



Disciplina de Gestão de Dados e de Bases de Dados

Ano lectivo 2013/2014

Modelo Relacional

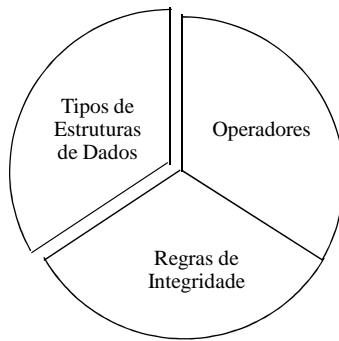
- Estrutura
- Operações
- Integridade

Modelo de Relacional de Dados



Estruturas de Dados do Modelo Relacional

RELACÕES
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
...

Domínio

Números inteiros positivos

Cadeias de caracteres de comprimento inferior ou igual a 20

Domínio

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, carácter, data, etc.
- ☞ Não comparáveis entre si, i.e., nunca poderemos dizer que um valor *NULL* é igual a outro valor *NULL*

Relação (ou Estado de uma Relação)(1/2)

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

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

Relação (2/2)

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

Cardinalidade \longleftrightarrow Número de n-uplos da relação

Grau da Relação \longleftrightarrow Número de colunas

Atributo \longleftrightarrow Coluna

Domínios de R \longleftrightarrow D1, D2,, Dn

Conceitos

Esquema de Relação ↔ Definição de uma relação

Exemplo:

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

Esquema Relacional ↔ Definição de uma Base de Dados Relacional

=

Conjunto de Esquemas de Relação e um conjunto de Regras de Integridade

Chave Candidata

Chave (candidata) de uma Relação ↔ Atributo (**chave simples**) ou conjunto de atributos (**chave composta**) 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
...

Chaves

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*.

Nenhum dos atributos que compõem a chave primária pode, em alguma circunstância, ter o valor null

Chave Estrangeira

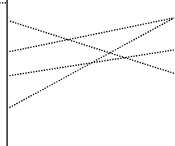
Chave Estrangeira de uma Relação \longleftrightarrow Em algumas relações, temos um atributo (ou conjunto de atributos) cujas ocorrências são referências a uma chave primária de uma outra relação. A esses atributos damos o nome de *Chaves Estrangeiras*

Empregado

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

Categoria

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



Chave Estrangeira

(Estado de uma) Base de Dados Relacional

Conjunto de Relações, que satisfazem as Regras de Integridade especificadas para o respectivo Esquema Relacional

Exemplo de Base de Dados Relacional

Factura				Produto			
num_factura	data_emissão	data_pagamento	num_cli	cod_prod	nome_prod	preço	qtd_existe
1	1-Jan-89	1-Fev-89	20	01	Lápis	100	2543
2	12-Fev-89		1	02	Caneta	150	1321
3	3-Mar-89	20-Mar-89	3	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

fatura (num_fatura, 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_fatura (num_fatura, cod_prod, quant)

Natural Keys vs Surrogate Keys

Depois de ter defendido que as chaves primárias deviam ter significado para o negócio, Codd, na década de 80, defendeu no seu modelo RM/T a utilização de **Surrogate Keys**, que não são mais do que "chaves primárias artificiais".

Uma **surrogate key** é um identificador único (**primary key**), gerado pelo sistema e sem qualquer significado para o negócio. Pode ser único na base de dados ou apenas na relação e nunca deve ser reutilizado.

No caso da **primary key** ser uma **surrogate key** deve ser definida uma **candidate key (UNIQUE)** que representa a "verdadeira chave primária" (com significado semântico).

