

SGBD relacional

Baseiam-se num conjunto de conceitos teóricos apresentados em 1970 por **E. F. Codd**.

- os conceitos que Codd introduziu foram desenvolvidos por outros autores
- actualmente são os mais utilizados

Vantagens dos SGBD relacionais:

- **simplicidade** dos conceitos que utilizam
- existência de **definições formais** para os conceitos
 - permitiram uma rápida divulgação
 - permitiram a adesão de diversos fabricantes de software;
- **adequação** à representação de muitos dos aspectos que constituem a realidade

Relação

- ☞ Nas bases de dados relacionais a estrutura fundamental é a **relação**.
- ☞ Uma relação é definida por um **esquema** e por uma **tabela**.
- ☞ Um esquema é composto:
 - pelo **nome da relação**
 - pelos nomes de cada um dos **atributos**
 - a definição de cada atributo depende do tipo de dados (inteiro, real, alfanumérico, data, lógico ...) que o atributo irá armazenar.

Exemplos de esquemas


Aluno(nAluno, nomeA, datanasc)

Professor(nProf, nomeP, categoria)

Atributos

- Um atributo A_i toma valores num conjunto D_i chamado **domínio** do atributo.
 - O domínio determina o tipo de valores que o atributo pode tomar.
- Dado $U = \{A_1, A_2, \dots, A_n\}$, uma relação R sobre U é um **subconjunto** de $D_1 \times D_2 \times \dots \times D_n$.
- A cada tuplo deste produto cartesiano dá-se o nome de **instância** da relação R .
 - as linhas da relação R são os **registos**
 - as colunas são os atributos (ou **campos**)

Tabela

 O conjunto das instâncias da relação R constitui uma **tabela** em que

- as linhas são as instâncias (ou **registos**)
- as colunas são os atributos (ou **campos**)

Exemplo de uma tabela

Atributos ou campos			
←	↓	→	
naluno	nomeA	datanasc	
11111	Ana Maria Santos	1978-12-03	↙
12345	José Silva	1980-11-30	←
22222	Natália Costa	1985-12-25	↙
...	

Instâncias
ou
registos

- Domínio de **naluno**: integer (**N**)
- Domínio de **nome**: string 50 (**A**)
- Domínio de **datanasc**: date (**D**)
- Cada registo (instância) é um tuplo

$(12345, \text{"José Silva"}, 1980-11-03) \in N \times A \times D$

Observações

- ☞ Numa instância o valor de um atributo é sempre **atômico**;
 - isto é, numa tabela, no cruzamento de uma linha com uma coluna só pode existir um valor de atributo;
 - ☞ numa relação **não** podem existir instâncias iguais;
 - ☞ a **ordem** porque se encontram as instâncias de uma relação e os seus atributos é **irrelevante**;
 - ☞ os valores de cada atributo pertencem a um mesmo **domínio**;
 - ☞ podem existir instâncias sem valores em alguns dos seus atributos; neste caso o atributo diz-se opcional e o seu valor é **null**;
 - ☞ os nomes (ou **identificadores**) dos atributos que constituem o esquema de uma relação são únicos nessa relação.
-


Chave(s) de uma relação

- Um atributo que toma valores diferentes para cada instância da relação é uma **chave candidata** da relação.
 - cada instância pode ser identificada pelo valor de um atributo chave: dado um determinado valor dessa chave só existe uma instância em que o atributo possui esse valor.
- Por vezes, uma chave pode ser composta por mais do que um atributo. E, no caso extremo, uma chave pode até ser composta por todos os atributos que definem a relação.

Chave primária

- 📄 **Chave primária** de uma relação é um subconjunto mínimo de atributos cujos valores permitam identificar de modo único cada uma das instâncias dessa relação.
 - 📄 Nos SGBD relacionais, para representar as associações existentes entre as várias entidades utilizam-se esquemas de relações em que figuram **atributos comuns**.
 - 📄 Uma **chave estrangeira** de uma relação é um conjunto de atributos que é chave primária de outra relação.
-

Restrições de integridade

 **Condições** que devem ser verificadas para que um elemento do produto cartesiano dos domínios dos atributos possa pertencer à relação.

