais
 - seleção
 - projeção
 - produto cartesiano
 - renomear
 - união
 - diferença de conjuntos
- Adicionais
 - intersecção de conjuntos
 - junção natural
 - divisão
 - agregação

SQL DML

```
SELECT <lista de atributos e funções>  
FROM <lista de tabelas>  
[ WHERE predicado ]  
[ GROUP BY <atributos de agrupamento> ]  
[ HAVING <condição para agrupamento> ]  
[ ORDER BY <lista de atributos> ] ;
```

Álgebra Relacional: Seleção

- **Seleciona tuplas** que satisfaçam à condição de seleção

$\sigma_{\text{condição_seleção}}$ (relação argumento)

- pode envolver operadores de comparação
(=, >, ≥, <, ≤, ≠)
- pode combinar condições usando-se \wedge , \vee , \neg

- relação
- resultado de alguma operação da álgebra relacional

Álgebra Relacional: Seleção

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

Álgebra Relacional: Projeção

- **Projeta** as **colunas** solicitadas (i.e. produz um subconjunto vertical)

$\pi_{\text{lista_atributos}}$ (relação argumento)

- lista de atributos
- os atributos são separados por vírgula

- relação
- resultado de alguma operação da álgebra relacional

Álgebra Relacional: Projeção

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

Álgebra Relacional: Produto Cartesiano

- **Combina tuplas** de duas relações
 - relações não precisam ter atributos comum
- Tuplas da relação resultante
 - todas as combinações de tuplas possíveis entre as relações participantes

relação argumento 1 \times relação argumento 2



- relação
- resultado de alguma operação da álgebra relacional

Relações Cliente e Vendedor

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

vendedor (cod_vend, nome_vend)

cod_vend	nome_vend
1	Adriana
2	Roberto

Cliente × Vendedor

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

grau: número de atributos de cliente + número de atributos de vendedor

número de tuplas: número de tuplas de cliente * número de tuplas de vendedor

SQL: SELECT-FROM-WHERE

```
SELECT <lista de atributos>  
FROM <lista de tabelas>  
[WHERE condições de seleção]
```

SQL	Álgebra Relacional
SELECT	projeção
FROM	produto cartesiano
WHERE	seleção

Álgebra Relacional: Junção

- Concatena tuplas relacionadas de duas relações
- Passos:
 - forma um produto cartesiano das relações
 - faz uma seleção forçando igualdade sobre os atributos que aparecem nas relações

Álgebra Relacional: Junção

- Sintaxe

relação argumento 1 $\bowtie_{\text{condição_junção}}$ relação argumento 2



- relação
- resultado de alguma operação da álgebra relacional

Junção (Exemplo)

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

- **Passo 1:**
 - formar um produto cartesiano das relações

Junção (Exemplo)

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

- Passo 2:
 - fazer uma seleção forçando igualdade sobre os atributos que aparecem nas relações

Junção (Exemplo)

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

SQL: Junção (Primeiras Versões)

- Cláusulas **SELECT** e **WHERE**
 - especificam atributos com mesmo nome usando o nome da relação e o nome do atributo (nome_relação.nome_atributo)
- Cláusula **FROM**
 - possui mais do que uma relação
- Cláusula **WHERE**
 - inclui as condições de junção (igualdade sobre os atributos que aparecem nas relações)

Junção (Exemplo)

```
SELECT nro_cli, nome_cli, end_cli,  
       saldo, vendedor.cod_vend,  
       nome_vend  
FROM cliente, vendedor  
WHERE cliente.cod_vend =  
       vendedor.cod_vend
```

SQL-92: Junção

```
SELECT nro_cli, nome_cli, end_cli,  
       saldo, vendedor.cod_vend,  
       nome_vend  
FROM cliente JOIN vendedor ON  
       cliente.cod_vend =  
       vendedor.cod_vend
```

Operações sobre Conjuntos

- Unem duas relações

- Operações

- união
- intersecção
- diferença

- Características

- atuam sobre relações compatíveis
- eliminam tuplas duplicadas da relação resultado

Duas relações são compatíveis se:

- possuem o mesmo grau
- seus atributos possuem os mesmos domínios (os domínios dos i -ésimos atributos de cada relação são os mesmos)

Álgebra Relacional: Operações sobre Conjuntos

- União ($R \cup S$)
 - gera uma relação que contém todas as tuplas pertencentes a R , a S , ou a ambas R e S
- Intersecção ($R \cap S$)
 - gera uma relação que contém todas as tuplas pertencentes tanto a R quanto a S
- Diferença ($R - S$)
 - gera uma relação que contém todas as tuplas pertencentes a R que não pertencem a S

SQL: Operações sobre Conjuntos

SQL	Álgebra Relacional
UNION	\cup
INTERSECT	\cap
MINUS	$-$

- Observação
 - operações oferecidas dependem do SGBD

Exemplo

- Liste os nomes dos clientes que possuem nomes iguais aos nomes de vendedores.

```
SELECT nome_cli
```

```
FROM cliente
```

```
INTERSECT
```

```
SELECT nome_vend
```

```
FROM vendedor
```