

# Álgebra Relacional

...

BD I

# Introdução

- Proposta por Edgar Codd como linguagem formal de consulta aos dados relacionais
- Possui 5 operadores primitivos (veremos 3)
- Assemelha-se a álgebra matemática elementar
  - Baseada em operadores **unários** e **binários**
  - Adição, por exemplo, é um operador binário  $A + B$ . Se quisermos somar 3 valores temos que utilizar 2 vezes o  $+$ :  $A + B + C$
  - Potenciação é unário:  $A^2$  (não podemos elevar mais de um valor ao mesmo tempo)

# Projeção

- Operadores primitivos
  - Projeção (unário): projeta alguns atributos da relação (faz um corte vertical na tabela)
    - $\pi_{(\text{atributos})}(\text{tabela})$
    - $\pi_{(\text{nome}, \text{pop})}(\text{cidade})$

nome	pop
Chapeco	200
Erechim	105
Passo Fundo	204
Realeza	17.07
Itajai	200.5
Ponta Grossa	355.3

```
select nome, pop  
from cidade;
```

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Seleção

- Operadores primitivos
  - Seleção (unário): seleciona tuplas de uma relação baseado em critérios de seleção (condição)
    - $\sigma_{(\text{condição})}(\text{tabela})$
    - $\sigma_{(\text{pop} > 200)}(\text{cidade})$

codcid	nome	pop	area	uf
3	Passo Fundo	204	783.4	RS
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

```
select * from cidade where pop > 200;
```

# Projeção e Seleção

- Os operadores da álgebra são fechados
  - O resultado de um pode (**e deve**) ser utilizado como entrada de um outro.
    - $\pi_{(\text{nome}, \text{pop})}(\text{cidade})$
    - $\sigma_{(\text{pop} > 200)}(\text{cidade})$
    - $\pi_{(\text{nome}, \text{pop})}(\sigma_{(\text{pop} > 200)}(\text{cidade}))$

nome	pop
Passo Fundo	204
Itajai	200.5
Ponta Grossa	355.3

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

```
select nome, pop from cidade where pop > 200;
```

# Projeção e Seleção

- A ordem, como na matemática, não altera o valor do resultado
  - 1:  $\pi_{(\text{nome}, \text{pop})}(\sigma_{(\text{pop} > 200)}(\text{cidade}))$
  - 2:  $\sigma_{(\text{pop} > 200)}(\pi_{(\text{nome}, \text{pop})}(\text{cidade}))$
  - Ambas produzem o mesmo resultado


nome	pop
Passo Fundo	204
Itajai	200.5
Ponta Grossa	355.3

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Projeção e Seleção

- Porém, CUIDADO:

- 1:  $\pi_{(\text{nome}, \text{pop})}(\sigma_{(\text{area} > 500)}(\text{cidade})) \leftarrow \text{OK}$
- 2:  $\sigma_{(\text{area} > 500)}(\pi_{(\text{nome}, \text{pop})}(\text{cidade})) \leftarrow \text{Não OK}$
- Por quê?
- A projeção da consulta 2 resulta:



nome	pop
Chapeco	200
Erechim	105
Passo Fundo	204
Realeza	17.07
Itajai	200.5
Ponta Grossa	355.3

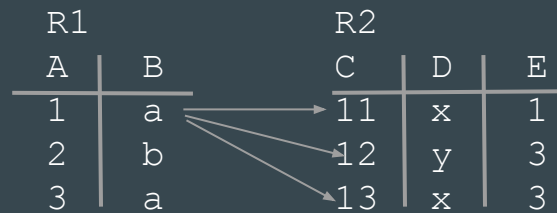
- A seleção precisa do atributo **area** que não está no resultado da projeção

# Produto Cartesiano

- Produto cartesiano (binário): multiplicação entre duas relações

- Existe apenas para explicar o join
- $tabela\_1 \times tabela\_2$
- $R1 \times R2$

A	B	C	D	E
1	a	11	x	1
1	a	12	y	3
1	a	13	x	3
2	b	11	x	1
2	b	12	y	3
2	b	13	x	3
3	a	11	x	1
3	a	12	y	3
3	a	13	x	3





# Produto Cartesiano

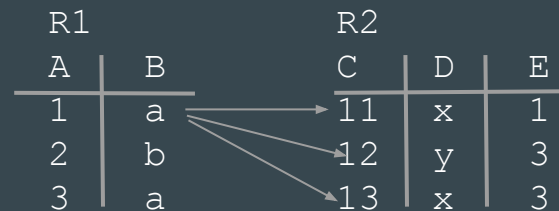
- Produto cartesiano (binário): multiplicação entre duas relações

- Existe apenas para explicar o join

- $tabela\_1 \times tabela\_2$

- $R1 \times R2$

A	B	C	D	E
1	a	11	x	1
1	a	12	y	3
1	a	13	x	3
2	b	11	x	1
2	b	12	y	3
2	b	13	x	3
3	a	11	x	1
3	a	12	y	3
3	a	13	x	3



- Como explica o join?

