

Banco de Dados I

09 - Álgebra Relacional

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

Introdução

- A *Álgebra Relacional* é uma linguagem muito importante utilizada no modelo relacional
- Ela é composta por um conjunto de operadores que, quando combinados, permitem realizar diversos tipos de operações sobre uma relação ou um conjunto de relação
- A álgebra relacional está relacionada com a linguagem SQL, sendo que os SGBD atuais traduzem as consultas SQL para expressões da álgebra relacional para realizar o processamento das consultas
- Toda operação da Álgebra Relacional possui uma ou mais relações como entrada e uma relação como saída

Principais Operações da Álgebra Relacional

- Seleção (σ)
- Projeção (π)
- Renomeação (ρ)
- Junção (\bowtie)
- Funções de agregação (γ)
- União (\cup)
- Interseção (\cap)
- Diferença ($-$)
- Produto Cartesiano (\times)

- A operação de seleção (σ) permite selecionar tuplas de uma relação que atendam a certas condições

$$\sigma_{\langle \text{condição} \rangle}(\langle \text{relação} \rangle)$$

- A condição pode conter comparações de atributos e valores usando os operadores $=$, \neq , $<$, \leq , $>$ e \geq
- As comparações podem ser combinadas com os conectivos \wedge , \vee e \neg

Exemplo de Seleção

Relação cliente

	id_cliente	nome_cliente	endereco	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe os clientes com saldo maior ou igual a 100”

SQL

```
SELECT * FROM cliente  
WHERE saldo >= 100;
```

Expressão Algébrica

$$\sigma_{\text{saldo} \geq 100}(\text{cliente})$$

Exemplo de Seleção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe os clientes com saldo maior ou igual a 100”

SQL

```
SELECT * FROM cliente  
WHERE saldo >= 100;
```

Expressão Algébrica

$$\sigma_{\text{saldo} \geq 100}(\text{cliente})$$

- A operação de projeção projeta as colunas selecionadas de uma relação (é uma espécie de *seleção vertical*)

$$\pi_{\langle \text{atributos} \rangle}(\langle \text{relação} \rangle)$$

- Os atributos são separados por vírgula

Exemplo de Projeção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe o nome e endereço dos clientes”

SQL

```
SELECT nome_cliente, endereço  
FROM cliente;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{endereço}}(\text{cliente})$$

Exemplo de Projeção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe o nome e endereço dos clientes”

SQL

```
SELECT nome_cliente, endereço  
FROM cliente;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{endereço}}(\text{cliente})$$

Exemplo de Seleção e Projeção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe o nome e endereço dos clientes com saldo maior ou igual 100”

SQL

```
SELECT nome_cliente, endereço
FROM cliente
WHERE saldo >= 100;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{endereço}}(\sigma_{\text{saldo} \geq 100}(\text{cliente}))$$

Exemplo de Seleção e Projeção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo
t_1	1	José	Rua X	90,00
t_2	2	Cristina	Avenida 1	110,00
t_3	3	Tadeu	Avenida 3	234,00
t_4	4	Rodrigo	Rua X	37,00

- Consulta: “Informe o nome e endereço dos clientes com saldo maior ou igual 100”

SQL

```
SELECT nome_cliente, endereço
FROM cliente
WHERE saldo >= 100;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{endereço}}(\sigma_{\text{saldo} \geq 100}(\text{cliente}))$$

Produto Cartesiano

- A operação de produto cartesiano combina as tuplas de duas relações

$$\langle \textit{relação1} \rangle \times \langle \textit{relação2} \rangle$$

- O produto cartesiano retorna todas as combinações de tuplas possíveis
- Se a *relação1* possui n tuplas e a *relação2* possui m tuplas, a relação resultante terá $n \times m$ tuplas
- As tuplas resultantes terão os atributos das duas relações de entrada

