

---

# Bancos (Bases) de Dados

Aula #7 – Álgebra Relacional  
Cálculo Relacional

---

**Prof. Eduardo R. Hruschka**

\* Slides baseados no material elaborado pelas professoras:

**Cristina D. A. Ciferri**

**Elaine P. M. de Souza**

---

# Álgebra Relacional

- Fornece um conjunto básico de operações que permitem ao usuário especificar solicitações de recuperações (novas relações);
  - Fundamento formal para operações realizadas no modelo relacional;
  - Base para implementar e otimizar consultas em SGBDs;
  - Alguns de seus conceitos são incorporados na linguagem de consulta padrão SQL.

---

# Álgebra Relacional

- Maneira teórica de se manipular um BD 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;
- ❑ Atribuição;

- Podem ser geradas a partir das operações fundamentais;
- Facilitam a construção de consultas.

# Classificação das Operações

## ■ Unárias:

- ❑ Seleção;
- ❑ Projeção;
- ❑ Renomear.

operam sobre uma  
única relação

## ■ Binárias:

- ❑ Produto cartesiano;
- ❑ União;
- ❑ Diferença de conjuntos;
- ❑ Intersecção de conjuntos;
- ❑ Junção natural;
- ❑ Divisão.

operam sobre duas  
relações

---

# Relações

Consideremos os seguintes esquemas de relações:

cliente (nro\_cli, nome\_cli, end\_cli, saldo, cod\_vend)

vendedor (cod\_vend, nome\_vend)

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

pedido\_peça (nro\_ped, nro\_peça)

peça (nro\_peça, descrição\_peça)

---

# Seleção

- Seleciona tuplas da relação argumento 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 de álgebra relacional.

# Relação Cliente

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



---

# Consulta 1

- Liste toda a informação da relação Cliente referente ao cliente de número 4.

$\sigma_{\text{nro\_cli} = 4} (\text{cliente})$

# Consulta 1

## ■ Relação resultado:

nro_cli	nome_cli	end_cli	saldo	cod_vend
4	Rodrigo	Rua X	137,00	2

grau: mesmo grau da relação argumento.

número de tuplas: menor ou igual ao número de tuplas da relação argumento.

---

## Consulta 2

- Liste toda a informação da relação Cliente para clientes que possuam saldo devedor inferior a R\$ 200,00 e que morem na Rua X.

$\sigma_{\text{saldo\_dev} < 200,00 \wedge \text{end\_cli} = \text{"Rua X"}} (\text{cliente})$

## Consulta 2

### ■ Relação resultado:

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
4	Rodrigo	Rua X	137,00	2

grau: mesmo grau da relação argumento.

número de tuplas: menor ou igual ao número de tuplas da relação argumento.

---

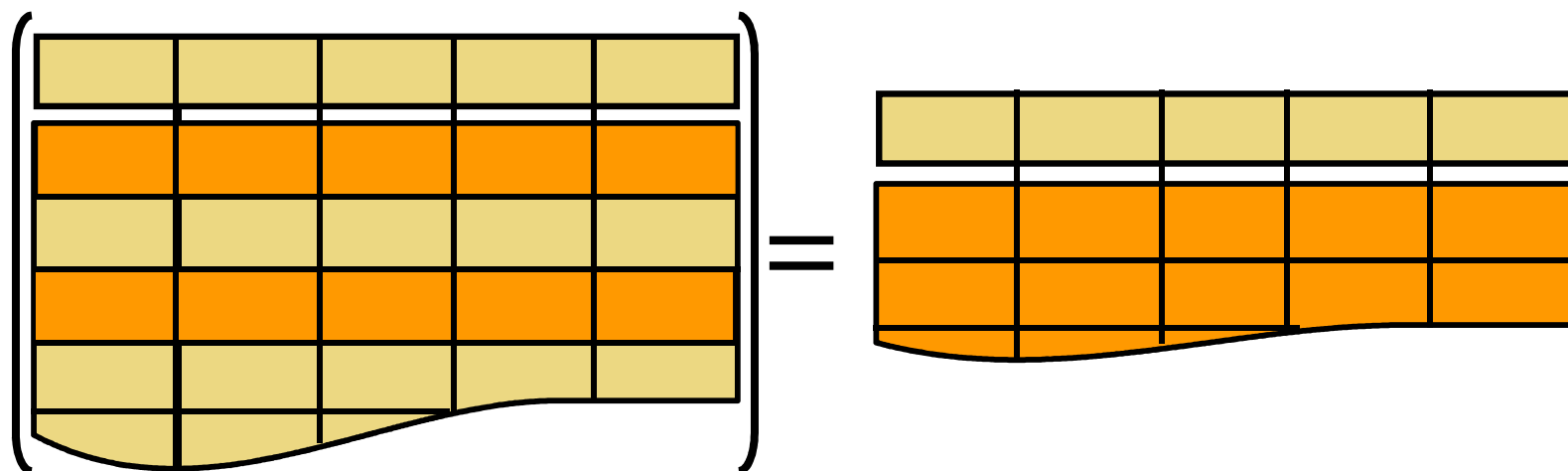
## Seleção ...

