



# Linguagens formais de Consulta

## Modelo Relacional

Profa. Maria Camila Nardini Barioni

[camila.barioni@ufu.br](mailto:camila.barioni@ufu.br)

Bloco B - sala 1B137

1º semestre de 2024

# Avisos

- ◆ Com o conteúdo da aula de hoje já é possível fazer a terceira etapa do projeto
- ◆ Haverá uma tarefa no Teams para a entrega parcial do projeto: 21/09

# Cronograma do projeto

## ◆ Projeto: HIPERMERCADO

◆ 24/08

- ◆ Início do desenvolvimento do projeto (Definição de requisitos adicionais e Modelo Entidade Relacionamento)
- ◆ Entrega dos nomes dos integrantes do grupo do projeto e da descrição dos requisitos adicionais

◆ 14/09

- ◆ Continuação do desenvolvimento do projeto (Mapeamento para o modelo relacional)

◆ 21/09

- ◆ Continuação do desenvolvimento do projeto (Álgebra relacional)
- ◆ Entrega da descrição das consultas

◆ 28/09

- ◆ Continuação do desenvolvimento do projeto (Consultas SQL)
- ◆ Entrega da descrição das consultas

◆ 19/10

- ◆ Continuação do desenvolvimento do projeto (SP e gatilhos)
- ◆ Entrega da descrição do SP e do gatilho

◆ 25/10

- ◆ Finalização do projeto
- ◆ Entrega do relatório final completo e scripts finais



# **CONTINUAÇÃO DA APRESENTAÇÃO – ÁLGEBRA RELACIONAL**

# Operações

## ◆ Fundamentais

- seleção
- projeção
- produto cartesiano
- renomear
- união
- diferença de conjuntos

## ◆ Adicionais

- intersecção de conjuntos
- junção natural
- divisão
- atribuição

- podem ser geradas a partir das operações fundamentais
- facilitam a construção de consultas

# Relações

cliente (nro\_cli, nome\_cli, end\_cli,  
saldo, vendedor)

vendedor (cod\_vend, nome\_vend)

pedido (nro\_ped, data, nro\_cliente)

pedido\_peça (nro\_ped, nro\_peça)

peça (nro\_peça, descrição\_peça)

# Junção

- ◆ Concatena tuplas relacionadas de duas relações em tuplas únicas
- ◆ Simplifica consultas que requerem produto cartesiano
  - forma um produto cartesiano dos argumentos
  - faz uma seleção forçando igualdade sobre os atributos que aparecem em ambos argumentos
  - remove colunas duplicadas

# Junção

## ◆ Concatenação

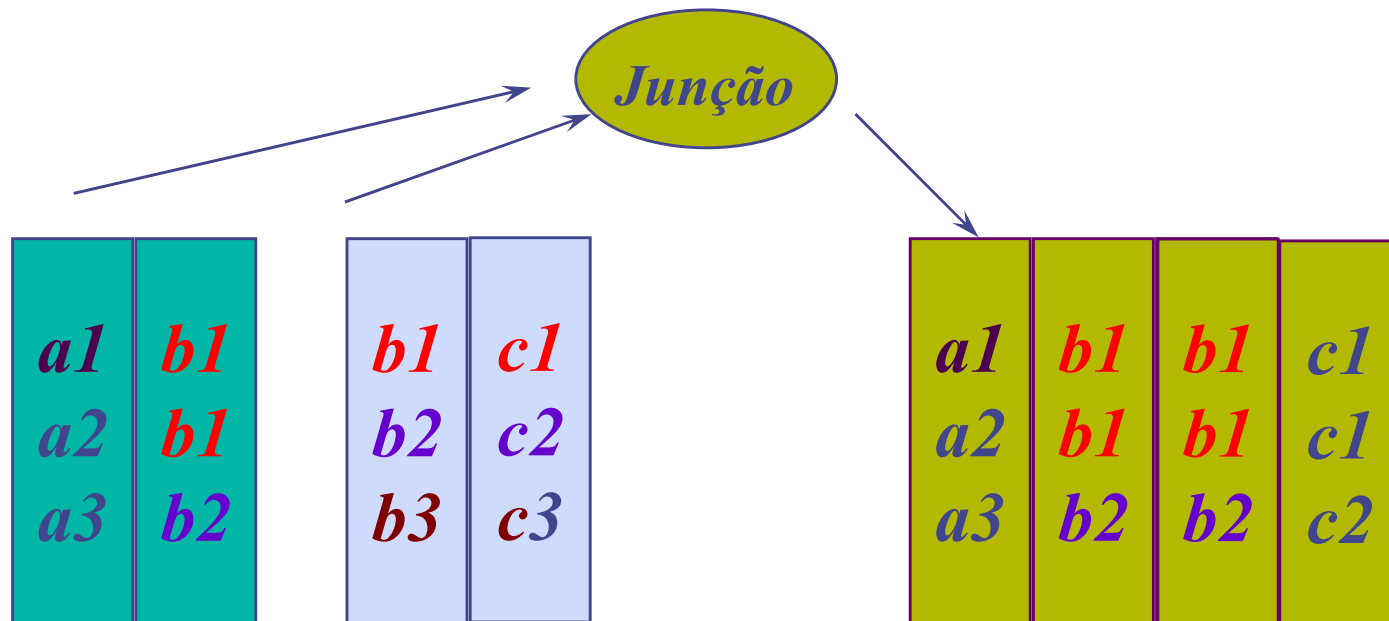
- dos atributos comuns
- dos atributos especificados na condição de junção