Natural Keys

Vantagens:

- Têm significado para o negócio
- As *queries* requerem normalmente um menor número de junções

Desvantagens:

- Por estarem relacionadas com o negócio, têm de ser alteradas quando os requisitos mudam
- Podem exigir a concatenação de vários atributos

Surrogate Keys

Vantagens:

- Como não têm significado para o negócio, não têm de ser alteradas quando os requisitos mudam
- Por serem chaves "curtas", com um só atributo, as *queries* tendem a ser mais rápidas

Desvantagens:

- Não têm significado para o negócio
- Dão origem a mais uma coluna na tabela
- As *queries* tendem a exigir um maior número de junções

Caso Prático com *Natural Keys*

Pergunta: Pergunta: qual o valor da(s) prestação(ões) devida(s) em Outubro de 2013 pelo cliente com NIF= 145342123

Cliente (nifCliente, nomeCliente, localidadeCliente, telefoneCliente)

Contrato (nifCliente, nContrato, codPeriodicidade, dataContrato, dataFimContrato, valorCredito)

Prestacao (nifCliente, nContrato, nPrestacao, dataVencimento, valorPrestacao, valorJuro, valorAmortizacao)

Select valorPrestacao

From Prestacao

Where nifCliente=145342123 and dataVencimento between to_date ('2013/10/01', 'yyyy/mm/dd') AND to_date ('2013/10/31', 'yyyy/mm/dd')

Caso Prático com *Surrogate Keys*

Pergunta: qual o valor da(s) prestação(ões) devida(s) em Outubro de 2013 pelo cliente com NIF= 145342123

Cliente (idCliente, nifCliente, nomeCliente, localidadeCliente, telefoneCliente)

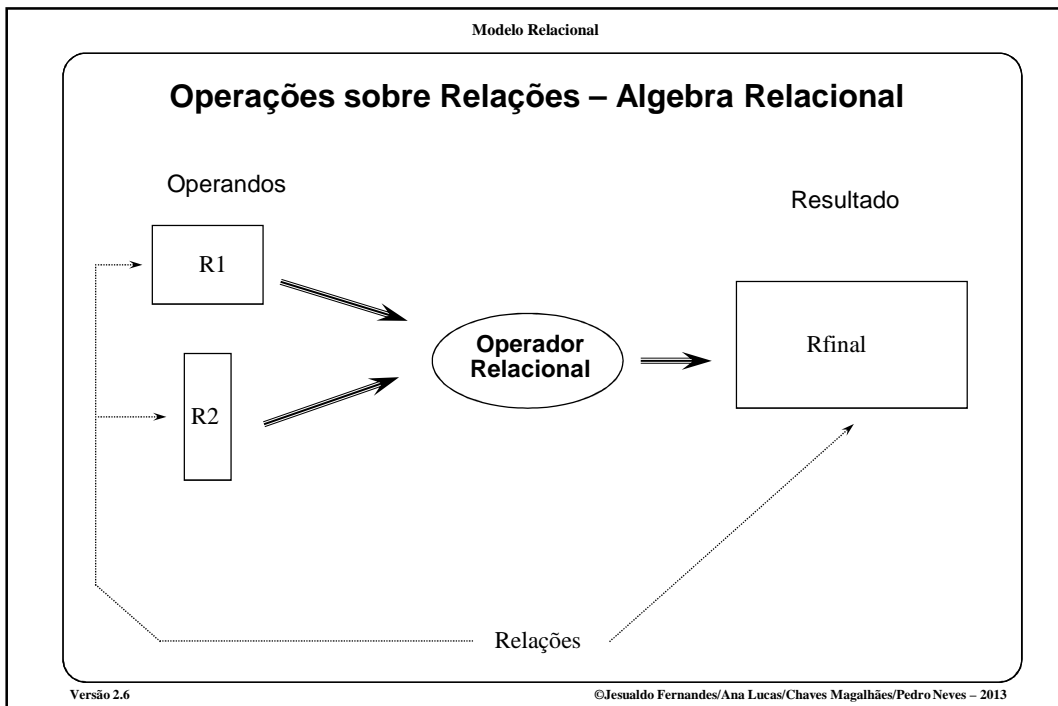
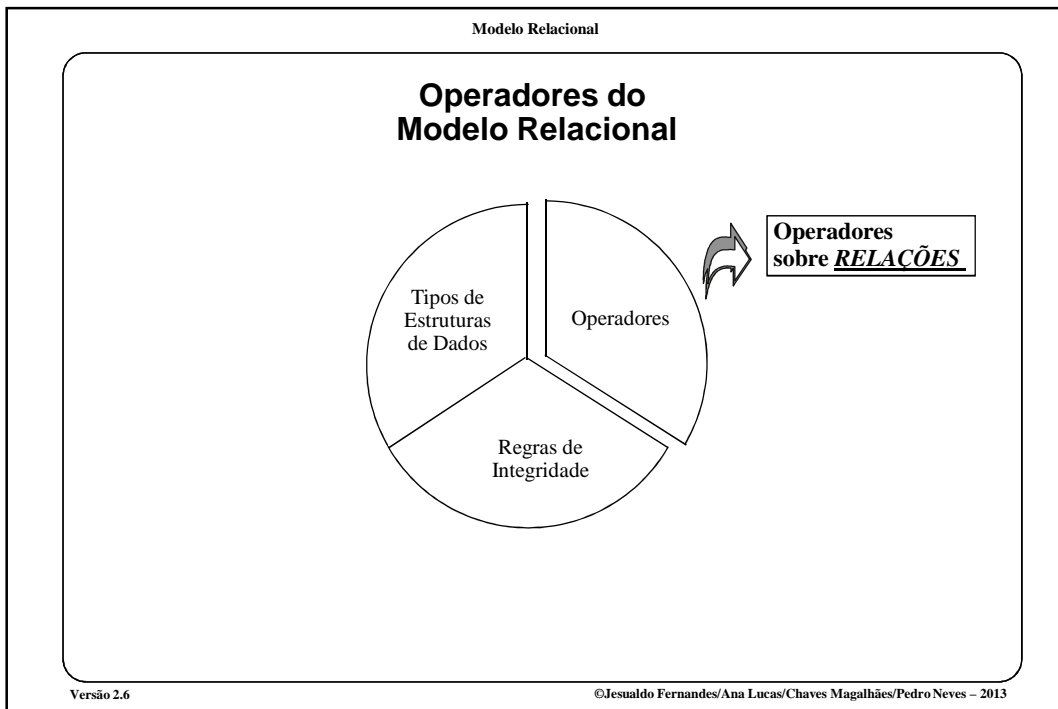
Contrato (idContrato, idCliente, nContrato, codPeriodicidade, dataContrato, dataFimContrato, valorCredito)

