

Modelos de Dados

1 - Introdução

2 - Modelo Relacional

2.1. Estrutura de Dados Relacional

2.1.1. Modelo Conceptual de Dados

*Um modelo conceptual de dados é a representação de um conjunto de
objectos e das suas associações*

Como qualquer representação é o resultado de um processo de abstracção.

. Durante esse processo de abstracção, objectos relevantes, associações entre eles e características (atributos) de objectos e associações são seleccionadas.

. A relevância de um objecto, de uma associação ou de um atributo é determinada pelos objectivos do modelo.

. Atributos de objectos e correspondentes associações têm valores específicos que pertencem a conjuntos denominados domínios.

. Um valor de um dado atributo pode variar ao longo do tempo mas pertencendo sempre ao domínio desse atributo.

O modelo relacional baseia-se no pressuposto de que os dados (que obedecem a certas restrições) podem ser tratados da mesma forma que as relações matemáticas.

2.1.2. Entidades, Atributos e Domínios

Objectos e respectivas associações são chamados ENTIDADES

Um conjunto E de entidades do mesmo tipo é caracterizado por um conjunto de ATRIBUTOS, A_1, A_2, \dots, A_n e denotado por

$$E (A_1, A_2, \dots, A_n)$$

onde

$A_i : E \rightarrow D_i$ é uma função cujo contradomínio D_i é denominado DOMÍNIO do atributo A_i

Dado \underline{e} em E , $A_i(\underline{e})$ em D_i é denominado o valor do atributo A_i da entidade \underline{e} .

2.1.3. Exemplo de um tipo de entidade

Seja o tipo de entidade Pessoa cujos atributos relevantes são:

Número de segurança social
Primeiro nome
Último nome
Idade

Pessoa (NSS, P_nome, U_nome, Idade)

Onde os domínios dos atributos são:

NSS – o conjunto, S, dos números de segurança social

P_nome – o conjunto, A, de sequências finitas de letras

U_nome – o conjunto, A, de sequências finitas de letras

Idade – o conjunto, N, dos inteiros positivos <150 !

2.1.4. Representação de entidades por tuplos

Dado um tipo de entidade

$$E (A_1, A_2, \dots, A_n)$$

o conjunto de funções

$$A_1 : E \rightarrow D_1, \quad A_2 : E \rightarrow D_2, \quad \dots, \quad A_n : E \rightarrow D_n$$

determina uma única função:

$$(A_1, A_2, \dots, A_n) : E \rightarrow D_1 \times D_2 \times \dots \times D_n \quad (i)$$

onde $D_1 \times D_2 \times \dots \times D_n$, denota o produto cartesiano dos conjuntos D_1, D_2, \dots, D_n

(isto é, o conjunto de todos os n-uplos (d_1, d_2, \dots, d_n) onde $d_i \in D_i$ para $i = 1, 2, \dots, n$)

A função (A_1, A_2, \dots, A_n) é definida como:

$$(A_1, A_2, \dots, A_n)(e) = (A_1(e), A_2(e), \dots, A_n(e)) \quad (ii)$$

onde para quaisquer duas entidades e_1 e e_2 do tipo E se verifique que

$$(A_1, A_2, \dots, A_n)(e_1) \neq (A_1, A_2, \dots, A_n)(e_2) \quad (iii)$$

A condição (iii) verifica-se se existir um $j = 1, 2, \dots, n$ para o qual

$$A_j(e_1) \neq A_j(e_2)$$

Diferentes entidades são representadas por tuplos diferentes.

Dois tuplos são diferentes se têm valores diferentes em pelo menos um atributo.

