

Bases de Dados

Modelação

Programação e Bases de Dados (CC4042)
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Adaptado de slides de
Eduardo R. B. Marques, Fernando Silva e Ricardo Rocha

Introdução

Alguns Conceitos

- **Base de dados (BD)**: conjunto de dados que se relacionam entre si.
- **Dados**: factos conhecidos que têm algum significado e que podem ser guardados de forma persistente.
- **Universo**: parte do mundo real sobre o qual os dados guardados na base de dados dizem respeito.
 - Empresa: empregados, departamentos, projetos, ...
 - Universidade: alunos, professores, unidades curriculares, inscrições, horários, ...
 - Contactos: nomes, endereços, telefones, ...

Exemplo de uma BD

CADEIRA	CodCad	Nome	Docente
	12347	Bases de Dados	José Aguiar Mota
	34248	Álgebra	Maria das Dores
	32439	Introdução aos Computadores	Carlos Duarte

ALUNO	NumMec	Nome	Curso
	798764544	João Pinto	LCC
	345673451	Carlos Semedo	MIERSI
	487563546	Maria Silva	LBIO
	452212348	Pedro Costa	LMAT

INSCRIÇÃO	NumMec	CodCad
	798764544	12347
	345673451	12347
	798764544	34248
	452212348	32439

- **Universo** = uma faculdade
- **Dados**: alunos, cadeiras, inscrições em cadeiras.

Inter-relacionamento entre dados

CADEIRA	CodCad	Nome	Docente
	<u>12347</u>	Bases de Dados	José Aguiar Mota
	<u>34248</u>	Álgebra	Maria das Dores
	<u>32439</u>	Introdução aos Computadores	Carlos Duarte

ALUNO	NumMec	Nome	Curso
	<u>798764544</u>	João Pinto	LCC
	<u>345673451</u>	Carlos Semedo	MIERSI
	<u>487563546</u>	Maria Silva	LBIO
	<u>452212348</u>	Pedro Costa	LMAT

INSCRIÇÃO	NumMec	CodCad
	<u>798764544</u>	12347
	<u>345673451</u>	12347
	<u>798764544</u>	<u>34248</u>
	<u>452212348</u>	<u>32439</u>

- **Os dados numa BD relacionam-se entre si de acordo com o universo modelado.**
- **As relações entre dados devem ser consistentes.** Por exemplo nesta BD não devem existir dois alunos com o mesmo nº mecanográfico ou duas cadeiras com o mesmo código, e uma inscrição refere-se a alunos e cadeiras que existam na base de dados.

Exemplo de uma BD (cont.)

ALUNO	NumMec	Nome	Curso
798764544	João Pinto	LCC	
345673451	Carlos Semedo	MIERSI	
487563546	Maria Silva	LBIO	
452212348	Pedro Costa	LMAT	

Tabela {

← **Registo**
(linha)

↑ **Campo** (coluna)

- Está em causa uma **BD relacional**, o tipo de BD mais comum e que iremos considerar nesta cadeira.
- **Intuição preliminar**: numa BD relacional a informação está organizada em **tabelas**, cada uma contendo um conjunto de **registos** (linhas), decomposto em vários **campos** (colunas).

Sistema de Gestão de Bases de Dados (SGBD)

- **SGBD**: software que permite criar e manipular uma base de dados.
 - Em inglês **DBMS** (“Database Management System”)
 - Também designado por **motor de base de dados** (“database engine”).
- Um SGBD lida abstrai a forma como os dados estão **estruturados e armazenados**.
- Por contraste, num sistema tradicional de ficheiros:
 - A estrutura de um ficheiro de dados está intrinsecamente espelhada nos programas que manipulam esses ficheiros.
 - A alteração de estrutura dos ficheiro de dados obriga à alteração de todos os programas.

Aspectos essenciais de um SGBD

■ Persistência e disponibilidade dos dados

- Uma BD é armazenada de forma **persistente** (durável) e está **disponível** para acesso em qualquer altura.

■ Mutabilidade, atomicidade, consistência, concorrência

- O estado de uma BD (cjto. de dados contidos na BD) é **mutável**, i.e., pode ser alterado ao longo do tempo.
- Cada alteração ao estado deve ser logicamente **atômica**.
- O estado deve ser sempre **consistente**, de acordo com as regras definidas para a BD.
- Podem haver **operações concorrentes** (simultâneas) sobre os dados, sem prejuízo da consistência da BD.

Aspectos essenciais de um SGBD (cont.)

■ Segurança

- Uma BD deve ser acedida de forma **segura** por exemplo mediante autenticação e suporte de diferentes níveis de acesso aos dados.

■ Acessibilidade

- O acesso a BDs pode ser feito por software externo, de forma programática e através de uma rede.

■ Administração

- O SGBD deve permitir a administração de BDs, de forma programática e/ou através de ferramentas específicas para o efeito.

Aplicações de BD – modelo típico “3-tier”

utilizador



apresentação dos dados



servidor



lógica aplicacional

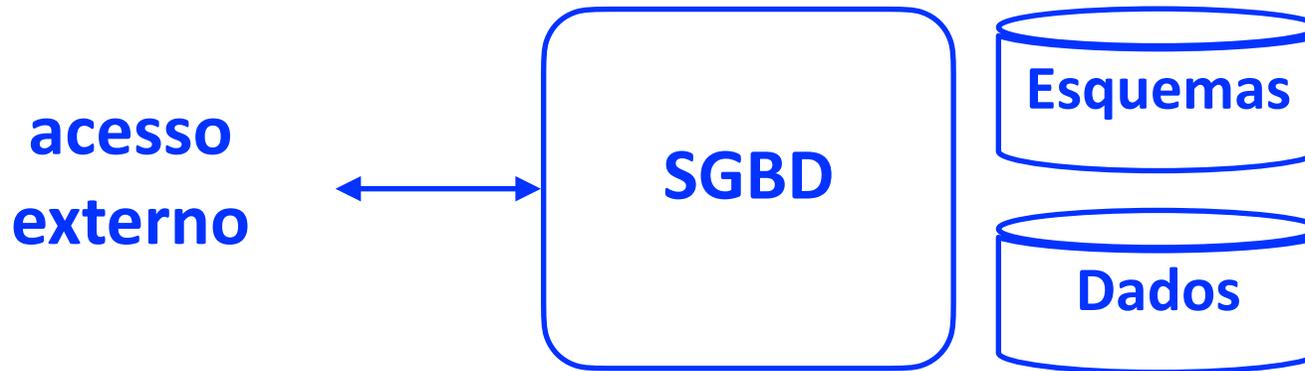


DBMS



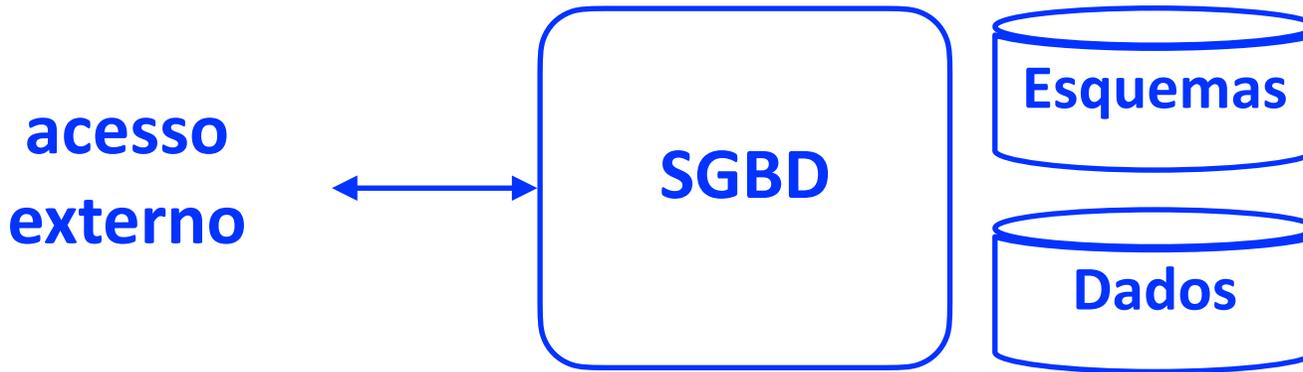
dados

SGBD: esquemas e dados



- Um SGDB permite a definição e manipulação de:
 - **Esquemas de base de dados**, que descrevem como uma base de dados está estruturada, também chamados de meta-dados.
 - **Bases de dados = instância de esquemas.**
- Um SGBD pode suportar BDs distintas com o mesmo esquema e a existência de vários esquemas distintos.

Como se “fala” com uma base de dados?



- Do ponto de vista de acesso externo, um SGBD “fala” dois tipos de linguagem:
 - **DDL (“Data Definition Language”)**: para lidar com **esquemas**.
 - **DML (“Data Manipulation Language”)**: para lidar com os **dados** de uma BD.
- Nesta cadeira iremos considerar BDs manipuladas usando a linguagem **SQL (Structured Query Language)** que cumpre estes dois papéis.

Um “cheirinho” de SQL

ALUNO	NumMec	Nome	Curso
-------	--------	------	-------

```
CREATE TABLE ALUNO (  
    NumMec INTEGER PRIMARY KEY NOT NULL,  
    Nome    VARCHAR(64) NOT NULL,  
    Curso   VARCHAR(20) NOT NULL  
);
```

- SQL usada como DDL – criação de tabela.

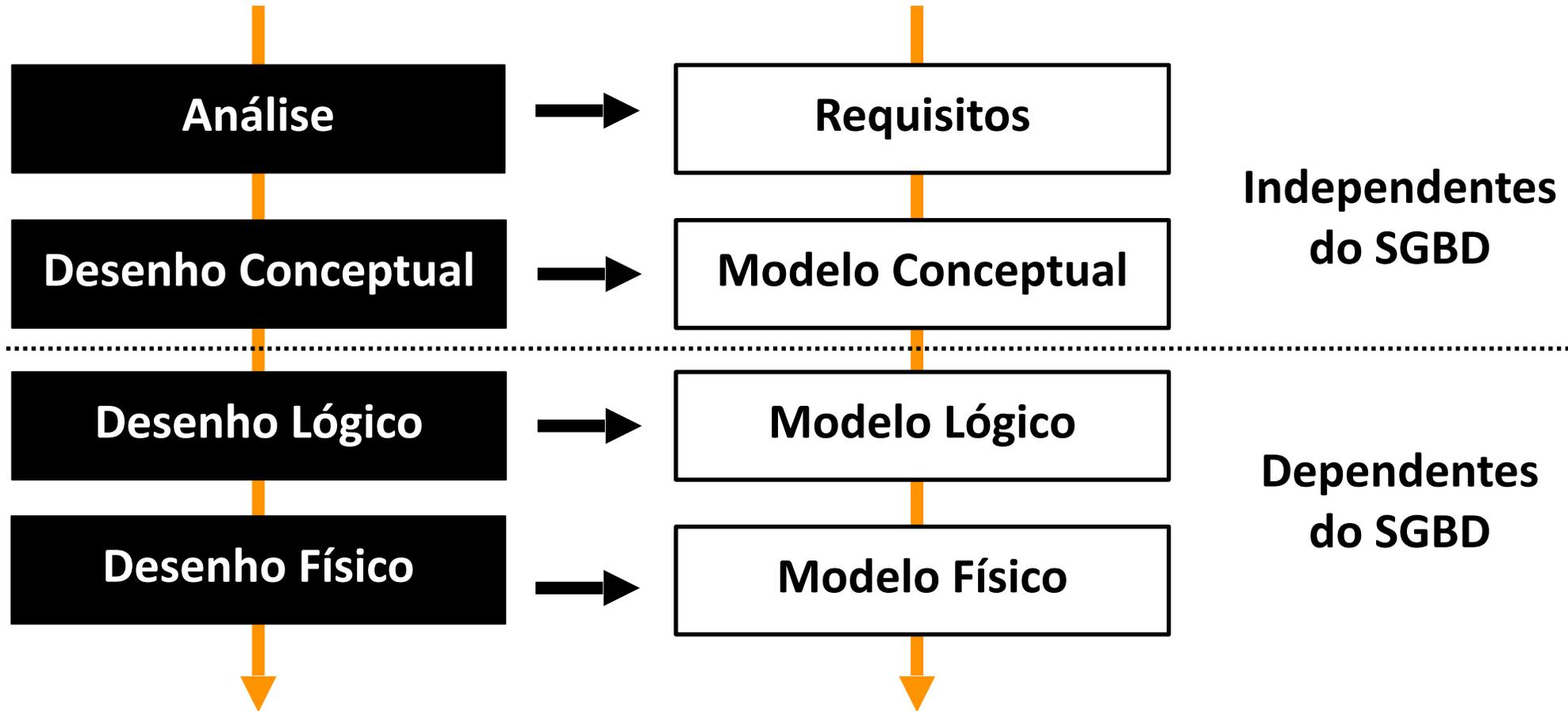
Um “cheirinho” de SQL (cont.)