relação argumento 1  $\bowtie$  condição\_junção relação argumento 2

- relação
- resultado de alguma operação da álgebra relacional



# Junção



Note que a **coluna comum** na duas relações é a coluna “**b**”

# Cliente $\bowtie$ Vendedor



nro_cli	nome_cli	end_cli	saldo	vende- dor	cod_vend	nome_ve nd
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

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

número de tuplas: entre  
zero e o (número de  
tuplas de cliente \* número  
de tuplas de vendedor)

# Junção

## ◆ Condição de junção

- $\langle \text{condição} \rangle \wedge \langle \text{condição} \rangle \wedge \dots \wedge \langle \text{condição} \rangle$

## ◆ $A_i \theta B_j$

- $A_i$ : atributo da relação argumento 1
- $B_j$ : atributo da relação argumento 2
- $A_i$  e  $B_j$  tem o mesmo domínio
- $\theta$  (theta) é um operador de comparação  $\{=, <, \leq, >, \geq, \neq\}$
- existem diversas variações sobre a operação de junção

# Junção

## ◆ Junção theta

- na qual pode ser usada qualquer operador  $\theta$  válido no domínio dos atributos comparados
  - ◆  $\{=, <, \leq, >, \geq, \neq\}$
- os atributos envolvidos na comparação aparecem ambos na relação resultado (gerando pares de atributos com valores idênticos), mas tuplas com valores nulos nos atributos envolvidos na junção não aparecem no resultado
- variação mais genérica

# Junção

## ◆ Junção theta

cliente (nro\_cli, nome\_cli, end\_cli, saldo, vendedor)

vendedor (cod\_vend, nome\_vend)

cliente ⋈<sub>vendedor ≠ cod\_vend</sub> vendedor

grau da relação resultante é igual a 7

# Junção

## ◆ Equi-Junção

- uso mais comum de junção
- o operador  $\theta$  é a igualdade
  - ◆  $\{=\}$
- os atributos envolvidos na comparação aparecem ambos na relação resultado (gerando pares de atributos com valores idênticos), mas tuplas com valores nulos nos atributos envolvidos na comparação não aparecem no resultado

# Junção

## ◆ Equi-Junção

cliente (nro\_cli, nome\_cli, end\_cli, saldo, vendedor)

vendedor (cod\_vend, nome\_vend)

cliente ⋈<sub>vendedor = cod\_vend</sub> vendedor

grau da relação resultante é igual a 7

# Junção

## ◆ Junção Natural – $R * S$

- semelhante à operação de Equi-Junção
- porém, dos atributos da junção, apenas os originários de uma das relações operadas aparecem na relação resultado
  - ◆ requer que os atributos comparados tenham nomes iguais nas duas relações
- tuplas com valores nulos nos atributos envolvidos na comparação também não aparecem no resultado



# Junção

## ◆ Junção Natural

cliente (nro\_cli, nome\_cli, end\_cli, saldo, vendedor)

vendedor (cod\_vend, nome\_vend)

vendedor \*  $\rho_{(nro\_cli, nome\_cli, end\_cli, saldo, cod\_vend)}$  cliente

grau da relação resultante é igual a 6

# Junção

R

A	B	C
1	a	x
2	b	y
3	a	y
4	c	y

S

A	D
1	d
2	d
5	e

$R \bowtie S$

A	B	C	D
1	a	x	d
2	b	y	d

## ◆ Interna

- somente as tuplas de R que têm tuplas correspondentes em S, e vice-versa, aparecem no resultado

