



# Álgebra Relacional

Disciplina: Banco de Dados I

**Prof: Aglaê Pereira Zaupa**

Unoeste – Universidade do Oeste Paulista  
**FIPP – Faculdade do Oeste Paulista**

# Sumário

- ♦ Introdução
- ♦ BD Exemplo
- ♦ Operações:
  - Seleção
  - Projeção
  - Operações da Teoria de Conjuntos
  - Renomeação
  - Junção
  - Operador de Ponto Fixo
  - Divisão
  - Designação
  - Funções Agregadas
- ♦ Conjunto Mínimo de Operações
- ♦ Junção Externa (*Outer Join*)

# Álgebra

- ♦ Na matemática uma álgebra é um conjunto de objetos e um conjunto de operações sobre estes objetos
- ♦ **Exemplo:**
  - Aritmética:
    - Conjunto de números e de operações sobre números (soma, subtração, ...)

# Álgebra Relacional

- ♦ Álgebra desenvolvida para descrever operações sobre uma base de dados relacional\*
- ♦ Os objetos sobre os quais a álgebra opera são **tabelas**
- ♦ Uma operação possui como **operandos** e como **resultado** tabelas
- ♦ **Proposta por E. F. Codd:**
  - A Relational Model of Data for Large Shared Data Banks, 1970 (ACM Classic of the Month) <http://www.acm.org/classics/nov95/>

\* Apesar de existirem operadores para inserção, alteração e exclusão de dados, trabalharemos apenas com **consultas** nestes slides

# Operadores da Álgebra Relacional

- ◆ Operadores originários da teoria dos conjuntos (uma tabela é um conjunto de linhas):
  - União
  - Interseção
  - Diferença
  - Produto Cartesiano
- ◆ Operadores específicos da álgebra relacional:
  - Seleção
  - Projeção
  - Junção
  - Divisão
  - Renomeação

# Álgebra Relacional

## ♦ Porque aprender:

- Compreendendo a álgebra relacional é mais fácil aprender SQL
- Não há SGBD que implemente álgebra diretamente como DML, mas SQL incorpora cada vez mais conceitos de álgebra
- Algoritmos de otimização de consultas definidos sobre álgebra (possível uso interno no SGBD)

# BD exemplo

Tabela Peca

<u>CodPeca</u>	NomePeca	CorPeca	PesoPeca	CidadePeca
P1	Eixo	Cinza	10	Porto Alegre
P2	Rolamento	Preto	16	Rio de Janeiro
P3	Mancal	Verde	30	São Paulo

Tabela Fornecedor

<u>CodFornec</u>	NomeFornec	StatusFornec	CidadeFornec
F1	Silva	5	São Paulo
F2	Souza	10	Rio de Janeiro
F3	Álvares	5	São Paulo
F4	Tavares	8	Rio de Janeiro

Tabela Embarque

<u>CodPeca</u>	<u>CodFornec</u>	QtdeEmbarc
P1	F1	300
P1	F2	400
P1	F3	200
P2	F1	300
P2	F4	350

# Operação de Seleção

- ♦ A seleção tem como operando ***uma tabela***
- ♦ O resultado é ***uma tabela*** que contém as linhas que obedecem a um determinado critério
- ♦ Sintaxe:
  - $\sigma$  <critério de seleção> ( <tabela> )
    - Onde <tabela> é o nome de uma tabela ou uma expressão de álgebra relacional que resulta em uma tabela e <critério de seleção> é uma expressão lógica que envolve literais e valores de atributos da tabela
- ♦ O resultado da seleção tem colunas com os mesmos nomes e domínios da tabela de entrada



# Exemplo de Seleção (BD de peças e fornecedores)

$\sigma_{\text{CodPeca} = \text{'P1'}} (\text{Peca})$

- ◆ Resulta em uma tabela que contém todos os dados da peça de código P1

CodPeca	NomePeca	CorPeca	PesoPeca	CidadePeca
P1	Eixo	Cinza	10	Porto Alegre

# Exemplo de Seleção

$\sigma$  (StatusFornec > 5 and CidadeFornec = 'Rio de Janeiro') (Fornecedor)