ALUNO	NumMec	Nome	Curso
	798764544	João Pinto	LCC
	345673451	Carlos Semedo	MIERSI
	487563546	Maria Silva	LBIO
	452212348	Pedro Costa	LMAT

```
INSERT INTO ALUNO (NumMec, Nome, Curso)
VALUES (798764544, 'João Pinto', 'LCC');
INSERT INTO ALUNO (NumMec, Nome, Curso)
VALUES (345673451, 'Carlos Semedo', 'MIERSI');
INSERT INTO ALUNO (NumMec, Nome, Curso)
VALUES (487563546, 'Maria Silva', 'LBIO');
INSERT INTO ALUNO (NumMec, Nome, Curso)
VALUES (452212348, 'Pedro Costa', 'LMAT');
```

- SQL usada como DML, neste caso para inserção de registos.

Desenho e implementação de uma BD



- O desenho e implementação de uma BD compreende várias fases e correspondentes níveis de modelação.

Desenho e implementação de uma BD (cont.)

■ Modelo de Dados

- Conjunto de conceitos que descrevem a estrutura da BD em termos da relação entre dados, seu significado, e restrições.

■ Tipos de modelo:

- **Conceptual:** descreve os conceitos a representar em uma BD de forma independente do SGBD.
- **Lógico:** descreve de forma lógica a implementação de uma BD num SGBD.
- **Físico:** descreve a organização física interna dos dados no SGBD.

Desenho e implementação de uma BD (cont.)

■ **Análise**

- Compreensão do universo para a BD, resultando num conjunto de requisitos documentados.
- A derivação de requisitos pode passar por reuniões entre peritos sobre o universo em causa, entrevistas com potenciais utilizadores / clientes, etc.

■ **Desenho conceptual (ou modelação)**

- Definição de um modelo conceptual a partir dos requisitos.
- O modelo deve descrever as entidades da BD e a forma como se relacionam, de forma independente do SGBD.
- Durante a cadeira iremos considerar os modelos conceptuais **Entidade-Relacionamento (ER)** e o modelo **Entidade-Relacionamento Estendido (EER)**.

Desenho e implementação de uma BD (cont.)

■ **Desenho lógico (ou implementação)**

- Mapeamento do modelo de dados conceptual num modelo de dados lógico concreto.
- Implementação da BD usando um SGBD.
- Na cadeira iremos considerar o **modelo relacional** e a sua implementação em **SQL**.

■ **Desenho físico**

- Mapear o modelo de dados lógico no modelo de dados físico interno ao SGBD, ex. em termos de parametrização do tipo de armazenamento a usar, optimização do seu uso tendo em conta padrões de acesso, operação em rede, redundância, ...

Modelo Entidade Relação

Modelação conceptual de BDs

■ De requisitos a modelo conceptual ...

- Tendo identificado os requisitos do universo de uma BD torna-se útil a sua modelação conceptual.
- A modelação conceptual tem por propósito definir um modelo para a BD **independente** do tipo de base de dados e SGBD específico que depois se empregue na fase de implementação.

■ Modelo Entidade-Relacionamento (ER)

- Modelo usado para desenho conceptual de uma BD, empregando os conceitos de **entidades**, **atributos** e **relacionamentos**.
- Tem associadas uma **sintaxe textual** e também uma **sintaxe visual na forma de diagramas ER**.

Conceitos do modelo ER

■ Entidades

- **Objetos ou conceitos do mundo real com uma existência independente.**
- Com existência física: PESSOA CARRO PRODUTO ...
- Com existência conceptual: EMPRESA PROFISSÃO CURSO ...

■ Atributos

- **Propriedades que caracterizam as entidades.**
- Atributos da entidade PESSOA: NumCC Nome Sexo DataNasc

■ Relacionamentos

- **Representam ligações entre duas ou mais entidades.**
- Relacionamento ESTUDA: um ESTUDANTE ESTUDA em uma ESCOLA.

Entidades e atributos

Entidade-tipo

- **Entidade-tipo:** Modelo para um conjunto de entidades que partilham a mesma estrutura, definido por um **nome** e uma **lista de atributos**.
- Exemplo: **PESSOA(NumCC, Nome, Sexo, DataNasc)**
 - **Nome:** **PESSOA**
 - **Atributos:** **NumCC, Nome, Sexo, DataNasc**
 - Entidade-tipo modelando o universo de pessoas com atributos: cartão de cidadão, nome, sexo e data de nascimento.
- Convenções
 - **Nome** em maiúsculas e singular.
 - Atributos: **PrimeiraLetraDeCadaPalavra** maiúscula. Abreviaturas são comuns (como em **NumCC** ou **DataNasc**).

Entidade

- Uma **entidade** é uma instância concreta do esquema modelado por uma entidade-tipo.
- Para `PESSOA(NumCC, Nome, Sexo, DataNasc)` podemos ter por exemplo as seguintes entidades:
 - `PESSOA1(19428771, 'José Silva', M, 17-11-2000)`
 - `PESSOA2(10447777, 'Maria Carvalho', F, 08-02-1976)`
 - `PESSOA3(12819123, 'Rita Assunção', F, 09-02-1987)`
 - `PESSOA4(12818456, 'Rita Assunção', F, 08-02-1987)`
- Obs.: duas pessoas diferentes podem ter o mesmo nome, sexo, e (pouco provável mas possível) até a mesma data de nascimento. O número do cartão de cidadão permite no entanto **identificar de forma única** uma entidade `PESSOA`.

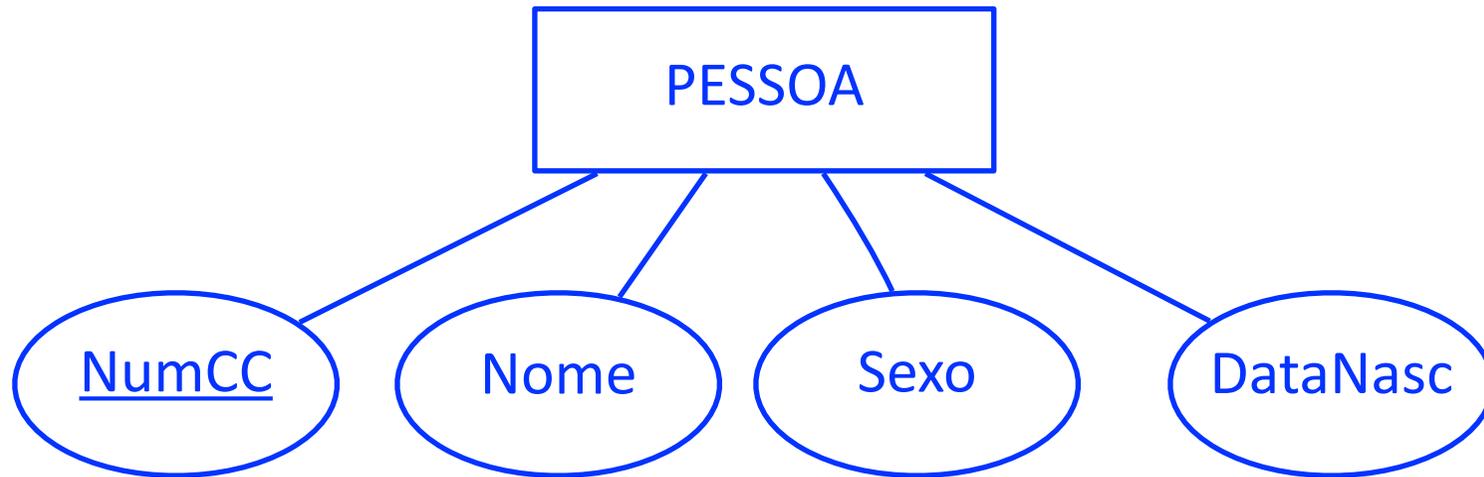
Atributos chave

- **Atributos chave:** conjunto mínimo de atributos que identificam de forma única cada entidade.
- **PESSOA(NumCC, Nome, Sexo, DataNasc)** tem apenas um atributo-chave: NumCC.
- O número de cartão de cidadão identifica de forma única uma pessoa.
- A convenção sintáctica é que um atributo chave apareça sublinhado.

Atributos chave (cont.)

- Consideremos agora o universo de uma faculdade ...
- **AULA(Dia, Hora, Sala, UnidadeCurricular, Turma)** tem três atributos chave, pois não podemos ter duas aulas diferentes no mesmo dia, à mesma hora, e na mesma sala. Podem no entanto haver por exemplo aulas simultâneas a decorrer em salas diferentes.
- Por outro lado, o conjunto de chaves deverá ser mínimo e o mais adequado ao universo em causa. Por exemplo, em **ESTUDANTE(NumMec, NumCC, ...)** o n.º mecanográfico (**NumMec**) identifica de forma única um aluno, embora o n.º de cartão de cidadão (**NumCC**) também identifique de forma única um estudante. O n.º mecanográfico é o mais adequado para chave, porque se relaciona directamente com o universo em causa.

Entidades-tipo – diagrama ER



- Além da representação textual o modelo ER emprega uma notação visual na forma de **diagrama ER**.
- Convenções:



Domínios de atributos

- O **domínio de um atributo** é o conjunto de valores que um atributo pode tomar.
- Dependendo do domínio em causa o valor de um atributo pode ser:
 - **Definido** por valor concreto OU possivelmente **indefinido** se opcional para a entidade, designado nesse caso por **NULL**.
 - **Simples** OU **composto** por vários sub-atributos.
 - **Derivado** se derivado do valor de outros atributos ou informação no modelo.
 - Ter **valor único** OU ser **multi-valor** (conjunto de valores)
 - **Complexo** se formado pela combinação de vários atributos multi-valor e/ou compostos.

Domínios de atributos

- Vamos considerar mais atributos para **PESSOA**, por ex.,
PESSOA(NumCC, Nome, Sexo, DataNasc, Idade, Morada(Rua,Num,Andar?,Localidade,CodPostal), {NumTelefone}, {Habilitação(Grau,Ano, Instituição)})
- Temos que:
 - **Idade** é um atributo **derivado** de **DataNasc**
 - **Morada** é um atributo **composto** por sub-atributos **Rua, Num, Andar, Localidade, CodPostal**.
Numa modelação alternativa poderíamos também decompor **DataNasc** em dia, mês e ano (usualmente datas são no entanto tratadas como valores simples por serem um tipo básico suportado por um SGBD) ou **Nome** em nome principal e apelidos.
 - O sub-atributo **Andar** (denotado com **?**) de **Morada** poderá tomar valor **NULL** (ser **indefinido**).
 - **{NumTelefone}** é **multi-valor**: considera-se que uma pessoa pode ter mais do que um n^o de telefone.
 - **{Habilitação(Grau,Ano,Instituição)}** é um atributo **complexo**, pois resulta da combinação de atributos multi-valor e composto. Uma pessoa pode ter várias habilitações académicas , cada um caracterizado por um grau, ano e instituição.