- $\sigma_{(A=E)}(R1 \times R2)$

A	B	C	D	E
1	a	11	x	1
3	a	12	y	3
3	a	13	x	3

`select * from R1,R2 where A=E;`

# Join

- Join (binário): operador não primitivo, formado pelo produto cartesiano e pela seleção:

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

- Inner join:  $\bowtie$
- Outer join:  $\ltimes$  (left),  $\rtimes$  (right) e  $\Join$  (full)

○ cidade  $\bowtie$  estado

$\text{cidade.uf} = \text{estado.uf}$

■ `select * from cidade join estado on cidade.uf=estado.uf;`

○ estado  $\Join$  cidade

$\text{estado.uf} = \text{cidade.uf}$

`select * from estado left join cidade  
on estado.uf=cidade.uf`

uf	nome	nome	pop	area	uf
SC	Santa Catarina	Chapeco	200	626.06	SC
RS	Rio Grande do Sul	Erechim	105	430.7	RS
RS	Rio Grande do Sul	Passo Fundo	204	783.7	RS
PR	Paraná	Realeza	17.07	353.4	PR
SC	Santa Catarina	Itajai	200.5	289.3	SC
PR	Paraná	Ponta Grossa	355.3	2054	PR

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Join

- Join (binário): operador não primitivo, formado pelo produto cartesiano e pela seleção:

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

- Inner join:  $\bowtie$
- Outer join:  $\ltimes$  (left),  $\rtimes$  (right) e  $\Join$  (full)
- cidade  $\bowtie$  estado

■ `select * from cidade join estado on cidade.uf=estado.uf;`

○ `estado  $\Join$  cidade` `select * from estado left join cidade on estado.uf=cidade.uf`

uf	nome	nome	pop	area	uf
SC	Santa Catarina	Chapeco	200	626.06	SC
RS	Rio Grande do Sul	Erechim	105	430.7	RS
RS	Rio Grande do Sul	Passo Fundo	204	783.7	RS
PR	Paraná	Realeza	17.07	353.4	PR
SC	Santa Catarina	Itajai	200.5	289.3	SC
PR	Paraná	Ponta Grossa	355.3	2054	PR
SP	São Paulo				

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Consultas

- Unindo os comandos:

- $\pi_{(estado.nome, cidade.nome, pop)}(cidade \bowtie_{cidade.uf=estado.uf} estado)$

nome	nome	pop
Santa Catarina	Chapecó	200
Rio Grande do Sul	Erechim	105
Rio Grande do Sul	Passo Fundo	204
Paraná	Realeza	17.07
Santa Catarina	Itajaí	200.5
Paraná	Ponta Grossa	355.3

estado	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

```
select e.nome, c.nome, c.pop
from cidade c join estado e on c.uf=e.uf
```

- $\pi_{(estado.nome, cidade.nome, pop)}(\sigma_{(area > 400)}(cidade \bowtie_{cidade.uf=estado.uf} estado))$

nome	nome	pop
Santa Catarina	Chapecó	200
Rio Grande do Sul	Erechim	105
Rio Grande do Sul	Passo Fundo	204
Paraná	Ponta Grossa	355.3

```
select e.nome, c.nome, c.pop
from cidade c join estado e on c.uf=e.uf
where area > 400;
```

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Operadores de conjunto

- A álgebra também oferece os seguintes operadores de conjunto (todos binários):

- União:  $\cup$
- Intersecção:  $\cap$
- Diferença:  $\setminus$  ou  $-$
- $\pi_{\text{nome}}(\text{cidade}) \cup \pi_{\text{nome}}(\text{estado})$
- Divisão:  $\div$

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

nome  
Santa Catarina  
Rio Grande do Sul  
Paraná  
Chapeco  
Erechim  
Passo Fundo  
Realeza  
Itajai  
Ponta Grossa

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Divisão ÷

- Divisão: útil para consultas do tipo “**todos que participaram de tudo**”
  - **cliente** (id, cnome) **banco** (id, bnome) **agencia**(idb(banco), id, anome)  
**contac**(idb(agencia), ida(agencia), idc(cliente), saldo)
  - Retornar os clientes que têm contas em todas as agências do banco **12**
  - $\rho(\text{cliente}, \text{agencia}) (\pi_{\text{cnome}, \text{anome}} (\text{cliente} \bowtie \text{agencia})) \div \rho(\text{agencia}) (\pi_{\text{anome}} (\sigma_{\text{id}=12} (\text{banco}) \bowtie \text{agencia}))$