 Consequências imediatas das definições dadas:

- **integridade de domínio**
 - impostas pelo domínio de cada atributo;
- **integridade de entidade**
 - não existem duas instâncias de uma relação cujas chaves tenham o mesmo valor
 - valor de uma chave primária (ou mesmo de uma parte dela) não pode ser *null*;
- **integridade de referência**
 - o valor de qualquer chave estrangeira de uma relação é um valor da chave primária da relação que refere

Outras restrições de integridade

Outra categoria de restrições:

- condições que devem ser verificadas pelos valores de dois ou mais atributos de uma mesma instância da relação.
 - Algumas restrições requerem a utilização de fórmulas matemáticas ou lógicas.

Dependências funcionais:

- Dada um relação R definida sobre um conjunto de atributos $U = \{A_1, A_2, \dots, A_n\}$, diz-se que o atributo A_j **depende funcionalmente do atributo A_i** ($A_i \rightarrow A_j$) quando é impossível terem-se duas instâncias de R com o mesmo valor para A_i e diferentes valores para A_j .

Esquema de uma base de dados

- ❏ O conjunto das relações que a compõem, definidas por meio dos seus atributos, e das restrições de integridade.
- ❏ Para cada base de dados pode existir um número quase ilimitado de esquemas.
- ❏ Um esquema é tanto melhor quanto mais fácil for o seu manuseamento, nomeadamente no que se refere à manutenção da sua coerência após as actualizações que venha a sofrer.
- ❏ Um esquema é tanto melhor quanto menor for o número de dependências funcionais nele existente.

Normalização

- ❏ O objectivo das regras de normalização é definir regras para o agrupamento de atributos de forma a minimizar o número de dependências funcionais existentes numa base de dados.
- ❏ Resumidamente, pode dizer-se que numa relação que verifica as primeiras três formas normais, qualquer atributo que não pertence à chave depende completamente e exclusivamente da totalidade da chave.

Primeira forma normal

☞ Uma relação R encontra-se na primeira forma normal se e só se todos os seus atributos contêm valores atômicos.

- na prática, todas as instâncias de uma relação têm o mesmo número de atributos e não existem duas instâncias de uma relação que sejam iguais.

Aluno (nAluno, nomeA, disciplina1, ..., disciplinaj, ..., dataNasc)

☞ não respeita a 1ª forma normal (nem é sequer uma relação) porque possui um número variável de atributos.

Segunda forma normal

- ☞ Uma relação R encontra-se na segunda forma normal se e só se:
- verifica a primeira forma normal e
 - os atributos que não são chave de R são dependentes da **totalidade** da chave primária de R.
 - Esta normalização visa diminuir a duplicação de dados.

Inscrito (nAluno, código, nomeA, dataNasc)

☞ não se encontra na 2ª forma normal porque ...

☞ Aplicação da 2ª forma normal a esta relação:

Inscrito (nAluno, código)

Aluno (nAluno, nomeA, dataNasc)

Terceira forma normal

- ☞ Uma relação R encontra-se na terceira forma normal se e só se:
 - verifica a segunda forma normal e
 - todos os atributos que não são chave de R são não-transitivamente dependentes dessa chave.
- ☞ Um atributo que não pertence à chave primária da relação não pode ser funcionalmente dependente de outro atributo que também não pertence à chave da relação.

Disciplina (código, designação, nProf, nomeP, categoria)

- ☞ não se encontra na 3ª forma normal porque ...