2.1.5. Exemplo da representação de um tipo de entidade por um conjunto de tuplos

A notação

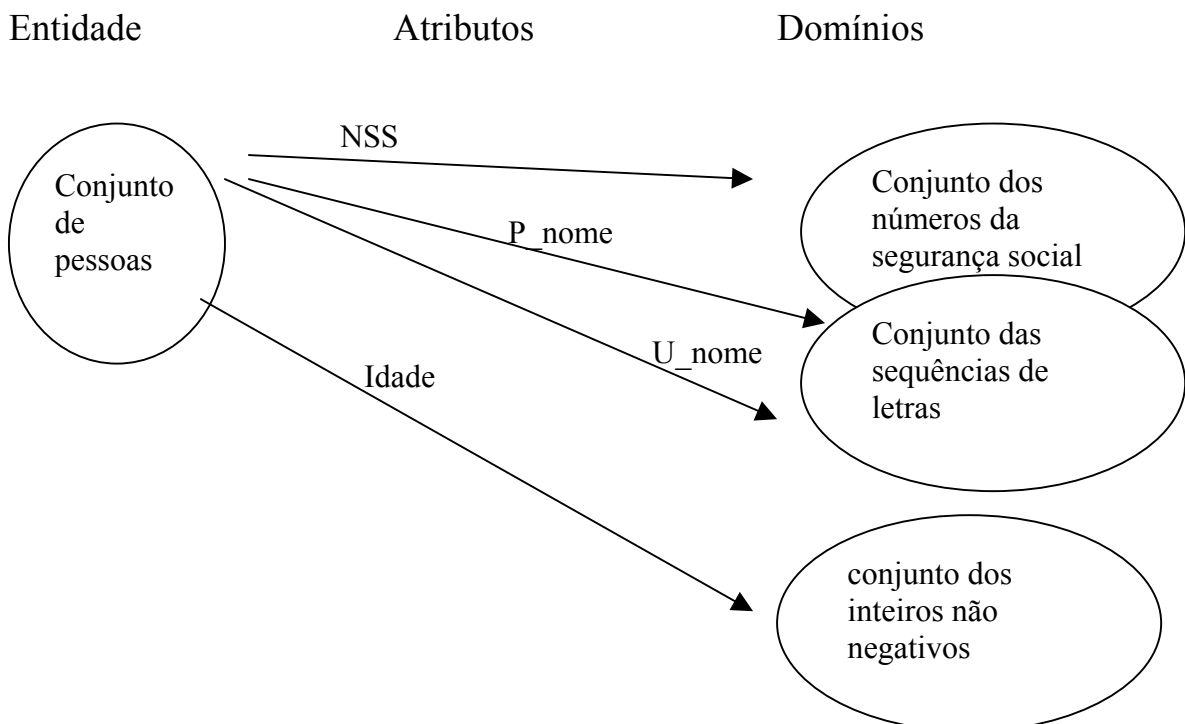
Pessoa (NSS, P_nome, U_nome, Idade) é interpretada como a função

$$(NSS, P_nome, U_nome, Idade) : Pessoa \rightarrow S \times A \times A \times N$$

Esta função determina para cada pessoa p um 4-uplo (n° de segurança social, primeiro nome, último nome, idade) que representa essa pessoa.

Diferentes pessoas p_1 e p_2 determinam diferentes tuplos.

Mesmo que tenham os mesmos primeiro e último nomes, o NSS é seguramente diferente.



Num dado instante, um conjunto de pessoas pode ser representado pela seguinte tabela:

NSS	P_nome	U_nome	Idade
941	Pedro	Silva	31
385	Mário	Sousa	24
102	Joana	Ferreira	64
243	Maria	Andrade	52
860	João	Almeida	24
543	Alice	Fonseca	45

2.1.6. Relação (ver página 31)

R é uma relação nos conjuntos D_1, D_2, \dots, D_n se e só se

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

A estrutura da relação R é descrita pela notação $R(A_1, A_2, \dots, A_n)$ onde

$$A_i: R \rightarrow D_i \text{ é uma atributo de R e } D_i \text{ o seu domínio para } i=1,2,\dots,n$$

2.1.7. Base de dados relacional e esquema relacional

Uma base de dados relacional é uma colecção de relações cujo conteúdo varia ao longo do tempo.

Um esquema relacional é a descrição da estrutura das relações numa base de dados relacional.

2.1.8. Exemplo de um esquema relacional

Departamento (Dep, Nome, Local)

Empregado (Emp, Nome, Categoria, Dep)

Projecto (Proj, Designação, Fundos)

Atribuição (Emp, Proj, Função)

- O esquema descreve 4 tipos de entidades:

Departamento, Empregado, Projecto, e Atribuição de empregados a projectos.