- ◆ Resulta em uma tabela com os dados de todos os fornecedores que tenham status maior que 5 e sejam do Rio

CodFornec	NomeFornec	StatusFornec	CidadeFornec
F2	Souza	10	Rio de Janeiro
F4	Tavares	8	Rio de Janeiro

# Seleção é Comutativa

$\sigma (\text{StatusFornec} > 5)$   
 $( \sigma (\text{CidadeFornec} = \text{'Rio de Janeiro'}) ( \text{Fornecedor} ))$

- ♦ O mesmo que o anterior, mas usando duas expressões de Seleção aninhadas
- ♦ O critério de seleção envolve somente valores de atributos de uma linha. Não é possível especificar critérios que envolvam múltiplas linhas nem critérios que envolvam diferentes tabelas

# Operação de Projeção

- ♦ A Projeção tem como operando uma tabela. O resultado é uma tabela que contém apenas as colunas selecionadas
- ♦ Sintaxe:  
 $\pi$  <lista de colunas> ( <tabela> )
  - Onde <tabela> é o nome de uma tabela ou uma expressão de álgebra relacional que resulta em uma tabela e <lista de colunas> é uma lista que contém os nomes de colunas da tabela operando

# Exemplo de Projeção (BD de peças e fornecedores)

$\pi$  CodPeca, NomePeca (Peca)

- ◆ Resulta em uma tabela que contém os códigos e os nomes de todas as peças

CodPeca	NomePeca
P1	Eixo
P2	Rolamento
P3	Mancal

# Exemplo de Projeção

- ♦ A Projeção pode resultar também na eliminação de linhas, caso colunas que são parte da chave primária forem eliminadas
  - Uma tabela é um conjunto de linhas
    - Se uma coluna cujos valores distinguem diferentes linhas é eliminada, surgem linhas duplicadas na tabela, que devem ser eliminadas
- ♦ Exemplo:  
 $\pi$  CidadeFornec (Fornecedor)
- ♦ Resulta em uma tabela que contém todas as cidades em que há fornecedores. Note-se que se houver múltiplos fornecedores na mesma cidade, as linhas duplicadas são eliminadas

CidadeFornec
São Paulo
Rio de Janeiro

# Projeção Generalizada

- ♦ De forma geral, expressões aritméticas podem ser usadas na lista de projeção:

$\pi$  CodPeca, NomePeca, PesoPeca\*1.1322 (Peca)

- ♦ Problema:
  - Qual o nome da terceira coluna?
  - Ver operador de renomeação adiante

# Aninhamento de Operadores

- ◆ Operações diferentes podem ser aninhadas

$\pi \text{ CodFornec, QtdeEmbarc } (\sigma \text{ CodPeca} = \text{'P1'} \text{ (Embarque)})$

- ◆ Resulta em uma tabela com código de fornecedor e quantidade embarcada para cada embarque da peça de código P1

CodFornec	QtdeEmbarc
F1	300
F2	400
F3	200



# Operações da Teoria de Conjuntos

- ♦ A álgebra relacional empresta da teoria de conjuntos quatro operadores: União, Interseção, Diferença e Produto Cartesiano
  - Sintaxe da operação União:  
 $\langle \text{tabela} \rangle_1 \cup \langle \text{tabela} \rangle_2$
  - Sintaxe da operação Interseção:  
 $\langle \text{tabela} \rangle_1 \cap \langle \text{tabela} \rangle_2$
  - Sintaxe da operação Diferença:  
 $\langle \text{tabela} \rangle_1 - \langle \text{tabela} \rangle_2$
  - Sintaxe da operação Produto Cartesiano:  
 $\langle \text{tabela} \rangle_1 \times \langle \text{tabela} \rangle_2$
- ♦ Essas operações possuem duas tabelas como operando. Nos três primeiros casos, essas tabelas devem ser **compatíveis para a operação**:
  - Possuir o mesmo número de colunas
  - O domínio da i-ésima coluna de uma tabela deve ser idêntico ao domínio da i-ésima coluna da outra
- ♦ Quando os nomes das colunas forem diferentes, adota-se a convenção de usar os nomes das colunas da primeira tabela como nomes das colunas do resultado