Passagem do modelo E-A para relacional

- ☞ Um modelo de tipo E-A descreve bem e de uma forma natural e simples a realidade.
 - ☞ Os modelos deste tipo são suficientemente flexíveis para poderem ser utilizados eficazmente numa fase em que a estruturação dos dados é ainda confusa.
 - ☞ A estas vantagens pode ainda adicionar-se a facilidade com que é possível efectuar a sua passagem para um SGBD relacional, bem formalizado e comercializado por diversos fornecedores de *software*.
-

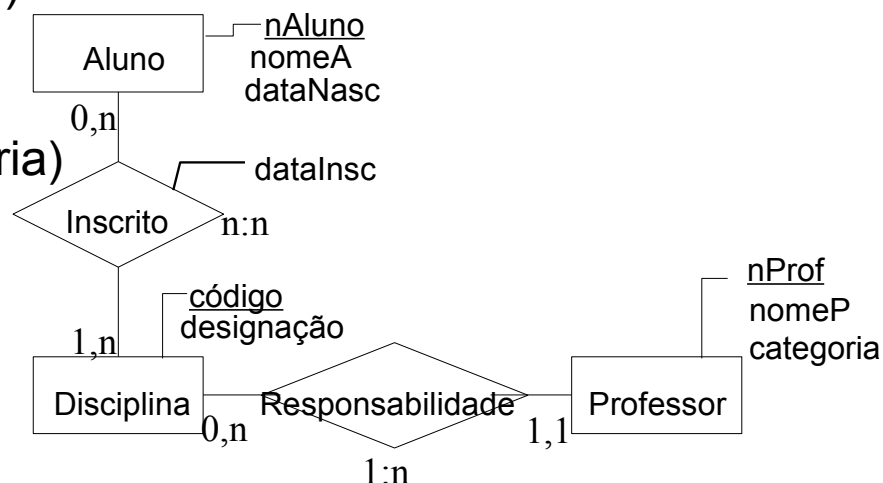
Regras para transformação de um modelo E-A num esquema relacional (1)

☞ Cada tipo de entidades do modelo E-A traduz-se por uma relação em que a chave primária e os atributos provêm do tipo de entidade.

Aluno (nAluno, nomeA, dataNasc)

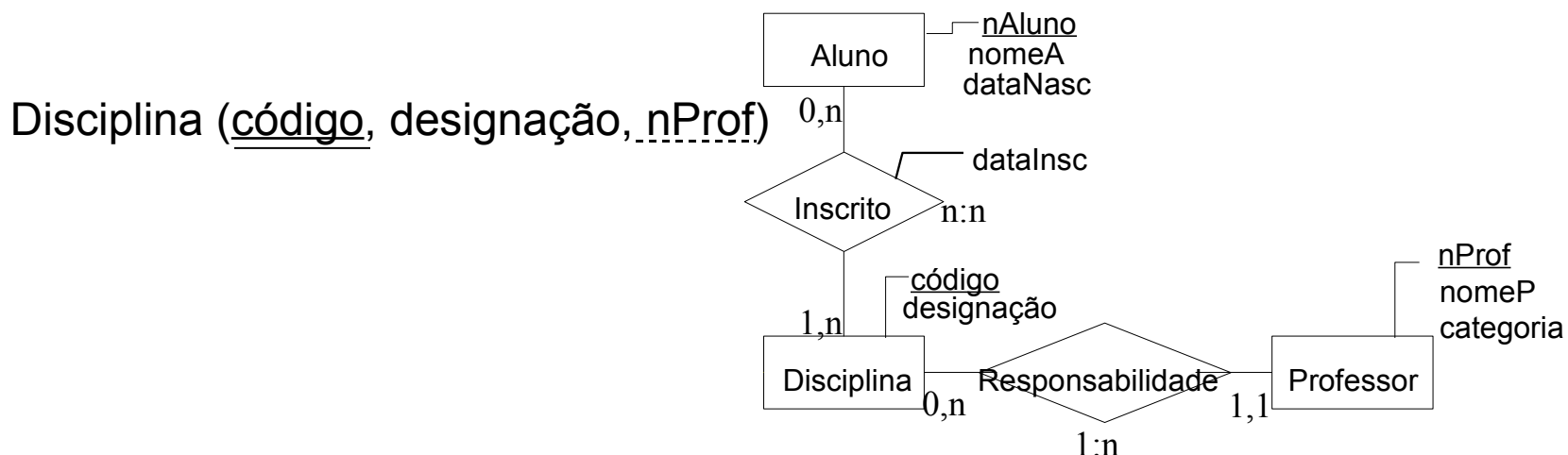
Disciplina (código, designação)

Professor (nProf, nomeP, categoria)



Regras para transformação de um modelo E-A num esquema relacional (2)


Um tipo de associação de 1:n (um para muitos) entre dois tipos de entidades E_i e E_j que tenha uma multiplicidade igual a 0,1 ou 1,1 para um tipo de entidade E_i traduz-se por uma chave estrangeira na relação R que é tradução de E_j .