- **Seleção** -  $\sigma_{(\text{condição})}\mathbf{R}$

- resultado: subconjunto das tuplas de R que satisfazem à condição de seleção  $\langle \text{condição} \rangle$
- condição de seleção: sempre é uma **operação de comparação** (=, >, <, etc) de um atributo da relação com:
  - uma constante;
  - com outro atributo da própria relação  $\Rightarrow$  comparação de valores de dois atributos da mesma tupla.

## Seleção ...

- Seleção  $\Rightarrow$  **particionamento horizontal**:
  - escolha de algumas “linhas” da tabela.



$\sigma_{(\text{cond})}$

## Seleção ...

- Operador **Seleção é Comutativo**
  - $\sigma_{(\text{condição A})}(\sigma_{(\text{condição B})}) = \sigma_{(\text{condição B})}(\sigma_{(\text{condição A})})$
- Uma seqüência de seleções pode ser executada em qualquer ordem, ou pode ser transformada numa única seleção
  - $\sigma_{(\text{condição 1})}(\sigma_{(\text{condição 2})}(\dots(\sigma_{(\text{condição n})}(\mathbf{R}))))$
  - $\sigma_{(\text{condição 1})} \mathbf{AND} (\text{condição 2}) \mathbf{AND} \dots (\text{condição n})(\mathbf{R})$

---

## Seleção ...

- **Operador Seleção:**
  - aplicado a cada tupla;
  - $(\text{grau de } \sigma_{(\text{condição})}(\mathbf{R})) = (\text{grau de } \mathbf{R});$
  - $|\sigma_{(\text{condição})}(\mathbf{R})| \leq |\mathbf{R}|;$
  - **seletividade** da condição de seleção: fração de tuplas selecionadas.



# Projeção

- Produz uma nova relação contendo um “subconjunto vertical” da relação argumento, sem “duplicações”:

$\pi_{\text{lista\_atributos}}$  ( relação argumento )

- lista de atributos;
- os atributos são separados por vírgula.

- relação;
- resultado de alguma operação da álgebra relacional.

---

## Consulta 3

- Liste o número e o nome de todos os clientes.

$\pi_{\text{nro\_cli, nome\_cli}}(\text{cliente})$

# Consulta 3

## ■ Relação resultado:



---

## Consulta 4

- Liste o *número* e o *nome* de todos os clientes que possuam saldo devedor inferior a 200,00 reais e morem na Rua X.

---

# Consulta 4

## ■ Passos:

- ❑ Realizar uma operação de seleção para criar uma nova relação que contém somente aqueles clientes com o saldo e o endereço apropriados;
- ❑ Realizar uma projeção sobre a relação resultante do passo anterior, restringindo o resultado desejado às colunas indicadas.