# Exemplo de Interseção

$\pi$  CodFornec (Embarque)

$\cap$

$\pi$  CodFornec ( $\sigma$  StatusFornec  $> 5$  (Fornecedor))

- ♦ Obtém os códigos de todos os fornecedores que tem embarques e que tem status maior que 5

CodFornec
F2
F4

# Operadores Deriváveis

- ♦ Observar que há operadores de álgebra que são deriváveis de outros
- ♦ Exemplo:
  - Operação de interseção é derivável de:

$$A \cap B = A - (A - B)$$

# Operação de Produto Cartesiano

- ♦ Sintaxe da operação Produto Cartesiano:  
 $\langle \text{tabela} \rangle_1 \times \langle \text{tabela} \rangle_2$
- ♦ O produto cartesiano possui como operandos duas tabelas
- ♦ O resultado é uma tabela cujas linhas são a combinação (todas as combinações possíveis) das linhas das tabelas  $\langle \text{tabela} \rangle_1$  e  $\langle \text{tabela} \rangle_2$  tomando-se uma linha da  $\langle \text{tabela} \rangle_1$  e concatenando-a com uma linha da  $\langle \text{tabela} \rangle_2$

Total de colunas do produto cartesiano =  
Número de colunas da primeira tabela +  
Número de colunas da segunda tabela

Total de linhas do produto cartesiano =  
Número de linhas da primeira tabela X  
Número de linhas da segunda tabela

a	x	=	a x
b	y		a y
c			b x
			b y
			c x
			c y

# Exemplo de Produto Cartesiano

## ♦ Peca X Embarque

Peca					Embarque		
CodPeca	NomePeca	CorPeca	PesoPeca	CidadePeca	CodPeca	CodForec	QtdeEmbarc
P1	Eixo	Cinza	10	Porto Alegre	P1	F1	300
P1	Eixo	Cinza	10	Porto Alegre	P1	F2	400
P1	Eixo	Cinza	10	Porto Alegre	P1	F3	200
P1	Eixo	Cinza	10	Porto Alegre	P2	F1	300
P1	Eixo	Cinza	10	Porto Alegre	P2	F4	350
P2	Rolamento	Preto	16	Rio de Janeiro	P1	F1	300
P2	Rolamento	Preto	16	Rio de Janeiro	P1	F2	400
P2	Rolamento	Preto	16	Rio de Janeiro	P1	F3	200
P2	Rolamento	Preto	16	Rio de Janeiro	P2	F1	300
P2	Rolamento	Preto	16	Rio de Janeiro	P2	F4	350
P3	Mancal	Verde	30	São Paulo	P1	F1	300
P3	Mancal	Verde	30	São Paulo	P1	F2	400
P3	Mancal	Verde	30	São Paulo	P1	F3	200
P3	Mancal	Verde	30	São Paulo	P2	F1	300
P3	Mancal	Verde	30	São Paulo	P2	F4	350

# Exemplo de Produto Cartesiano

- ♦ A operação de Produto Cartesiano não costuma ser usada isoladamente. Normalmente, ela é combinada com uma seleção que envolve as diversas tabelas multiplicadas

$\sigma_{\text{Peca.CodPeca}=\text{Embarque.CodPeca}} (\text{Peca} \times \text{Embarque})$

Peca					Embarque		
CodPeca	NomePeca	CorPeca	PesoPeca	CidadePeca	CodPeca	CodForec	QtdeEmbarc
P1	Eixo	Cinza	10	Porto Alegre	P1	F1	300
P1	Eixo	Cinza	10	Porto Alegre	P1	F2	400
P1	Eixo	Cinza	10	Porto Alegre	P1	F3	200
P2	Rolamento	Preto	16	Rio de Janeiro	P2	F4	350
P2	Rolamento	Preto	16	Rio de Janeiro	P2	F1	300

