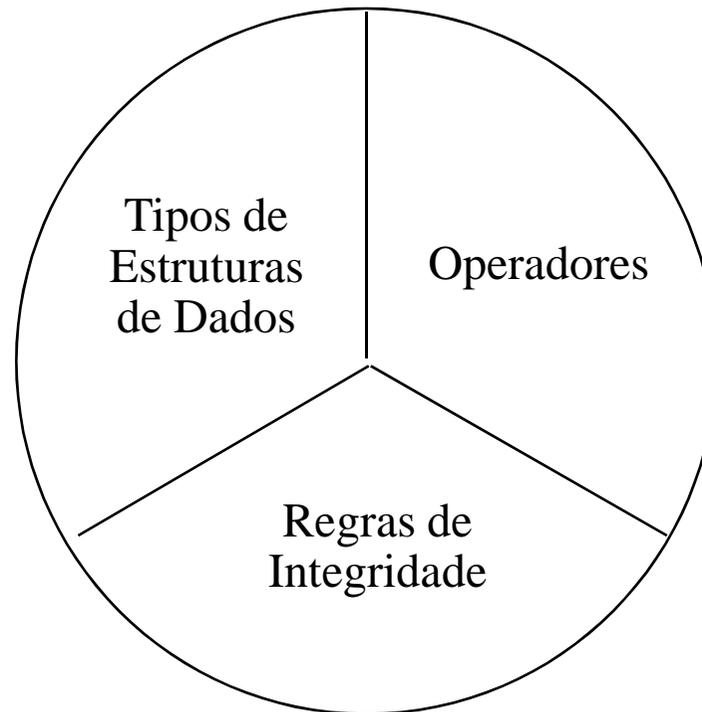


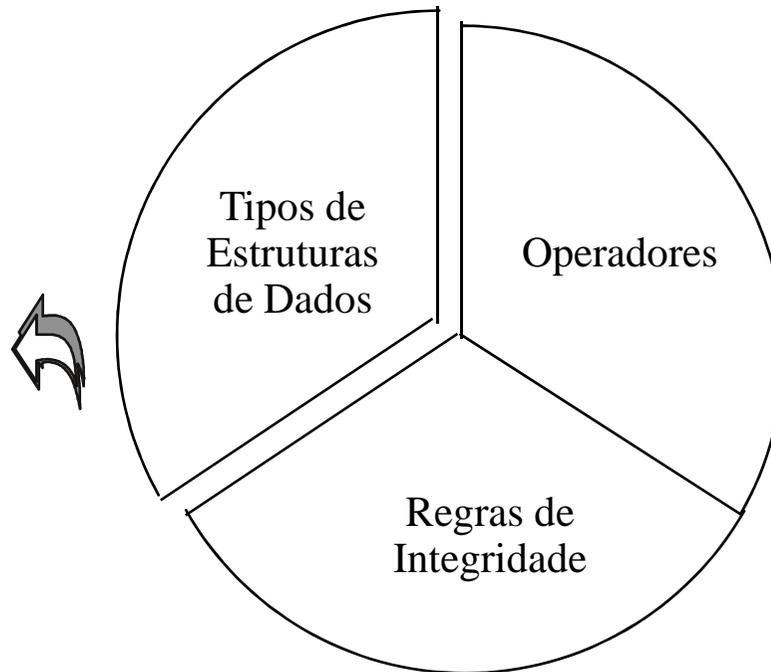
Modelo Relacional

Modelo de Dados



Estruturas de Dados do Modelo Relacional

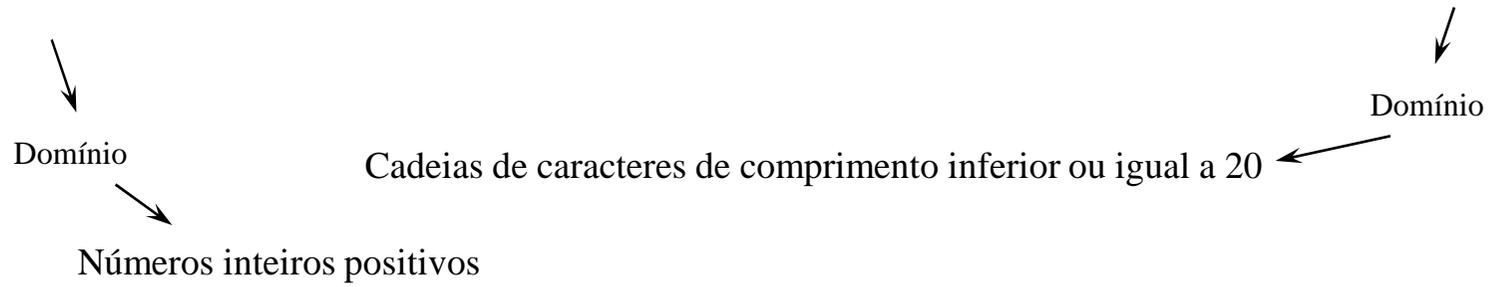
RELAÇÕES
implementadas
na forma de
TABELAS



Exemplo de Relação

Empregado

NºEmp	Nome	Telef	Categ	Salario	Comissão	Função
10	Antunes	12554	B	100	15	Analista
20	Bento	54321	A	250	50	Director
30	Correia		E	60.5		Porteiro
40	Dias	23457	C	90.5		Programador
50	Edmundo		B	120	12.5	Contabilista
...