Atributos de entidades do tipo Departamento são:

Dep – número de departamento

Nome – nome do departamento

Local – localização do departamento

Atributos de entidades do tipo Empregado são:

Emp – nº de segurança social do empregado

Nome – nome do empregado

Categoria – Categoria do empregado

Dep – número do departamento a que pertence o empregado

Atributos de entidades do tipo Projecto são:

Proj – Código do projecto

Designação – designação do projecto

Fundos – fundos atribuídos ao projecto

Atributos de entidades do tipo Atribuição

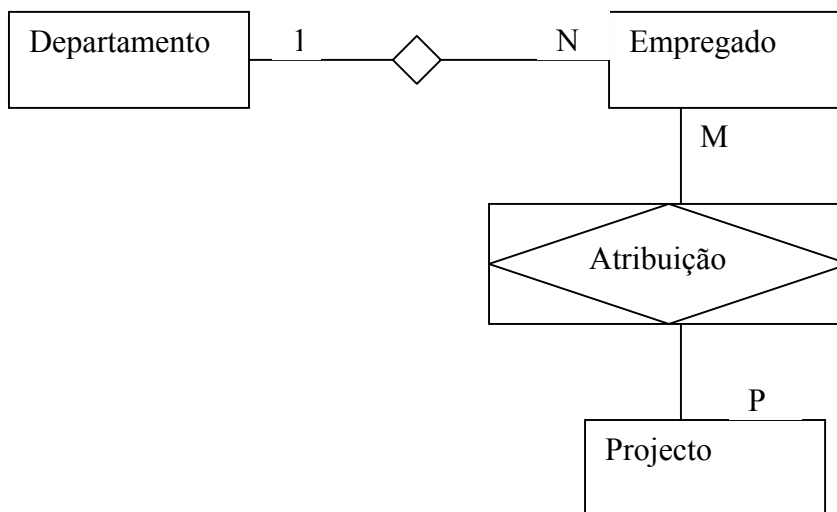
Emp – número de segurança social do empregado

Proj – código do projecto

Função – função que o empregado desempenha no projecto

- São assumidas as seguintes restrições
 - um empregado pertence a um único departamento
 - um empregado pode ser designado para vários projectos e um projecto tem vários empregados atribuídos

Representação do Modelo de dados (*a estudar posteriormente*):



Como obteríamos resposta à seguinte interrogação:

Quais os nomes dos empregados cuja categoria é “Programador” e que pertencem a Departamentos localizados em “Lisboa” ?

(Com o que já sabemos de Álgebra relacional)

(E em SQL?)

2.2. Álgebra Relacional

Um modelo por si próprio não pode realizar qualquer unidade de trabalho útil. É apenas uma representação da realidade.

Para realizar interrogações acerca das propriedades das entidades representadas no modelo precisamos de uma linguagem apropriada.

Existem várias linguagens eficientemente implementadas e amplamente aceites. Do ponto de vista conceptual todas tiveram origem numa linguagem formal denominada Álgebra Relacional.

Para aprofundar as linguagens base do modelo relacional ver Capítulo IV - “Relacional Álgebra and Relacional Calculus” de [Connolly99],

[Connolly99] Connolly, Thomas, Carolyn Begg and Anne Strachan, Database Systems, A Practical Approach to Design, Implementation and Management, Addison-Wesley 2nd Edition, 1999.
Biblioteca da UBI: I-6.2-53

A álgebra relacional consiste numa colecção de operadores sobre relações:

Operações usuais sobre conjuntos:

- União
- Intersecção
- Diferença
- Produto cartesiano

Outras operações:

- Projecção
- Restrição
- Junção
- Divisão

1 - Projecção

Seja $R(X,Y)$ com $X = A_1, A_2, \dots, A_k$

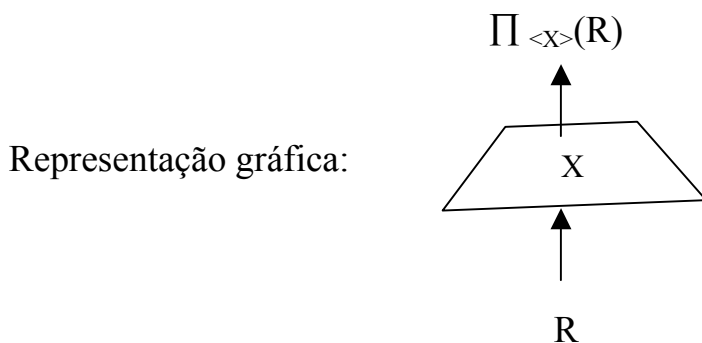
$Y = A_{k+1}, \dots, A_n$

Projecção de R sobre os atributos X :

$$\Pi_{\langle X \rangle}(R) = \{ x : \text{existe um } y \text{ tal que } (x,y) \in R(X,Y) \}$$

Se a relação R é representada como uma tabela, a operação de projecção de R sobre o conjunto de atributos X é interpretada como a selecção das colunas de R que correspondem aos atributos de X e a eliminação das linhas duplicadas na tabela obtida.

	//////////	
	//////////	
	//////////	
	//////////	



Exemplo:

Empregado (Emp#, Nome, Categoria, Dep#)

. Emp# é chave da relação empregado

Empregado

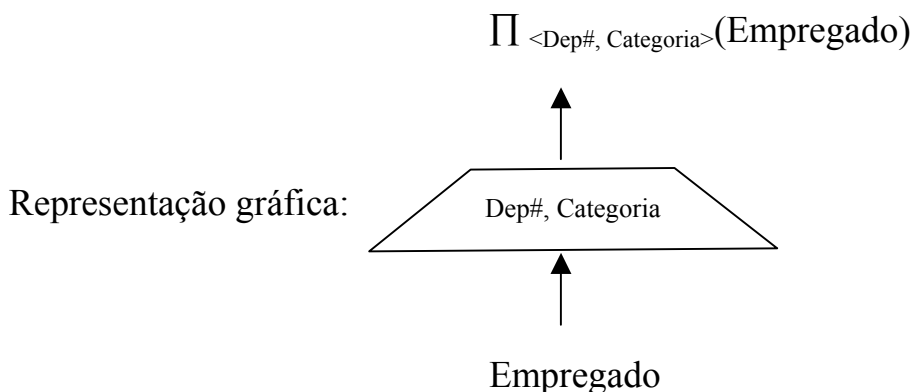
Emp#	Nome	Categoria	Dep#
e1	n1	c1	d1
e2	n2	c2	d2
e3	n3	c3	d1
e4	n4	c1	d2
e5	n5	c2	d3
e6	n6	c2	d3
e7	n7	c1	d1

Projectão da tabela Empregado sobre os atributos Dep# e Categoria,

$$\Pi_{\langle \text{Dep\#}, \text{Categoria} \rangle} (\text{Empregado})$$

dá origem à tabela:

Dep#	Categoria
d1	c1
d1	c3
d2	c1
d2	c2
d3	c2



2- Restrição (ou selecção)

Seja a relação $R (A_1, A_2, \dots, A_n)$ e p uma expressão lógica definida sobre $D_1 \times D_2, \dots, \times D_n$, com D_i domínio de A_i

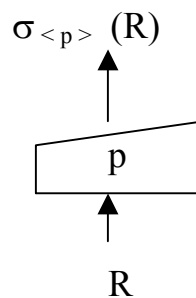
A restrição de R a respeito da condição p ,

$$\sigma_{\langle p \rangle} (R) = \{ z: z \text{ é tuplo de } R \text{ e } p(z) \text{ é verdadeiro} \}$$

Sendo R representada como uma tabela a operação de restrição pode ser interpretada como a eliminação das linhas da tabela R que não satisfazem a condição p .

