

**INSTITUTO DE EXATAS E TECNOLÓGICAS**  
Curso de Ciência da Computação

**BANCO DE DADOS**  
aula 6 – álgebra relacional -

professor: Juliano Varella de Carvalho

# 5.1. Álgebra Relacional – relações de exemplo

conta

nome_agencia	numero_conta	saldo
SAL-1	0001	1200
SAL-1	0002	3000
NOH-1	0003	4500
POA-1	0004	4000
POA-1	0005	1500
NOH-1	0006	200
SAL-2	0007	3750
SAL-2	0008	1800

cliente

nome	rua	cidade
João	Getúlio Vargas	São Leopoldo
Pedro	Getúlio Vargas	São Leopoldo
Francisco	Olavo Bilac	Novo Hamburgo
Maria	João Pessoa	Porto Alegre
Paulo	Cecília Meireles	São Leopoldo
José	João Goulart	Novo Hamburgo
Ana	Assis Brasil	Porto alegre
Beatriz	Florian Peixoto	Novo Hamburgo

agencia

nome_agencia	cidade_agencia	saldo
NOH-1	Novo Hamburgo	260050
SAL-1	São Leopoldo	455580
POA-1	Porto Alegre	1250369
SAL-2	São Leopoldo	125588

depositante

nome_cliente	numero_conta
João	0001
Pedro	0002
Francisco	0003
Maria	0004
Paulo	0007
José	0006
Ana	0005

devedor

nome_cliente	num_emprestimo
João	E-001
Ana	E-005
Helena	E-008

emprestimo

nome_agencia	num_emprestimo	total
SAL-1	E-001	40000
POA-1	E-005	25400
NOH-1	E-008	5420

## 5.2. Álgebra Relacional – conceitos

• **Linguagens de Consulta:** linguagem por meio da qual os usuários obtêm informações do banco de dados. Podem ser:

- procedurais: o usuário descreve uma seqüência de operações para obter o resultado desejado; **Ex:** pascal, pl-sql, transact-sql, etc.
- não procedurais: o usuário descreve a informação desejada, sem instruir como esta será obtida. **Ex:** sql, QBE (Query by example), etc.

• **Álgebra Relacional:** linguagem de consulta que consiste em um conjunto de operações, tendo como entrada uma ou duas relações e produzindo como resultado uma nova relação. Operações mais importantes:

- |                |                      |
|----------------|----------------------|
| - select;      | - set difference;    |
| - project;     | - cartesian product; |
| - Union;       | - rename;            |
| - namely       | - set intersection;  |
| - natural join | - division;          |
| - assignment;  |                      |

- Operações primárias: operam sobre uma única relação: select, project, rename.

- Operações binárias: operam sobre duas relações: union, set difference, cartesian product.

## 5.3. Álgebra Relacional – operações fundamentais

- **Operação SELECT:** seleciona as tuplas que satisfaçam um determinado predicado. A letra sigma representa a seleção, o predicado aparece subscrito a sigma, e a relação utilizada é colocada entre parênteses.

**Ex:**  $\sigma_{\text{nome\_agencia} = \text{"SAL-1"}}(\text{conta})$  : seleciona as tuplas da relação conta cuja agência é igual a "SAL-1".

O resultado da seleção é

nome_agencia	numero_conta	saldo
SAL-1	0001	1200
SAL-1	0002	3000

- podem ser usados comparações do tipo:  $=, \neq, \geq, >, \leq, <$  e os conectivos:  $\wedge$  (e)  $\vee$  (ou)

- **Operação PROJECT:** denotada pela letra grega pi ( $\pi$ ), esta operação retorna a relação com somente os atributos selecionados, eliminando as linhas em duplicidade. Os atributos aparecem ao lado de pi, subscrito, e a relação aparece entre parênteses.