Domínios de atributos (cont.)

Para

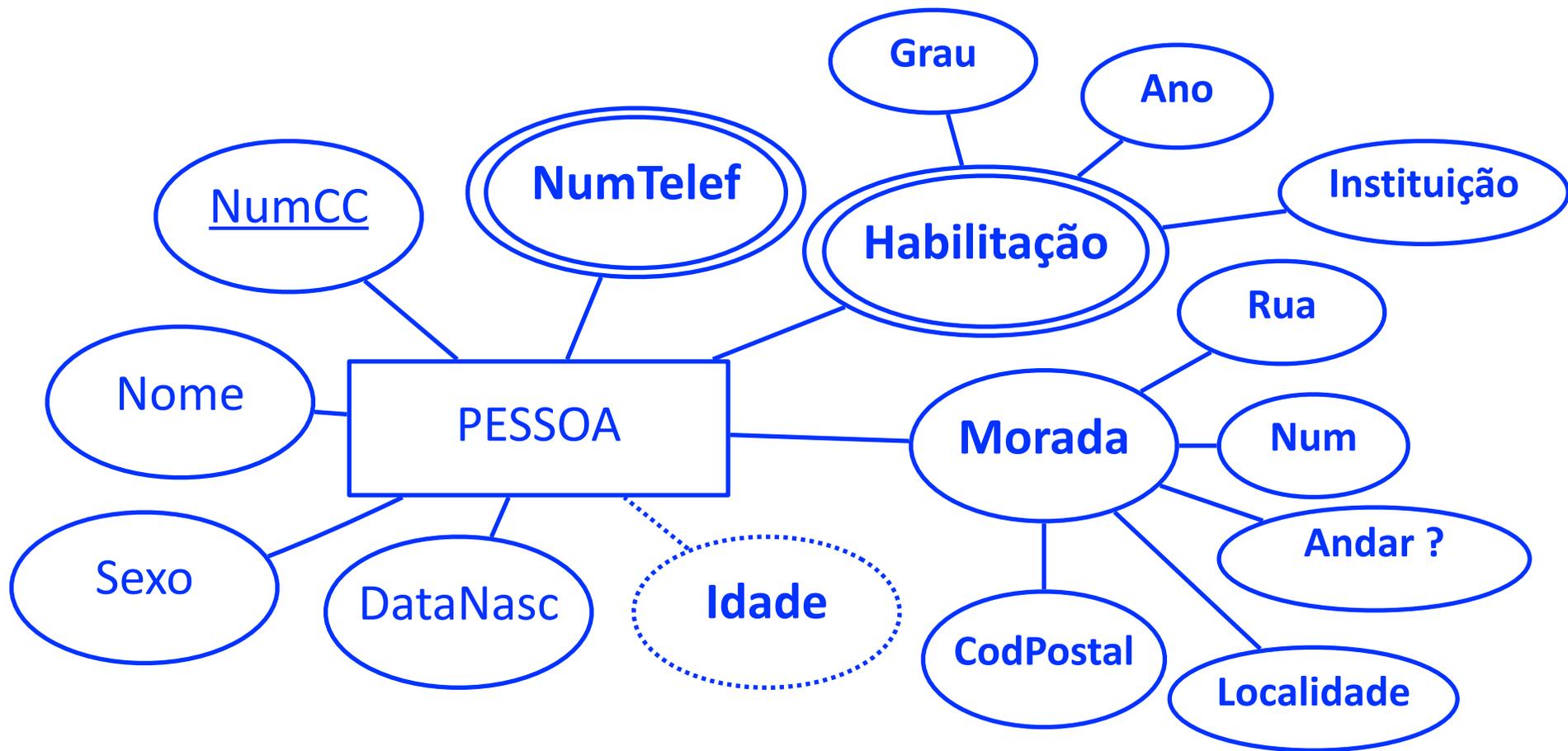
```
PESSOA(NumCC, Nome, Sexo, DataNasc, Idade,  
Morada(Rua,Num,Andar?,Localidade,CodPostal), {NumTelefone},  
{Habilitação(Grau, Ano, Instituição)})
```

podemos ter os seguintes exemplos de entidades:

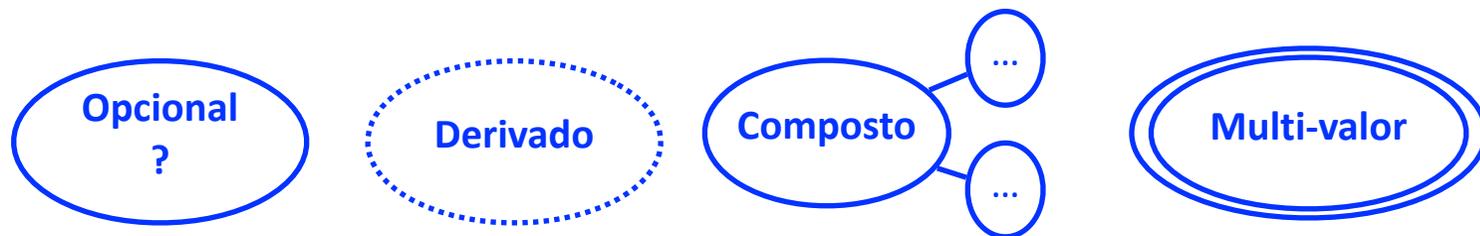
```
PESSOA1(19428771, 'José Silva', M, 17-11-2000, 18, ('Rua Fim do Mundo', 783,  
'R/C', 'Finisterra', '4444-555'), {987654321, 222333444}, { ('Ens. Secundário',  
2017, 'Escola Sec. Dr. Estranho-Amor') } }
```

```
PESSOA2(10447777, 'Maria Carvalho', F, 08-02-1976, 43, ('Rua das Bases de  
Dados', 1555, NULL, 'Vila Nova de Informática', '4000-123'), {933933933},  
{ ('Ens. Secundário', 1994, 'Escola Sec. Vila Nova de Informática'), ('Lic.  
Física', 1998, 'Fac. de Ciências Univ. Porto') }
```

Diagramas ER – notação adicional p/ atributos



Convenções



Relacionamentos

Relacionamento – forma geral

- Um **relacionamento** exprime uma interação conceptual entre entidades.
- Forma geral:
NOME(ENTIDADE-TIPO₁, ... , ENTIDADE-TIPO_N, Atributo₁, ..., Atributo_k)
- **NOME** é o **nome** do relacionamento, por convenção predicado que faz uso de verbos/adjectivos e qualificadores (ex. **FILHO_DE**, **TRABALHA_PARA**).
- **ENTIDADE-TIPO₁, ... , ENTIDADE-TIPO_N** são as **entidades-tipo participantes**.
- **N** : **grau do relacionamento** (número de participantes)
- Um relacionamento pode (opcionalmente) ter também associados **atributos Atributo₁, ..., Atributo_k**

Relacionamentos – exemplo simples

- Considerando o exemplo anterior de
`PESSOA(NumCC, Nome, Sexo, DataNasc, ...)`

podemos considerar possíveis relacionamentos entre pessoas, por exemplo:

`FILHO_DE(PESSOA, PESSOA)`

`CASADO_COM(PESSOA, PESSOA, DataCasamento)`

- Os relacionamentos acima dizem-se **binários** porque associam duas entidades (têm grau 2) e **recursivos** porque as entidades são do mesmo tipo.
- `CASADO_COM` exemplifica um relacionamento com atributos, no caso `DataCasamento`
- O nome do relacionamento exprime implicitamente papéis para as entidades em causa: filho e pai/mãe em `FILHO_DE` , conjugues em `CASADO_COM`.

Relacionamentos – exemplo mais rico

- Consideremos o universo de uma empresa em que temos as seguintes entidades-tipo e restrições (com **relacionamentos implícitos**):
 - **FUNCIONÁRIO** com os seguintes atributos: n.º de CC, nome, email opcional, data de nascimento, salário, horas semanais de dedicação a projectos, **funcionário supervisor opcional**, e **departamento definido a que pertence**.
 - **DEPARTAMENTO**: com nome único, **um funcionário gerente definido**, várias localizações possíveis associadas (moradas como anteriormente). **Um funcionário pode ser gestor de apenas um departamento**.
 - **PROJECTO**: com nome único, data de início, data de conclusão, **departamento definido que controla o projecto**, e **funcionário definido para director do projecto**.
 - **Um conjunto de funcionários pode trabalhar num projecto, sendo necessário identificar o número de horas semanais que cada funcionário dedica a este. Pode acontecer que um projecto não tenha funcionários associados além do director.**

Relacionamentos – exemplo mais rico (cont.)

Podemos tentar exprimir o universo anterior apenas usando entidades-tipo

```
FUNCIONÁRIO(NumCC, Nome, Email?, DataNasc, Salário,,  
HorasProj, Supervisor?, Departamento)
```

```
DEPARTAMENTO(Nome, Gerente,  
{ Localização(Rua, Num, Andar?, Localidade, CodPostal) } )
```

```
PROJECTO(Nome, DataInício, DataFim, Departamento, Director,  
{ Trabalho(Funcionário, Horas) } )
```

mas ao termos **referências** entre entidades, será em vez disso adequado o uso de relacionamentos. Que relacionamentos estão implícitos?

Relacionamentos – exemplo mais rico (cont.)

Reformulando, podemos ter como entidades-tipo

FUNCIONÁRIO(NumCC, Nome, Email?, DataNasc, Salário, HorasProj)

DEPARTAMENTO(Nome, { Localização})

PROJECTO(Nome, DataInício, DataFim)

e os seguintes relacionamentos

SUPERVISIONA(FUNCIONÁRIO, FUNCIONÁRIO)

TRABALHA_PARA(FUNCIONÁRIO, DEPARTAMENTO)

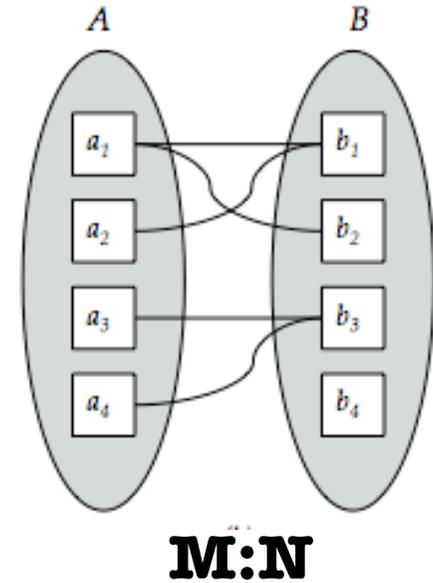
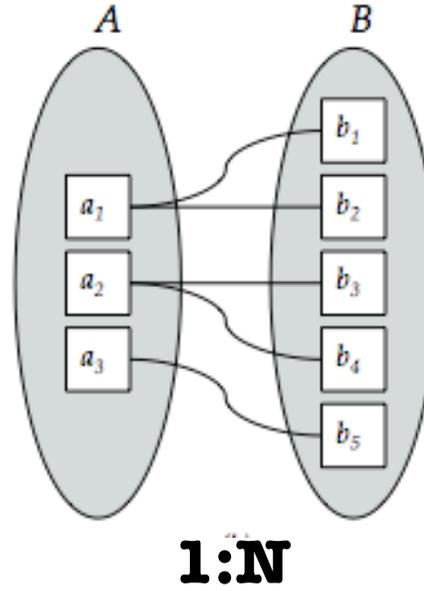
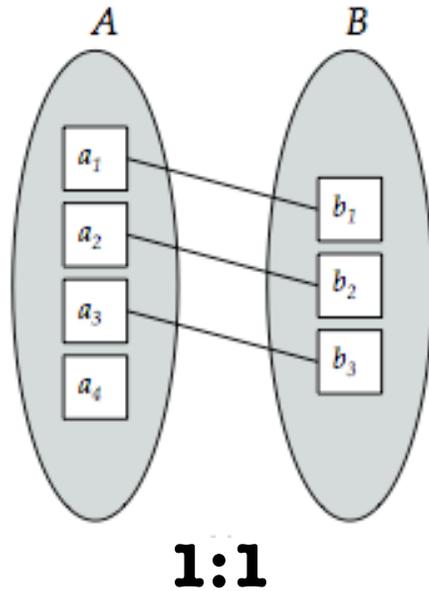
GERE(FUNCIONÁRIO, DEPARTAMENTO)