Relação (1)

Dada uma coleção de conjuntos D_1, D_2, \dots, D_n (não necessariamente disjuntos), R é uma relação naqueles conjuntos se for constituída por um conjunto de n -uplos ordenados $\langle d_{i1}, d_{i2}, \dots, d_{in} \rangle$ tais que:

$\forall i \forall j \ d_{ij} \in [D_j \text{ ou } \text{é nulo (null)}]$

Cardinalidade \longleftrightarrow Número de n -uplos da relação

Domínios de R \longleftrightarrow D_1, D_2, \dots, D_n

Relação (2)

Relação  Tabela de valores, com algumas características especiais:

- Os valores de cada coluna têm todos o mesmo domínio (conjunto de valores possíveis) e cada coluna tem um **título**
- Não existe a noção de "*posição*" em relação às colunas, i.e., a ordem das colunas é arbitrária - não existe a noção de "*próxima coluna*" ou "*coluna anterior*"
- Não existe a noção de "número de linha", i.e., a ordem das linhas é arbitrária - não existe a noção de "*próxima linha*" ou "*linha anterior*"

Podemos alterar a ordem das linhas ou das colunas, sem que isso afecte o conteúdo da relação

Conceitos

Grau da Relação \longleftrightarrow Número de colunas

Atributo \longleftrightarrow Coluna

Esquema de Relação \longleftrightarrow Definição de uma relação

Exemplo:

Empregado (NºEmp, Nome, Telef, Categ, Salario, Comissão, Função)

Esquema Relacional \longleftrightarrow Definição de uma Base de Dados Relacional

=

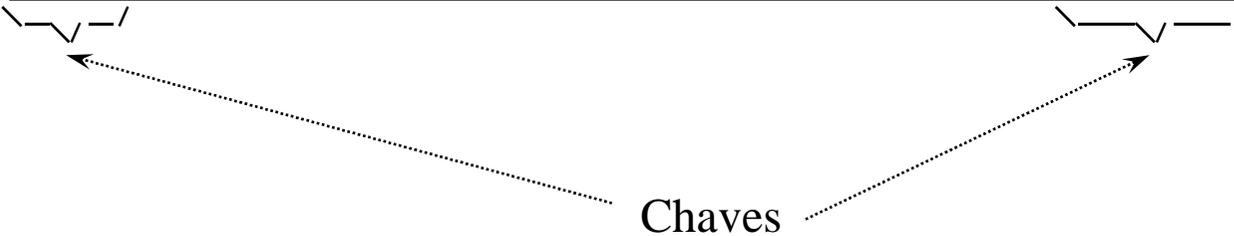
Conjunto de Esquemas de Relação

Chave Candidata

Chave (candidata) de uma Relação \longleftrightarrow Atributo ou conjunto de atributos que identificam cada ocorrência da relação, i.e., não podem existir duas linhas da relação com o mesmo conjunto de valores neste(s) atributo(s)

Ex: Cliente

num_cli	nome_cli	morada	cidade	país	tipo_cli	n_contrib
1	António Abreu	R.Alberto Antunes	Andorra	Andorra	1	123 456 789
2	Bernardo Bento	R.Beta Brás	Bruxelas	Belgica	2	789 123 456
3	Carlos Castro	R. Clara Campos	Camberra	Canadá	1	456 789 123
...
20	Manuel Matos	R.Marco Moita	Maputo	Moçamb	2	111 222 333
...



Chave Primária

Chave Primária de uma Relação \longleftrightarrow De entre as chaves candidatas de uma relação, escolhe-se uma para ser a chave efectiva da mesma (a que for mais útil, no sistema em questão). A essa chave, dá-se o nome de *Chave Primária*

Chave Estrangeira

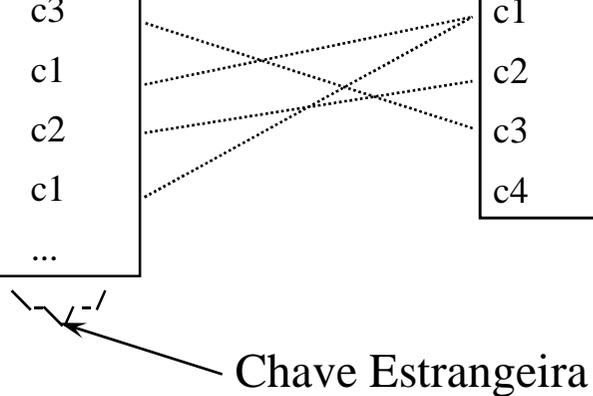
Chave Estrangeira de uma Relação \longleftrightarrow Em algumas relações, temos um atributo (ou conjunto de atributos) cujas ocorrências são referências a uma chave primária de uma outra relação. A esses atributos damos o nome de *Chaves Estrangeiras*

Empregado

n_emp	nome	...	cod_categ
01	João	...	c3
02	José	...	c1
03	Gil	...	c2
04	Júlio	...	c1
...

Categoria

cod_cat	design	venc_base
c1	Director	300
c2	Técnico	200
c3	Admin	120
c4	Auxil	90

 Chave Estrangeira

Surrogate Keys

Depois de ter defendido obstinadamente que as chaves primárias deviam ter significado semântico, Codd, na década de 80 (?), defendeu no seu modelo RM/T a utilização de ***Surrogate Keys***, que não são mais do que “chaves primárias cegas”.

Uma ***surrogate key*** é um identificador único (***primary key***), gerado pelo sistema e sem qualquer significado semântico. Pode ser único na base de dados ou apenas na tabela. Nunca deve ser reutilizado.

Deve obviamente ser definida uma ***candidate key (UNIQUE)*** que representa a “verdadeira chave primária” (com significado semântico).

Vantagens?

Inconvenientes?

Base de Dados Relacional

Conjunto de Relações, cujo conteúdo varia ao longo do tempo

Exemplo de Base de Dados Relacional

Factura

num_factura	data_emissão	data_pagamento	num_cli
1	1-Jan-89	1-Fev-89	20
2	12-Fev-89		1
3	3-Mar-89	20-Mar-89	3
...

Produto

cod_prod	nome_prod	preço	qtd_existe
01	Lápis	100	2543
02	Caneta	150	1321
03	Régua	500	354
...

Cliente

num_cli	nome_cli	morada	cidade	país	tipo_cli
1	António Abreu	R.Alberto Antunes	Andorra	Andorra	1
2	Bernardo Bento	R.Beta Brás	Bruxelas	Belgica	2
3	Carlos Castro	R. Clara Campos	Camberra	Canadá	1
...
20	Manuel Matos	R.Marco Moita	Maputo	Moçamb	2
...

