

Banco de Dados

Capítulo 2: Modelo Relacional

Bach. em Ciência da Computação

UFPB/CCT

Cláudio Baptista, PhD

baptista@dsc.ufpb.br

2.1 Histórico Modelo Relacional

- Foi introduzido por Codd (1970)
- Tornou-se um padrão de fato para aplicações comerciais, devido a sua simplicidade e performance.
- É um modelo formal, baseado na teoria matemática das relações
- Um dos SGBD's precursores que implementaram este modelo foi o System R (IBM). Baseado em seus conceitos surgiram: DB2 (IBM), SQL-DS (IBM), Oracle, Informix, Ingres, Sybase entre outros.

2.2 Conceitos do Modelo Relacional

- O modelo relacional representa os dados num BD como uma coleção de tabelas (relações).
- Cada tabela terá um nome, que será único, e um conjunto de atributos com seus respectivos nomes e domínios.
- Todos os valores de uma coluna são do mesmo tipo de dados.

2.2 Conceitos do Modelo Relacional

- Exemplo de uma tabela de empregados de uma empresa

Ex.: Empregado

Matr	Nome	Endereço	Função	Salário	Depart
100	Ana	R. da Juçaras, 12, Alto	Secretária	500,00	D1
250	Pedro	R. J. Silva, 24, Calhau	Engenheiro	1500,00	D1
108	André	R. Itália, 33, B. Nações	Técnico	950,00	D2
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00	D2
105	Sônia	R. Oliveira, 76, Alto	Engenheiro	2500,00	D1

2.2 Conceitos do Modelo Relacional

- **Terminologia:**

- Na terminologia de BD Relacional uma linha é chamada tupla, um nome de coluna é chamado de atributo e cada tabela é chamada de relação.

- **Definições:**

- **Domínio:** Um domínio D é um conjunto de valores atômicos.
 - Ex.: Fone: conjunto de 10, dígitos
CPF: conjunto de 7 dígitos
Idade_Empregado: $16 \leq \text{idade} \leq 70$
Departamentos: conjunto de departamentos de uma empresa.

2.2 Conceitos do Modelo Relacional

- A cada domínio está associado um tipo de dados ou formato.
Ex.: Fone: (ddd) ddd-dddd onde $d = \{0,1,2,\dots,9\}$
IdadeEmpregado: inteiro entre 16 e 70.
- Um esquema de relação R , denotado por $R(A_1, A_2, \dots, A_n)$, é um conjunto de atributos $R = \{A_1, A_2, \dots, A_n\}$. Cada atributo A_i é o nome de um papel realizado por algum domínio D na relação R .
- O *grau* de uma relação é o número de atributos que seu esquema contém.
 - Ex.: Seja o esquema Estudante (matrícula, nome, fone, idade, curso) \Rightarrow grau = 5

2.2 Conceitos do Modelo Relacional

- Um instante (snapshot) de relação r , do esquema $R(A_1, A_2, \dots, A_n)$, denotado por $r(R)$, é o conjunto de n -tuplas $r = \{t_1, t_2, \dots, t_n\}$. Cada tupla t é uma lista ordenada de valores $t = \langle v_1, v_2, \dots, v_n \rangle$
- Os termos *intenção* para o esquema R e *extensão* para instante de relação são também empregados.
- Uma instância $r(R)$ é um subconjunto do produto cartesiano dos domínios de R .
- $r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$

2.2 Conceitos do Modelo Relacional

- **Características de Relações:**
 - A ordem das tuplas e dos atributos não tem importância
 - Todo atributo possui valor atômico
 - Cada atributo numa relação tem um nome que é único dentro da relação.
 - Todas as tuplas devem ser únicas (conjunto).

2.2 Conceitos do Modelo Relacional

- Chaves de uma Relação

- Valem os mesmos conceitos de superchave, chave candidata e chave primária vistos no capítulo 2.
- Convenciona-se sublinhar os atributos que compõem a chave primária. Ex.: Empregado (Matrícula, Nome, Endereço, Função, Salário)
- Um mesmo atributo pode ter nomes diferentes nas diversas relações em que participa. Ex.: Empregado (Matrícula, Nome, Endereço, Função, Salário, Dep) e Departamento (CodDepart, Nome, Endereço)
- Atributos que representam diferentes conceitos podem ter o mesmo nome. Ex.: Ver os atributos Nome do exemplo anterior.