---

## Consulta 4

- Primeiro passo:

$\sigma_{\text{saldo\_dev} < 200,00 \wedge \text{end\_cli} = \text{“Rua X”}}$  (cliente)

- Segundo passo:

$\pi_{\text{nro\_cli}, \text{nome\_cli}}$  (primeiro passo)

## Consulta 4

- Listando o *número* e o *nome* de todos os clientes que possuam saldo inferior a R\$ 200,00 e que morem na Rua X de maneira mais compacta:

$$\pi_{\text{nro\_cli, nome\_cli}} (\sigma_{\text{saldo\_dev} < 200,00 \wedge \text{end\_cli} = \text{"Rua X"}} (\text{cliente}))$$

---

## Projeção ...

- **Projeção** -  $\pi_{\langle \text{atributos} \rangle} \mathbf{R}$ 
  - resultado: relação que tem apenas os atributos indicados na lista de atributos.
  - $\langle \text{atributos} \rangle$ : subconjunto do conjunto de atributos da relação.



---

## Projeção ...

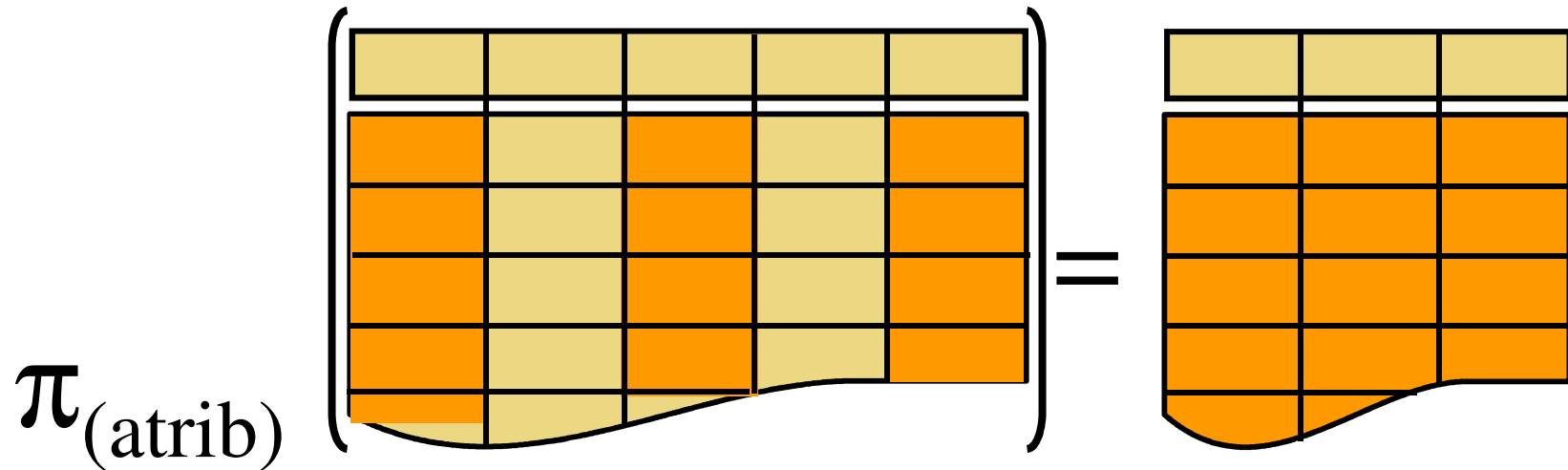
- O resultado de uma operação de projeção é uma relação  $\Rightarrow$  não devem existir tuplas repetidas;
  - Se *<atributos>* contém chave da relação  $\Rightarrow$  resultado não tem tuplas repetidas;
  - Se *<atributos>* não contém chave  $\Rightarrow$  possibilidade de tuplas repetidas.



**eliminação de repetições**

## Projeção ...

- Projeção  $\Rightarrow$  **particionamento vertical**:
  - escolha de algumas “Colunas” da tabela.



---

## Projeção ...

- Operador de **Projeção**:
  - **não é Comutativo**;
  - se  $\langle lista\ B \rangle$  contém  $\langle lista\ A \rangle$ , então vale a igualdade:
    - $\pi_{\langle lista\ A \rangle}(\pi_{\langle lista\ B \rangle}(\mathbf{R})) = \pi_{\langle lista\ A \rangle}(\mathbf{R})$
  - (grau de  $\pi_{\langle lista \rangle}(\mathbf{R})$ ) =  $|\langle lista \rangle|$
  - $|\pi_{\langle lista \rangle}(\mathbf{R})| \leq |\mathbf{R}|$

# Atribuição

- Funcionalidades:
  - Associa uma relação argumento a uma relação temporária;
  - Permite o uso da relação temporária em expressões subsequentes.

relação temporária ← relação argumento

- resultado de alguma operação da álgebra relacional.