**Ex:**  $\pi_{\text{nome\_agência}, \text{saldo}}(\text{agencia})$

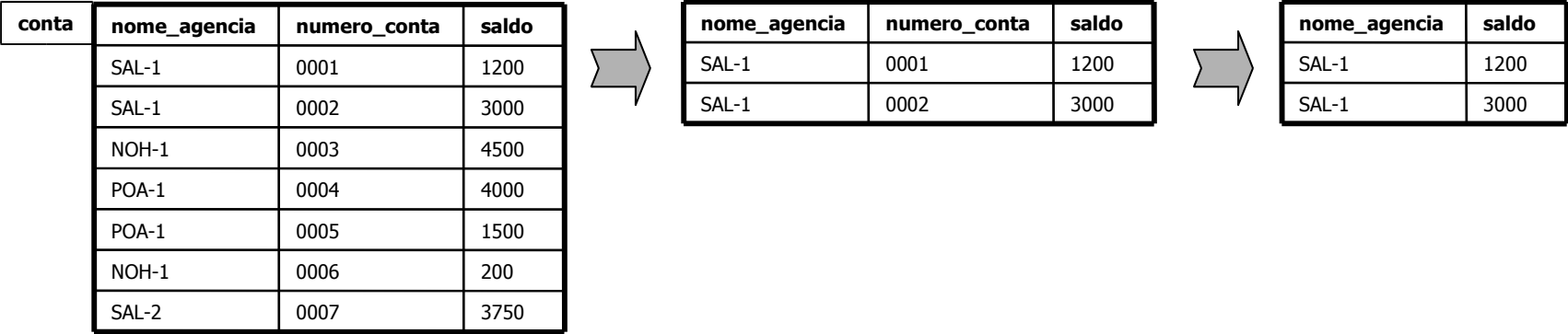
nome_agencia	cidade_agencia	saldo	agencia
NOH-1	Novo Hamburgo	260050	
SAL-1	São Leopoldo	455580	
POA-1	Porto Alegre	1250369	
SAL-2	São Leopoldo	125588	

nome_agencia	saldo
NOH-1	260050
SAL-1	455580
POA-1	1250369
SAL-2	125588

# 5.3. Álgebra Relacional – operações fundamentais

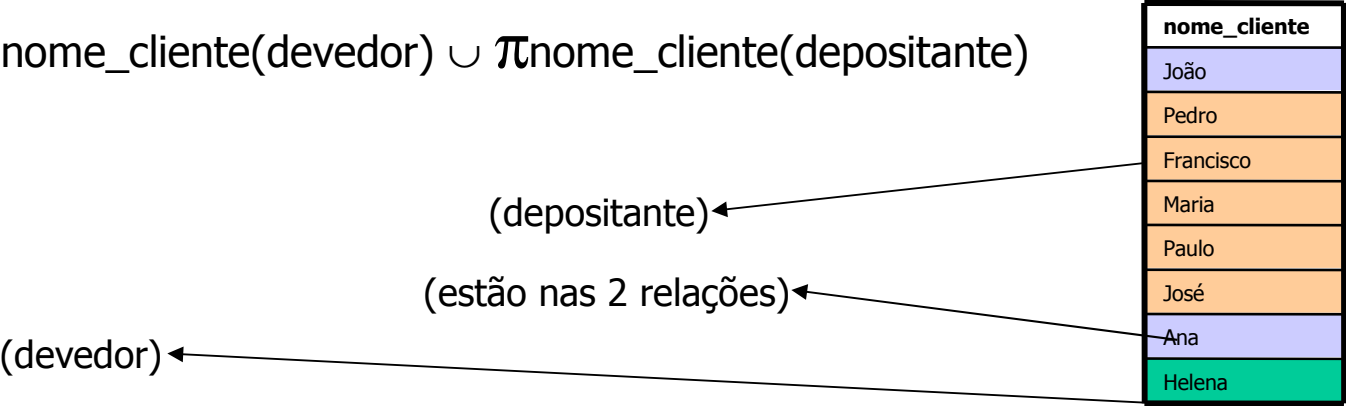
- Juntando operações SELECT E PROJECT: pode-se unir operações desde que elas tratem de relações.

$$\pi_{\text{nome\_agencia}, \text{saldo}}(\sigma_{\text{nome\_agencia} = \text{"SAL-1"}}(\text{conta}))$$



• **Operação UNION:** considere a seguinte consulta: selecionar todos os nomes de clientes que tenham um empréstimo, uma conta ou ambos. Deve-se utilizar as relações “depositante” e “devedor”.

$$\pi_{\text{nome\_cliente}}(\text{devedor}) \cup \pi_{\text{nome\_cliente}}(\text{depositante})$$



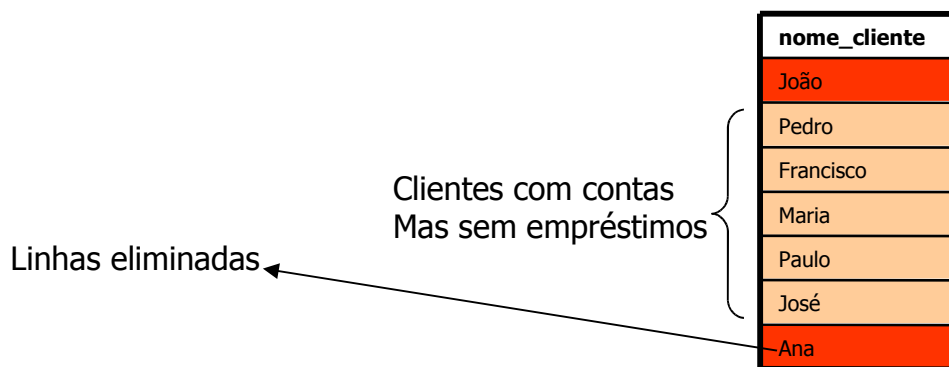
## 5.3. Álgebra Relacional – operações fundamentais

- Uma operação UNION entre duas relações "r" e "s" só é possível se:
  - i. as relações "r" e "s" têm o mesmo número de atributos;
  - ii. os domínios dos i-ésimo atributo de "r" e o i-ésimo atributo de "s" devem ser os mesmos para todo "i".