2.2 Conceitos do Modelo Relacional

- **Restrições de integridade**
 - **Integridade de Chave:** Toda tupla tem um conjunto de atributos que a identifica de maneira única na relação.
 - **Integridade de Entidade:** Nenhum valor de chave primária poderá ser NULO.
 - **Integridade Referencial:** Uma relação pode ter um conjunto de atributos que contém valores com mesmo domínio de um conjunto de atributos que forma a chave primária de uma outra relação. Este conjunto é chamado *chave estrangeira*.

2.2 Conceitos do Modelo Relacional

Exemplo de Integridade Referencial

Departamento

<i>CodDep</i>	<i>Nome</i>	<i>MatrGerent</i>
<i>D2</i>	Produção	210
<i>D1</i>	Custos	105
<i>D5</i>	Pessoal	NULL



Empregado

<i>Mat</i>	<i>Nome</i>	<i>Endereço</i>	<i>Função</i>	<i>Salário</i>	<i>Dep</i>
<i>100</i>	Ana	R. Pedro I, 12, Alto	Secretária	500,00	D1
<i>250</i>	Pedro	R. J. Silva, 24, Calhau	Engenheiro	1500,00	D1
<i>108</i>	André	R. Itália, 33, Nações	Técnico	950,00	D2
<i>210</i>	Paulo	R. Pará, 98, Estados	Engenheiro	1810,00	D2
<i>105</i>	Sônia	R. Olivas, 76, Alto	Engenheiro	2500,00	D1

2.2 Conceitos do Modelo Relacional

- Um conjunto de atributos de uma relação $R1$ é uma chave estrangeira se satisfaz às seguintes regras:
 - Os atributos da chave estrangeira têm o mesmo domínio dos atributos da chave primária de outra relação $R2$.
 - Um valor da chave estrangeira numa tupla $t1$ de $R1$ possui o mesmo valor da chave primária para alguma tupla $t2$ em $R2$ ou é NULO.
- A integridade referencial estabelece que todo valor de chave estrangeira numa relação deve corresponder a um valor de chave primária de uma segunda relação ou deve ser nulo.

2.2 Conceitos do Modelo Relacional

- Uma chave estrangeira pode referenciar-se a sua própria relação (auto-relacionamento).
Ex.: `Empregado(matrícula, nome, salário, matr_supervisor)`
- As restrições de integridade devem ser implementadas pelo SGBD. Muitos SGBD's implementam integridade de chave e de entidade, mas não implementam integridade referencial.
- Estudaremos as integridades semânticas mais a frente.
- Ex.: "Nenhum empregado pode ganhar mais que seu gerente"

2.2.1 Operações de atualização em relações

- *Inserção:*

- 1. Inserir $\langle '102', 'André', \text{null}, 'Engenheiro', '1.980', 'D2' \rangle \Rightarrow$ é aceito sem problemas
- 2. Inserir $\langle '100', 'Maria', \text{null}, 'Técnica', '950', 'D1' \rangle \Rightarrow$ viola a restrição de chave.
- 3. Inserir $\langle \text{null}, 'Cecília', \text{null}, 'Engenheiro', '1.950', 'D1' \rangle \Rightarrow$ viola restrição de integridade de entidade.
- 4. Inserir $\langle '108', 'Mauro', 'Rua 4', 'Técnico', '980', 'B6' \rangle \Rightarrow$ viola a restrição de integridade referencial.

2.2.1 Operações de atualização em relações

- *Inserção:*

- O que fazer quando se detectar uma violação de integridade?
 - Rejeitar a inserção (podendo explicar o porquê)
 - Tentar corrigir a anomalia para depois inserir.

2.2.1 Operações de atualização em relações

- Remoção:

- 1. Remover da tabela empregado a tupla com matrícula = '100'. => remoção aceita sem problemas.
- 2. Remover da tabela departamento a tupla com CodDep = 'D1'. => viola a regra de integridade referencial. Pois existem empregados que estão alocados neste departamento.

2.2.1 Operações de atualização em relações

- **Remoção:**

- O que fazer quando uma violação ocorrer numa remoção?
 - Rejeitar a remoção
 - Dar o efeito cascata na remoção, removendo todas as tuplas referenciadas por aquela tupla que está sendo removida.
 - Modificar os atributos referenciados para novos valores ou nulos (caso não façam parte da chave primária).
- Dos três tipos de restrições de integridade discutidas, uma operação de remoção poderá violar apenas a integridade referencial.