Prestacao (idPrestacao, idContrato, nPrestacao, dataVencimento, valorPrestacao, valorJuro, valorAmortizacao)

Select valorPrestacao

From Prestacao P, Contrato Ct, Cliente C

Where P.idContrato=Ct.idContrato and Ct.idCliente= C.idCliente and nifCliente=145342123 and dataVencimento between to_date ('2013/10/01', 'yyyy/mm/dd') AND to_date ('2013/10/31', 'yyyy/mm/dd')



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

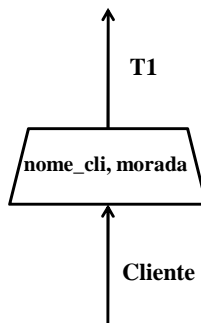
Cliente[nome_cli, morada]

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

Projeção – Representação Gráfica



Projeção

Seja R (X,Y) com X = A1, A2, , Ak
e Y = Ak+1, , An

A projeção 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 Seleção

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

Linha_Factura

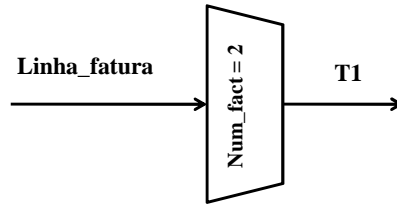
num_fact	cod_prod	quant
1	01	20
1	02	15
1	07	50
2	06	10
2	01	25
3	02	13
3	05	15
3	01	17
3	02	26

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 – Representação Gráfica



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)^(quant ≥ 20)]

num_fact	cod_prod	quant
2	01	25

Produto Cartesiano (Cross Product)

"Combinação de TODAS as linhas de uma relação 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)