## Consulta 4

- Liste o número e o nome de todos os clientes que possuam saldo inferior a R\$ 200,00 e que morem na Rua X.

$$\pi_{\text{nro\_cli, nome\_cli}} (\sigma_{\text{saldo\_dev} < 200,00 \wedge \text{end\_cli} = \text{“Rua X”}} (\text{cliente}))$$

- Usando atribuição:
  - $\text{temp} \leftarrow \sigma_{\text{saldo\_dev} < 200,00 \wedge \text{end\_cli} = \text{“Rua X”}} (\text{cliente})$
  - $\pi_{\text{nro\_cli, nome\_cli}} (\text{temp})$

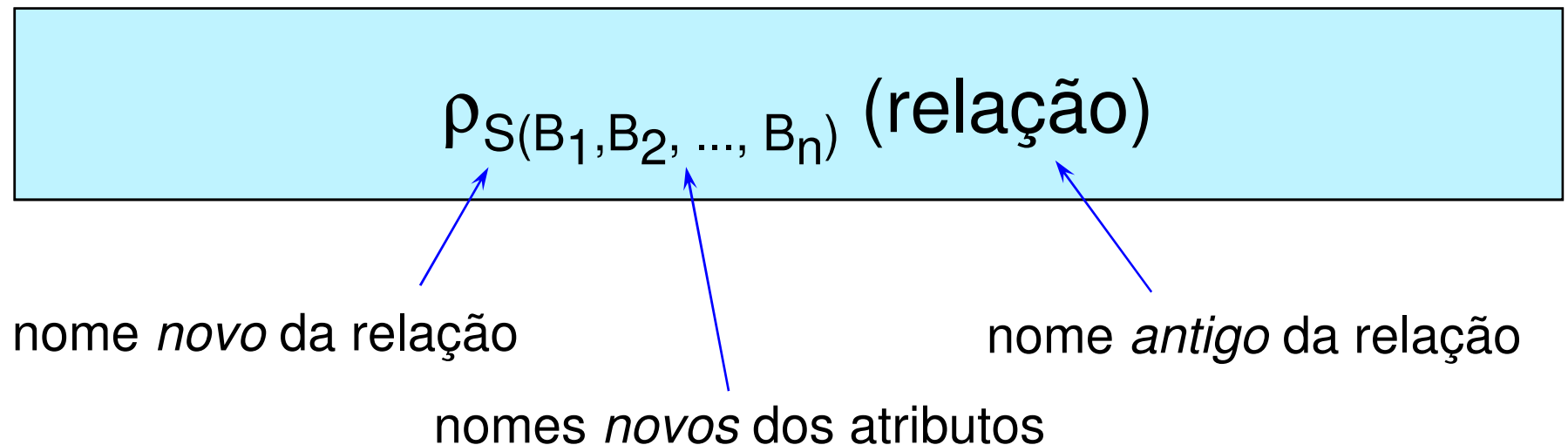
---

# Atribuição

- Características adicionais:
  - permite renomear os atributos de relações intermediárias e final. Exemplo:
  - $R(\text{código}, \text{nome}) \leftarrow \pi_{\text{nro\_cli}, \text{nome\_cli}}(\text{temp})$
- Observações:
  - não adiciona poder à álgebra relacional;
  - geralmente utilizada para expressar consultas complexas.

# Renomear

- Alterar nomes de relações e/ou atributos:
  - Nome da relação;
  - Nomes dos atributos da relação;
  - Nome da relação e nomes dos atributos.