2.2.1 Operações de atualização em relações

- *Modificação:*

- 1. Modificar o salário do empregado com matrícula='250' => operação aceita sem problemas.
- 2. Modificar o número do departamento da tupla de empregado com matrícula '210' para 'D1' => operação aceita sem problemas.
- 3. Modificar o número do departamento de empregado '108' para 'D9' => viola a integridade referencial
- 4. Modificar a matrícula do empregado '100' para '250' => viola regra de integridade de chave.

2.3 Álgebra Relacional

- É uma linguagem de banco de dados procedural e formal.
 - Seja o esquema relacional de uma empresa hipotética a seguir:
 - Empregado(matr, nomeE, endereço, sexo, salário, supervisor, depto)
 - Departamento(codDepto, nomeD, matrGerente)
 - DepLocalizações(codDepto, Localização)
 - Alocação(matrEmp, codProj, numHoras)
 - Projetos(codProj, nome, localização, deptoControla)
 - Dependentes(matrEmp, nomeDep, sexo, dataNasc, parentesco)

2.3 Álgebra Relacional

- Usaremos o esquema acima para exemplificar os diversos operadores da álgebra relacional.
- **Operação de Seleção (σ)**
 - Seleciona um subconjunto de tuplas de uma relação, de acordo com uma condição
 - Sintaxe: $\sigma_{\langle \text{predicado} \rangle} (\langle \text{Relação} \rangle)$
 - Onde: σ = Operador de seleção
 - Predicado: $\langle \text{atributo} \rangle \langle \text{op} \rangle \langle \text{constante} \rangle$ ou $\langle \text{atributo} \rangle \langle \text{op} \rangle \langle \text{atributo} \rangle$
 $\langle \text{op} \rangle = \{ =, >, <, \leq, \geq, \neq \}$
 - No predicado podemos ter as cláusulas conectadas pelos conectivos Booleanos AND, OR e NOT.

2.3 Álgebra Relacional

- Exemplo 1: Selecione os empregados que trabalham no departamento 4.

$\sigma_{\text{DEPTO} = 4}$ (Empregado)

- Exemplo 2: Selecione os empregados que ganham mais de R\$ 3.000,00

$\sigma_{\text{salário} > 3000}$ (Empregado)

- Exemplo 3: Selecione os empregados que ganham mais de R\$2.000,00 e trabalham no departamento 4, ou ganham menos de R\$500,00 e trabalham no departamento 5.

$\sigma_{(\text{salário} > 3000 \text{ AND } \text{depto} = 4) \text{ OR } (\text{salário} < 500 \text{ AND } \text{depto} = 5)}$ (Empregado)

2.3 Álgebra Relacional

- O operador de seleção é unário (aplicado a uma única relação)
- O grau da relação resultante é o mesmo da relação original
- O número de tuplas da relação resultante é menor ou igual ao número de tuplas da relação original.

- A seleção é comutativa:

$$\sigma_{\langle \text{COND1} \rangle} (\sigma_{\langle \text{COND2} \rangle} (R)) = \sigma_{\langle \text{COND2} \rangle} (\sigma_{\langle \text{COND1} \rangle} (R))$$

- $\sigma_{\langle \text{COND1} \rangle} (\sigma_{\langle \text{COND2} \rangle} (\dots (\sigma_{\langle \text{CONDn} \rangle} (R)) \dots))$

2.3 Álgebra Relacional

- Operação de Projeção
 - Seleciona um subconjunto de atributos de uma dada relação.
 - Sintaxe: $\pi_{\langle \text{LISTA DE ATRIBUTOS} \rangle} (\text{Relação})$
 - Obs: Se a lista de atributos inclui apenas atributos não-chave, tuplas duplicadas poderão aparecer no resultado, porém, a operação de projeção elimina esta duplicação.
 - Exemplo: $\pi_{\langle \text{FUNÇÃO, SALÁRIO} \rangle} (\text{Empregado})$

2.3 Álgebra Relacional

- O número de tuplas da relação resultante será menor ou igual ao da relação original

$$\pi_{\langle \text{LISTA-1} \rangle} (\pi_{\langle \text{LISTA-2} \rangle} (\text{Relação})) = (\pi_{\langle \text{LISTA-1} \rangle} (\text{Relação})) \Leftrightarrow \langle \text{LISTA-2} \rangle \supset \langle \text{LISTA-1} \rangle$$