Equi-Join

A partir de duas relações, 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

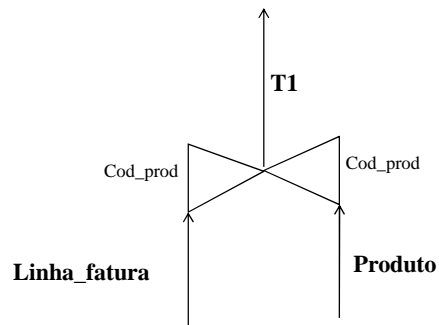
cod_prod	nome_prod	preço	qtd_existe
01	Lápis	100	2543
02	Caneta	150	1321
03	Régua	500	354

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

Equi Join – Representação Gráfica



Equi-Join

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

A *Equi Join* 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_fatura

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

Produto

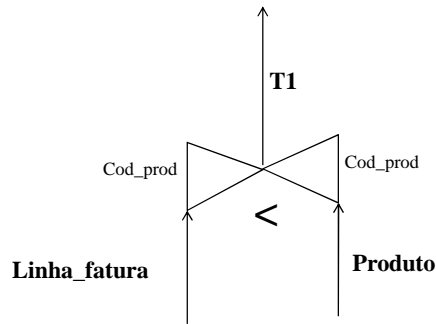
cod_prod	nome_prod	preço	qtd_existe
01	Lápis	100	2543
02	Caneta	150	1321
03	Régua	500	354

θ -> "<"

Linha_fatura [cod_prod < cod_prod] Produto

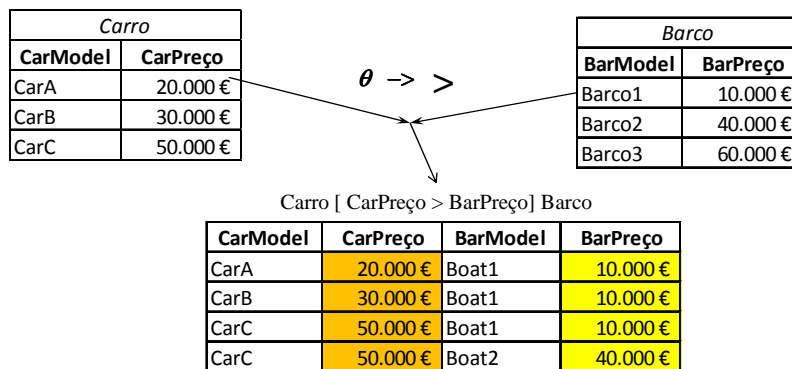
num_fact	quant	cod_prod	cod_prod	nome_prod	preço	qtd_existe
1	20	01	02	Caneta	150	1321
1	20	01	03	Régua	500	354
1	15	02	03	Régua	500	354

θ - Junção - Representação Gráfica



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 "



Exemplo traduzido da wikipédia , em 2010-10-05
http://en.wikipedia.org/wiki/Relational_algebra#.CE.B8-join_and_equijoin

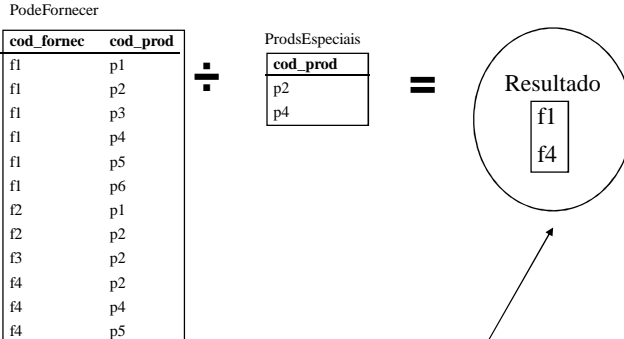
θ - Junção

Seja $R_1(A, B_1)$ e $R_2(B_2, C)$ duas relações, B_1 e B_2 conjuntos de atributos compatíveis e θ um operador binário $\{ =, <, <=, <>, >, >= \}$ aplicável aos domínios de B_1 e B_2 .