CONTROLA(DEPARTAMENTO, PROJECTO)

DIRIGE(FUNCIONÁRIO, PROJECTO)

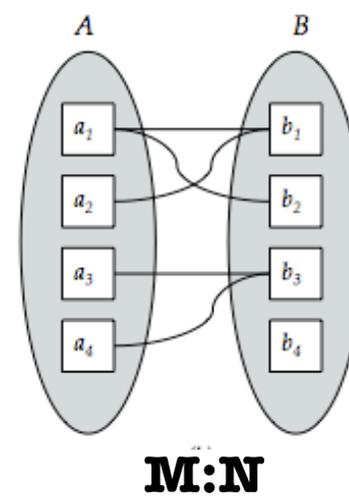
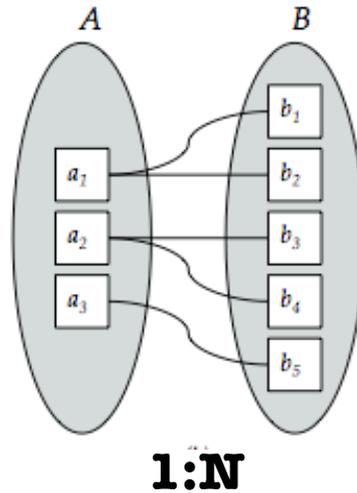
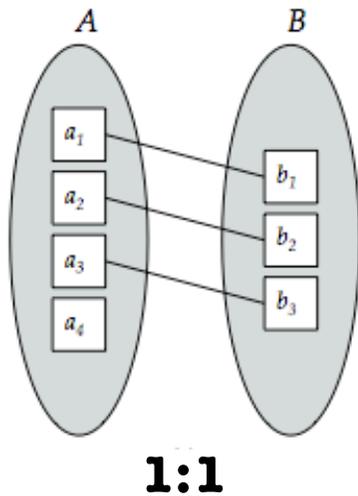
TRABALHA_EM(FUNCIONÁRIO, PROJECTO, Horas)

Restrições de cardinalidade



- A um relacionamento binário podemos ter associadas restrições de **cardinalidade**:
 - **1:1** – um-para-um
 - **1:N**, **N:1** – um-para-muitos, muitos-para-um
 - **M:N** – muitos-para-muitos

Restrições de cardinalidade (cont.)



- No exemplo da empresa

SUPERVISIONA(FUNCIONÁRIO, FUNCIONÁRIO) **1:N**

TRABALHA_PARA(FUNCIONÁRIO, DEPARTAMENTO) **N:1**

GERE(FUNCIONÁRIO, DEPARTAMENTO) **1:1**

CONTROLA(DEPARTAMENTO, PROJECTO) **1:N**

DIRIGE(FUNCIONÁRIO, PROJECTO) **1:N**

TRABALHA_EM(FUNCIONÁRIO, PROJECTO, Horas) **M:N**

Restrições de participação

- A um relacionamento podemos ter também associadas restrições de **participação**. A participação é **total** para uma entidade-tipo se a existência de uma entidade desse tipo obriga que a que participe no relacionamento, e apenas **parcial** caso contrário.
- Participação total – exemplos:
 - **TRABALHA_PARA(FUNCIÓNÁRIO, DEPARTAMENTO)** – total para entidades dos dois tipos envolvidas: todo o funcionário trabalha necessariamente para um departamento e todos os departamentos têm necessariamente funcionários.
 - **GERE(FUNCIÓNÁRIO, DEPARTAMENTO)** – total para um departamento: todo o departamento tem um gerente.
- Participação parcial – exemplos:
 - **SUPERVISIONA(FUNCIÓNÁRIO, FUNCIÓNÁRIO)** – nem todos os funcionários supervisionam outros, nem todos os funcionários têm supervisor.
 - **GERE(FUNCIÓNÁRIO, DEPARTAMENTO)** – parcial para um funcionário: nem todos os funcionários gerem um departamento.

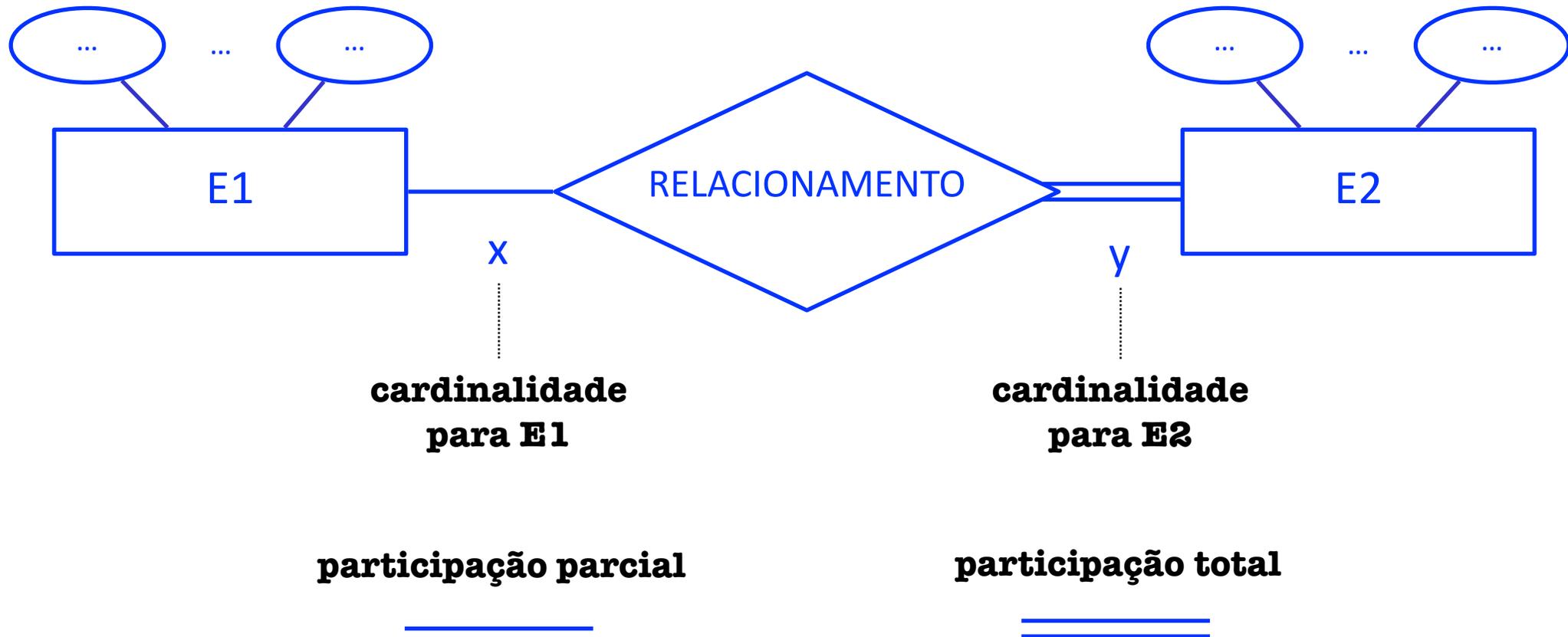
Restrições estruturais

- Uma **restrição estrutural** (**min,max**) para cada entidade-tipo num relacionamento **R(A,B)** pode resumir tanto as restrições de cardinalidade como de participação.
- Quando **min = 0** o nível de participação é **parcial**, e se **min > 0** a participação é **total**.
- Exemplos
 - **TRABALHA_PARA(FUNCIONÁRIO,DEPARTAMENTO)** – (1,N) para funcionário, (1,1) para departamento – cardinalidade N:1, participação total de ambas as entidades.
 - **GERE(FUNCIONÁRIO, DEPARTAMENTO)** – (0,1) para **FUNCIONÁRIO**, (1,1) para **DEPARTAMENTO** – cardinalidade 1:1, participação total apenas de **DEPARTAMENTO**.

Restrições – resumo do exemplo da empresa

Relacionamento	P	C	E
SUPERVISIONA (FUNCIONÁRIO, FUNCIONÁRIO)	parcial > parcial	1 : N	(0,1) > (0,N)
TRABALHA_PARA (FUNCIONÁRIO,DEPARTAMENTO)	total > total	N : 1	(1,N) > (1,1)
GERE (FUNCIONÁRIO, DEPARTAMENTO)	parcial > total	1 : 1	(0,1) > (1,1)
CONTROLA (DEPARTAMENTO, PROJECTO)	parcial > total	1 : N	(0,1) > (1,N)
DIRIGE (FUNCIONÁRIO, PROJECTO)	parcial > total	1 : N	(0,1) > (1,N)
TRABALHA_EM (FUNCIONÁRIO,PROJECTO, Horas)	parcial > parcial	M : N	(0,M) > (0,N)