# Junção

R

A	B	C
1	a	x
2	b	y
3	a	y
4	c	y

S

A	D
1	d
2	d
5	e

$R \bowtie S$

A	B	C	D
1	a	x	d
2	b	y	d
3	a	y	Null
4	c	y	Null

## ◆ Externa à esquerda

- mantém cada tupla de R em  $R \bowtie S$
- preenche com valores nulos os atributos de S que não correspondem às tuplas em R

# Junção

R

A	B	C
1	a	x
2	b	y
3	a	y
4	c	y

S

A	D
1	d
2	d
5	e

$R \bowtie S$

A	B	C	D
1	a	x	d
2	b	y	d
5	Null	Null	e

## ◆ Externa à direita

- mantém cada tupla de S em  $R \bowtie S$
- preenche com valores nulos os atributos de R que não correspondem às tuplas em S

# Junção

R

A	B	C
1	a	x
2	b	y
3	a	y
4	c	y

S

A	D
1	d
2	d
5	e

$R \bowtie S$

A	B	C	D
1	a	x	d
2	b	y	d
3	a	y	Null
4	c	y	Null
5	Null	Null	e

## ◆ Externa completa

- mantém as tuplas de R e S em  $R \bowtie S$
- preenche com valores nulos os atributos que não correspondem à coluna de junção

# Exercícios

- ◆ Refaça os exemplos da aula anterior, utilizando a operação de junção natural ao invés da operação de produto cartesiano

# Exemplo 1

◆ Considere as seguintes relações

- usuário ( cliente\_nome, gerente\_nome )
- cliente ( cliente\_nome, rua, cidade )

cliente_nome	gerente_nome
Márcia	Manoel
Rodrigo	Maria

cliente_nome	rua	cidade
Márcia	Rua X	Itambé
Rodrigo	Rua X	Maringá

◆ Liste o nome de todos os usuários atendidos pelo gerente Manoel, assim como as cidades nas quais eles vivem

# Exemplo 1

- ◆ Considere as seguintes relações
  - usuário ( cliente\_nome, gerente\_nome )
  - cliente ( cliente\_nome, rua, cidade )
- ◆ Liste o nome de todos os usuários atendidos pelo gerente Manoel, assim como as cidades nas quais eles vivem

$\text{temp}_1 \leftarrow \pi_{\text{cliente\_nome}} (\sigma_{\text{gerente\_nome} = \text{"Manoel"}} (\text{usuário}))$

$\text{temp}_2 \leftarrow \text{temp}_1 \times \text{cliente}$

$\text{temp}_3 \leftarrow \sigma_{\text{temp1.cliente\_nome} = \text{cliente.cliente\_nome}} (\text{temp}_2)$

$\pi_{\text{temp1.cliente\_nome}, \text{cidade}} (\text{temp}_3)$



# Exemplo 1

- ◆ Considere as seguintes relações
  - usuário ( cliente nome, gerente\_nome )
  - cliente ( cliente nome, rua, cidade )
- ◆ Liste o nome de todos os usuários atendidos pelo gerente Manoel, assim como as cidades nas quais eles vivem

$\text{temp}_1 \leftarrow \pi_{\text{cliente\_nome}} (\sigma_{\text{gerente\_nome} = \text{"Manoel"}} (\text{usuário}))$

$\text{temp}_2 \leftarrow \text{temp}_1 * \text{cliente}$

$\pi_{\text{temp1.cliente\_nome}, \text{cidade}} (\text{temp}_2)$

# Exemplo 2

◆ Considere a seguinte relação

■ cliente ( cliente\_nome, rua, cidade )

cliente_nome	rua	cidade
Márcia	Rua X	Itambé
Rodrigo	Rua X	Maringá
Cristina	Rua XTZ	Maringá
Sofia	Rua X	Maringá
Ricardo	Rua AAA	Itambé

◆ Liste o nome dos clientes que moram na mesma rua e na mesma cidade que Rodrigo

# Exemplo 2

◆ Considere a seguinte relação

■ cliente ( cliente\_nome, rua, cidade )