# Seleção Combinada com Produto Cartesiano

$\pi$  NomePeca ( $\sigma$  Peca.CodPeca=Embarque.CodPeca (Peca X Embarque ) )

- ♦ Obtém os nomes de todas as peças para as quais há embarques

NomePeca
Eixo
Rolamento

# Renomeação

- ♦ Operador para atribuir (dentro de uma consulta) um novo nome a uma tabela
- ♦ Sintaxe  
 $\rho$  <novo nome> ( <nome de tabela> )
- ♦ A tabela denominada <nome de tabela> recebe, apenas dentro da consulta, a denominação <novo nome>
- ♦ Utilizado quando, em uma consulta, é necessário acessar mais de uma linha da mesma tabela ao mesmo tempo



# Renomeação

- ♦ Tabela Exemplo

EMP(CodEmp, NomeEmp, CodEmpGer)

CodEmpGer referencia EMP

- ♦ Obter o nome de cada empregado, que tem gerente, seguido do nome de seu gerente

$$\pi \text{ EMP.NomeEmp, EMPGER.NomeEmp}$$
$$(\sigma \text{ EMP.CodEmpGer} = \text{EMPGER.CodEmp}$$
$$(\text{EMP} \times \rho \text{ EMPGER} (\text{EMP}))$$

# Renomeação

- ♦ Tabela Exemplo

EMP(CodEmp, NomeEmp, CodEmpGer)

CodEmpGer referencia EMP

- ♦ Obter o nome de cada empregado seguido do nome do gerente de seu gerente, caso ele o possua

$\pi$  EMP.NomeEmp, EMPGERGER.NomeEmp  
( $\sigma$  EMP.CodEmpGer = EMPGER.CodEmp AND  
EMPGER.CodEmpGer = EMPGERGER.CodEmp  
(EMP X  $\rho$  EMPGER (EMP) X  $\rho$  EMPGERGER (EMP)))

# Renomeação

- ♦ Qual o resultado desta consulta?

$$\begin{aligned} &\pi \text{ EMP.NomeEmp, EMPGER.NomeEmp} \\ &\quad (\sigma \text{ EMP.CodEmpGer} = \text{EMPGER.CodEmp} \\ &\quad (\text{EMP X } \rho \text{ EMPGER (EMP)}) \\ &\cup \\ &\pi \text{ EMP.NomeEmp, EMPGERGER.NomeEmp} \\ &\quad (\sigma \text{ EMP.CodEmpGer} = \text{EMPGER.CodEmp AND} \\ &\quad \text{EMPGER.CodEmpGer} = \text{EMPGERGER.CodEmp} \\ &\quad (\text{EMP X } \rho \text{ EMPGER (EMP) X } \rho \text{ EMPGERGER (EMP)})) \end{aligned}$$

# Renomeação de Colunas

- ♦ As colunas de uma tabela também podem ser renomeadas
- ♦ Exemplo:

$\rho$  EMPGER(CodEmpGer, NomeEmpGer, CodEmpGerGer) (EMP)

# Operador de Junção

- ♦ A combinação de uma operação de seleção aplicada sobre uma operação de produto cartesiano é usual em aplicações de BD relacionais. É através dela que dados de tabelas relacionadas são associados. Por isso, foi criada a operação de Junção que corresponde exatamente à seqüência de operações em questão
- ♦ **Sintaxe:**
  - $\langle \text{tabela} \rangle_1 \bowtie \langle \text{critério} \rangle \langle \text{tabela} \rangle_2$ 
    - Onde  $\langle \text{tabela} \rangle_x$  é o nome de uma tabela ou uma expressão de álgebra relacional que resulta em uma tabela e  $\langle \text{critério} \rangle$  é uma expressão lógica envolvendo literais e valores de atributos das duas tabelas
- ♦ A Junção tem como operandos duas tabelas. O resultado é equivalente a executar:
  - $\sigma \langle \text{critério} \rangle (\langle \text{tabela} \rangle_1 \times \langle \text{tabela} \rangle_2)$