Relacionamentos – diagramas ER



Relacionamentos – diagramas ER

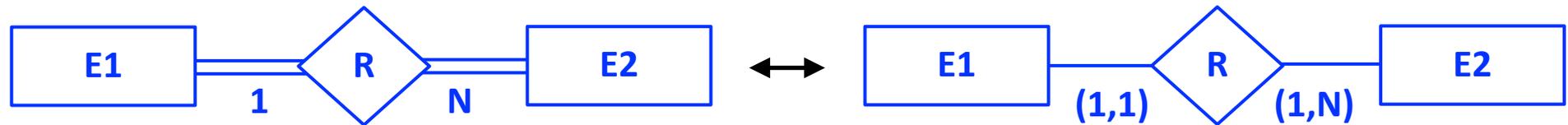
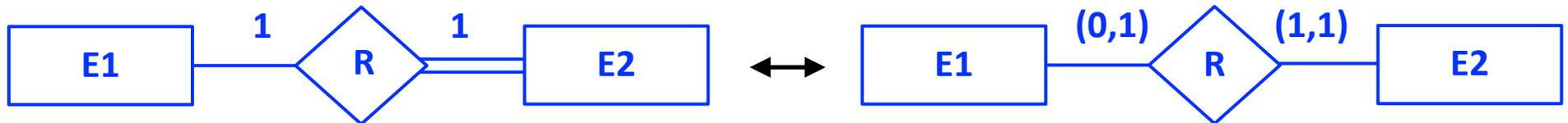
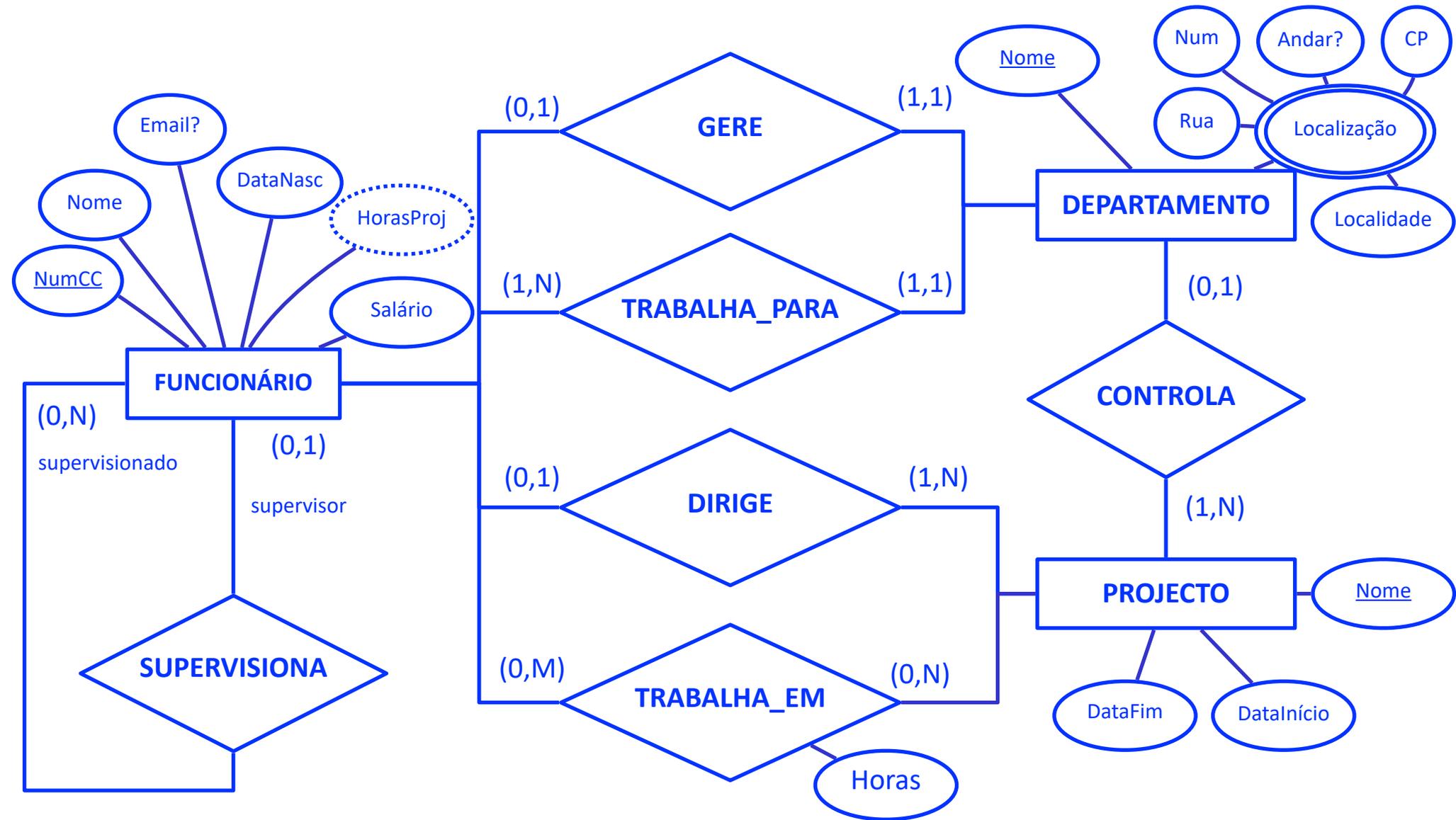


Diagrama ER – exemplo da empresa



Relacionamentos de grau superior a 2

- Até agora vimos apenas relacionamentos de grau 2 (binários), os mais comuns. Podemos no entanto ter relacionamentos de grau superior.
- No exemplo de uma empresa considere o fornecimento de produtos no contexto de um projecto em que:
 - estão envolvidas **3 entidades**: o **projecto**, o **produto** fornecido, e o **fornecedor** do produto;
 - um fornecimento é caracterizado por uma certa quantidade do produto em causa.

Relacionamentos de grau superior a 2 (cont.)

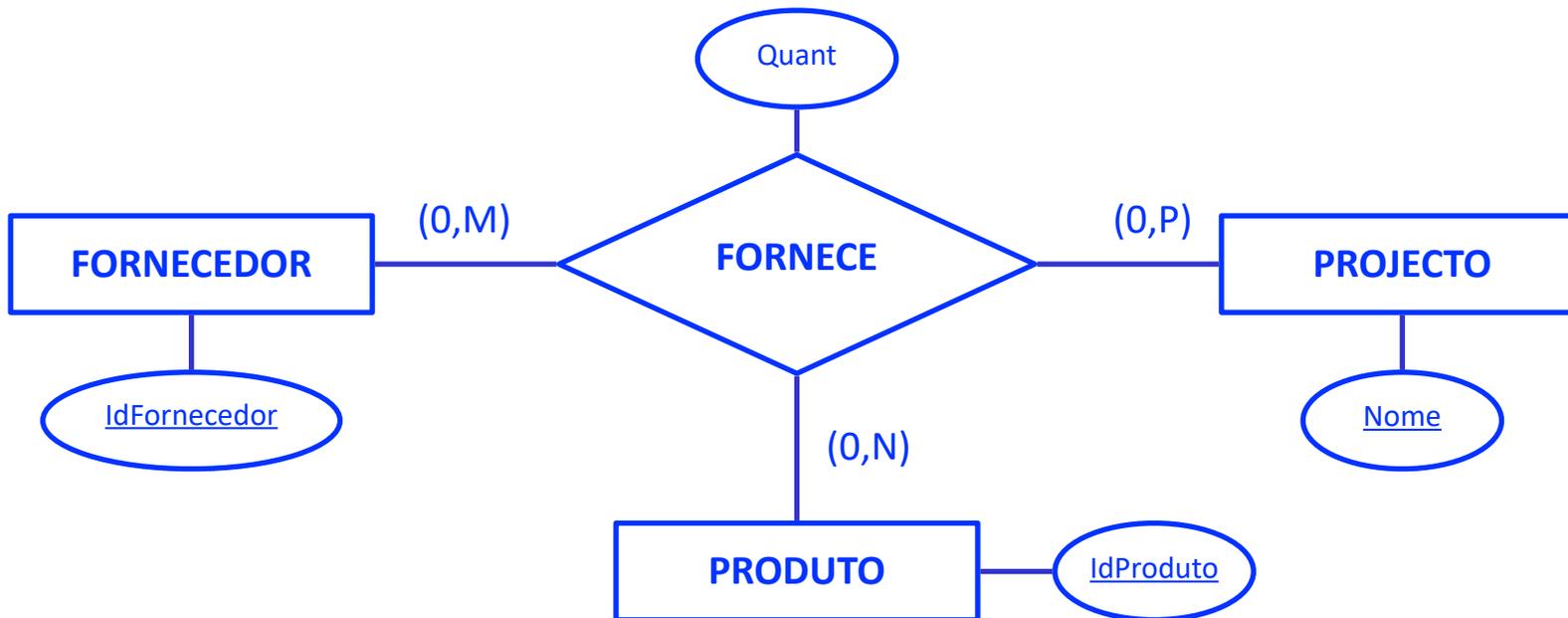
Para o exemplo anterior, assumindo que temos as entidade-tipos adicionais no modelo da empresa:

PRODUTO(IdProduto, ...)

FORNECEDOR(IdFornecedor, ...)

podemos ter a relação ternária:

FORNECE(FORNECEDOR, PRODUTO, PROJECTO, Quant)



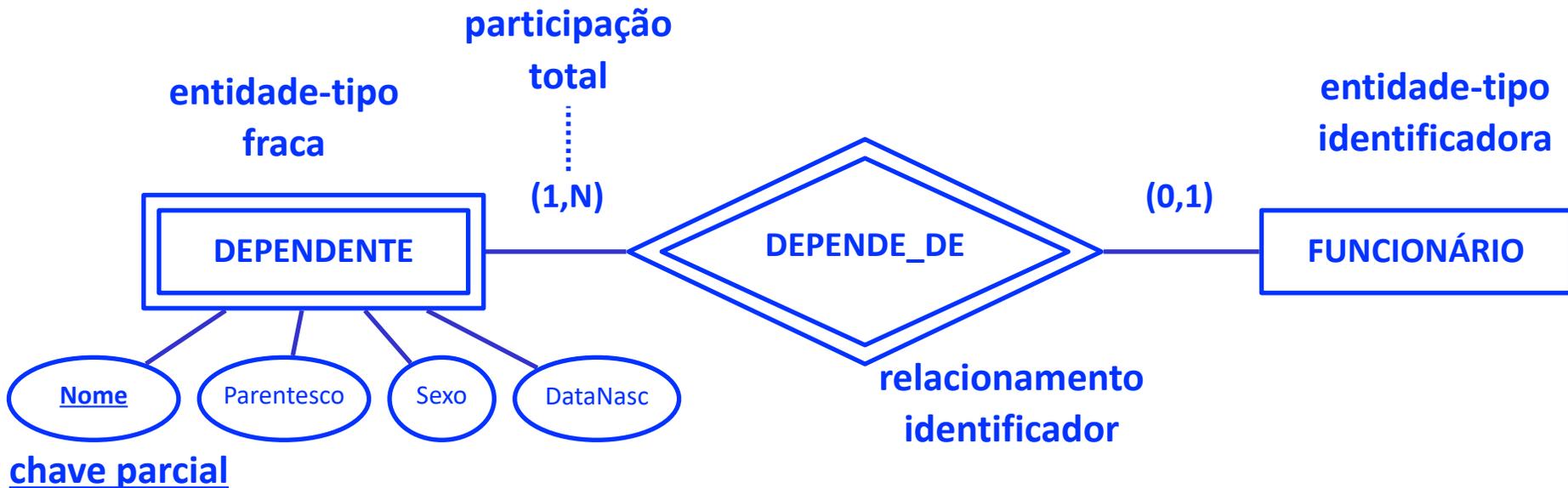
Entidades-tipo fracas



■ Entidades-tipo fracas:

- são assim chamadas as entidades-tipo sem chave
- uma entidade fraca (instância de entidade-tipo fraca) está necessariamente ligada a uma única instância de uma **entidade-tipo identificadora**, por meio de um **relacionamento identificado** com participação total da entidade-tipo fraca
- entidades fracas podem ter associadas uma **chave parcial**, que permita distinguir as instâncias da entidade-tipo fraca (apenas) no contexto da entidade identificadora

Entidades-tipo fracas – exemplo



- No contexto da BD empresa poderíamos considerar a noção de pessoa dependente (parte do núcleo familiar) de um funcionário.
- Entidades-tipo fraca ser vistas / modeladas alternativamente como um atributo complexo multi-valor da entidade-tipo identificadora, ou vice-versa (**HABILITAÇÕES** no exemplo **PESSOA** poderia alternativamente ser modelada como entidade-tipo fraca).

Modelo Relacional

Modelação lógica de BDs

■ Desenho lógico de BDs

- A modelação conceptual, como vimos para o modelo ER, define um modelo para a BD **independente** do tipo de base de dados.
- Um modelo lógico considera já o tipo de BD em causa, sem necessariamente ser dependente do SGBD.

■ Modelo Relacional

- Modelo lógico para **BDs relacionais**, baseadas no conceito de **relação**, também chamado de **tabela**.
- **ER (e EER) > Modelo relacional**: Entidades-tipo e relacionamentos no modelo ER (ou também no modelo EER, como veremos mais tarde) podem ser mapeados em tabelas no modelo relacional.
- **Modelo relacional > SQL**: um modelo relacional pode ser depois concretizado num SGBD baseado na linguagem SQL (como veremos também depois).

Conceitos base

Conceito de relação

Atributos
(colunas)



Registos
(linhas)



ALUNO			
NumMec	NumCC	Nome	Curso
798764544	12345678	João Pinto	LCC
345673451	17222303	Carlos Semedo	MIERSI
487563546	12021999	Maria Silva	LBIO
452212348	18392100	Pedro Costa	LMAT

← Nome

- Uma **relação** é um conjunto de tuplos, que pode ser representada na forma de **tabela**, com um **esquema** associado definido por um **nome** e **sequência de atributos**.
- Cada **tuplo**, também chamado **registo ou linha**, é definido por uma sequência de valores para atributos da tabela.
- Não existe uma ordem associada aos tuplos, a ordem em que poderão aparecer numa representação textual/visual é irrelevante.