◆ Liste o nome dos clientes que moram na mesma rua e na mesma cidade que Rodrigo

$\text{temp}_1 \leftarrow \pi_{\text{rua}, \text{cidade}} (\sigma_{\text{cliente\_nome} = \text{"Rodrigo"}} (\text{cliente}))$

$\text{temp}_2 \leftarrow \text{temp}_1 \times \text{cliente}$

$\text{temp}_3 \leftarrow \sigma_{\text{cliente\_nome} \neq \text{"Rodrigo"}} (\text{temp}_2)$

$\pi_{\text{cliente\_nome}} (\sigma_{\text{temp}_1.\text{rua} = \text{cliente.rua} \wedge \text{temp}_1.\text{cidade} = \text{cliente.cidade}} (\text{temp}_3))$

# Exemplo 2

◆ Considere a seguinte relação

■ cliente ( cliente\_nome, rua, cidade )

◆ Liste o nome dos clientes que moram na mesma rua e na mesma cidade que Rodrigo

$\text{temp}_1 \leftarrow \pi_{\text{rua}, \text{cidade}} (\sigma_{\text{cliente\_nome} = \text{"Rodrigo"}} (\text{cliente}))$

$\text{temp}_2 \leftarrow \text{temp}_1 * \text{cliente}$

$\text{temp}_3 \leftarrow \sigma_{\text{cliente\_nome} \neq \text{"Rodrigo"}} (\text{temp}_2)$

$\pi_{\text{cliente\_nome}} (\text{temp}_3)$

# Operações sobre Conjuntos

## ◆ Operações

- união
- intersecção
- diferença

## ◆ Características

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

Duas relações são compatíveis quando:

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

# União de Conjuntos

- ◆ Une duas relações R e S compatíveis em uma relação que contém todas as tuplas pertencentes a R, a S, ou a ambas R e S

relação argumento 1  $\cup$  relação argumento 2

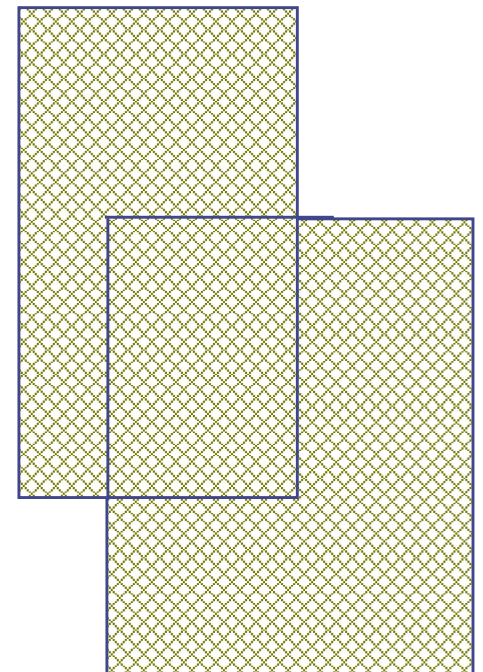
- relação
- resultado de alguma operação da álgebra relacional

# União de Conjuntos

- relações cujos domínios dos atributos são iguais, na mesma ordem de definição das colunas

**Notação:**

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



# Intersecção de Conjuntos

- ◆ Une duas relações R e S compatíveis em uma relação que contém todas as tuplas pertencentes tanto a R quanto a S

relação argumento 1  $\cap$  relação argumento 2

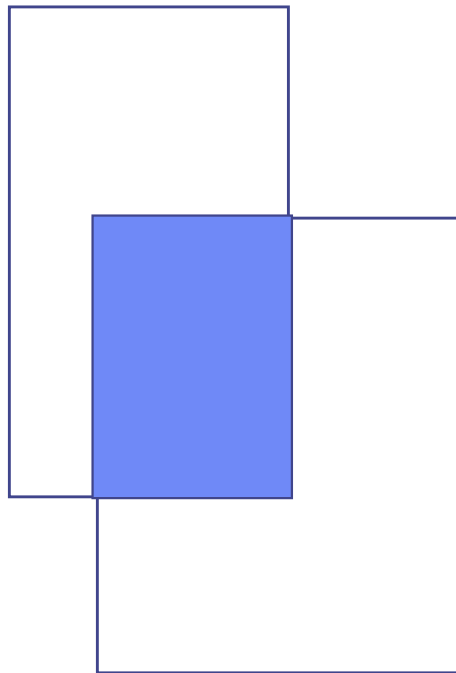
- relação
- resultado de alguma operação da álgebra relacional