Linha_Factura Linha de Factura

num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10
2	01	25
3	02	13
3	05	15
3	01	17
3	02	26



Exemplo Esquema Relacional

Factura (num_factura, data_emissão, data_pagamento, num_cli)

Cliente (num_cli, nome_cli, morada, cidade, país, tipo_cli)

Produto (cod_prod, nome_prod, preço, qtd_existe)

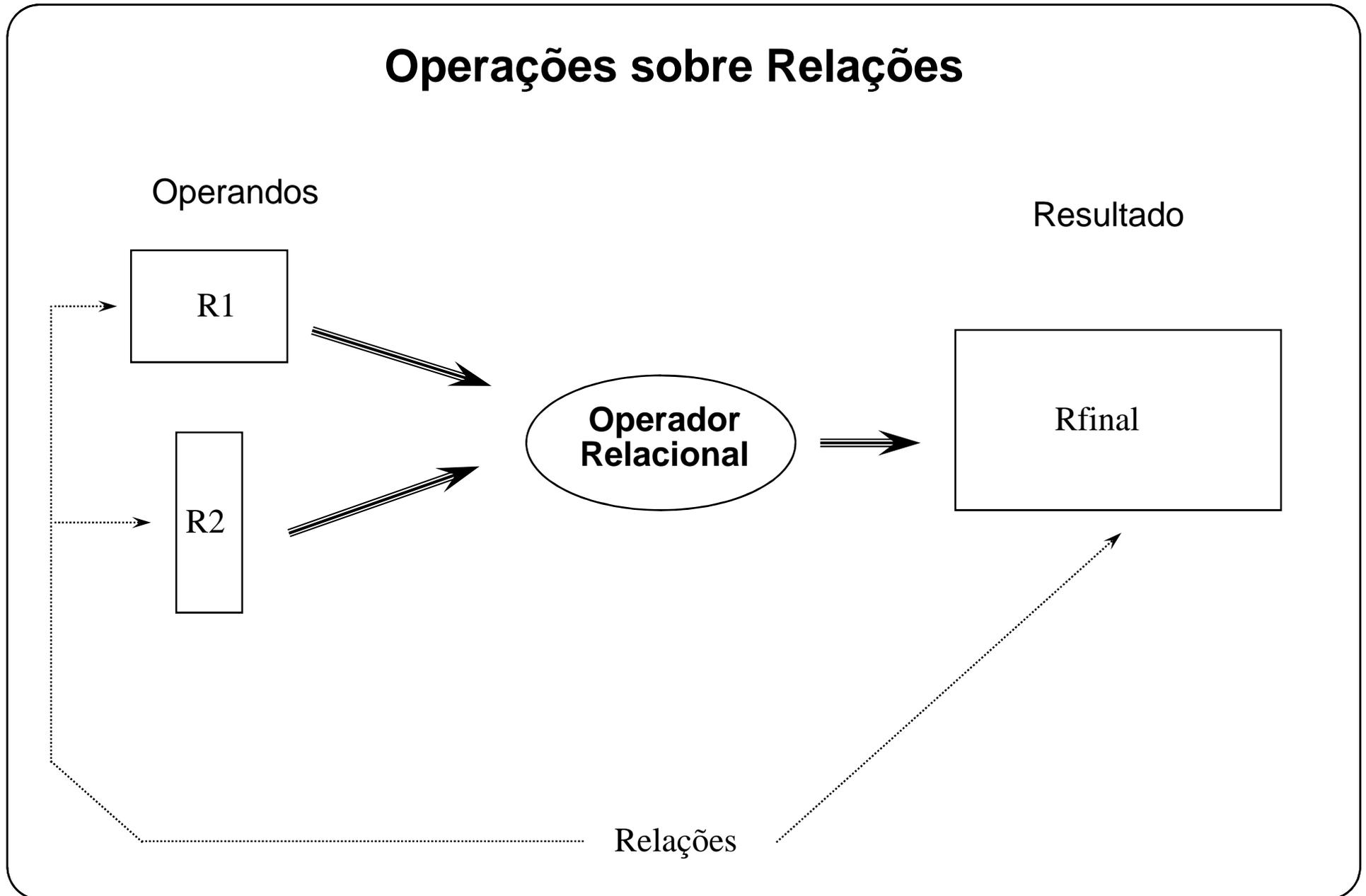
Linha_Factura (num_factura, cod_prod, quant)

Operadores do Modelo Relacional



Operadores
sobre RELAÇÕES

Operações sobre Relações



Projeção

A partir de UMA relação, obter *apenas algumas colunas* (com eliminação de repetições)

Cliente

num_cli	nome_cli	morada	cidade	país	tipo_cli
1	António Abreu	R.Alberto Antunes	Andorra	Andorra	1
2	Bernardo Bento	R.Beta Brás	Bruxelas	Bélgica	2
3	Carlos Castro	R. Clara Campos	Camberra	Canadá	1
...
20	Manuel Matos	R.Marco Moita	Maputo	Moçamb	2
21	Mario Martins	R.Maria Mendes	Manga	Moçamb	3

Cliente[nome_cli, morada]

nome_cli	morada
António Abreu	R.Alberto Antunes
Bernardo Bento	R.Beta Brás
Carlos Castro	R. Clara Campos
...	...
Manuel Matos	R.Marco Moita
Mario Martins	R.Maria Mendes

Cliente[País]

País
Andorra
Bélgica
Canadá
Moçamb

Projecção

Seja $R(X, Y)$ com $X = A_1, A_2, \dots, A_k$
e $Y = A_{k+1}, \dots, A_n$

A projecção de R sobre os atributos X é representada pelo conjunto

$$R[X] = \{ x : \exists y : (x, y) \text{ existe em } R(X, Y) \}$$

Restrição ou Selecção

A partir de UMA relação, obter *apenas algumas linhas* (eliminando as restantes), utilizando para isso uma qualquer condição booleana

Linha_Factura

num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10
2	01	25
3	02	13
3	05	15
3	01	17
3	02	26

Podemos substituir este operador por um outro qualquer, a que chamaremos "operador θ "

Linha_Factura[num_fact = 2]

num_fact	cod_prod	quant
2	06	10
2	01	25

Restrição ou Seleção

Seja $R (A_1, A_2, \dots, A_n)$ e P uma expressão lógica definida sobre $D_1 * D_2 * \dots * D_n$, com D_i domínio de A_i .