A operação θ -junção das relações R_1 e R_2 sobre os conjuntos de atributos B_1 e B_2 é representada pelo conjunto

$$R_1[B_1 \theta B_2] R_2 = \{ (a, b_1, b_2, c) : (a, b_1) \text{ em } R_1 \text{ e } (b_2, c) \text{ em } R_2 \text{ e } b_1 \theta b_2 \}$$

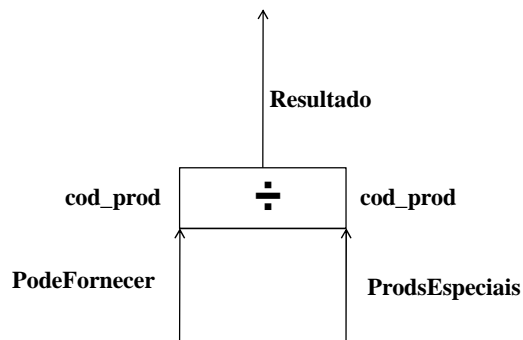
Divisão Relacional



Fornecedores que podem fornecer **TODOS** os produtos da relação "ProdsEspeciais"

Notação → Resultado ← PodeFornecer[cod_fornec, cod_prod / cod_prod]ProdsEspeciais

Divisão – Representação Gráfica



Divisão Relacional

Seja $R_1 (A, B)$ e $R_2 (C)$ duas relações e B e C conjuntos de atributos compatíveis

A Divisão de R_1 por R_2 sobre B e C é representada pelo conjunto

$R_1[B : C] R_2 = R_3 (A) : r_3 (a) \in R_3 (A) \text{ se } \forall c \in R_2 (C), R_1 (a,c) \in R_1 \}$

União

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

Cientes

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

Cientes - UNION - Fornecedores

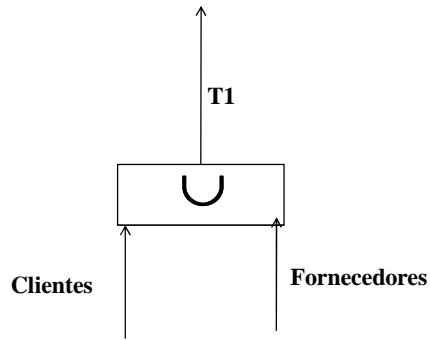
Nome	Morada
Alberto Alves	Alameda
Carlos Odorico	Campo d'Ourique
Alves Bernardo	Av. Berna
Fernando Fontes	Figueira da Foz
Manuel Mendes	Madeira
Júlio Jaime	Jamaica
Rui Rodrigues	Ribeirinha