• **Operação Diferença entre conjuntos:** sendo "r" e "s" duas relações, "r-s" tem como resultado o conjunto de tuplas que estão na relação "r", mas não encontram-se na relação "s".

- **Ex:** selecionar todos os clientes que tem conta mas não tem empréstimo.

$\pi_{\text{nome\_cliente}}(\text{depositante}) - \pi_{\text{nome\_cliente}}(\text{devedor})$



# 5.4. Álgebra Relacional – operações adicionais

• **Operação de Interseção de conjuntos:** captura todas as tuplas que encontram-se em uma relação “r”, e que também encontram-se na relação “s”.

- **Ex:** Selecionar os nomes de clientes que possuem uma conta e um empréstimo.

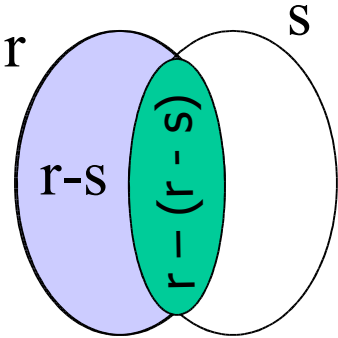
$$\pi_{\text{nome\_cliente}}(\text{devedor}) \cap \pi_{\text{nome\_cliente}}(\text{depositante})$$

depositante	
nome_cliente	numero_conta
João	0001
Pedro	0002
Francisco	0003
Maria	0004
Paulo	0007
José	0006
Ana	0005

nome_cliente
João
Ana

devedor	
nome_cliente	num_emprestimo
João	E-001
Ana	E-005
Helena	E-008

$$r \cap s = r - (r - s)$$



## 5.3. Álgebra Relacional – operações fundamentais

• **Operação Produto Cartesiano:** permite combinar informações de duas relações quaisquer. Representado por “r x s”.

**Ex:** Qual o produto cartesiano entre as relações devedor e emprestimo.

(devedor)x(emprestimo) = (nome\_cliente, num\_emprestimo, nome\_agencia, num\_emprestimo, total)

Obs: Para evitar o problema da duplicidade de nomes de atributos insere-se o nome da relação antes do atributo:

(nome\_cliente, devedor.num\_emprestimo, nome\_agencia, emprestimo.num\_emprestimo, total)

- Quantas e quais tuplas aparecerão na relação (devedor x emprestimo) ?  
Se a relação (devedor) possui “n” tuplas e a relação empréstimo tem “m” tuplas, então o produto cartesiano (devedor x empréstimo) tem n\*m tuplas.

devedor x emprestimo				
nome_cliente	devedor.num_emprestimo	nome_agencia	emprestimo.num_emprestimo	total
João	E-001	SAL-1	E-001	40000
João	E-001	POA-1	E-005	25400
João	E-001	NOH-1	E-008	5420
Ana	E-005	SAL-1	E-001	40000
Ana	E-005	POA-1	E-005	25400
Ana	E-005	NOH-1	E-008	5420
Helena	E-008	SAL-1	E-001	40000
Helena	E-008	POA-1	E-005	25400
Helena	E-008	NOH-1	E-008	5420



# 5.3. Álgebra Relacional – operações fundamentais

- Agora, deve-se encontrar todos os nomes de clientes que tenham um empréstimo na agência "SAL-1", através da operação:

Apenas João fez empréstimo em SAL-1.

$$\sigma_{\text{nome\_agência} = \text{"SAL-1"}}(\text{devedor} \times \text{empréstimo})$$

nome_client	devedor.num_emprestimo	nome_agencia	emprestimo.núm_emprestimo	total
<sup>e</sup> João	E-001	SAL-1	E-001	40000
Ana	E-005	SAL-1	E-001	40000
Helena	E-008	SAL-1	E-001	40000

OPS !!!

Obs: Como a operação produto cartesiano associa todas as tuplas de empréstimos a todas as tuplas de devedor, então existem empréstimos que estão relacionados erroneamente a alguns clientes. Para contornar este problema, devemos exigir que o atributo devedor.num\_emprestimo seja igual ao atributo emprestimo.num\_emprestimo.