A restrição de R a respeito de P é representada pelo conjunto

$$R[P] = \{ x: x \text{ em } R \text{ e } P(x) \text{ é verdadeira} \}$$

O Operador " θ "

" θ " poderá ser um dos seguintes operadores:

1. Igualdade
2. Desigualdade
3. Menor que
4. Menor ou igual que
5. Maior que
6. Maior ou igual que

Restrição ou Seleção (cont)

Poderemos também restringir o conjunto das linhas que queremos obter, utilizando os operadores lógicos ("*e*", "*ou*", "*negação*") sobre duas ou mais condições

Linha_Factura

num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10
2	01	25
3	02	13
3	05	15
3	01	17
3	02	26

Linha_Factura[(num_fact=2) \wedge (quant \geq 20)]

num_fact	cod_prod	quant
2	01	25

Produto Cartesiano (*Cross Join*)

"Combinação de TODAS as linhas de uma tabela com TODAS as linhas da outra"

Cliente

num_cli	nome_cli
1	António Abreu
2	Bernardo Bento
3	Carlos Castro

Produto

cod_prod	nome_prod
01	Lápis
02	Caneta

Cliente-X-Produto

num_cli	nome_cli	cod_prod	nome_prod
1	António Abreu	01	Lápis
2	Bernardo Bento	01	Lápis
3	Carlos Castro	01	Lápis
1	António Abreu	02	Caneta
2	Bernardo Bento	02	Caneta
3	Carlos Castro	02	Caneta

(pouca
utilidade
prática)

Junção Natural - *Equi-Join* ou *Inner Join*

A partir de duas tabelas, obter uma terceira, concatenando as colunas e restringindo apenas às linhas com o mesmo valor em algum atributo

Linha_Factura

num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10
2	01	25
3	02	13
3	05	15
3	01	17
3	02	26

Produto

cod_prod	nome_prod	preço	qtd_existe
01	Lápis	100	2543
02	Caneta	150	1321
03	Régua	500	354

Linha_Factura [cod_prod = cod_prod] Produto

num_fact	quant	cod_prod	cod_prod	nome_prod	preço	qtd_existe
1	20	01	01	Lápis	100	2543
1	15	02	02	Caneta	150	1321
2	25	01	01	Lápis	100	2543
3	13	02	02	Caneta	150	1321
3	17	01	01	Lápis	100	2543
3	26	02	02	Caneta	150	1321

Nota: Junção = Produto Cartesiano + Restrição

Junção Natural ou *Equi-Join*

Seja $R1 (A, B1)$ e $R2 (B2,C)$ duas relações e $B1$ e $B2$ conjuntos de atributos compatíveis.

A Junção Natural das relações $R1$ e $R2$ sobre os conjuntos de atributos $B1$ e $B2$ é representada pelo conjunto

$$R1[B1 = B2] R2 = \{ (a, b1, b2, c) : (a, b1) \text{ em } R1 \text{ e } (b2, c) \text{ em } R2 \text{ e } b1 = b2 \}$$

Seja $X = \{X1, \dots, Xn\}$ com DXi domínio de Xi e $Y = \{Y1, \dots, Yn\}$ com domínio DYj domínio de Yj , conjuntos de atributos. X e Y **denominam-se conjuntos de atributos compatíveis** se $DXi = DYi$

Natural join é o *equi-join* em que se retira o $b2$ (porque $=b1$)

θ - Junção

Podemos estender a noção de Junção, de forma que a condição de restrição das linhas não seja apenas a igualdade, mas uma qualquer operação " θ "

Linha_Factura

num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10

Produto

cod_prod	nome_prod	preço	qtd_existe
01	Lápis	100	2543
02	Caneta	150	1321
03	Régua	500	354

$\theta \rightarrow "<"$

Linha_Factura [cod_prod < cod_prod] Produto

num_fact	quant	cod_prod	cod_prod	nome_prod	preço	qtd_existe
1	20	01	02	Caneta	150	1321
1	20	01	03	Régua	500	354
1	15	02	03	Régua	500	354

θ - Junção

Seja $R1 (A, B1)$ e $R2 (B2,C)$ duas relações, $B1$ e $B2$ conjuntos de atributos compatíveis e Θ um operador binário $\{ =, <, <=, <>, >, >= \}$ aplicável aos domínios de $B1$ e $B2$.

A operação Θ -junção das relações $R1$ e $R2$ sobre os conjuntos de atributos $B1$ e $B2$ é representada pelo conjunto

$$R1[B1 \Theta B2] R2 = \{ (a, b1, b2, c) : (a, b1) \text{ em } R1 \text{ e } (b2, c) \text{ em } R2 \text{ e } b1 \Theta b2 \}$$

Divisão Relacional

Forneceu

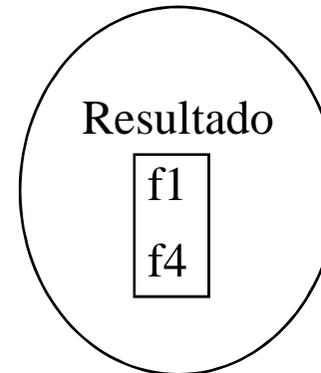
cod_fornec	cod_prod
f1	p1
f1	p2
f1	p3
f1	p4
f1	p5
f1	p6
f2	p1
f2	p2
f3	p2
f4	p2
f4	p4
f4	p5