---

# Renomear

## ■ Exemplos:

- ❑ cliente (nro\_cli, nome\_cli, end\_cli, saldo, cod\_vend)
- ❑  $\rho_{\text{comprador}}$  (cliente);
- ❑  $\rho_{(\text{código}, \text{nome}, \text{rua}, \text{saldo}, \text{vendedor})}$  (cliente);
- ❑  $\rho_{\text{comprador}}(\text{código}, \text{nome}, \text{rua}, \text{saldo}, \text{vendedor})$  (cliente).



---

# Produto Cartesiano

- Combina tuplas de duas relações (quaisquer);
- Tuplas da relação resultante:
  - todas as combinações de tuplas possíveis entre as relações participantes.

relação argumento 1  $\times$  relação argumento 2

## 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.co d_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.

## Exercício:

- 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.

# Solução

- Primeiro passo:

- Determinar quem são os usuários atendidos pelo gerente Manoel

$temp_1 \leftarrow \pi_{cliente\_nome} (\sigma_{gerente\_nome = \text{“Manoel”}} (\text{usuário}))$

- Relação resultado  $temp_1$ :

cliente_nome
Márcia

# Solução

- Segundo passo:

- Realizar o produto cartesiano das relações:

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

- relação resultado  $\text{temp}_2$  :

$\text{temp}_1$ . cliente_nome	cliente. cliente_nome	rua	cidade
Márcia	Márcia	Rua X	Itambé
Márcia	Rodrigo	Rua X	Maringá

# Solução

## ■ Terceiro passo:

- eliminar informações inconsistentes:

$$\text{temp}_3 \leftarrow \sigma_{\text{temp}_1.\text{cliente\_nome} = \text{cliente}.\text{cliente\_nome}} (\text{temp}_2);$$

- relação resultado  $\text{temp}_3$  :

$\text{temp}_1.$ cliente_nome	cliente. cliente_nome	rua	cidade
Márcia	Márcia	Rua X	Itambé

# Solução

- Quarto passo:

- Exibir as informações solicitadas:

$\pi_{\text{temp1.cliente\_nome, cidade}}(\text{temp3});$

- Relação resultado:

temp <sub>1</sub> . cliente_nome	cidade
Márcia	Itambé



# Exercício

- Considere a 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* de Rodrigo usando a operação “produto cartesiano”.

# Solução

- Primeiro passo:
  - Determinar o nome da rua e o nome da cidade na qual Rodrigo mora:

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

- Relação resultado  $\text{temp}_1$ :

rua	cidade
Rua X	Maringá

# Solução

Segundo passo:

– realizar o produto cartesiano das relações:

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

– relação resultado  $\text{temp}_2$ :

$\text{temp}_1.\text{rua}$	$\text{temp}_1.\text{cidade}$	cliente_nome	cliente.rua	cliente.cidade
Rua X	Maringá	Márcia	Rua X	Itambé
Rua X	Maringá	Rodrigo	Rua X	Maringá
Rua X	Maringá	Cristina	Rua XTZ	Maringá
Rua X	Maringá	Sofia	Rua X	Maringá
Rua X	Maringá	Ricardo	Rua AAA	Itambé

# Solução

Terceiro passo:

– eliminar informações indesejadas:

$temp_3 \leftarrow \sigma_{\text{cliente\_nome} \neq \text{“Rodrigo”}}(temp_2);$

– relação resultado  $temp_3$ :

$temp_1.rua$	$temp_1.cidade$	cliente_nome	cliente.rua	cliente.cidade
Rua X	Maringá	Márcia	Rua X	Itambé
Rua X	Maringá	Cristina	Rua XTZ	Maringá
Rua X	Maringá	Sofia	Rua X	Maringá
Rua X	Maringá	Ricardo	Rua AAA	Itambé

---

# Solução

Quarto passo:

– exibir as informações solicitadas:

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

– relação resultado:

cliente_nome
Sofia

---

# Junção Natural

- 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  $\bowtie_{\text{condição\_junção}}$  relação argumento 2

Exercício: fazer a junção natural de cliente X vendedor.

---

## 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 $\bowtie$ Vendedor

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

grau: número de atributos diferentes de cliente e de vendedor + (número de atributos comuns)

número de tuplas: entre zero e (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
  - $\theta : \{=, <, >, \leq, \geq, \neq\}$ : *theta join*.  
 $\{ = \}$ : *equijoin*.