- Assim temos:

$$\sigma_{\text{devedor.num\_emprestimo} = \text{emprestimo.num\_emprestimo}}(\sigma_{\text{nome\_agência} = \text{"SAL-1"}}(\text{devedor} \times \text{empréstimo}))$$

nome_client	devedor.num_emprestimo	nome_agencia	emprestimo.num_emprestimo	total
<sup>e</sup> João	E-001	SAL-1	E-001	40000

$$\pi_{\text{nome\_cliente}}(\sigma_{\text{devedor.num\_emprestimo} = \text{emprestimo.num\_emprestimo}}(\sigma_{\text{nome\_agência} = \text{"SAL-1"}}(\text{devedor} \times \text{empréstimo})))$$

nome_client
<sup>e</sup> João

## 5.3. Álgebra Relacional – operações fundamentais

- **Operação Rename:** representado pela letra grega rho ( $\rho$ ), permite dar um nome a uma determinada expressão.

Assim, a expressão em álgebra relacional E abaixo tem por resultado uma expressão sob o nome x.

$$\rho_x(E)$$

**Ex:** Desejamos o nome dos clientes que moram na mesma cidade de Francisco.

Primeiro, é necessário descobrir em qual cidade Francisco mora, fazendo uma seleção:

$$\sigma_{\text{nome\_cliente} = \text{"Francisco"}}(\text{cliente}),$$

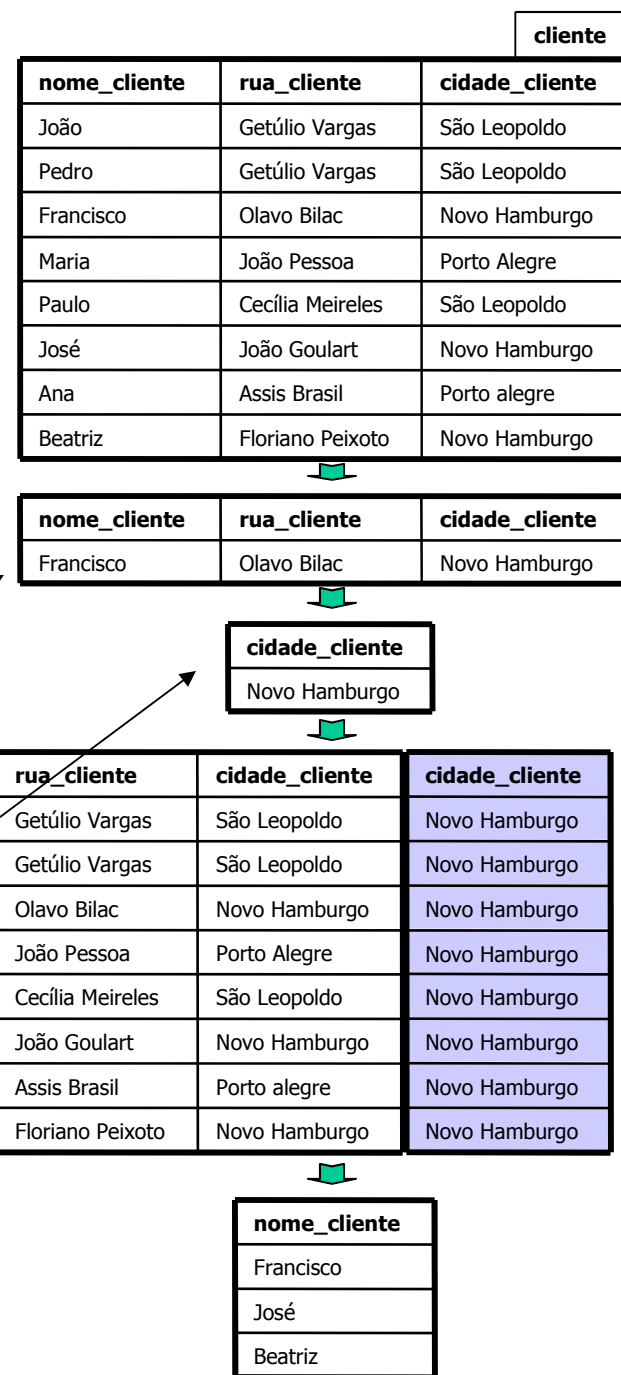
E depois uma projeção:

$$\pi_{\text{cidade\_cliente}}(\sigma_{\text{nome\_cliente} = \text{"Francisco"}}(\text{cliente}))$$

Para encontrar os outros clientes, é necessário fazer uma varredura na relação cliente, comparando o atributo cidade\_cliente com o atributo da relação obtida após a projeção anterior, ficando assim:

$$\pi_{\text{nome\_cliente}}(\sigma_{\text{cliente.cidade\_cliente} = \text{cid\_francisco.cidade\_cliente}}$$

$$(\text{cliente} \times \rho_{\text{cid\_francisco}}(\pi_{\text{cidade\_cliente}}(\sigma_{\text{nome\_cliente} = \text{"Francisco"}}(\text{cliente}))))))$$



## 5.4. Álgebra Relacional – operações adicionais

• **Operação de Junção Natural:** a junção natural ou natural join é uma operação binária que permite combinar seleções e um produto cartesiano em uma só operação.