÷

Prods-especiais

cod_prod
p2
p4

=



**Fornecedores que forneceram
TODOS os produtos da tabela
"Prods-especiais"**

Notação →

Resultado ← Forneceu[cod_fornec, cod_prod / cod_prod]Prods-especiais

Divisão Relacional

Seja $R1 (A, B)$ e $R2 (C)$ duas relações e B e C conjuntos de atributos compatíveis

A Divisão de $R1$ por $R2$ sobre B e C é representada pelo conjunto

$R1[B : C] R2 = R3 (A) : r3 (a) \in R3 (A) \text{ se } \forall c \in R2 (C), R1 (a,c) \in R1 \}$

União

União de duas relações é a relação composta por todas as linhas de ambas
(com eliminação de repetições)

Clientes

Nome	Morada
Alberto Alves	Alameda
Carlos Odorico	Campo d´Ourique
Alves Bernardo	Av. Berna
Fernando Fontes	Figueira da Foz

Fornecedores

Nome	Morada
Manuel Mendes	Madeira
Carlos Odorico	Campo d´Ourique
Júlio Jaime	Jamaica
Alves Bernardo	Av. Berna
Rui Rodrigues	Ribeirinha

Clientes - **UNION** - Fornecedores

Nome	Morada
Alberto Alves	Alameda
Carlos Odorico	Campo d´Ourique
Alves Bernardo	Av. Berna
Fernando Fontes	Figueira da Foz
Manuel Mendes	Madeira
Júlio Jaime	Jamaica
Rui Rodrigues	Ribeirinha



Os clientes e os Fornecedores

Nota: as relações terão que ser compatíveis

Intersecção

Intersecção de duas relações é a relação composta pelas linhas que pertencem a ambas as relações

Clientes

Nome	Morada
Alberto Alves	Alameda
Carlos Odorico	Campo d´Ourique
Alves Bernardo	Av. Berna
Fernando Fontes	Figueira da Foz

Fornecedores

Nome	Morada
Manuel Mendes	Madeira
Carlos Odorico	Campo d´Ourique
Júlio Jaime	Jamaica
Alves Bernardo	Av. Berna
Rui Rodrigues	Ribeirinha

Os clientes que também são fornecedores



Clientes - **Intersecção** - Fornecedores

Nome	Morada
Carlos Odorico	Campo d´Ourique
Alves Bernardo	Av. Berna

Nota: as relações terão que ser compatíveis

Diferença

Diferença de duas relações é a relação composta pelas linhas que pertencem à primeira e NÃO pertencem à segunda

Clientes

Nome	Morada
Alberto Alves	Alameda
Carlos Odorico	Campo d´Ourique
Alves Bernardo	Av. Berna
Fernando Fontes	Figueira da Foz

Fornecedores

Nome	Morada
Manuel Mendes	Madeira
Carlos Odorico	Campo d´Ourique
Júlio Jaime	Jamaica
Alves Bernardo	Av. Berna
Rui Rodrigues	Ribeirinha

Os clientes que não são fornecedores



Clientes — Fornecedores

Nome	Morada
Alberto Alves	Alameda
Fernando Fontes	Figueira da Foz

Nota: as relações terão que ser compatíveis

O Conceito de valor *NULL*

Numa relação podemos, em determinado momento, ter um atributo cujo valor é desconhecido ou não aplicável no momento. Diz-se então que esse elemento tem o valor *NULL*

Características dos valores *NULL*'s:

-  Independente do domínio - inteiro, real, caracter, data, etc.
-  Não comparáveis entre si, i.e., nunca poderemos dizer que um valor *NULL* é igual a outro valor *NULL*

Outer Equi-Join

Considere as seguintes relações:

S (A B)

a1 b1

a2 b2

a3 b3

T (C D)