Os clientes e os Fornecedores

Nota: as relações terão que ser compatíveis

União – Representação Gráfica



Interseção

Interseção 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

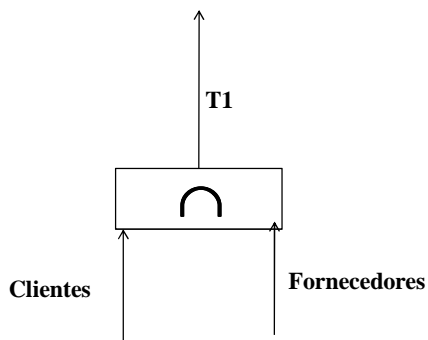
Os clientes que também são fornecedores

Clientes - **Interseção** - Fornecedores

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

Nota: as relações terão que ser compatíveis

Interseção – Representação Gráfica



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

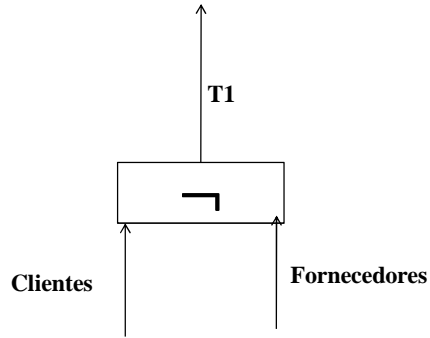
Os clientes que não são fornecedores

Clientes — Fornecedores

Nome	Morada
Alberto Alves	Alameda
Fernando Fontes	Figueira da Foz

Nota: as relações terão que ser compatíveis

Diferença – Representação Gráfica



Outer Equi-Join

Considere as seguintes relações

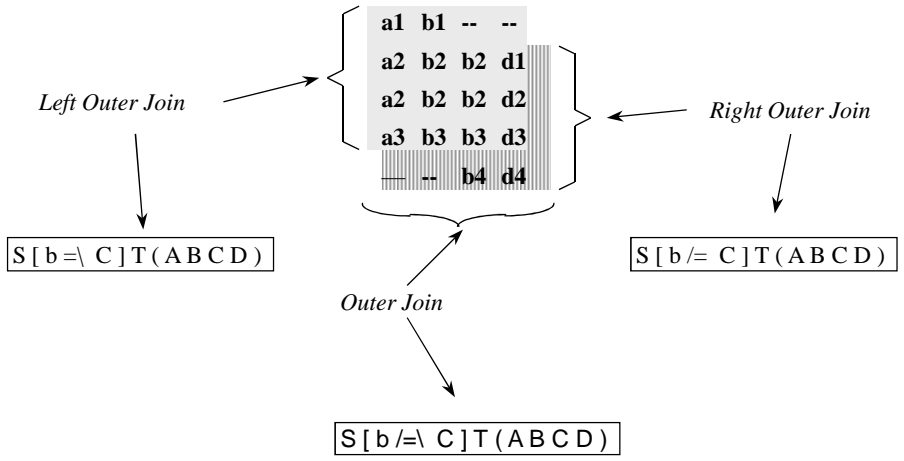
Empregado			Departamento	
EmpNome	EmpId	DeptNome	DeptNome	Director
João	3415	Financeiro	Vendas	Rui
Maria	2241	Vendas	Produção	Carlos
Jorge	3401	Financeiro		
Rui	2202	Vendas		
André	1123	SI		

Junção Natural
ou
Equi-Join

Empregado outer join Dept			
EmpNome	EmpId	DeptNome	Director
João	3415	Financeiro	Null
Jorge	3401	Financeiro	Null
André	1123	SI	Null
Maria	2241	Vendas	Rui
Rui	2202	Vendas	Rui
Null	Null	Produção	Carlos

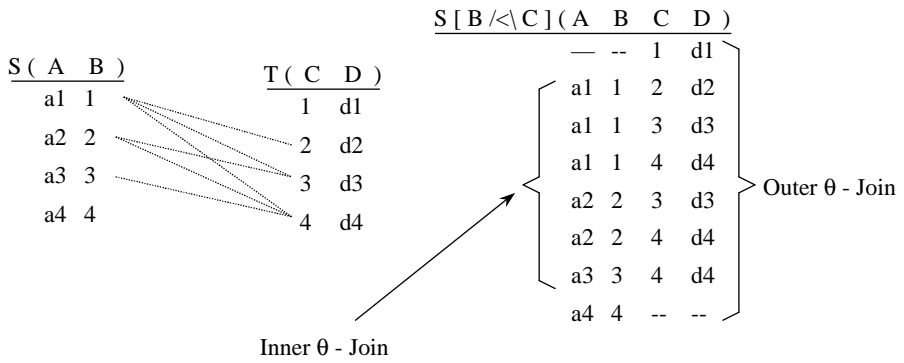
Outer Equi-Join

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



Views

Considere a relação **fatura**, obtida por *inner join* do **Cabeçalho** e das **Linhas**

CabFactura		
numFact	codCli	Data
100	1050	21-09-2010
105	2100	26-09-2010
108	1080	01-10-2010

LinhaFactura			
numFact	codProd	PreçoUnit	quant
100	10100	2,5	10
100	10150	5	50
105	10200	7,5	5
108	10211	3	45
108	10100	2,5	35
108	10200	7,5	20

Factura -> CabFactura [numFact=numFact] LinhaFactura							
100	1050	21-09-2010	100	10100	2,5	10	
100	1050	21-09-2010	100	10150	5	50	
105	2100	26-09-2010	105	10200	7,5	5	
108	1080	01-10-2010	108	10211	3	45	
108	1080	01-10-2010	108	10100	2,5	35	
108	1080	01-10-2010	108	10200	7,5	20	