# Exemplo de Junção (BD de peças e fornecedores)

Embarque ⋈ (Embarque.CodFornec = Fornecedor.CodFornec) Fornecedor

- ♦ Associa cada linha de embarque com a correspondente linha de fornecedor

Embarque			Fornecedor			
CodPeca	CodForec	QtdeEmbarc	CodFornec	NomeFornec	StatusFornec	CidadeFornec
P1	F1	300	F1	Silva	5	São Paulo
P1	F2	400	F2	Souza	10	Rio de Janeiro
P1	F3	200	F3	Álvares	5	São Paulo
P2	F1	300	F1	Silva	5	São Paulo
P2	F4	350	F4	Tavares	8	Rio de Janeiro

# Junção theta, Equijunção e Junção Natural

- ♦ Critério de junção:
  - Qualquer expressão lógica, inclusive com comparações do tipo  $<$ ,  $>$ ,  $<>$ , ... entre os valores de atributos das tabelas envolvidas na junção
  - Essa operação genérica de Junção é chamada de ***Junção theta***
- ♦ Na maior parte dos casos, o  $<\text{critério}>$  de junção é uma expressão como mostrada no exemplo
  - Envolve apenas **igualdade de valores** de atributos de diferentes tabelas
  - Esse tipo de junção é chamada ***Equijunção***

# Equijunção

## ♦ *Sintaxe da Equijunção:*

$\langle \text{tabela} \rangle_1 \bowtie (\langle \text{lista} \rangle_1) , (\langle \text{lista} \rangle_2) \langle \text{tabela} \rangle_2$

- $\langle \text{lista} \rangle_1$  e  $\langle \text{lista} \rangle_2$  são as listas dos nomes das colunas as tabelas 1 e 2 respectivamente cujos valores são comparados um a um, para fazer a junção
- ♦ A operação de Equijunção distingue-se da Junção theta pelo fato de eliminar a segunda coluna em cada um dos pares que são comparados (já que os valores da segunda coluna são idênticos aos primeiros)



# Exemplo de Equijunção (BD de peças e fornecedores)

Embarque  $\bowtie$  (CodFornec), (CodFornec) Fornecedor

- ♦ Associa cada linha de embarque com a correspondente linha de fornecedor

Embarque			Fornecedor		
CodPeca	CodForec	QtdeEmbarc	NomeFornec	StatusFornec	CidadeFornec
P1	F1	300	Silva	5	São Paulo
P1	F2	400	Souza	10	Rio de Janeiro
P1	F3	200	Álvares	5	São Paulo
P2	F1	300	Silva	5	São Paulo
P2	F4	350	Tavares	8	Rio de Janeiro

# Junção Natural

- ♦ No caso anterior, as colunas igualadas possuem os mesmos nomes
- ♦ Para estes casos existe a ***Junção Natural***, na qual as listas de nomes de colunas não necessitam ser especificadas
- ♦ Sintaxe da Junção Natural:  
 $\langle \text{tabela} \rangle_1 \bowtie \langle \text{tabela} \rangle_2$
- ♦ Exemplo de Junção Natural:
  - Associa cada linha de embarque com a correspondente linha de fornecedor  
Embarque  $\bowtie$  Fornecedor
  - O resultado é idêntico ao da Equijunção do exemplo anterior

# Operador de Ponto Fixo

- ♦ Obter o nome de cada empregado seguido do nome de seu gerente, bem como, obter o nome de cada empregado seguido do nome do gerente de seu gerente e assim por diante
- ♦ Caso o número de níveis de gerência seja fixo, é possível resolver com uma série de uniões (como feito anteriormente)
- ♦ Caso o número de níveis de gerência seja variável, é necessária uma operação que implemente ***recursividade***
- ♦ A álgebra relacional original não implementa recursividade, mas há extensões que implementam o **operador de ponto fixo ( $\phi$ )**.