# Junção ...

R			S		$R \bowtie S$			
A	B	C	A	D	A	B	C	D
1	a	x	1	d	1	a	x	d
2	b	y	2	d	2	b	y	d
3	a	y						
4	c	y	5	e				

- Interna:

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

# Junção ...

R			S		$R \bowtie S$			
A	B	C	A	D	A	B	C	D
1	a	x	1	d	1	a	x	d
2	b	y	2	d	2	b	y	d
3	a	y	5	e	3	a	y	Null
4	c	y			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			S		$R \bowtie S$			
A	B	C	A	D	A	B	C	D
1	a	x	1	d	1	a	x	d
2	b	y	2	d	2	b	y	d
3	a	y	5	e	5	Null	Null	e
4	c	y						

- 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			S		$R \bowtie S$			
A	B	C	A	D	A	B	C	D
1	a	x	1	d	1	a	x	d
2	b	y	2	d	2	b	y	d
3	a	y			3	a	y	Null
4	c	y			4	c	y	Null
			5	e	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.

# 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 correspondentes possuem os mesmos domínios.

---

# 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



---

# 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

---

# 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

# 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 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 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

- Listar 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:

<b>saldo</b>
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

## 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*.



---

# Exercícios:

- Considere os seguintes esquemas de relações:
  - ❑ empregado (cod\_empregado, primeiro\_nome\_emp, último\_nome\_emp, data\_niver\_emp, end\_emp, sexo\_emp, salário\_emp, cod\_supervisor, nro\_departamento)
  - ❑ departamento (nro\_departamento, nome\_depto, cod\_gerente, data\_início\_gerente)
  - ❑ projeto (nro\_projeto, nome\_projeto, local\_projeto, nro\_departamento)
  - ❑ trabalha\_para (cod\_empregado, nro\_projeto, horas\_trabalhadas)
  - ❑ dependente (cod\_empregado, nome\_dependente, sexo\_depte, data\_niver\_depte, parentesco)

---

■ Pede-se:

1. Liste as informações dos empregados que trabalham para o departamento 4 e que recebem salário maior do que R\$25.000,00 ou que trabalham para o departamento 5 e que recebem salário maior do que R\$30.000,00.
  2. Liste o primeiro nome, o último nome e o salário dos empregados que trabalham para o departamento 4 e que recebem salário maior do que R\$25.000,00.
  3. Liste o código dos empregados que trabalham para o departamento 5 ou que supervisionam um empregado que trabalha para o departamento 5.
  4. Recupere, para cada empregado do sexo feminino, o seu nome completo e os nomes dos seus dependentes. Use a operação de produto cartesiano.
  5. Recupere, para cada departamento, o seu nome e o nome completo de seu gerente.
  6. Recupere o nome completo dos empregados que trabalham em todos os projetos no qual o empregado João Silva trabalha.
  7. Recupere os nomes completos dos empregados que não têm dependentes.
-

---

# Cálculo Relacional de Tupla

- Uma consulta é expressa como:

$$\{ t \mid P(t) \}$$

- O conjunto de tuplas  $t$  tal que o predicado  $P$  é verdadeiro para  $t$ :
  - $t[A]$ : valor da tupla  $t$  para o atributo  $A$ ;
  - $t \in r$ : tupla  $t$  pertence à relação  $r$ .

---

# Esquema Base

agência (nome\_agência, cidade\_agência, fundos)

cliente (nome\_cliente, rua\_cliente, cidade\_cliente)

empréstimo (número\_empréstimo, nome\_agência, total)

devedor (nome\_cliente, número\_empréstimo)

conta (número\_conta, nome\_agência, saldo)

depositante (nome\_cliente, número\_conta)

---

## Consulta 1

- Liste as informações da relação *empréstimo* para os empréstimos com totais superiores a R\$ 1.200.
- Álgebra relacional:

$$\sigma_{\text{total} > 1200} (\text{empréstimo})$$

- Cálculo relacional de tupla:

$$\{ t \mid t \in \text{empréstimo} \wedge t[\text{total}] > 1200 \}$$

## Consulta 2

- Liste os números dos empréstimos para os empréstimos com totais superiores a R\$1.200.
- Álgebra relacional:

$$\pi_{\text{número\_empréstimo}} ( \sigma_{\text{total} > 1200} ( \text{empréstimo} ) )$$