- Exemplo:

$$\pi_{\text{NOME, SALÁRIO}} (\pi_{\text{NOME, FUNÇÃO, SEXO, SALÁRIO}} (\text{Empregado})) = \pi_{\text{NOME, SALÁRIO}} (\text{Empregado})$$

2.3 Álgebra Relacional

- Combinando Seleção e Projeção:
 - Exemplo: Obtenha o nome e salário dos empregados do departamento 5
 $\pi_{\text{NOME, SALÁRIO}}(\sigma_{\text{DEPTO} = 5}(\text{Empregado}))$
 - Alternativamente podemos usar uma notação que usa uma sequência dos resultados dando nome as relações intermediárias:
 $\text{EmpDepto5} \leftarrow \sigma_{\text{DEPTO} = 5}(\text{Empregado})$
 $\text{Resultado} \leftarrow \pi_{\text{NOME, SALÁRIO}}(\text{EmpDepto5})$

2.3 Álgebra Relacional

- União
- A união de duas relações, $R \cup S$, é o conjunto de tuplas que está em R ou S ou em ambas. Duplicatas são eliminadas.

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i>	b	c
<i>d</i>	a	f
<i>c</i>	b	d

S

<i>D</i>	<i>E</i>	<i>F</i>
<i>b</i>	g	a
<i>d</i>	a	f

R \cup S

<i>a</i>	<i>b</i>	<i>c</i>
<i>d</i>	a	f
<i>c</i>	b	d
<i>b</i>	g	a

2.3 Álgebra Relacional

- Exemplo: Obtenha a matrícula dos empregados que trabalham no departamento 5 ou supervisionam empregados que trabalham no departamento 5.

$$\text{EmpDepto5} \leftarrow \sigma_{\text{depto} = 5}(\text{Empregados})$$
$$\text{Temp1} \leftarrow \pi_{\text{matricula}}(\text{EmpDepto5})$$
$$\text{Temp2} \leftarrow \pi_{\text{supervisor}}(\text{EmpDepto5})$$
$$\text{Resultado} \leftarrow \text{Temp1} \cup \text{Temp2}$$

2.3 Álgebra Relacional

- **Interseção**

- A interseção de duas relações, $R \cap S$, é uma relação que inclui todas as tuplas que estão em R e em S.

R		
<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i>	b	c
<i>d</i>	a	f
<i>c</i>	b	d

S		
<i>D</i>	<i>E</i>	<i>F</i>
<i>b</i>	g	a
<i>d</i>	a	f

$R \cap S$

$R \cap S$		
<i>d</i>	a	f

2.3 Álgebra Relacional

- **Diferença**

- A diferença entre duas relações $R - S$, é o conjunto de tuplas que estão em R mas não estão em S .

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i>	<i>b</i>	<i>c</i>
<i>d</i>	<i>a</i>	<i>f</i>
<i>c</i>	<i>b</i>	<i>d</i>

S

<i>D</i>	<i>E</i>	<i>F</i>
<i>b</i>	<i>g</i>	<i>a</i>
<i>d</i>	<i>a</i>	<i>f</i>

R-S

<i>a</i>	<i>b</i>	<i>c</i>
<i>c</i>	<i>b</i>	<i>d</i>

2.3 Álgebra Relacional

- Observações:
 - 1) As operações de conjunto (União, Interseção, Diferença) devem ser compatíveis de união. Duas relações $R(A_1, A_2, \dots, A_n)$ e $S(B_1, B_2, \dots, B_n)$ são compatíveis de união se têm o mesmo grau e $\text{domínio}(A_i) = \text{domínio}(B_i)$, para $1 \leq i \leq n$.
 - 2) $R \cup S = S \cup R$
 $R \cap S = S \cap R$
 $R - S \neq S - R$
 $R \cup (S \cap T) = (R \cup S) \cap T$
 $R \cap (S \cup T) = (R \cap S) \cup T$

2.3 Álgebra Relacional

- **Produto Cartesiano**

- O produto Cartesiano de duas relações R X S combina cada tupla de R com cada tupla de S .
- O resultado de $R(A_1, A_2, \dots, A_n)$ X $S(B_1, B_2, \dots, B_m)$ é uma relação Q com $n + m$ atributos $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$.
- Se R tem x tuplas e S tem y tuplas $\Rightarrow R$ X S terá $x \cdot y$ tuplas