- **Ex:** encontrar todos os nomes de clientes que tenham um empréstimo no banco, bem como o total emprestado.

Utilizando apenas as operações fundamentais, poderia ser resolvido assim:

$$\pi_{\text{nome\_cliente, emprestimo.num\_emprestimo, total}}(\sigma_{\text{devedor.num\_emprestimo = emprestimo.num\_emprestimo}}(\text{devedor} \times \text{emprestimo}))$$

No entanto, é possível simplificar esta operação utilizando o símbolo da junção natural:  $\bowtie$ . A junção natural forma um produto cartesiano das duas relações, e já executa uma seleção, fazendo a equivalência dos atributos que aparecem em ambos os esquemas da relação, e por último, remove os atributos em duplicidade.

$$\pi_{\text{nome\_cliente, num\_emprestimo total}}(\text{devedor} \bowtie \text{empréstimo})$$

nome_client	num_emprestimo	total
João	E-001	40000
Ana	E-005	25400
Helena	E-008	5420

## 5.4. Álgebra Relacional – operações adicionais

**Formalizando:** Sendo R e S dois esquemas de relação, ou seja, uma lista de atributos e  $r(R)$  e  $s(S)$  duas relações, a junção natural de r e s ( $r \bowtie s$ ) é uma relação formalmente definida assim:

$$r \bowtie s = \pi_{R \cup S}(\sigma_{r.A1=s.A1 \wedge r.A2=s.A2 \wedge \dots \wedge r.An=s.An}(r \times s))$$

$$\text{Em que } R \cap S = \{A1, A2, \dots, An\}$$

### Propriedades:

- a junção natural é associativa, portanto:

$$\text{cliente} \bowtie \text{conta} \bowtie \text{depositante} = (\text{cliente} \bowtie \text{conta}) \bowtie \text{depositante} = \text{cliente} \bowtie (\text{conta} \bowtie \text{depositante})$$

- se  $r(R)$  e  $s(S)$  são relações sem atributos comuns, então  $r \bowtie s = r \times s$

## 5.4. Álgebra Relacional – operações adicionais

• **Operação de Divisão:** a divisão de duas relações  $R \div S$ , onde os atributos de  $S$  estão contidos nos atributos de  $R$ ,  $A(R) \subseteq A(S)$ , resulta na relação  $T$ , onde  $A(T) = \{ A(R) - A(S) \}$ , onde para cada tupla  $t$  que aparece no resultado, os valores de  $t$  devem aparecer em  $R$ , combinando com cada tupla de  $S$ . Esta operação é utilizada nas consultas em que se emprega a frase “para todos”;

S
A1
a1
a2
a3

$R \div S$
A2
b1
b4

R		
	A1	A2
	a1	b1
	a2	b1
	a3	b1
	a4	b1
	a1	b2
	a3	b2
	a2	b3
	a3	b3
	a4	b3
	a1	b4
	a2	b4
	a3	b4

# 5.4. Álgebra Relacional – operações adicionais

- **Ex:** encontrar todos os clientes que tenham conta em todas as agências localizadas em São Leopoldo.

conta

nome_agencia	numero_conta	saldo
SAL-1	0001	1200
SAL-1	0002	3000
NOH-1	0003	4500
POA-1	0004	4000
POA-1	0005	1500
NOH-1	0006	200
SAL-2	0007	3750
SAL-2	0008	1800

agencia

nome_agencia	cidade_agencia	saldo
NOH-1	Novo Hamburgo	260050
SAL-1	São Leopoldo	455580
POA-1	Porto Alegre	1250369
SAL-2	São Leopoldo	125588

depositante

nome_cliente	numero_conta
João	0001
Pedro	0002
Francisco	0003
Maria	0004
Paulo	0007
José	0006
Ana	0005
João	0008

$\pi_{\text{nome\_agencia}}(\sigma_{\text{cidade\_agencia} = \text{"São Leopoldo"}}(\text{agencia}))$

nome_agência	cidade_agência	saldo
SAL-1	São Leopoldo	455580
SAL-2	São Leopoldo	125588