# Operador de Ponto Fixo

- ♦ Exemplo:

EMP(CodEmp, NomeEmp, CodEmpGer)  
CodEmpGer referencia EMP

$\phi ($

CHEFIA =

$\pi$  EMP.CodEmp, EMP.CodEmpGer (EMP)

$\cup$

$\pi$  EMP.CodEmp, CHEFIA.CodEmpGer

( EMP  $\bowtie$  (CodEmpGer), (CodEmp) CHEFIA)

)

# Operador de Ponto Fixo (Avaliação)

- Exemplo:

EMP(CodEmp, NomeEmp, CodEmpGer)  
CodEmpGer referencia EMP

**Passo 1:** CHEFIA inicia vazia;  
Relação CHEFIA é definida  
como igual à projeção de EMP

$\phi ($

CHEFIA =

$\pi$  EMP.CodEmp, EMP.CodEmpGer (EMP)

$\cup$

$\pi$  EMP.CodEmp, CHEFIA.CodEmpGer

( EMP  $\bowtie$  (CodEmpGer), (CodEmp) CHEFIA)

)

# Operador de Ponto Fixo (Avaliação)

- Exemplo:

EMP(CodEmp, NomeEmp, CodEmpGer)  
CodEmpGer referencia EMP

**Passo 1+i (i=1,...):** É feita a junção entre a relação EMP e a relação CHEFIA definida no passo anterior

$\phi ($

CHEFIA =

$\pi$  EMP.CodEmp, EMP.CodEmpGer (EMP)

$\cup$

$\pi$  EMP.CodEmp, CHEFIA.CodEmpGer

( EMP  $\bowtie$  (CodEmpGer), (CodEmp) CHEFIA)

)

# Operador de Ponto Fixo (Avaliação)

- Exemplo:

EMP(CodEmp, NomeEmp, CodEmpGer)  
CodEmpGer referencia EMP

**Final:** O processo termina quando o resultado não muda pela iteração

$\phi ($

CHEFIA =

$\pi$  EMP.CodEmp, EMP.CodEmpGer (EMP)

$\cup$

$\pi$  EMP.CodEmp, CHEFIA.CodEmpGer

( EMP  $\bowtie$  (CodEmpGer), (CodEmp) CHEFIA)

)

# Divisão

- ♦ Como a Junção, a Divisão é uma operação de álgebra relacional que pode ser construída a partir de outras, e é útil para casos que aparecem frequentemente

- ♦ **Sintaxe:**

$\langle \text{tabela} \rangle_1 \div \langle \text{tabela} \rangle_2$

a x	$\div$	X	=	a
a y		y		
a z				
b x				
c y				

- ♦ **Semântica:**

- A operação de divisão tem duas tabelas como operandos
- Os nomes das colunas e respectivos domínios da  $\langle \text{tabela} \rangle_2$  (C2) devem estar contidos dentro dos nomes das colunas e respectivos domínios da  $\langle \text{tabela} \rangle_1$  (C1)
- A tabela resultante tem como nomes de colunas e domínios aqueles que aparecem na  $\langle \text{tabela} \rangle_1$ , mas não aparecem na  $\langle \text{tabela} \rangle_2$  (C1-C2). Para que uma linha apareça no resultado, é necessário que a sua concatenação com cada linha da  $\langle \text{tabela} \rangle_2$  apareça também na  $\langle \text{tabela} \rangle_1$ .



# Exemplo de Divisão (BD de peças e fornecedores)

$(\pi \text{ CodFornec, NomeFornec, CodPeca (Embarque} \bowtie \text{Fornecedor))}$

$\div$

$(\pi \text{ CodPeca}$

$(\sigma \text{ CidadePeca='Porto Alegre' OR}$

$\text{CidadePeca='Rio de Janeiro' (Peca)))$

- ♦ A consulta obtém os códigos e nomes dos fornecedores que possuem embarques ***para todas*** as peças de 'Porto Alegre' ou 'Rio de Janeiro'
- ♦ As palavras “***para todos***” muitas vezes está associada à operação de divisão