# Intersecção de Conjuntos

Notação:

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



# Diferença de Conjuntos

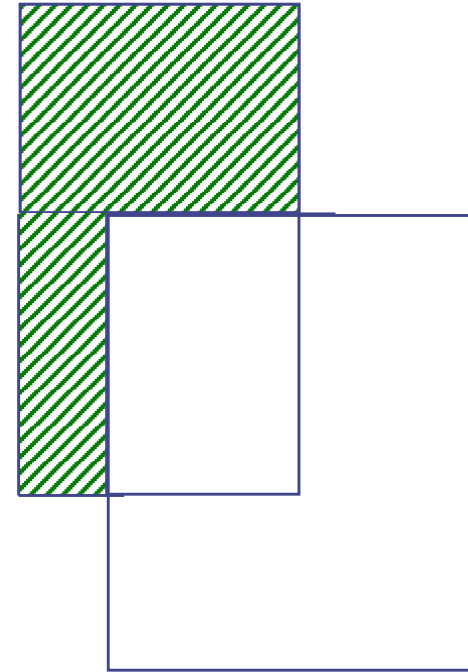
- ◆ Une duas relações R e S compatíveis em uma relação que contém todas as tuplas pertencentes a R que não pertencem a S

relação argumento 1 – relação argumento 2

- relação
- resultado de alguma operação da álgebra relacional

# Diferença de Conjuntos

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



Exemplo : Listar os vendedores que não atendem nenhum cliente, ou seja, que estão na tabela Vendedor mas que não estão na tabela de “Clientes”

# Relações Cliente e Pedido

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

pedido (nro\_ped, data, nro\_cliente)

nro_ped	data	nro_cliente
1	10/12/2004	1
2	11/12/2004	4

# Consultas 5, 6 e 7

- ◆ Liste os números dos clientes que
  5. ou têm pedido, ou foram atendidos pelo vendedor 2, ou ambos
  6. têm pedido, e que foram atendidos pelo vendedor 2
  7. têm pedido, mas que não foram atendidos pelo vendedor 2

# Sub-Consultas

- ◆ Liste os números dos clientes que têm pedido.

$\text{Temp}_1 \leftarrow \pi_{\text{nro\_cliente}} (\text{pedido})$

- ◆ Liste os números dos clientes que foram atendidos pelo vendedor 2.

$\text{temp}_2 \leftarrow \pi_{\text{nro\_cliente}} (\sigma_{\text{cod\_vend} = 2} (\text{cliente}) )$

# Consulta 5

- ◆ Liste os números dos clientes que ou têm pedido, ou foram atendidos pelo vendedor 2, ou ambos.

$temp_1$

nro_cliente
1
4

$temp_2$

nro_cliente
4

$temp_1 \cup temp_2$

nro_cliente
1
4

# Consulta 6



Liste os números dos clientes que têm pedido, e que foram atendidos pelo vendedor 2 .

$temp_1$

nro_cliente
1
4

$temp_2$

nro_cliente
4

$temp_1 \cap temp_2$

nro_cliente
4



# Consulta 7



Liste os números dos clientes que têm pedido, mas que não foram atendidos pelo vendedor 2.

$temp_1$

nro_cliente
1
4

$temp_2$

nro_cliente
4

$temp_1 - temp_2$

nro_cliente
1

# Exercício

◆ Considere a seguinte relação

■ conta ( nro\_conta, saldo )

<b>nro_conta</b>	<b>saldo</b>
01-010101-01	100,00
01-020202-02	200,00
01-030303-03	300,00
01-040404-04	400,00

◆ Liste o maior saldo

# Solução

## ◆ Primeiro passo

- realizar o produto cartesiano da relação conta com ela mesma

$$\text{temp}_1 \leftarrow \text{conta} \times \rho_{\text{conta2}}(\text{conta})$$