 $\Rightarrow$ 

nome_agência
SAL-1
SAL-2

$\pi_{\text{nome\_cliente, nome\_agencia}}(\text{depositante} \bowtie \text{conta})$

nome_cliente	nome_agencia
João	SAL-1
Pedro	SAL-1
Francisco	NOH-1
Maria	POA-1
Paulo	SAL-2
José	NOH-1
Ana	POA-1
João	SAL-2

$\pi_{\text{nome\_cliente, nome\_agencia}}(\text{depositante} \bowtie \text{conta}) \div \pi_{\text{nome\_agencia}}(\sigma_{\text{cidade\_agencia} = \text{"São Leopoldo"}}(\text{agencia}))$

nome_cliente
João

## 5.4. Álgebra Relacional – operações adicionais

- **Operação de Atribuição (Assignment Operation):** trabalha de maneira similar à atribuição das linguagens de programação. Representada por  $\leftarrow$

- Atribui-se a relação resultante de uma operação à direita de  $\leftarrow$ , a uma variável temporária, à esquerda, a qual poderá ser utilizada em relações subsequentes.

- **Ex:** operação de divisão anterior poderia ser descrita como:

$\text{temp1} \leftarrow \pi_{\text{nome\_agencia}}(\sigma_{\text{cidade\_agencia} = \text{"São Leopoldo"}}(\text{agencia}))$

$\text{temp2} \leftarrow \pi_{\text{nome\_cliente}, \text{nome\_agencia}}(\text{depositante} \bowtie \text{conta})$

$\text{resultado} \leftarrow \text{temp2} \div \text{temp1}$

## 5.5. Álgebra Relacional – operações estendidas

- **Projeção generalizada:** estendem as projeções, permitindo que operações aritméticas sejam usadas nas projeções.

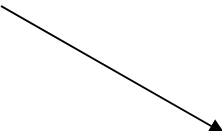
$$\pi_{F_1, F_2, \dots, F_n}(E)$$

Onde E é uma expressão em álgebra relacional e cada  $F_i$  são expressões aritméticas envolvendo constantes e atributos no esquema de E.

- **Ex:** Dada a relação crédito, descobrir quanto cada cliente ainda pode utilizar do seu crédito pessoal.

$$\pi_{\text{nome\_cliente}, \text{limite\_credito} - \text{credito\_usado}}(\text{credito})$$

credito		
nome_cliente	limite_credito	credito_usado
José	4000	2200
Ana	3200	100
Pedro	1200	1500
Maria	5800	5000
Flávia	3600	3600



nome_cliente	limite_credito - credito_usado
José	1800
Ana	3100
Pedro	-300
Maria	800
Flávia	0



## 5.5. Álgebra Relacional – operações estendidas

- **Junção externa (outer join):** extensão da operação de join para resolver informações omitidas.

- **Ex:** Dadas as relações a seguir com os seguintes esquemas:

empregado(nome\_empregado, rua, cidade)

trabalhador\_integral(nome\_empregado, nome\_agencia, salario)

empregado		
nome_empregado	rua	cidade
José	João Goulart	Novo Hamburgo
Ana	Assis Brasil	Porto Alegre
Pedro	Getúlio Vargas	São Leopoldo
Maria	João Pessoa	Porto Alegre

trabalhador_integral		
nome_empregado	nome_agencia	salario
José	NOH-1	5000
Ana	POA-1	4800
Flávia	SAL-1	3200
Maria	POA-1	6500

Descobrir a rua, cidade, agência e salário de todos os empregados.

Alternativa natural é fazer uma seleção com uma junção natural:

$\pi_{\text{nome\_empregado, rua, cidade, nome\_agencia, salario}}(\text{empregado} \bowtie \text{trabalhador\_integral})$

nome_empregado	rua	cidade	nome_agencia	salario
José	João Goulart	Novo Hamburgo	NOH-1	5000
Ana	Assis Brasil	Porto Alegre	POA-1	4800
Maria	João Pessoa	Porto Alegre	POA-1	6500

problema: os dados de rua e cidade do empregado Pedro foram perdidos, bem como os dados de agência e salário foram perdidos da empregada Flávia.

solução: utilizar as operações de junções externas para evitar perda de informações, que podem ser:

# 5.5. Álgebra Relacional – operações estendidas

## Junção externa a esquerda

empregado  $\bowtie$  trabalhador\_integral

nome_empregado	rua	cidade	nome_agencia	salario
José	João Goulart	Novo Hamburgo	NOH-1	5000
Ana	Assis Brasil	Porto Alegre	POA-1	4800
Maria	João Pessoa	Porto Alegre	POA-1	6500
Pedro	Getúlio Vargas	São Leopoldo	Nulo	Nulo

A junção à esquerda pega todas as tuplas da relação à esquerda que não encontraram par entre as tuplas da relação à direita, preenchendo com “nulo” os valores não encontrados. Estas tuplas são adicionadas ao resultado da junção natural.