# Exemplo de Divisão

$(\pi \text{ CodFornec}, \text{NomeFornec}, \text{CodPeca} \text{ (Embarque } \bowtie \text{ Fornecedor)})$

$\div$

$(\pi \text{ CodPeca}$

$(\sigma \text{ CidadePeca} = \text{'Porto Alegre'} \text{ OR}$   
 $\text{CidadePeca} = \text{'Rio de Janeiro'} \text{ (Peca)))}$

CodFornec	NomeFornec	CodPeca
F1	Silva	P1
F2	Souza	P1
F3	Álvares	P1
F1	Silva	P2
F4	Tavares	P2

$\div$ 

CodPeca
P1
P2

 $=$ 

CodFornec	NomeFornec
F1	Silva

# Designação

- ♦ É conveniente, às vezes, escrever expressões em álgebra relacional com uma designação para a relação, de modo a usá-la como uma variável temporária
- ♦ **Sintaxe:**
  - $\langle \text{temp} \rangle \leftarrow \langle \text{tabela} \rangle$ 
    - Onde  $\langle \text{temp} \rangle$  é o nome da variável temporária a ser criada e  $\langle \text{tabela} \rangle$  é o nome de uma tabela do BD ou uma expressão de álgebra relacional que resulta em uma tabela
- ♦ A designação não adiciona nada à álgebra relacional. Entretanto, ela é conveniente para expressar consultas complexas

## Exemplo de Designação (BD de peças e fornecedores)

- ♦ A consulta utilizada como exemplo de divisão poderia também ser escrita da seguinte maneira:

$\text{temp1} \leftarrow \pi \text{ CodFornec}, \text{ NomeFornec}, \text{ CodPeca} (\text{Embarque} \bowtie \text{Fornecedor})$

$\text{temp2} \leftarrow \pi \text{ CodPeca} (\sigma \text{ CidadePeca} = \text{'Porto Alegre'} \text{ OR } \text{CidadePeca} = \text{'Rio de Janeiro'} (\text{Peca}))$

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

# Funções Agregadas

- ♦ Funções agregadas são aquelas que, quando aplicadas, tomam uma coleção de valores e retornam um valor simples como resultado. A álgebra relacional define as seguintes funções agregadas:
  - **sum**: soma de uma coleção de valores
  - **avg**: média aritmética de uma coleção de valores
  - **min**: menor valor de uma coleção de valores
  - **max**: maior valor de uma coleção de valores
  - **count**: quantidade de valores de uma coleção de valores
- ♦ **Sintaxe:**  
 $F_1 A_1, F_2 A_2, \dots, F_m A_m (<tabela>)$ 
  - Onde  $F_i$  representa o nome de uma função agregada,  $<tabela>$  representa o nome de uma tabela do BD ou uma expressão de álgebra relacional que resulta em uma tabela, e  $A_i$  representa o nome de um atributo dessa tabela ou uma expressão aritmética sobre atributos da tabela

# Exemplos de Funções Agregadas (BD de peças e fornecedores)

**avg** StatusFornec ( Fornecedor )

- ♦ Status médio de fornecedores

**sum** QtdeEmbarc, **avg** QtdeEmbarc ( Embarque )

- ♦ Soma e média de quantidades embarcadas

**count-distinct** CodPeca (Embarque)

- ♦ Quantidade de peças que possuem embarque

# Funções Agregadas com Agrupamento

- ♦ Em alguns casos aplicamos as funções agregadas não somente a um único conjunto de tuplas, mas a diversos grupos, em que cada grupo é um conjunto de tuplas
  - Operação chamada **Agrupamento** (Operador de agregação  $\mathcal{G}$ )
- ♦ **Sintaxe:**  
 $g_1, g_2, \dots, g_n \mathcal{G} F_1 A_1, F_2 A_2, \dots, F_m A_m (<\text{tabela}>)$ 
  - Onde  $<\text{tabela}>$  é o nome de uma tabela do BD ou uma expressão de álgebra relacional que resulta em uma tabela,  $g_1, g_2, \dots, g_n$  constitui uma lista de atributos que serão agrupados, cada  $F_i$  é uma função agregada e cada  $A_i$  é o nome de um atributo da tabela ou uma expressão aritmética sobre os atributos da tabela