2.3 Álgebra Relacional

- Exemplo: Obtenha para cada empregado do sexo feminino, uma lista dos nomes de seus dependentes
- $Mulher \leftarrow \sigma_{\text{sexo}='F'}(\text{Empregados})$
- $\text{NomesMulheres} \leftarrow \pi_{\text{matrícula}, \text{nome}}(\text{Mulher})$
- $\text{DependentesMulher1} \leftarrow \text{NomesMulheres} \times \text{Dependentes}$
- $\text{DependentesMulher2} \leftarrow \sigma_{\text{matr} = \text{matrEmp}}(\text{DependentesMulher1})$
- $\text{Resultado} \leftarrow \pi_{\text{nomeE}, \text{nomeDep}}(\text{DependentesMulher2})$

2.3 Álgebra Relacional

- **Junção**

- Uma junção de duas relações $R(A_1, A_2, \dots, A_n)$ e $S(B_1, B_2, \dots, B_m)$, denotada por $R \bowtie_{\langle \text{condição de junção} \rangle} S$, é usada para combinar tuplas de duas relações numa única tupla.
- O resultado de uma junção é uma relação Q com $n + m$ atributos $Q(A_1, \dots, A_n, B_1, \dots, B_m)$
- Q contém uma tupla para cada combinação de tuplas $(R \times S)$ que satisfaz a condição de junção
- Uma condição de junção tem a forma:
 $\langle \text{cond} \rangle \text{ AND } \langle \text{cond} \rangle \text{ AND } \dots \text{ AND } \langle \text{cond} \rangle$
onde, $\langle \text{cond} \rangle$: $A_i \theta B_i$, A_i é atributo de R , B_i é atributo de S , $\text{dom}(A_i) = \text{dom}(B_i)$ e $\theta = \{ =, <, >, \geq, \leq, \neq \}$

2.3 Álgebra Relacional

- Exemplo: Obtenha o nome do gerente de cada departamento

DeptoGer \leftarrow Departamento $\mid x \mid$ matrGer = matr
Empregado

Resultado $\leftarrow \pi_{\text{nomeD}, \text{nomeE}}(\text{DeptoGer})$

- Quando a condição de junção é uma igualdade a junção é chamada de equijoin.
- Junção Natural é uma equijoin onde um dos atributos com valores repetidos (condição de junção) é eliminado.

2.3 Álgebra Relacional

- **Divisão**

- A divisão de duas relações $R \div S$, onde $\text{atributos}(S) \subseteq \text{atributos}(R)$, resulta na relação T com $\text{atributos}(T) = \{\text{atributos}(R) - \text{atributos}(S)\}$, onde para cada tupla t que aparece no resultado, os valores de t devem aparecer em R combinado com cada tupla de S .

2.3 Álgebra Relacional

- Exemplo de Divisão

R	
A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S
A
a1
a2
a3

R ÷ S
B
b1
b4

2.3 Álgebra Relacional

- Obs.: Quase sempre, a divisão é usada quando temos nas consultas frases do tipo “para todos”
- Exemplo: Obtenha o nome dos empregados que trabalham em todos os projetos que Silva trabalha

Silva $\leftarrow \sigma_{\text{nome} = \text{'silva'}}(\text{Empregados})$

ProjSilva $\leftarrow \pi_{\text{codProj}} (\text{Alocação} \mid x \mid_{\text{matrEmp} = \text{matrSilva}})$

ProjEmp $\leftarrow \pi_{\text{codProj}, \text{matrEmp}} (\text{Alocação})$

TrabProjSilva $\leftarrow (\text{ProjEmp} \div \text{ProjSilva})$

Result $\leftarrow \pi_{\text{nome}} (\text{TrabProjSilva} \mid x \mid_{\text{matrEmp} = \text{matrEmpregado}})$

2.3 Álgebra Relacional

- Operação Rename
 - Permite que renomeemos Relações e/ou atributos para que se evite a ambigüidade na hora de compararmos atributos com mesmo nome de diferentes relações.
 - Pode renomear uma relação ou os atributos da relação ou ambos.
 - A operação Rename quando aplicada a uma relação R de grau N é denotada por:
 - $\rho_{S(B_1, B_2, \dots, B_n)}(R)$ ou
 - $\rho_S(R)$ ou
 - $\rho_{(B_1, B_2, \dots, B_n)}(R)$ onde:
 - ρ denota o operador Rename
 - S é o nome da nova relação
 - B_1, \dots, B_n são os nomes dos novos atributos