Exemplo de Produto Cartesiano I

Relação cliente

	id_cliente	nome_cliente	endereço	saldo	id_vend
t_1	1	José	Rua X	90,00	10
t_2	2	Cristina	Avenida 1	110,00	10
t_3	3	Tadeu	Avenida 3	234,00	20
t_4	4	Rodrigo	Rua X	37,00	20

Relação vendedor

	id_vend	nome_vend
t'_1	10	João
t'_2	20	Maria

- Consulta: “Informe os cliente combinados com os vendedores”

Exemplo de Produto Cartesiano II

SQL

```
SELECT *  
FROM cliente, vendedor;
```

Expressão Algébrica

cliente \times vendedor

Resultado

	id_cliente	nome_cliente	endereço	saldo	id_vend	id_vend'	nome_vend
$t_1 \times t_1$	1	José	Rua X	90,00	10	10	João
$t_2 \times t_1$	2	Cristina	Avenida 1	110,00	10	10	João
$t_3 \times t_1$	3	Tadeu	Avenida 3	234,00	20	10	João
$t_4 \times t_1$	4	Rodrigo	Rua X	37,00	20	10	João
$t_1 \times t_2$	1	José	Rua X	90,00	10	20	Maria
$t_2 \times t_2$	2	Cristina	Avenida 1	110,00	10	20	Maria
$t_3 \times t_2$	3	Tadeu	Avenida 3	234,00	20	20	Maria
$t_4 \times t_2$	4	Rodrigo	Rua X	37,00	20	20	Maria

- A operação de renomeação permite renomear relações e atributos

$$\rho_{\langle \text{relação}' \rangle}(\langle \text{relação} \rangle)$$

$$\rho(\langle A'_1 \rangle, \dots, \langle A'_n \rangle)(\langle \text{relação} \rangle)$$

$$\rho_{\langle \text{relação}' \rangle}(\langle A'_1 \rangle, \dots, \langle A'_n \rangle)(\langle \text{relação} \rangle)$$

- A renomeação é especialmente útil quando uma tabela precisa de ser usada mais de uma vez em uma expressão

Exemplo de Renomeação

SQL

```
SELECT * FROM cliente AS c;
```

Expressão Algébrica

$$\rho_c(\text{cliente})$$

SQL

```
SELECT id_cliente AS id,  
       nome_cliente AS n,  
       endereco AS e,  
       saldo  
FROM cliente;
```

Expressão Algébrica

$$\rho_{(id,n,e,saldo)}(\text{cliente})$$

SQL

```
SELECT id_cliente AS id,  
       nome_cliente AS n,  
       endereco AS e,  
       saldo  
FROM cliente AS c;
```

Expressão Algébrica

$$\rho_{c(id,n,e,saldo)}(\text{cliente})$$

Operações com Conjuntos

- As operações com conjuntos permitem realizar a união, interseção ou diferença entre duas relações

$$\langle \text{relação1} \rangle \cup \langle \text{relação2} \rangle$$

$$\langle \text{relação1} \rangle \cap \langle \text{relação2} \rangle$$

$$\langle \text{relação1} \rangle - \langle \text{relação2} \rangle$$

- As duas relações envolvidas na operação precisam ter o mesmo número de atributos e os domínios dos atributos correspondentes precisam ser idênticos

Exemplo de Operações com Conjuntos

SQL

```
SELECT nome_cliente FROM cliente
UNION
SELECT nome_vend FROM vendedor;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}}(\text{cliente}) \cup \pi_{\text{nome_vend}}(\text{vendedor})$$

SQL

```
SELECT nome_cliente FROM cliente
INTERSECT
SELECT nome_vend FROM vendedor;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}}(\text{cliente}) \cap \pi_{\text{nome_vend}}(\text{vendedor})$$

SQL

```
SELECT nome_cliente FROM cliente
EXCEPT
SELECT nome_vend FROM vendedor;
```

Expressão Algébrica

$$\pi_{\text{nome_cliente}}(\text{cliente}) - \pi_{\text{nome_vend}}(\text{vendedor})$$