b2 d1

b2 d2

b3 d3

b4 d4

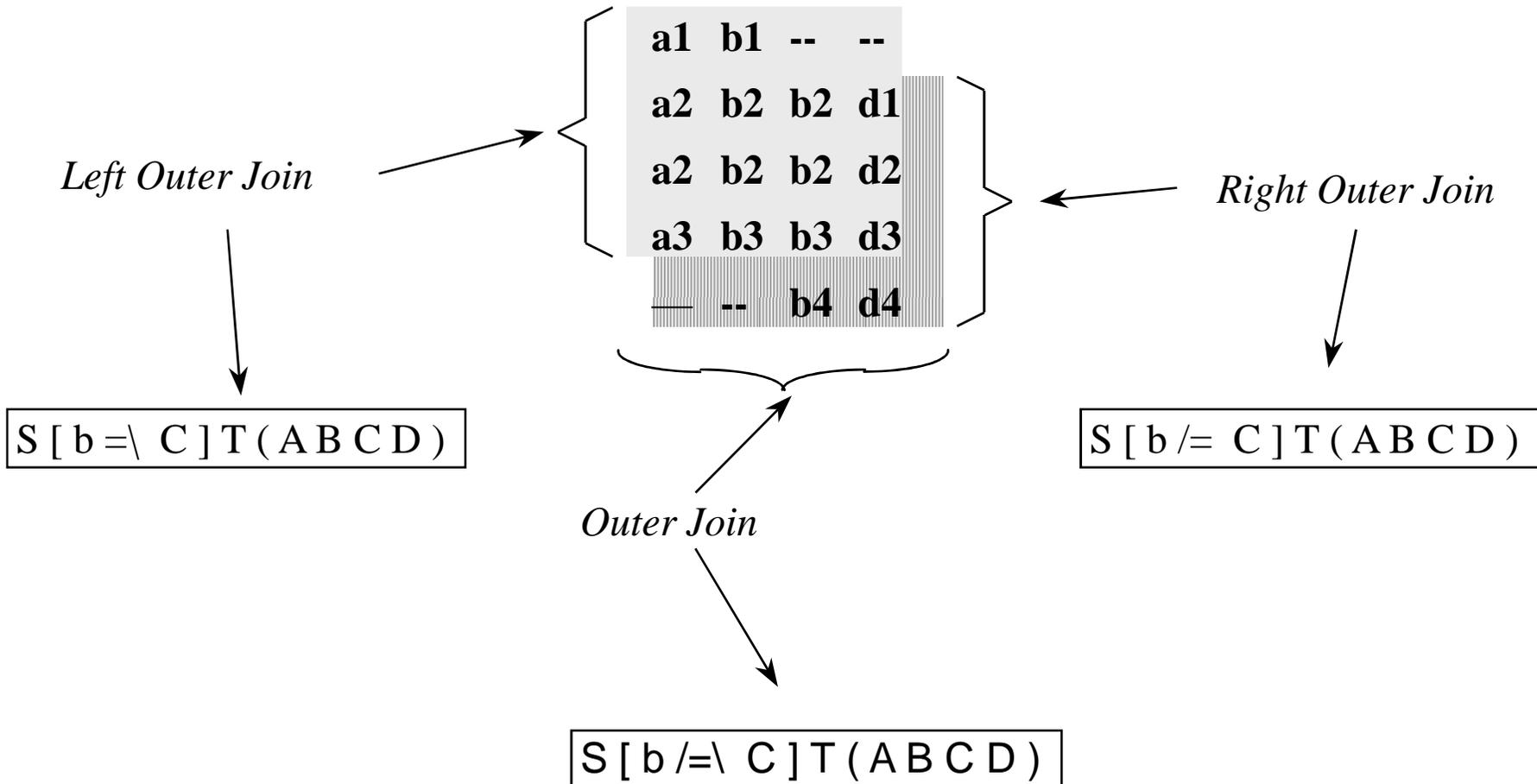
Junção Natural
ou
Inner Equi-Join

a1	b1	--	--
a2	b2	b2	d1
a2	b2	b2	d2
a3	b3	b3	d3
--	--	b4	d4

Outer Equi-Join

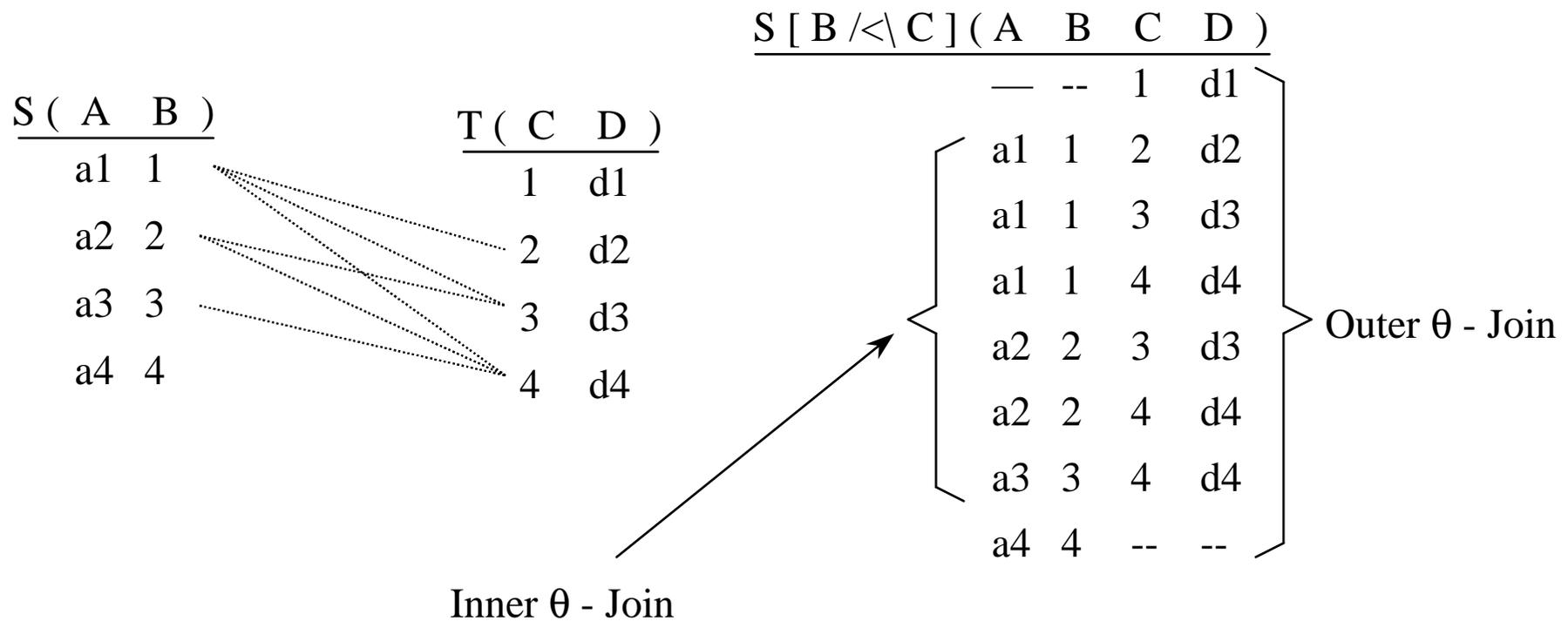
Valores Inexistentes
(NULL)

Left e Right Outer Join



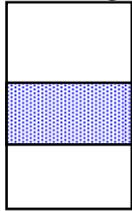
Outer θ - Join

Podemos estender o conceito de Outer-Join para os casos em que a comparação é outra além da simples igualdade

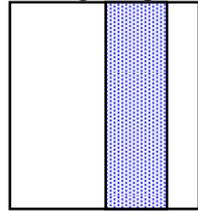


Resumo das Operações Relacionais abordadas

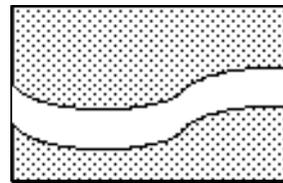
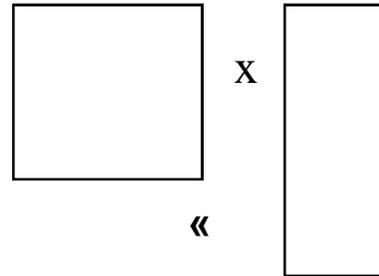
Seleccção