Terminologia: relação vs. tabela

- Em termos estritos ...
 - Uma **relação** é um conjunto de tuplos não-ordenados no sentido matemático do termo.
 - Uma **tabela** é um modelo para a representação física de uma relação em um SGBD.
- Vamos no entanto usar o termo **tabela** para a noção de relação:
 - para evitar alguma confusão possível entre a noção de relacionamento (no modelo ER) e a de relação
 - ... e por ser também o termo usado para designar/implementar uma relação no contexto concreto de um de SGBDs relacionais baseados em SQL.

Definições e notação genérica

Uma **tabela**, denotada por $T(A_1, \dots, A_n)$ tem **nome** T e **atributos** A_1, \dots, A_n
por ex.

ALUNO(NumMec, NumCC, Nome, Curso)

A cada atributo A_i está associado um **domínio de valores** $\text{dom}(A_i)$. Os valores no domínio de um atributo são **atômicos** e podem incluir o valor especial **NULL** para denotar a ausência de valor definido, i.e., podemos ter $\text{NULL} \in \text{dom}(A_i)$. Contrariamente ao modelo ER, **no modelo relacional os atributos não podem ser compostos ou multi-valor.**

Um **registo** r de uma tabela $T(A_1, \dots, A_n)$ é um tuplo

$$r = (v_1, \dots, v_n)$$

tal que $v_i \in \text{dom}(A_i)$. Cada valor v_i pode ser denotado por $r[A_i]$.

Atributos chave

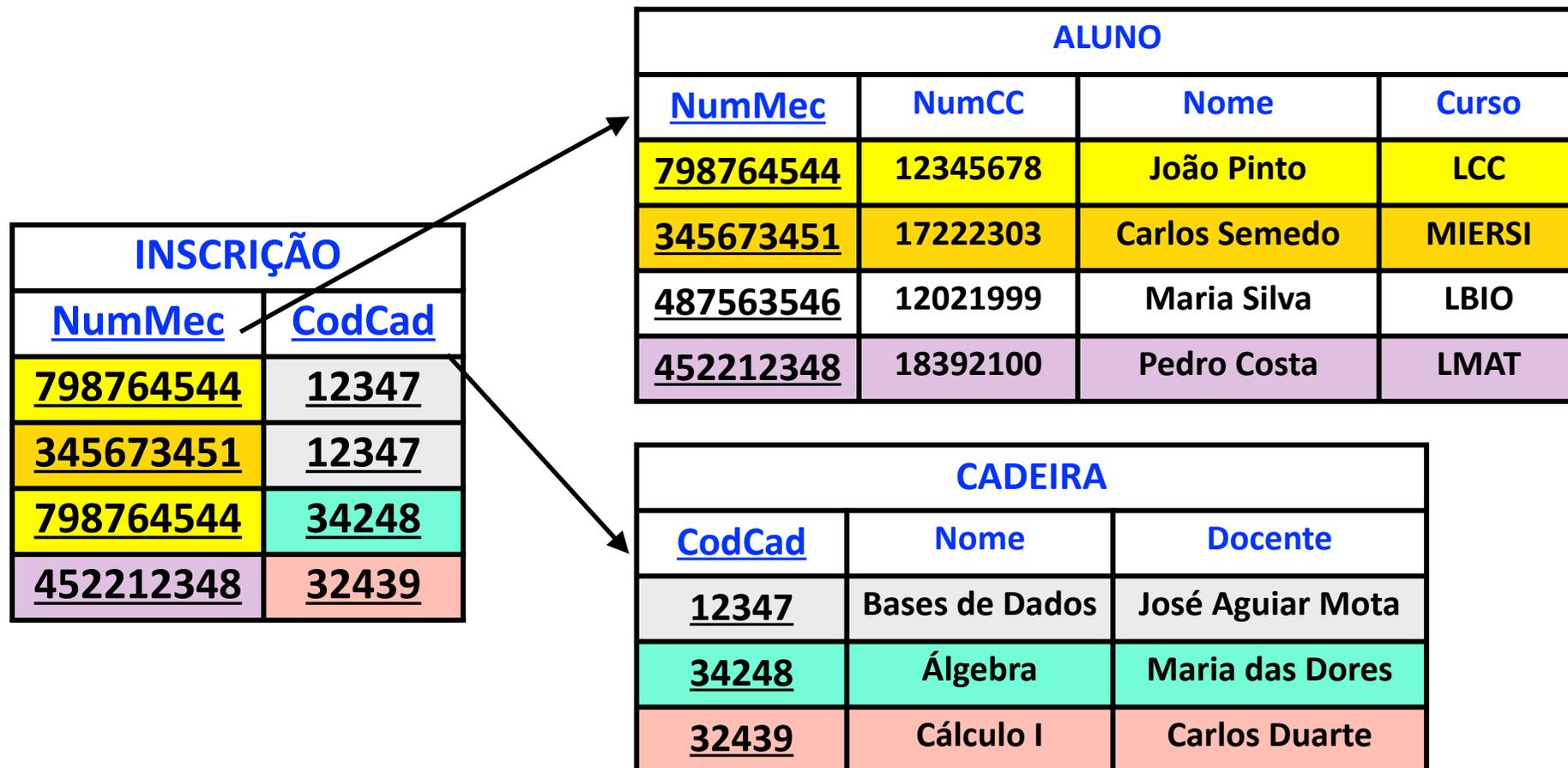
Analogamente ao modelo ER, uma **chave** para a tabela é um conjunto mínimo de atributos que permitem identificar de forma única um registo. Poderão haver várias chaves possíveis, sendo a chave mais adequada ao universo em causa escolhida como a chave primária (que aparece sublinhada).

Por exemplo

ALUNO(NumMec, NumCC, Nome, Curso)

diz-nos que a chave primária de **ALUNO** é formada apenas pelo atributo **NumMec**. Uma chave para **ALUNO** também definida apenas pelo atributo **NumCC**, mas seria menos adequada do que **NumMec** como chave primária.

Chaves externas



- **Chaves externas** numa tabela são atributos que se referem a chaves primárias de outras tabelas.

[Q: consegue vislumbrar o modelo ER correspondente a esta BD?]

Representação de esquema

ALUNO(NumMec, NumCC, Nome, Curso)



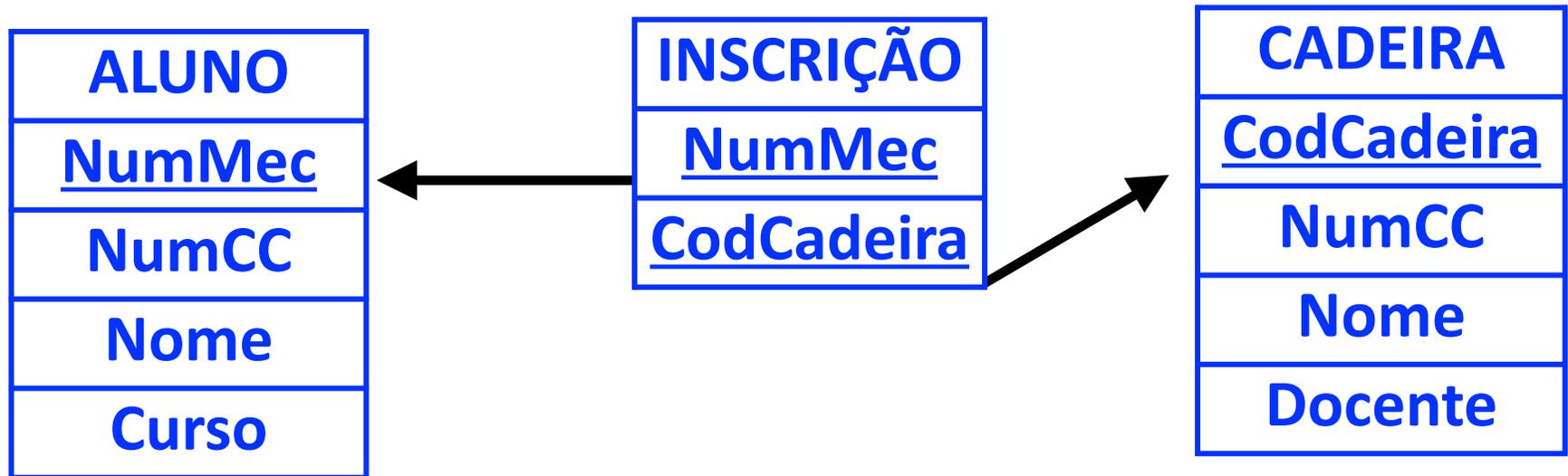
INSCRIÇÃO(NumMec, CodCadeira)



CADEIRA(CodCadeira, Nome, Docente)

- **INSCRIÇÃO** tem 2 chaves externas: **NumMec** refere-se à chave primária de **ALUNO** e **CodCadeira** à chave primária de **CADEIRA**. Os dois atributos definem também a chave primária de **INSCRIÇÃO** neste caso (no caso geral a chave externa não precisa de fazer parte da chave primária).
- **Na representação visual de um esquema relacional, representam-se as referências usando uma ligação (seta ou arco dirigido) entre chave externa e chave primária.**

Representação de esquema – alternativa



- **Alternativa:** poderá ser mais conveniente desenhar o esquema representando os atributos na vertical.

Dados consistentes?

INSCRIÇÃO	
<u>NumMec</u>	<u>CodCad</u>
111111111	<u>12347</u>
<u>345673451</u>	<u>12347</u>
<u>798764544</u>	<u>34248</u>
<u>452212348</u>	<u>12347</u>

?

ALUNO			
<u>NumMec</u>	<u>NumCC</u>	Nome	Curso
<u>798764544</u>	12345678	João Pinto	LCC
<u>345673451</u>	17222303	Carlos Semedo	MIERSI
<u>487563546</u>	12021999	Maria Silva	LBIO
<u>452212348</u>	ABCDEF	Pedro Costa	LMAT

?

CADEIRA		
<u>CodCad</u>	Nome	Docente
<u>12347</u>	Bases de Dados	José Aguiar Mota
<u>34248</u>	Álgebra	Maria das Dores
<u>34248</u>	Cálculo I	Carlos Duarte
<u>NULL</u>	Programação	Alberto Manuel

?

?

Esta BD não faz sentido. Quais são os **problemas**? O que está em causa em cada caso?

Restrições de integridade numa BD relacional

■ Integridade de domínio

- O valor de um atributo faz parte do domínio do atributo.

■ Integridade de chave

- Dois registos da mesma tabela não podem ter valores iguais para uma chave, em particular para a chave primária.

■ Integridade referencial

- O valor de um atributo que seja chave externa deve referir-se a uma chave primária da tabela a que a chave externa se refere.

■ Integridade de entidade

- O valor da chave primária não pode ser **NULL** (sob pena de não conseguirmos identificar registos).

Violação de restrições de integridade

X integridade referencial ?

INSCRIÇÃO	
<u>NumMec</u>	<u>CodCad</u>
<u>111111111</u>	<u>12347</u>
<u>345673451</u>	<u>12347</u>
<u>798764544</u>	<u>34248</u>
<u>452212348</u>	<u>12347</u>

ALUNO			
<u>NumMec</u>	<u>NumCC</u>	Nome	Curso
<u>798764544</u>	12345678	João Pinto	LCC
<u>345673451</u>	17222303	Carlos Semedo	MIERSI
<u>487563546</u>	12021999	Maria Silva	LBIO
<u>452212348</u>	ABCDEF	Pedro Costa	LMAT

? integridade de domínio **X**

X integridade de chave ?

X integridade de entidade ?