## Junção externa a direita

empregado  $\bowtie$  trabalhador\_integral

nome_empregado	rua	cidade	nome_agencia	salario
José	João Goulart	Novo Hamburgo	NOH-1	5000
Ana	Assis Brasil	Porto Alegre	POA-1	4800
Maria	João Pessoa	Porto Alegre	POA-1	6500
Flávia	Nulo	Nulo	SAL-1	3200

A junção à direita pega todas as tuplas da relação à direita que não encontraram par entre as tuplas da relação à esquerda, preenchendo com “nulo” os valores não encontrados. Estas tuplas são adicionadas ao resultado da junção natural.

## Junção externa total

empregado  $\bowtie$  trabalhador\_integral

nome_empregado	rua	cidade	nome_agencia	salario
José	João Goulart	Novo Hamburgo	NOH-1	5000
Ana	Assis Brasil	Porto Alegre	POA-1	4800
Maria	João Pessoa	Porto Alegre	POA-1	6500
Pedro	Getúlio Vargas	São Leopoldo	Nulo	Nulo
Flávia	Nulo	Nulo	SAL-1	3200

A junção externa total faz ambas operações acima.

# 5.5. Álgebra Relacional – operações estendidas

- **Funções agregadas:** são aquelas que, quando aplicadas, tomam uma coleção de valores e retornam um valor simples como resultado.

- **função sum:** descobrir a soma total dos salários de todos os empregados de tempo integral.

sum <sub>salario</sub> (trabalhador_integral)	<table><tr><th>sum salario</th></tr><tr><td>14505</td></tr></table>	sum salario	14505
sum salario			
14505			

trabalhador_integral		
nome_empregado	nome_agencia	salario
José	NOH-1	5000
Ana	POA-1	4800
Flávia	SAL-1	3200
Maria	POA-1	6500

-**função count:** descobrir o número de agências existentes na tabela de tempo integral.

count <sub>nome_agencia</sub> (trabalhador_integral)	<table><tr><th>count nome_agencia</th></tr><tr><td>4</td></tr></table>	count nome_agencia	4
count nome_agencia			
4			

count-distinct <sub>nome_agencia</sub> (trabalhador_integral)	count-distinct nome_agencia
	3

- **função avg:** descobrir a média dos salários.

avg_salario (trabalhador_integral)	<table><tr><th>avg salario</th></tr><tr><td>3626.25</td></tr></table>	avg salario	3626.25
avg salario			
3626.25			

- **função min:** descobrir o menor salário.

**ção min:** descobrir o menor salário.

min<sub>salario</sub> (trabalhador\_integral)

min salario
3200

- **função max:** descobrir o maior salário.

**ção max:** descobrir o maior salário.

max<sub>salario</sub> (trabalhador\_integral)

max salario
6500

# 5.5. Álgebra Relacional – operações estendidas

- Em alguns casos, aplicam-se as funções agregadas não somente a um único conjunto de tuplas, mas a diversos grupos, onde cada grupo é um conjunto de tuplas. Isto é realizado através da operação chamada **agrupamento**.

**Ex:** descobrir a soma total dos salários dos empregados de tempo integral, por agência.

solução: particionar a relação trabalhador\_integral em grupos, com base na agência as quais os empregados fazem parte e, por fim aplicar a função agregada a cada um destes grupos.

trabalhador_integral		
nome_empregado	nome_agencia	salario
José	NOH-1	5000
Ana	POA-1	4800
Flávia	SAL-1	3200
Maria	POA-1	6500
Fernando	POA-1	4000
Beatriz	SAL-1	1000
Bruno	NOH-1	3000

nome\_agencia **g** sum<sub>salario</sub> (trabalhador\_integral)

nome_agencia	sum salario
NOH-1	8000
POA-1	15300
SAL-1	4200

nome\_agencia **g** sum<sub>salario</sub> min<sub>salario</sub> (trabalhador\_integral)

nome_agencia	sum salario	min salario
NOH-1	8000	3000
POA-1	15300	4000
SAL-1	4200	1000

**- Formalmente:**  $G_1, G_2, \dots, G_n \text{ g } F_1A_1, F_2A_2, F_mA_m (E), \text{ onde:}$

- E: é uma expressão qualquer em álgebra relacional;
- Cada  $G_i$  constitui um atributo que será agrupado;
- Cada  $F_i$  é uma função agregada;
- Cada  $A_i$  é o nome de um atributo;