# Exemplos de Funções Agregadas

CodPeca, NomePeca  $\mathcal{G}$  **sum** QtdeEmbarc (Embarque  $\bowtie$  Peca)

- ♦ Soma das quantidades embarcadas de cada peça

CodPeca	NomePeca	Sum QtdeEmbarc
P1	Eixo	900
P2	Rolamento	650

CidadeFornec,  $\mathcal{G}$  **sum** StatusFornec, **avg** StatusFornec (Fornecedor)

- ♦ Soma e média dos status de fornecedores de uma mesma cidade

CidadeFornec	Sum StatusFornec	Avg StatusFornec
São Paulo	10	5
Rio de Janeiro	18	9



# Conjunto Mínimo de Operações

- ♦ Muitas operações em álgebra relacional podem ser derivadas de outras
- ♦ Foi identificado um conjunto mínimo (completo) de operações, das quais todas as demais operações da álgebra original podem ser derivadas:
  - Seleção
  - Projeção
  - União
  - Diferença
  - Produto Cartesiano
- ♦ Observe que a operação de Ponto Fixo não faz parte da álgebra relacional original e, portanto, não é derivável deste conjunto de operações

# Restrição da Operação de Junção

- ♦ A operação de Junção concatena duas linhas das tabelas que estão sendo “juntadas” com base no critério de junção (normalmente por igualdade de valores de atributos)
- ♦ Uma linha que não possua nenhuma linha na outra tabela associada pelo critério de junção não aparece na tabela de resultado
- ♦ Há situações em que é necessário garantir que todas as linhas de uma das tabelas de junção (ou de ambas) apareçam no resultado

# Exemplo de Restrição de Junção

Empregado( CodEmp, NomeEmp)

Departamento( CodDep, NomeDep, CodEmpGer)

CodEmpGer referencia Empregado

- ♦ Obter os dados de todos os empregados junto com o nome de seu departamento, caso o empregado seja gerente do departamento
- ♦ Esta consulta não pode ser resolvida com uma única Junção, já que do resultado participariam apenas as linhas de empregados que são gerentes e não dos demais

# Junção Externa (Outer Join)

- ♦ **Exemplo:**  
Empregado  $\bowtie$  (CodEmp=CodEmpGer) Departamento
- ♦ O operador  $\bowtie$  chamado de **Junção Externa Esquerda** (“Left Outer Join”)
- ♦ **Semântica:**
  - A Junção Externa Esquerda contém ao menos uma vez cada linha da tabela à esquerda (no caso a tabela Empregado). Esta linha aparece concatenada com uma linha vazia, caso o critério de junção não seja verdadeiro para nenhuma linha da tabela à direita do operador de junção. Caso o critério de junção seja verdadeiro para uma ou mais linhas da tabela à direita, a linha da tabela à esquerda aparecerá concatenada com uma ou mais linhas da tabela à direita
- ♦ De forma similar podem ser definidas:
  - **Junção Externa Direita** (símbolo  $\bowtie\!-\!$  )
  - **Junção Externa Plena** (símbolo  $\bowtie\!-\!\bowtie$  )

# Combinações de Junções

- ♦ Os três tipos de junções:  
**(theta, equijunção e natural)**  
podem ser combinados com as quatro alternativas:  
**(junção interna, junção externa esquerda,  
junção externa direita e junção externa total)**  
formando assim 12 combinações de junção possíveis

## Referências

- ♦ SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. Makron Books, 3ª Ed., 1999.
- ♦ ELMASRI, Ramez; NAVATHE, S. B. **Sistemas de Banco de Dados**. Addison Wesley, 4ª Ed., 2005.
- ♦ HEUSER, Carlos Alberto. **Fundamentos de Banco de Dados** (Notas de Aula)
- ♦ RANAKRISHMAN, R.; GEHRKE, J. **Database Management Systems**, McGraw-Hill, 2003.