Regras para transformação de um modelo E-A num esquema relacional (2')

- Um tipo de associação de 1:1 (um para um) entre dois tipos de entidades E_i e E_j é tratado como um caso especial do tipo de associação 1:n
- traduz-se por uma chave estrangeira na relação que é tradução de E_j ou de E_i ;
 - se apenas para um destes tipos entidades a multiplicidade for 1,1 (sendo para a outra 0,1) dá-se preferência à relação que traduz este tipo de entidade,
 - no caso contrário é indiferente qual a relação que é escolhida.

Regras para transformação de um modelo E-A num esquema relacional (3)

-  Um tipo de associação de n:n (muitos para muitos) traduz-se por uma relação em que a chave primária inclui as chaves estrangeiras que são chave primária dos tipos de entidade que a constituem; os outros atributos são a tradução dos tipos de atributos da associação (se esta possuir algum).
- No caso das chaves estrangeiras não serem suficientes para formar a chave primária da relação, na constituição desta devem também ser utilizados outros atributos de forma a ser obtida uma chave primária adequada à relação.
-

Regras para transformação de um modelo E-A num esquema relacional (3) (exemplo)

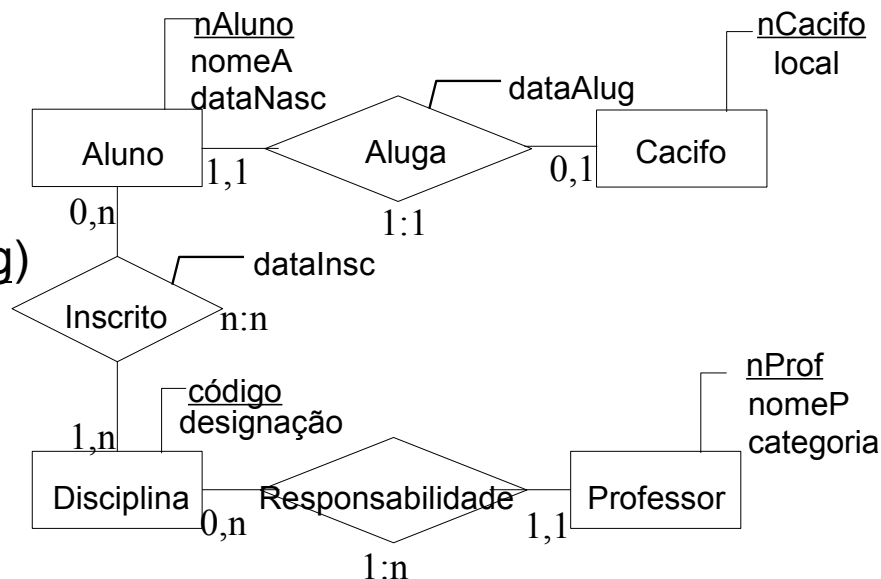
Aluno (nAluno, nomeA, dataNasc)

Disciplina (código, designação, nProf)

Professor (nProf, nomeP, categoria)

Inscrito (nAluno, código, dataInsc)

Cacifo (nCacifo, local, nAluno, dataAlug)



Na notação utilizada

- um duplo sublinhado indica que o conjunto de atributos é a chave primária da relação
- um sublinhado simples tracejado indica uma chave estrangeira.

Automatismo ...!!!

- Estas regras permitem que se proceda de uma forma rápida (e quase automática) à passagem de um modelo Entidade-Associação em que só existam associações binárias para um esquema relacional.