- Cálculo relacional de tupla:

$$\{ t \mid \exists s \in \text{empréstimo} ( t[\text{número\_empréstimo}] = s[\text{número\_empréstimo}] \wedge s[\text{total}] > 1200 ) \}$$

## Consulta 3

- Liste os nomes dos clientes que têm um empréstimo na agência Centro.
- Álgebra relacional:

$$\pi_{\text{nome\_cliente}}(\sigma_{\text{nome\_agência} = \text{"Maringá Centro"}}(\text{devedor} \bowtie \text{empréstimo}))$$

- Cálculo relacional de tupla:

$$\{ t \mid \exists s \in \text{devedor} ( t[\text{nome\_cliente}] = s[\text{nome\_cliente}] \wedge \exists u \in \text{empréstimo} ( u[\text{número\_empréstimo}] = s[\text{número\_empréstimo}] \wedge u[\text{nome\_agência}] = \text{"Centro"} )) ) \}$$

## Consulta 4

- Encontre todos os clientes que possuem uma conta, um empréstimo ou ambos.
- Álgebra relacional:

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

- Cálculo relacional de tupla:

$$\{ t \mid \exists s \in \text{depositante} ( t[\text{nome\_cliente}] = s[\text{nome\_cliente}] ) \vee \exists u \in \text{devedor} ( t[\text{nome\_cliente}] = u[\text{nome\_cliente}] ) \}$$



## Consulta 5

- Encontre todos os clientes que possuem uma conta e um empréstimo.
- Álgebra relacional:

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

- Cálculo relacional de tupla:

$$\{ t \mid \exists s \in \text{depositante} ( t[\text{nome\_cliente}] = s[\text{nome\_cliente}] ) \wedge \exists u \in \text{devedor} ( t[\text{nome\_cliente}] = u[\text{nome\_cliente}] ) \}$$

## Consulta 6

- Encontre todos os clientes que possuem contas, mas não possuem empréstimos.
- Álgebra relacional:

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

- Cálculo relacional de tupla:

$$\{ t \mid \exists s \in \text{depositante} ( t[\text{nome\_cliente}] = s[\text{nome\_cliente}] ) \wedge \neg \exists u \in \text{devedor} ( t[\text{nome\_cliente}] = u[\text{nome\_cliente}] ) \}$$

---

# Definição Formal

- Expressão em cálculo relacional de tupla:

$$\{ t \mid P(t) \}$$

- sendo que P é uma fórmula.
- Variáveis tuplas:
  - variável livre: não quantificada por  $\exists$  ou  $\forall$ ;
  - variável limite: quantificada por  $\exists$  ou  $\forall$ .

---

# Fórmula

- Construída por meio de átomos;
  - Formas de um átomo:
    - $s \in r$ 
      - $s$ : variável tupla
      - $r$ : uma relação
    - $s[x] \Theta u[y]$  e/ou  $s[x] \Theta c$ 
      - $s, u$ : variáveis tuplas
      - $x$ : atributo no qual  $s$  é definido;
      - $y$ : atributo no qual  $u$  é definido
      - $c$ : constante
      - $\Theta = (<, \leq, >, \geq, =, <>)$
-

# Uso de Átomos nas Fórmulas

## ■ Regras

- um átomo é uma fórmula
- se  $P_1$  é uma fórmula  
então  $\neg P_1$  e  $(P_1)$  são fórmulas
- se  $P_1$  e  $P_2$  são fórmulas  
então  $P_1 \vee P_2$ ,  $P_1 \wedge P_2$ ,  $P_1 \Rightarrow P_2$  são fórmulas
- se  $P_1(s)$  é uma fórmula contendo uma variável tupla livre  $s$  e  $r$  é uma relação  
então  $\exists s \in r (P_1(s))$  e  $\forall s \in r (P_1(s))$  são fórmulas