Junção

- A operação de junção permite combinar tuplas de duas relações considerando comparações entre os atributos destas relações
- Existem diversas variantes na operação de junção, mas vamos considerar primeiro a junção condicional

$$\langle \text{relação1} \rangle \bowtie_{\langle \text{condição} \rangle} \langle \text{relação2} \rangle$$

- A $\langle \text{condição} \rangle$ da junção é semelhante a condição da seleção, mas devem haver comparações entre os atributos da $\langle \text{relação1} \rangle$ e da $\langle \text{relação2} \rangle$
- Quando as relações participantes possuem atributos homônimos, pode ser feita a *junção natural* sobre estes atributos sem precisar de nenhuma condição
- Também existem traduções para as junções externas da linguagem SQL:
 - $A \bowtie B$: **LEFT JOIN**
 - $A \ltimes B$: **RIGHT JOIN**
 - $A \Join B$: **LEFT RIGHT JOIN**

Exemplo de Junção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo	id_vend
t_1	1	José	Rua X	90,00	10
t_2	2	Cristina	Avenida 1	110,00	10
t_3	3	Tadeu	Avenida 3	234,00	20
t_4	4	Rodrigo	Rua X	37,00	20

Relação vendedor

	id_vend	nome_vend
t'_1	10	João
t'_2	20	Maria

- Consulta: "Informe os cliente e seus respectivos vendedores"

SQL

```
SELECT * FROM cliente, vendedor
WHERE cliente.id_vend = vendedor.id_vend;
```

Expressão Algébrica

$$\text{cliente} \bowtie_{(\text{cliente.id_vend}=\text{vendedor.id_vend})} \text{vendedor}$$

Exemplo de Junção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo	id_vend
t ₁	1	José	Rua X	90,00	10
t ₂	2	Cristina	Avenida 1	110,00	10
t ₃	3	Tadeu	Avenida 3	234,00	20
t ₄	4	Rodrigo	Rua X	37,00	20

Relação vendedor

	id_vend	nome_vend
t' ₁	10	João
t' ₂	20	Maria

- Consulta: “Informe os cliente e seus respectivos vendedores”

SQL

```
SELECT * FROM cliente, vendedor
WHERE cliente.id_vend = vendedor.id_vend;
```

Expressão Algébrica

cliente $\bowtie_{(\text{cliente.id_vend}=\text{vendedor.id_vend})}$ vendedor

Resultado

	id_cliente	nome_cliente	endereço	saldo	id_vend	id_vend'	nome_vend
t'' ₁	1	José	Rua X	90,00	10	10	João
t'' ₂	2	Cristina	Avenida 1	110,00	10	10	João

Exemplo de Junção

Relação cliente

	id_cliente	nome_cliente	endereço	saldo	id_vend
t_1	1	José	Rua X	90,00	10
t_2	2	Cristina	Avenida 1	110,00	10
t_3	3	Tadeu	Avenida 3	234,00	20
t_4	4	Rodrigo	Rua X	37,00	20

Relação vendedor

	id_vend	nome_vend
t'_1	10	João
t'_2	20	Maria

- Consulta: “Informe os cliente e seus respectivos vendedores”

SQL

```
SELECT * FROM cliente, vendedor
WHERE cliente.id_vend = vendedor.id_vend;
```

Expressão Algébrica

$\text{cliente} \bowtie_{(\text{cliente.id_vend}=\text{vendedor.id_vend})} \text{vendedor}$

Resultado

	id_cliente	nome_cliente	endereço	saldo	id_vend	id_vend'	nome_vend
t''_1	1	José	Rua X	90,00	10	10	João
t''_2	2	Cristina	Avenida 1	110,00	10	10	João
t''_3	3	Tadeu	Avenida 3	234,00	20	20	Maria
t''_4	4	Rodrigo	Rua X	37,00	20	20	Maria