CADEIRA		
<u>CodCad</u>	Nome	Docente
<u>12347</u>	Bases de Dados	José Aguiar Mota
<u>34248</u>	Álgebra	Maria das Dores
<u>34248</u>	Cálculo I	Carlos Duarte
<u>NULL</u>	Programação	Alberto Manuel

Estado e operações sobre uma base de dados

- **Esquema da BD** = { **Esquema de tabelas** }
- **Estado da BD** = { **Conteúdo das tabelas** }
- **O estado da BD é mutável** sendo normal considerar as seguintes operações nucleares:
 - **INSERE(T, r)** : insere novo registo **r** na tabela **T**
 - **REMOVE(T,k)**: remove registo (que já exista) com chave primária **k** de **T**.
 - **ACTUALIZA(T,k,r)**: actualiza registo com chave primária **k** em **T** pelo registo **r** com a mesma chave primária (pode ser vista como uma remoção seguida de uma inserção, mas com efeito atómico).
- **Nota**: a estas operações irão corresponder às formas mais simples dos comandos SQL **INSERT**, **DELETE**, e **UPDATE**.

Operações e restrições de integridade

- As operações consideradas podem ser inválidas se violarem restrição de integridade:
 - **INSERE(T, r)** : insere novo registo **r** na tabela **T** – pode violar qualquer um dos tipos de restrições (domínio, entidade, chave, referencial).
 - **REMOVE(T,k)**: remove registo (que já exista) com chave primária **k** de **T** – pode violar a integridade referencial se existir uma referência a **k** por via de uma chave externa.
 - **ACTUALIZA(T,k,r)**: actualiza registo com chave primária **k** em **T** pelo registo **r** com a mesma chave primária – pode violar qualquer um dos tipos de restrição.

Exemplos de operações inválidas

INSCRIÇÃO	
<u>NumMec</u>	<u>CodCad</u>
798764544	12347
345673451	12347
798764544	34248
452212348	32439

ALUNO			
<u>NumMec</u>	<u>NumCC</u>	Nome	Curso
798764544	12345678	João Pinto	LCC
345673451	17222303	Carlos Semedo	MIERSI
487563546	12021999	Maria Silva	LBIO
452212348	18392100	Pedro Costa	LMAT

CADEIRA		
<u>CodCad</u>	Nome	Docente
12347	Bases de Dados	José Aguiar Mota
34248	Álgebra	Maria das Dores
32439	Cálculo I	Carlos Duarte

- $INSERE(ALUNO, r)$ tal que $r[NumMec] = 798764544$ violaria integridade de chave p/ $ALUNO$.
- $REMOVE(CADEIRA, 32439)$ violaria integridade referencial p/ $INSCRIÇÃO.CodCad$.
- $INSERE(INSCRIÇÃO, r)$ com $r[NumMec] = 999999$ violaria integridade referencial p/ $INSCRIÇÃO.NumMec$.
- $ACTUALIZA(ALUNO, 798764544, r)$ com $r[NumCC] = 'ABCDE'$ violaria a integridade de domínio p/ $ALUNO.NumCC$.
- $INSERE(ALUNO, r)$ com $r[NumMec] = NULL$ violaria a integridade de entidade p/ $ALUNO$.

SGBDs e restrições de integridade

- Um SGBD deverá rejeitar uma operação que viole restrições de integridade, assinalando o erro.
- SGBDs maduros normalmente suportam todos os tipos de restrições de integridade que consideramos (domínio, entidade, chave, referencial).
- Há no entanto exceções que se prendem com escolhas feitas p/ implementação de SGBDs, tipicamente por questões de complexidade de implementação/contexto de uso/desempenho. Por exemplo:
 - SQLite não valida restrições de domínio.
 - Versões antigas de MySQL não tinham suporte p/integridade referencial.

Conversão do Modelo ER para o Modelo relacional

Modelo ER > Modelo Relacional

- Perspectiva geral
 - Entidades-tipo e relacionamentos são convertidos em tabelas.
- Mapeamento de entidades-tipo em tabelas
 - Passa pelo mapeamento directo de atributos, excepto no caso de atributos multi-valor em que precisamos de recorrer a “tabelas auxiliares”.
- Mapeamento de relacionamentos – conforme a cardinalidade e participação das entidades-tipos de um relacionamentos:
 - Relacionamentos podem resultar em uma tabela nova para o relacionamento ou na adição de atributos às tabelas correspondentes a entidade-tipo.
 - Chaves primárias, externas são definidas em cada caso apropriadamente.
 - Atributos de relacionamentos têm de ser também considerados no mapeamento.

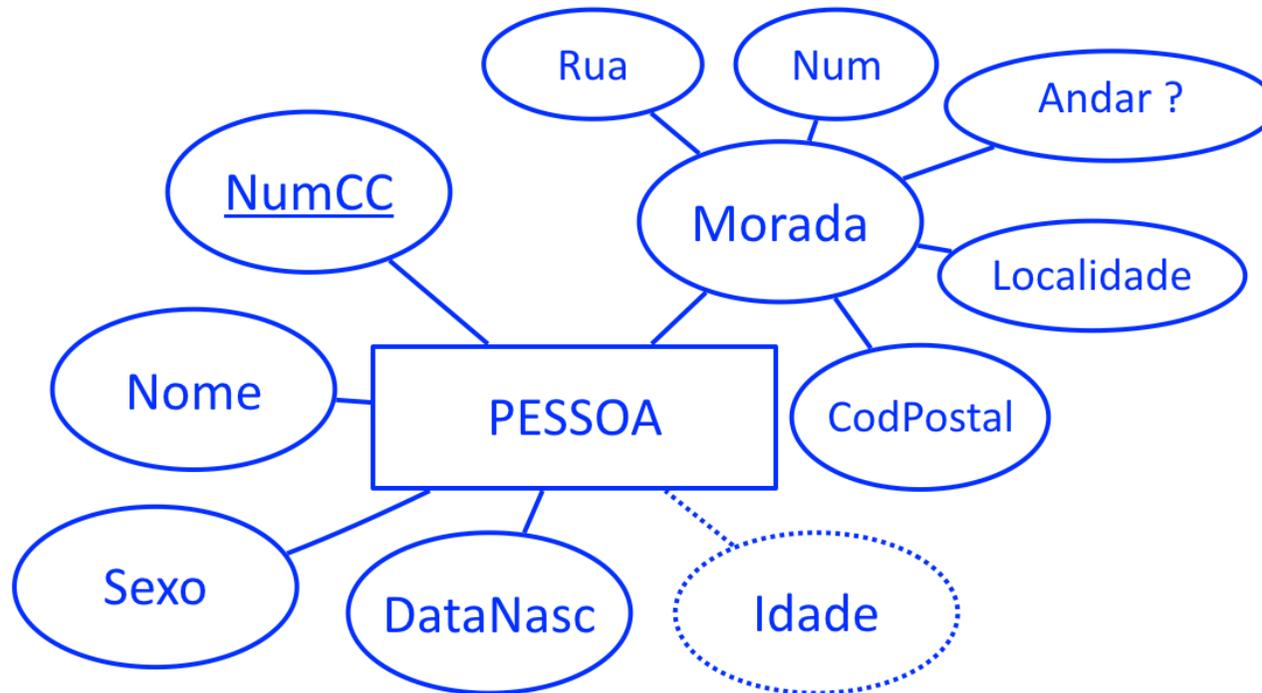
Mapeamento de entidades-tipo

■ Entidade-tipo $E \ggg$ tabela T

- Nome de $E \ggg$ nome de T
- Chave de $E \ggg$ chave primária de T
- Atributos derivados de E não são mapeados.
- Atributo simples de $E \ggg$ atributo simples de T
- Atributos simples de atributo composto de $E \ggg$ atributos de T
- Atributos opcionais levam simplesmente à inclusão de **NULL** no domínio do atributos.
- Atributo multi-valor **MV** de $E \ggg$ tabela auxiliar **MV** com chave externa referenciando a chave primária de T . Chave externa e restantes atributos definem a chave de **MV**.