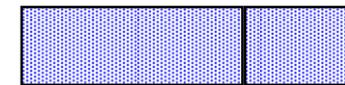
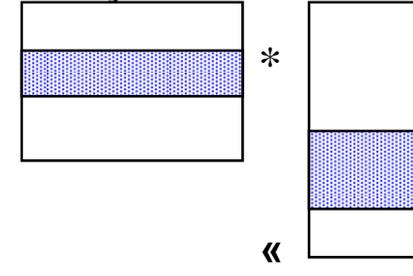
Projecção



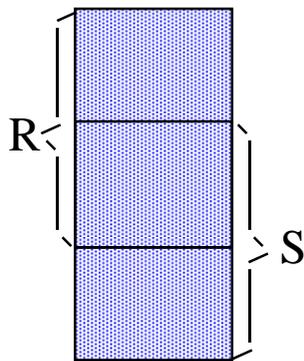
Produto



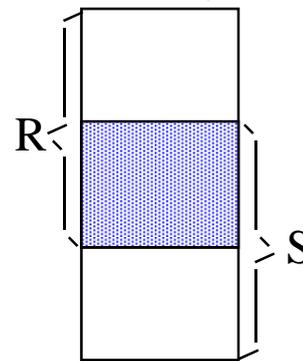
Junção *inner* e *outer*



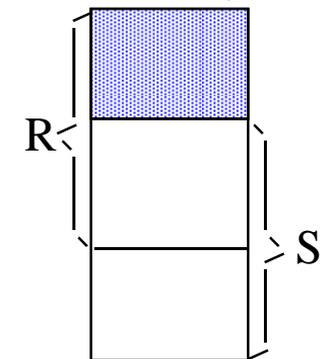
União



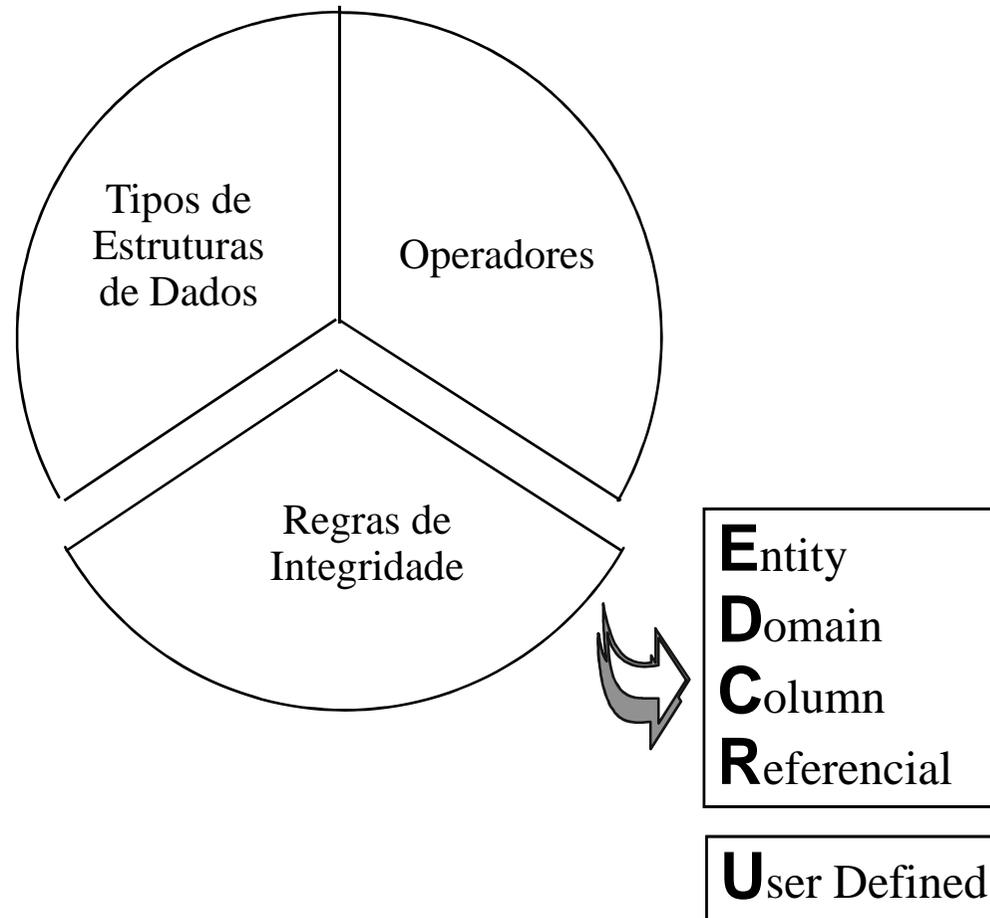
Intersecção



Diferença



Regras de Integridade



Integridade de Entidades



Nenhuma componente de uma chave primária de uma relação poderá em algum momento ter valor *NULL*

Integridade de Domínio



Cada coluna de cada relação tem um domínio, isto é, um conjunto (limitado ou não) de valores possíveis. Em todas as linhas dessa relação, o valor dessa coluna terá que pertencer SEMPRE a esse domínio.

Ex: **Create Domain** D_Contribuente **as** number(9) **Check** (**VALUE** > 0);

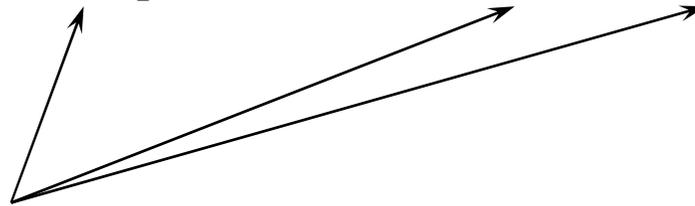
Integridade de Colunas



A integridade de colunas permite definir, para cada coluna, o conjunto dos valores possíveis.