## 5.5. Álgebra Relacional – modificações no BD

- **exclusão:** permite a exclusão de tuplas inteiras de uma relação. É expressa por:

$$r \leftarrow r - E, \text{ onde:}$$

r: é uma relação;

E: é uma consulta em álgebra relacional

**Ex:** excluir todas as contas do cliente João.

$$\text{conta} \leftarrow \text{conta} - \sigma_{\text{nome\_cliente} = \text{"João"}} (\text{conta})$$

**Ex:** excluir todas as contas das agências localizadas em São Leopoldo.

$$r1 \leftarrow \sigma_{\text{cidade\_agencia} = \text{"São Leopoldo"}} (\text{conta} \bowtie \text{agencia})$$

$$r2 \leftarrow \pi_{\text{nome\_agencia, numero\_conta, saldo}} (r1)$$

$$\text{conta} \leftarrow \text{conta} - r2$$

## 5.5. Álgebra Relacional – modificações no BD

- **inserção:** permite a inserção de um conjunto de tuplas em uma relação. É expressa por:  $r \leftarrow r \cup E$ , onde  $r$  é uma relação e  $E$ : é uma consulta em álgebra relacional

Obs:

- os valores dos atributos das tuplas a inserir devem ser membros do domínio dos atributos;
- As tuplas inseridas devem ter o mesmo grau da relação.

**Ex:** inserir que o cliente Saulo tem 3800 na conta 010 da agência NOH-1.

$\text{conta} \leftarrow \text{conta} \cup \{(\text{"NOH-1"}, 010, 3800)\}$

$\text{depositante} \leftarrow \text{depositante} \cup \{(\text{"Saulo"}, 010)\}$

**Ex:** inserir uma nova caderneta de poupança de 100 reais para todos os clientes da agência SAL-1, sendo o número do empréstimo o mesmo número usado para esta conta poupança.

$r1 \leftarrow \sigma_{\text{cidade\_agencia} = \text{"SAL-1"}} (\text{devedor} \times \text{empréstimo})$

$r2 \leftarrow \pi_{\text{nome\_agencia}, \text{num\_emprestimo}} (r1)$

$\text{conta} \leftarrow \text{conta} \cup (r2 \times \{(100)\})$

$\text{depositante} \leftarrow \text{depositante} \cup \pi_{\text{nome\_cliente}, \text{num\_emprestimo}} (r1)$

		conta
nome_agencia	numero_conta	saldo
SAL-1	0001	1200
SAL-1	0002	3000
NOH-1	0003	4500
POA-1	0004	4000
POA-1	0005	1500
NOH-1	0006	200
SAL-2	0007	3750
SAL-2	0008	1800

depositante	
nome_cliente	numero_conta
João	0001
Pedro	0002
Francisco	0003
Maria	0004
Paulo	0007
José	0006
Ana	0005

devedor	
nome_client	num_emprestimo
João	E-001
Ana	E-005
Helena	E-008

emprestimo		
nome_agencia	num_emprestimo	total
SAL-1	E-001	40000
POA-1	E-005	25400
NOH-1	E-008	5420

## 5.5. Álgebra Relacional – modificações no BD

- **atualização:** permite a mudança de uma tupla sem mudar todos os valores dela. Para isso usa-se o operador de projeção generalizada:

$r \leftarrow \pi_{F_1, F_2, \dots, F_n}(r)$ , em que cada  $F_i$  é o  $i$ -ésimo atributo de “ $r$ ” se o  $i$ -ésimo atributo não for atualizado, ou, se o  $i$ -ésimo atributo for atualizado, então cada  $F_i$  é uma expressão envolvendo somente constantes e os atributos de “ $r$ ”, que dá novos valores a esses atributos.

**Ex:** A taxa de juros aumentou e o saldo de todas as contas aumentou 5%.

$$\text{conta} \leftarrow \pi_{\text{nome\_agencia}, \text{numero\_conta}, \text{saldo} \leftarrow \text{saldo} * 1.05}(\text{conta})$$

**Ex:** As contas com saldo acima de 2500 recebem 10%, e os demais 5%.

$$\text{conta} \leftarrow \pi_{\text{nome\_agencia}, \text{numero\_conta}, \text{saldo} \leftarrow \text{saldo} * 1.1}(\sigma_{\text{saldo} > 2500}(\text{conta})) \cup$$

$$\pi_{\text{nome\_agencia}, \text{numero\_conta}, \text{saldo} \leftarrow \text{saldo} * 1.05}(\sigma_{\text{saldo} \leq 2500}(\text{conta}))$$