- relação resultado  $\text{temp}_1$

<b>conta.nro_conta</b>	<b>conta.saldo</b>	<b>conta2.nro_conta</b>	<b>conta2.saldo</b>
01-010101-01	100,00	01-010101-01	100,00
01-010101-01	100,00	01-020202-02	200,00
01-010101-01	100,00	01-030303-03	300,00
01-010101-01	100,00	01-040404-04	400,00
01-020202-02	200,00	01-010101-01	100,00
01-020202-02	200,00	01-020202-02	200,00
01-020202-02	200,00	01-030303-03	300,00
01-020202-02	200,00	01-040404-04	400,00
01-030303-03	300,00	01-010101-01	100,00
01-030303-03	300,00	01-020202-02	200,00
01-030303-03	300,00	01-030303-03	300,00
01-030303-03	300,00	01-040404-04	400,00
01-040404-04	400,00	01-010101-01	100,00
01-040404-04	400,00	01-020202-02	200,00
01-040404-04	400,00	01-030303-03	300,00
01-040404-04	400,00	01-040404-04	400,00

# Solução

## ◆ Segundo passo

- listar os saldos que não são os mais altos

$$\text{temp}_2 \leftarrow \pi_{\text{conta.saldo}} (\sigma_{\text{conta.saldo} < \text{conta2.saldo}} (\text{temp}_1))$$

- relação resultado  $\text{temp}_2$

<b>conta.saldo</b>
100,00
200,00
300,00

# Solução

## ◆ Terceiro passo

- listar todos os saldos da relação conta

$$\text{temp}_3 \leftarrow \pi_{\text{saldo}}(\text{conta})$$

- relação resultado  $\text{temp}_3$

saldo
100,00
200,00
300,00
400,00

# Solução

## ◆ Quarto passo

- fazer a diferença entre “todos os saldos da relação conta” e “os saldos que não são os mais altos”

$$\text{temp}_3 - \text{temp}_2$$

- relação resultado

saldo
400,00

# Divisão

## ◆ Divisão de duas relações R e S

- todos os valores de um atributo de R que fazem referência a todos os valores de um atributo de S

relação argumento 1  $\div$  relação argumento 2

- relação
- resultado de alguma operação da álgebra relacional



# Consulta 8

- ◆ Liste todos os pedidos que referenciam todas as peças listadas na relação peça.

pedido\_peça

nro_ped	nro_peça
9	12
1	04
1	66
4	03
5	11
8	04
8	74

$\pi_{\text{nro\_peça}}(\text{peça})$

nro_peça
66
04

pedido\_peça  $\div$  peça

nro_pedido
1

divisão: utilizada para consultas que incluam o termo *para todos* ou *em todos*

# Funções e operações agregadas

- ◆ As funções agregadas tomam uma coleção de valores e retornam um único valor como resultado
  - MÉDIA: valor médio
  - MÍNIMO: valor mínimo
  - MÁXIMO: valor máximo
  - SOMA: soma dos valores
  - CONTAGEM: número de valores
- ◆ Consideram valores repetidos, se existirem

# Funções e operações agregadas

- ◆ Ignoram “null”, exceto contagem
- ◆ Operação agregada na álgebra relacional
- ◆  $G_1, G_2, \dots, G_n \mathcal{F} F_{1A_1}, F_{2A_2}, \dots, F_{m_{A_m}}(E)$
- ◆ E é qualquer expressão de álgebra relacional
  - $G_1, G_2 \dots, G_n$  uma lista de atributos em que agrupar (pode ser vazia)
  - Cada  $F_i$  é uma função agregada
  - Cada  $A_i$  é um nome de atributo

# Operação agregada – Exemplo

◆ Relação  $r$ :

$A$	$B$	$C$
$\alpha$	$\alpha$	7
$\alpha$	$\beta$	7
$\beta$	$\beta$	3
$\beta$	$\beta$	10

◆  $\mathcal{F}_{\text{SOMA}_C}(r)$

$\text{SOMA}_C$
27

# Operação agregada – Exemplo

- ◆ Relação conta agrupada por nome\_agência:

<i>nome_agência</i>	<i>número_conta</i>	<i>saldo</i>
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

- nome\_agência  $\mathcal{F}$  SOMA saldo (conta)

<i>nome_agência</i>	<b>SOMA_saldo</b>
Perryridge	1300
Brighton	1500
Redwood	700

# Bibliografia

- ◆ Elmasri, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**. 4 ed. São Paulo: Addison Wesley, 2005, 724 p. Bibliografia: p. [690]-714.
- ◆ Material Didático produzido pelos professores Cristina Dutra de Aguiar Ciferri e Caetano Traina Júnior