Ex:

empregado(n-emp, nome, morada, salário, diuturnidades , ...)



Domínio : Inteiros

Regras de Integridade de colunas :

$\left\{ \begin{array}{l} \text{n-emp} \rightarrow \text{Positivo} \\ \text{salário} \rightarrow \text{Positivo} \\ \text{diuturnidades} \rightarrow \geq 0 \text{ e } \leq 5 \end{array} \right.$

....diuturnidades **INTEGER CHECK** (diuturnidades BETWEEN 0 and 5)

Integridade Referencial



Numa relação, qualquer ocorrência de uma chave estrangeira deverá obrigatoriamente existir como ocorrência de uma chave primária da relação à qual se refere (ou ser nula no caso de não obrigatoriedade)

Integridade Definida pelo Utilizador



Qualquer outra regra a que as ocorrências de uma determinada base de dados deverão obedecer. Estão intimamente ligadas a regras do negócio.

Estas restrições definidas pelo utilizador, sendo independentes do modelo de dados usado, são conhecidas como “regras de integridade explícitas” (ao contrário das implícitas, próprias de cada modelo de dados).

São classificadas em dois tipos:

- restrições estáticas
- restrições dinâmicas

Integridade Definida pelo Utilizador



Restrições estáticas

Correspondem a restrições que asseguram a integridade dos vários estados pelos quais passa a BD.

Exemplo:

O preço de venda de um determinado artigo não pode ser inferior ao valor de compra desse artigo acrescido de 3%.

Integridade Definida pelo Utilizador



Restrições dinâmicas

Correspondem a restrições que asseguram que a transição entre dois estados é válida e preserva a integridade da BD face às regras de negócio.

Exemplo:

*O estado civil de um indivíduo, não pode assumir o valor de **solteiro**, após ter sido **divorciado**, **viúvo** ou **casado**.*

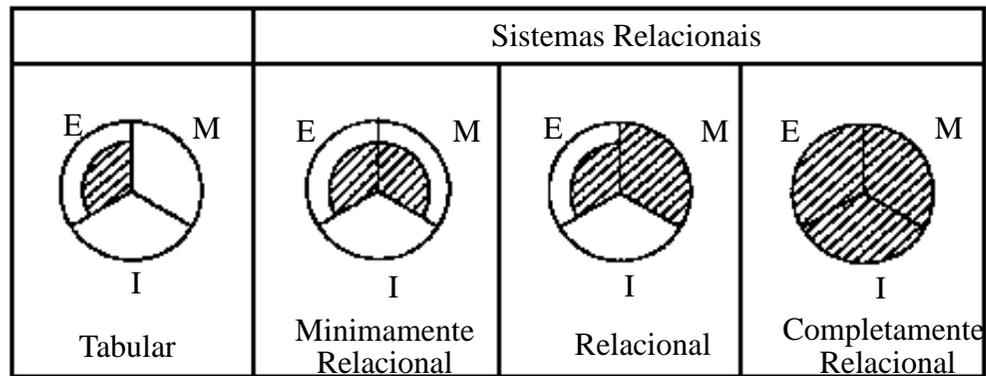
O Que é realmente um SGBD relacional?

Segundo Codd (*) um SGBD pode ser considerado minimamente relacional se satisfaz as seguintes condições:

- **Estrutura**: Ao utilizador são apresentadas apenas tabelas e nada mais do que tabelas
- **Manipulação**: Operações de restrição, projecção e junção natural, sem necessidade de definição de caminhos físicos de acesso que suportem essas operações

(*) E.F. Codd. "Relational Database: A Practical Foundation for Productivity." *CACM* 25, Nº 2 (Fevereiro 1982)

O que é realmente um SGBD relacional?



(*) C.J.Date. *An Introduction to Database Systems* . Reading, Mass: Addison Wesley, 5^a ed. (1990)

O que é realmente um SGBD relacional?

A partir da definição de Codd, Chris Date(*) sugere uma classificação para SGBD relacionais:

- **Tabular**: Suporta a estrutura tabular, mas não os operadores relacionais
- **Minimamente relacional**: Suporta a estrutura tabular e os operadores de restrição, projecção e junção (apenas estes)
- **Relacional**: Suporta a estrutura tabular e todos os operadores da algebra relacional
- **Completamente Relacional**: Suporta todos os aspectos do modelo relacional (estrutura tabular e domínios, todos os operadores da algebra relacional e a integridade de entidades e referencial)

(*) C.J.Date. *An Introduction to Database Systems* . Reading, Mass: Addison Wesley, 5ª ed. (1990)

Em que medida é um SGBD Relacional

Segundo um artigo posterior de Codd(*) um SGBD é completamente relacional (*fully relational*) se gere a base de dados utilizando apenas e só as possibilidades oferecidas pelo modelo relacional.

Assim sendo tem de satisfazer as seguintes 12 regras

1. Informação
2. Garantia de acesso
3. Tratamento sistemático de valores nulos
4. Catálogo on-line e activo baseado no modelo relacional
5. Sublinguagem de manipulação de dados compreensível
6. Actualização de views

(continua)

(*) E.F. Codd. "Is Your DBMS Really Relational?." (*Computerworld*, Outubro 14, 1985); "Does Your DBMS Run by the Rules." (*Computerworld*, Outubro 21, 1985)

Em que medida é um SGBD Relacional

(continuação)

7. Insert, Update e Delete de alto nível
8. Independência física dos dados
9. Independência lógica dos dados
10. Independência da Integridade
11. Independência da distribuição
12. Não subversão

(*) E.F. Codd. "Is Your DBMS Really Relational?." (*Computerworld*, Outubro 14, 1985); "Does Your DBMS Run by the Rules." (*Computerworld*, Outubro 21, 1985)