//////////	//////////	//////////
//////////	//////////	//////////

Representação gráfica:

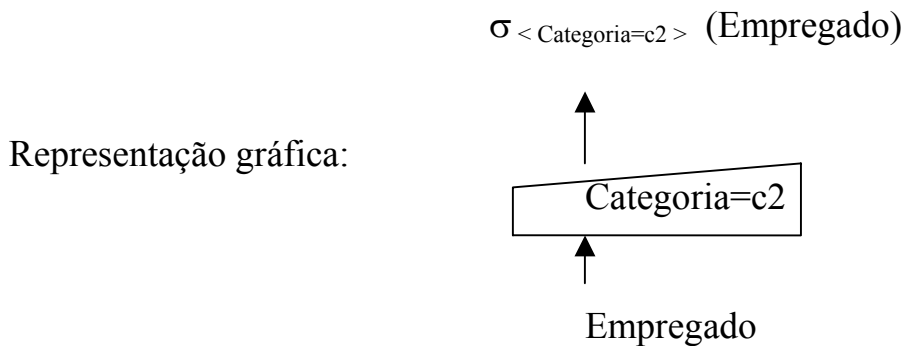


Exemplo: Restrição da tabela Empregado tal que Categoria= c2

$$\sigma_{\langle \text{Categoria=c2} \rangle} (\text{Empregado})$$

dá origem à tabela,

Emp#	Nome	Categoria	Dep#
e2	n2	c2	d2
e5	n5	c2	d3
e6	n6	c2	d3



3 – Junção (*equijunção*)

Seja $A(Z,X)$ e $B(Y,W)$

Com Z,X,Y,W conjuntos de atributos tais que

- X e Y têm o mesmo número de atributos e atributos correspondentes têm o mesmo domínio

Junção

das relações A e B sobre os atributos X e Y :

$$A \underset{x=y}{\bowtie} B = \{ (z,x,w) : (z,x) \in A \text{ and } (y,w) \in B \text{ and } x = y \}$$

O resultado é uma relação cujo conjunto de atributos é a união dos conjuntos Z, X (ou Y) e W.

Os tuplos da tabela são obtidos pela concatenação dos tuplos de A com os tuplos de B sempre que os valores dos atributos de X são iguais aos valores dos atributos de Y.

Atributos duplicados (X ou Y) são eliminados.

Exemplo:

Empregados (Emp#, Nome, Categoria, Dep#)

Departamento (Dep#, Nome, Local)

A Expressão:

$$\Pi \langle \text{Empregado.Nome, Local} \rangle (\text{Empregado} \underset{\text{Dep\# = Dep\#}}{\bowtie} \text{Departamento})$$

denota a composição de duas operações:

- A junção das relações Empregado e Departamento sobre os atributos Dep#
- A projecção do resultado da junção sobre os atributos Nome do empregado e Local

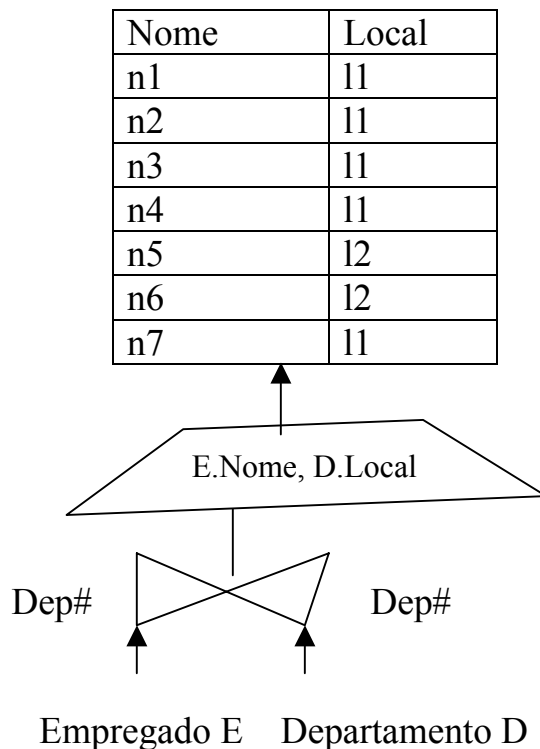
Empregado

Emp#	Nome	Categoria	Dep#
e1	n1	c1	d1
e2	n2	c2	d2
e3	n3	c3	d1
e4	n4	c1	d2
e5	n5	c2	d3
e6	n6	c2	d3
e7	n7	c1	d1

Departamento

Dep#	Nome	Local
d1	N1	l1
d2	N2	l1
d3	N3	l2

Representação gráfica:



- Qual a pergunta a que esta operação responde?