# Funções de agregação (grupo)

- Funcionamento similar ao SQL porém não existe o group by

- $\mathcal{G}_{\text{count}(\text{atr})}(\text{tabela})$ : conta as tuplas do resultado da consulta
- $\mathcal{G}_{\text{sum}(\text{atr})}(\text{tabela})$ : soma os valores do **atributo**
- $\mathcal{G}_{\text{max}(\text{atr})}(\text{tabela})$ : retorna o maior valor do **atributo**
- $\mathcal{G}_{\text{min}(\text{atr})}(\text{tabela})$ : retorna o menor valor do **atributo**
- $\mathcal{G}_{\text{avg}(\text{atr})}(\text{tabela})$ : retorna a média dos valores do **atributo**
- $\mathcal{G}_{\text{count}(*)}(\text{estado})$        $\mathcal{G}_{\text{max}(\text{pop})}(\text{cidade})$

$\frac{\text{count}}{4}$

```
select count(*)  
from estado;
```

$\frac{\text{max}}{355.3}$

```
select max(pop)  
from cidade;
```

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Funções de agregação (grupo)

- Agrupamento dos resultados:

- As funções de agregação podem ser agrupadas por atributos

- atributo(s)  $\mathcal{G}_{\text{count(atr)}}(\text{tabela})$

- $\text{estado.uf} \mathcal{G}_{\text{count(*)}}(\text{cidade} \bowtie_{\text{cidade.uf=estado.uf}} \text{estado})$

uf	count
SC	2
RS	2
PR	2

```
select e.uf, count(*)
from cidade c join estado e on c.uf=e.uf
group by e.uf;
```

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

$\text{uf} \mathcal{G}_{\text{max(pop)}}(\text{cidade})$

uf	max
SC	200.5
RS	204
PR	355.3

```
select uf, max(pop)
from cidade
group by uf;
```



# Funções de agregação (grupo)

- Detalhes de valores nulos

- As funções de agregação podem ser agrupadas por atributos

- $\text{atributo(s)} \mathcal{G}_{\text{count(atr)}}(\text{tabela})$

- $\text{estado.uf} \mathcal{G}_{\text{count(*)}}(\text{cidade} \bowtie_{\text{cidade.uf=estado.uf}} \text{estado})$

uf	count
SC	2
RS	2
PR	2

```
select e.uf, count(*)  
from cidade c join estado e on c.uf=e.uf  
group by e.uf;
```

$\text{estado.uf} \mathcal{G}_{\text{count(cidade.uf)}}(\text{estado} \bowtie)$

uf	count
SC	2
RS	2
PR	2
SP	0

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

```
select e.uf, count(c.uf)  
from estado e left join cidade c on e.uf=c.uf  
group by e.uf;
```

# Funções de agregação

- Consultas finais

- Retornar o nome da cidade (pode haver mais de uma) com a maior população

nome  
Ponta Grossa

- $\pi_{(\text{nome})}(\sigma_{(\text{pop} = (\mathcal{G}_{\text{max}(\text{pop})}(\text{cidade}))})(\text{cidade}))$

```
select nome from cidade  
where pop=(select max(pop) from cidade);
```

- Segunda forma (mais complexa) baseada em join:

$$\pi_{(\text{nome})}(\mathcal{G}_{\text{max}(\text{pop})}(\text{cidade}) \bowtie_{\text{max}=\text{pop}} \text{cidade})$$



```
select c.nome from cidade  
join (select max(pop) as m from cidade) t  
on c.pop=t.m;
```

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR

# Criando uma Relação

estado	
uf	nome
SC	Santa Catarina
PR	Paraná
RS	Rio Grande do Sul
SP	São Paulo

- Pode-se criar relações intermediárias para gerar a consulta final
  - Utiliza-se o operador  $\leftarrow$ 
    - $\text{CEstado} \leftarrow \rho_{(\text{cid}, \text{cnome}, \text{cpop}, \text{carea}, \text{cuf}, \text{euf}, \text{enome})}(\text{cidade} \bowtie \text{Estado})$
    - $\pi_{(\text{cnome}, \text{enome})}(\text{Estado})$

```
with CEstado as (select c.codcid as cid, c.nome as cnome,
                        c.pop as cpop, c.area as carea,
                        c.uf as cuf, e.uf as euf, e.nome as enome
                    from cidade c join estado e
                    on c.uf=e.uf)
select cnome, enome from CEstado;
```

cidade				
codcid	nome	pop	area	uf
1	Chapeco	200	626.06	SC
2	Erechim	105	430.7	RS
3	Passo Fundo	204	783.4	RS
4	Realeza	17.07	353.4	PR
5	Itajai	200.5	289.3	SC
6	Ponta Grossa	355.3	2054	PR