Função de Agregação

- A operação de função de agregação agrupar tuplas e sumarizar dados de atributos

$$\langle A_1 \rangle, \dots, \langle A_n \rangle \gamma \langle F_1(A'_1) \rangle, \dots, \langle F_m(A'_m) \rangle (\langle \text{relação} \rangle)$$

- Os atributos $\langle A_1 \rangle, \dots, \langle A_n \rangle$ são usados para agrupar os dados e as funções $\langle F_1(A'_1) \rangle, \dots, \langle F_m(A'_m) \rangle$ realizar a sumarização sobre os atributos A'_1, \dots, A'_m
- Podem ser usadas as mesmas funções da linguagem SQL (AVG, SUM, MAX, MIN, COUNT, etc.)

Exemplo de Função de Agregação

Relação venda

id_cliente	mes	valor
1	2015-02	470,00
1	2015-03	390,00
3	2015-03	230,00
2	2015-04	210,00
3	2015-04	140,00
2	2015-05	110,00
2	2015-05	480,00

- Consulta: “Informe total de vendas de cada mês”

SQL

```
SELECT mes, SUM(valor)
FROM venda
GROUP BY mes;
```

Expressão Algébrica

$mes \gamma \text{SUM(valor)} (venda)$

Resultado

mes	SUM(valor)
2015-02	470,00
2015-03	620,00
2015-04	350,00
2015-05	590,00

Exemplo de Função de Agregação

Relação venda

id_cliente	mes	valor
1	2015-02	470,00
1	2015-03	390,00
3	2015-03	230,00
2	2015-04	210,00
3	2015-04	140,00
2	2015-05	110,00
2	2015-05	480,00

- Consulta: “Informe total de vendas de cada mês”

SQL

```
SELECT mes, SUM(valor)
FROM venda
GROUP BY mes;
```

Expressão Algébrica

$\text{mes} \searrow \text{SUM}(\text{valor})(\text{venda})$

Resultado

mes	SUM(valor)
2015-02	470,00
2015-03	620,00
2015-04	350,00
2015-05	590,00

Planos de Execução de Consultas

- Uma das importantes aplicações da álgebra relacional está relacionada ao processamento de consultas
- Os SGBD traduzem as consultas para expressões algébricas e, posteriormente, para planos de execução representados por árvores
- Exemplo:

Consulta

```
SELECT nome_cliente, nome_vendedor
FROM cliente, vendedor
WHERE cliente.id_vendedor =
  ↪ vendedor.id_vendedor;
```

Plano de Execução



Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{nome_vendedor}}(\sigma_{\text{saldo} > 100}(\text{cliente} \bowtie_{\text{id_vendedor}=\text{id_vendedor}} \text{vendedor}))$$

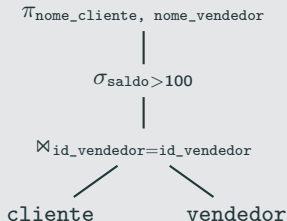
Planos de Execução de Consultas

- Uma das importantes aplicações da álgebra relacional está relacionada ao processamento de consultas
- Os SGBD traduzem as consultas para expressões algébricas e, posteriormente, para planos de execução representados por árvores
- Exemplo:

Consulta

```
SELECT nome_cliente, nome_vendedor
FROM cliente, vendedor
WHERE cliente.id_vendedor =
  ↪ vendedor.id_vendedor;
```


Plano de Execução




Expressão Algébrica

$$\pi_{\text{nome_cliente}, \text{nome_vendedor}}(\sigma_{\text{saldo} > 100}(\text{cliente} \bowtie_{\text{id_vendedor}=\text{id_vendedor}} \text{vendedor}))$$

 (2012).
Postgresql documentation.

 Elmasri, R. and Navathe, S. B. (2011).
Sistemas de banco de dados.
Pearson Addison Wesley, São Paulo, 6 edition.

 Ramakrishnan, R. and Gehrke, J. (2008).
Sistemas de gerenciamento de banco de dados.
McGrawHill, São Paulo, 3 edition.