4 – Divisão

Seja as relações $A(X,Y)$ e $B(Z)$ com X,Y, Z conjuntos de atributos.

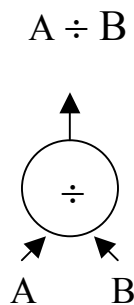
Y e Z contêm igual número de atributos e os domínios correspondentes são iguais .

A divisão de A por B sobre Y e Z é

$$A \div B = \{ x : \forall z \in B, (x,z) \in A \}$$

Valores de x tais que o par (x,z) ocorre em A para todos os valores de z que ocorrem em B .

Representação gráfica:



Exemplo: Atribuição \div ($\Pi_{\langle \text{Proj}\# \rangle}(\text{Projecto})$)

Denota a divisão da relação Atribuição pela projecção da relação Projecto sobre o atributo Proj#

Dadas as relações,

Projecto

Proj#	Designação	Fundos
p1	t1	f1
p2	t2	f2
p3	t3	f3

Atribuição

Emp#	Proj#	Função
e1	p1	r1
e2	p3	r1
e2	p2	r2
e3	p2	r1
e3	p3	r1
e4	p1	r1
e5	p3	r2
e6	p1	r3
e6	p2	r3
e6	p3	r3
e7	p1	r1

O resultado de $\Pi_{\langle \text{Proj\#} \rangle}(\text{Projecto})$ é

Proj#
p1
p2
p3

O resultado da divisão é:

Emp#	Função
e6	r3

- Qual a pergunta a que esta operação responde?

Exercício: dadas as tabelas,

D

S	P
s1	p1
s1	p2
s1	p3
s1	p4
s1	p5
s1	p6
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4
s4	p5

d1

P
p1

d2

P
p2
p4

d3

P
p1
p2
p3
p4
p5
p6

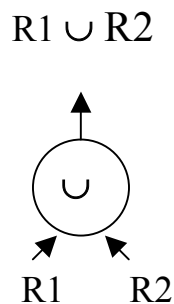
- Calcule 1 - $D \div d1$, 2 - $D \div d2$, 3 - $D \div d3$

União, Intersecção e Diferença

Dados R1 e R2 tais que têm igual número de atributos e os domínios dos atributos correspondentes são os mesmos (esquemas relacionais compatíveis)

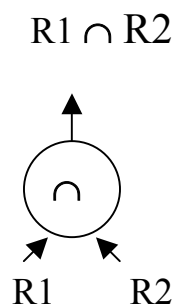
5 - União

$R1 \cup R2$ é o conjunto dos tuplos de R1 e R2



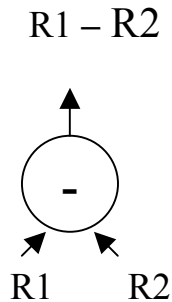
6 – Intersecção

$R1 \cap R2$ é o conjunto de tuplos comuns a R1 e R2.



7 – Diferença

$R1 - R2$ é o conjunto de tuplos de $R1$ que não pertencem a $R2$



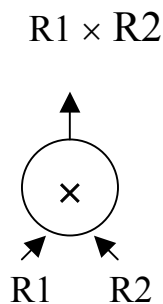
Para 5, 6 e 7 a relação resultado tem os mesmos atributos que o 1º operando

8 – Produto Cartesiano

Dadas $R1$ e $R2$ com qualquer esquema,

$R1 \times R2$ é concatenação dos atributos de $R1$ e $R2$.

Cada tuplo de $R1$ é concatenado com cada tuplo de $R2$.



Comparar com a junção ...

A representação gráfica permite construir uma árvore para exprimir questões à Base de Dados.

Seja a base de dados exemplo (página 32) e as questões

I1: Quem forneceu o material M1 para a obra O1?

I2: Que materiais (nomes) forneceu o fornecedor F2 e para que obras (nomes)?

- Construir a resposta em álgebra relacional.

Exercício:

A junção não é a uma operação essencial, podendo ser definida em termos de operações mais primitiva. O mesmo é válido para a intersecção e divisão

(As primitivas da linguagem são: União, Diferença, Produto, Selecção e Projecção)

- Definir junção, intersecção e divisão em termos dessas 5 primitivas.