A relação **fatura** é uma **Relação Derivada**, ou seja, é uma relação cujo conteúdo é derivado do conteúdo de outras Relações

Views

Numa base de dados (BD) relacional podem existir **dois Tipos de Relações**:

- 1. Relações de Base**, cujo conteúdo é autónomo e está armazenado na BD
- 2. Relações Derivadas (Views)**, cujo conteúdo é derivado do conteúdo de outras relações (de Base e/ou Derivadas)

Existem fundamentalmente **dois Tipos de Relações Derivadas (Views)**:

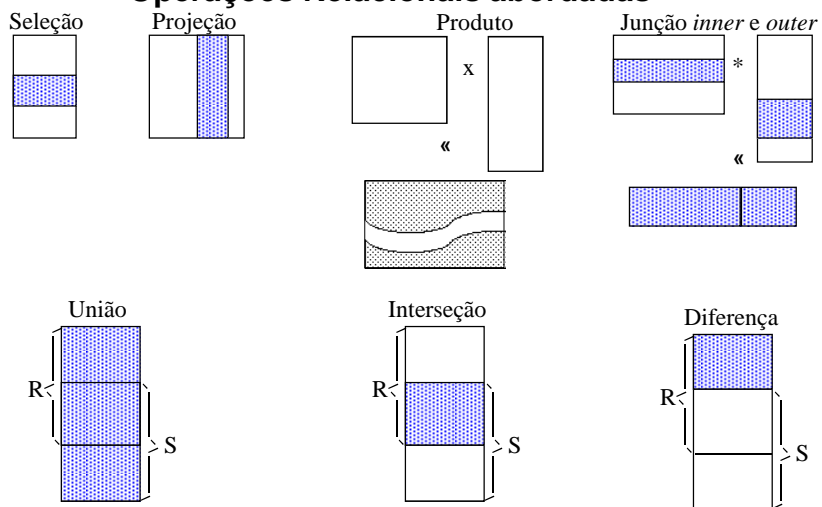
- 1. Materialized Views**, cujo conteúdo é armazenado na BD
- 2. Views**, definidas através de instruções SQL, que não estão armazenadas na BD, embora sejam utilizadas como se estivessem

Views - Utilização

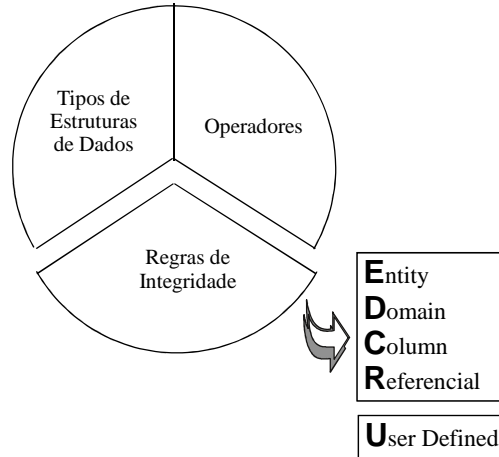
As *Views* são utilizadas por vários motivos, designadamente:

1. **Simplificação** – apresentar ao utilizador apenas os dados que lhe interessam, em formato apropriado
2. **Em substituição de *queries* muito complexas**, permitindo a sua reutilização
3. **Por razões de segurança**, autorizando apenas *views* devidamente parametrizadas, versus relações de base
4. **Por razões de flexibilidade**, nomeadamente em casos de reestruturação da BD, em que se podem manter as *views* com o mesmo esquema, embora construídas sobre diferentes relações de base

Resumo das Operações Relacionais abordadas



Regras de Integridade



Integridade de Entidades

E

Toda a relação tem uma *chave primária*, em que nenhum dos atributos constituintes pode, em algum momento, ter o valor *NULL*

Integridade de Domínio

D

Cada coluna de uma 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 ou ser *null* (no caso em que valores *null* são autorizados)

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

Integridade de Colunas

C

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

Ex:

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

Domínio : Inteiros

Regras de Integridade de colunas :

{ n-emp → Positivo
salário → Positivo

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 *null* 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**.*