2.3 Álgebra Relacional

- Exemplo de consultas
 - 1) Obtenha o nome e o endereço de todos os empregados do departamento de 'Pesquisa'

$$\begin{array}{l} \text{Pesquisa} \\ \text{'Pesquisa'}(\text{Departamento}) \end{array} \leftarrow \sigma_{\text{nomeD} = \text{'Pesquisa'}} \text{Resultado} \leftarrow \pi_{\text{nomeE}, \text{endereço}} \text{Empregado} \leftarrow \sigma_{\text{depto} = \text{codDepto} = \text{'Pesquisa'}}$$

2.3 Álgebra Relacional

- 2) Para cada projeto localizado em 'Natal', liste o código do projeto, o código do departamento que controla o projeto e o nome, endereço e salário do gerente deste departamento

ProjNatal $\leftarrow \sigma_{\text{localização} = \text{'Natal'}} (\text{Projetos})$

Result1 $\leftarrow (\text{ProjNatal} \bowtie_{\text{depControla} = \text{codDepto}} \text{Departamento})$

Result2 $\leftarrow (\text{Result1} \bowtie_{\text{matrGerente} = \text{matr}} \text{Empregado})$

Resultado $\leftarrow \pi_{\text{codProj}, \text{codDepto}, \text{nomeE}, \text{endereço}, \text{salário}} (\text{Result2})$

2.3 Álgebra Relacional

- 3) Encontre os nomes dos empregados que trabalham em todos os projetos do departamento 6

$\text{ProjDep6} \leftarrow \pi_{\text{codProj}} (\sigma_{\text{depControla} = 6} (\text{Projetos}))$

$\text{Temp_1} \leftarrow \pi_{\text{matrEmp}, \text{codProj}} (\text{Alocação})$

$\text{Temp_2} \leftarrow (\text{Temp1} \div \text{ProjDep6})$

$\text{Resultado} \leftarrow \pi_{\text{nomeE}} (\text{Temp_2} \mid x \mid_{\text{matrEmo}} = \text{matr} \text{ Empregado})$

2.3 Álgebra Relacional

- 4) Faça uma lista dos códigos dos projetos que envolvem um empregado cujo nome é 'Silva' como trabalhador ou como gerente do departamento que controla o projeto.

$Silva \leftarrow \pi_{matr} (\sigma_{nomeE='Silva'} (Empregado))$

$ProjSilvaPart \leftarrow \pi_{codProj} (Silva \mid x \mid$
 $matr=matrEmp \text{ Alocação})$

$DepSilvaGer \leftarrow \pi_{codDepto} (Silva \mid x \mid$
 $matr=matrGerente \text{ Departamento})$

$ProjDepSilGer \leftarrow \pi_{codProj} (DepSilvaGer$
 $\mid x \mid codDepto=deptoControla \text{ Projeto})$

$Resultado \leftarrow ProjSilvaPart \cup$
 $ProjSilvaPart$

2.3 Álgebra Relacional

- 5) Liste os nomes dos empregados que não têm dependentes

$\text{TodosEmpr} \leftarrow \pi_{\text{matr}} (\text{Empregado})$

$\text{EmpComDep} \leftarrow \pi_{\text{matrEmp}} (\text{Dependentes})$

$\text{EmpSemDep} \leftarrow (\text{TodosEmpr} - \text{EmpComDep})$

$\text{EmpSemDep2} \leftarrow \rho_{(\text{matricula})} (\text{EmpSemDep})$

$\text{Resultado} \leftarrow \pi_{\text{nomeE}} (\text{EmpSemDep2} \mid x \mid \text{matricula} = \text{matr} \text{Empregado})$

2.3 Álgebra Relacional

- 6) Liste os nomes dos gerentes que têm pelo menos 1 dependente

Gerentes $\leftarrow \pi_{\text{matrGerente}}(\text{Departamento})$

EmpComDepend $\leftarrow \pi_{\text{matrEmp}}(\text{Dependentes})$

GerComDepend $\leftarrow \rho_{(\text{matrG})}(\text{Gerentes} \cap \text{EmpComDepend})$

Resultado $\leftarrow \pi_{\text{nomeE}}(\text{GerComDepend} \mid x \mid \text{matrG} = \text{matr Empregado})$