Entidades-tipo – atributos de valor único

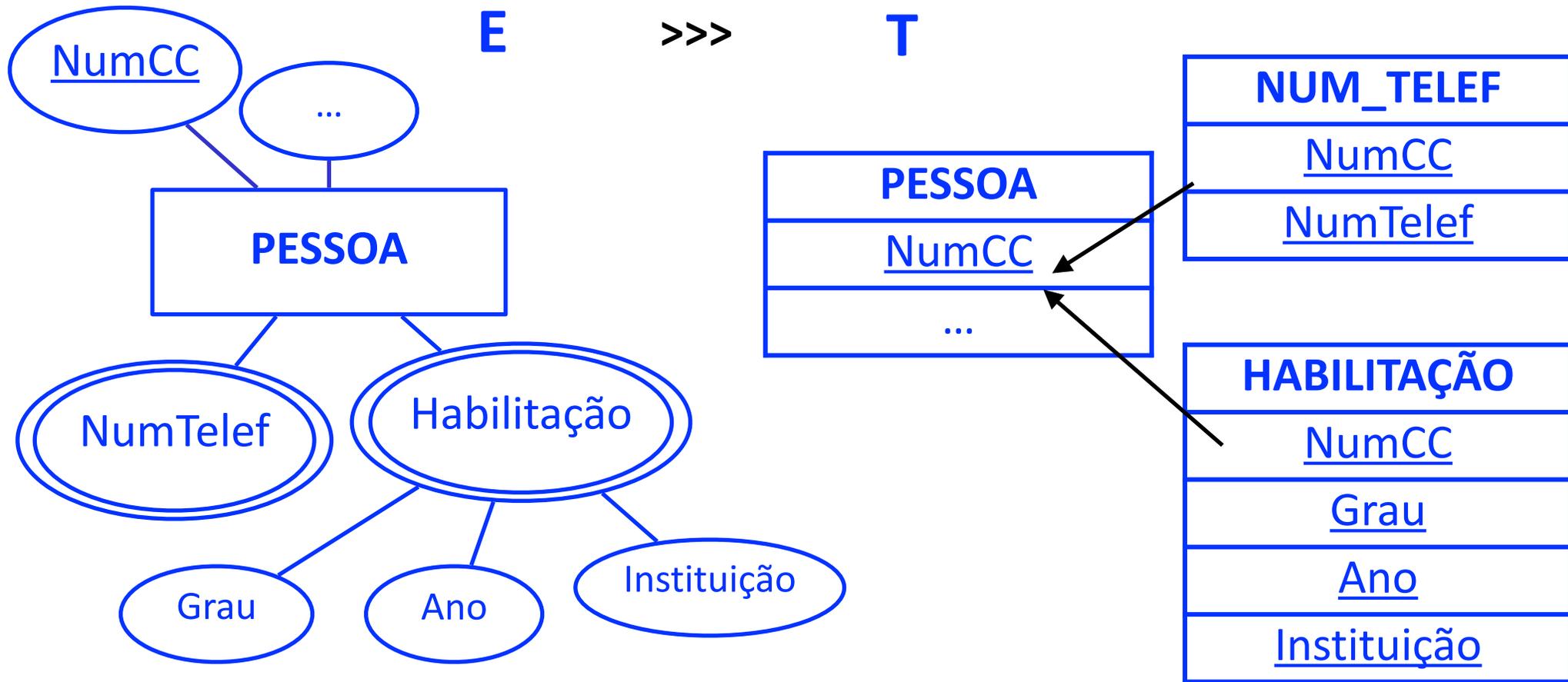
E >>> T



PESSOA
<u>NumCC</u>
Nome
Sexo
DataNasc
MRua
MNum
MAndar
MLocalidade
MCodPostal

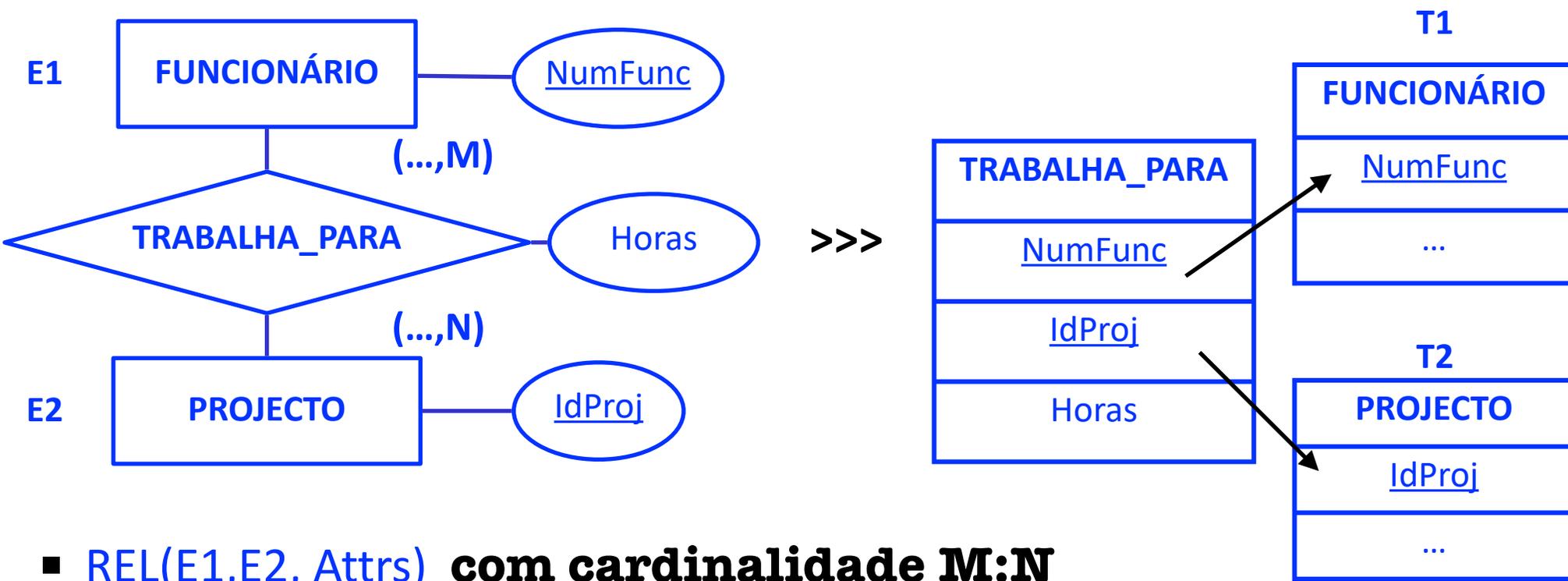
Nota: não usaremos notação especial para atributos opcionais no modelo relacional. Está implícito no entanto que deveremos ter $NULL \in \text{dom}(M\text{Andar})$

Entidades-tipo – atributos multivalor



É criada nova tabela “auxiliar” com chave externa referenciando a chave primária de **T**. Chave externa e restantes atributos definem a chave da tabela auxiliar.

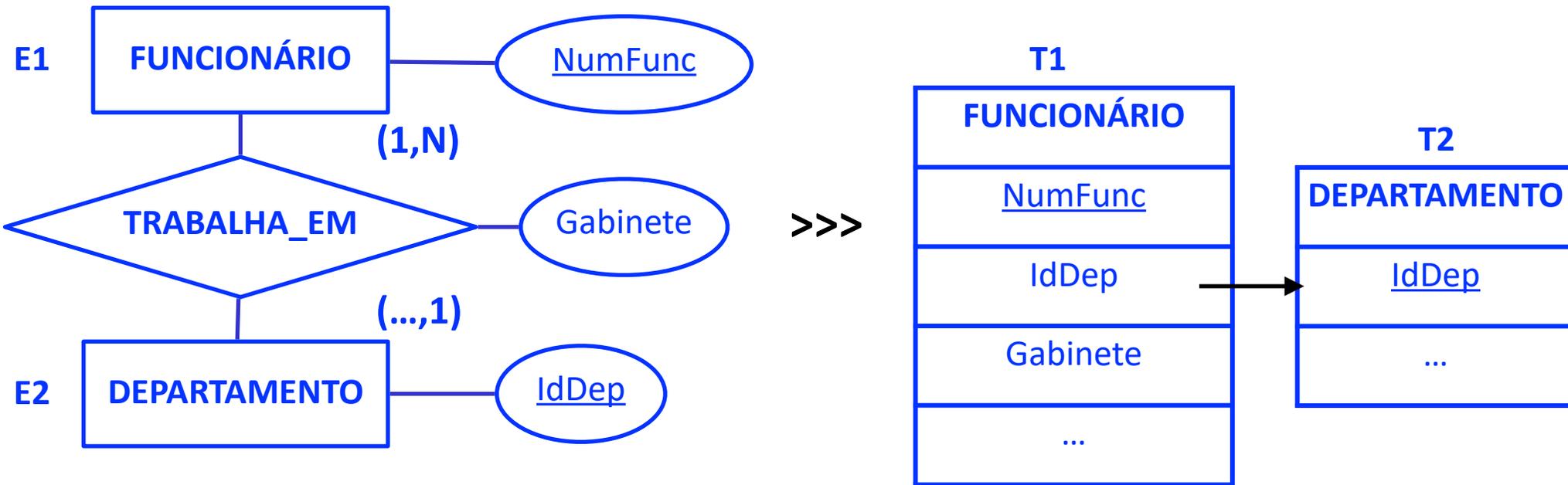
Mapeamento de relacionamentos M:N



■ REL(E1,E2, Attrs) com cardinalidade M:N

- Cria-se tabela de “referência-cruzadas” específica a REL.
- Chave primária de REL = Chave primária de T1 + Chave primária de T2 (ambas chaves externas)
- Attrs mapeados na tabela para REL

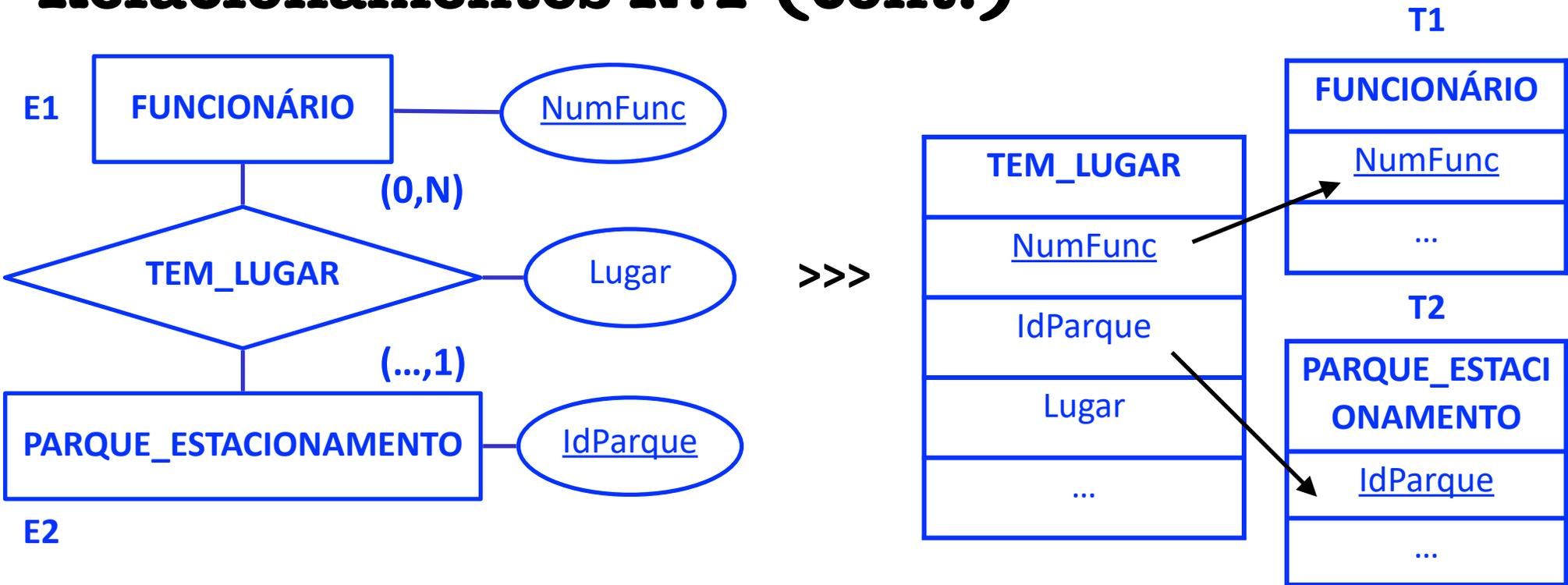
Relacionamentos N:1



■ REL(E1,E2, Attrs) com cardinalidade N:1 e participação total de E1

- chave externa definida em T1 para a chave primária de T2
- Attrs mapeados em T1

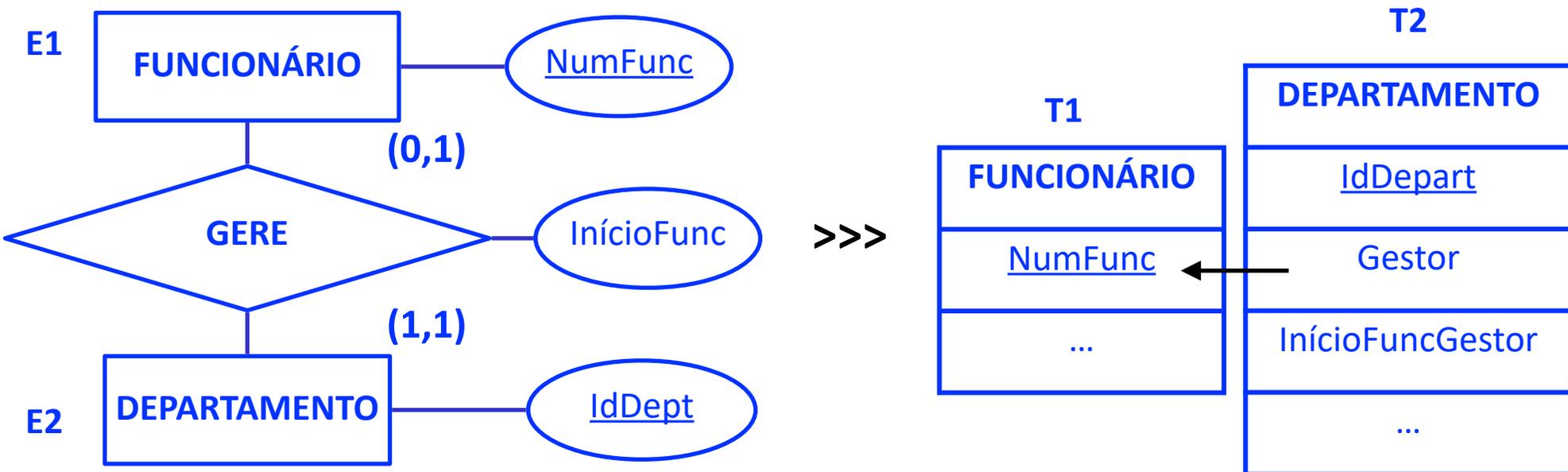
Relacionamentos N:1 (cont.)



■ REL(E1,E2, Attrs) c/cardinalidade N:1 e participação parcial de E1

- Cria-se tabela específica a REL incorporando Attrs
- Chave primária de T1 é chave primária na tabela e chave externa.
- Chave primária de T2 é chave externa (apenas).
- Estratégia anterior também válida, mas onerosa se apenas algumas instâncias se envolverem em REL levando a demasiados valor NULL.

Relacionamentos 1:1

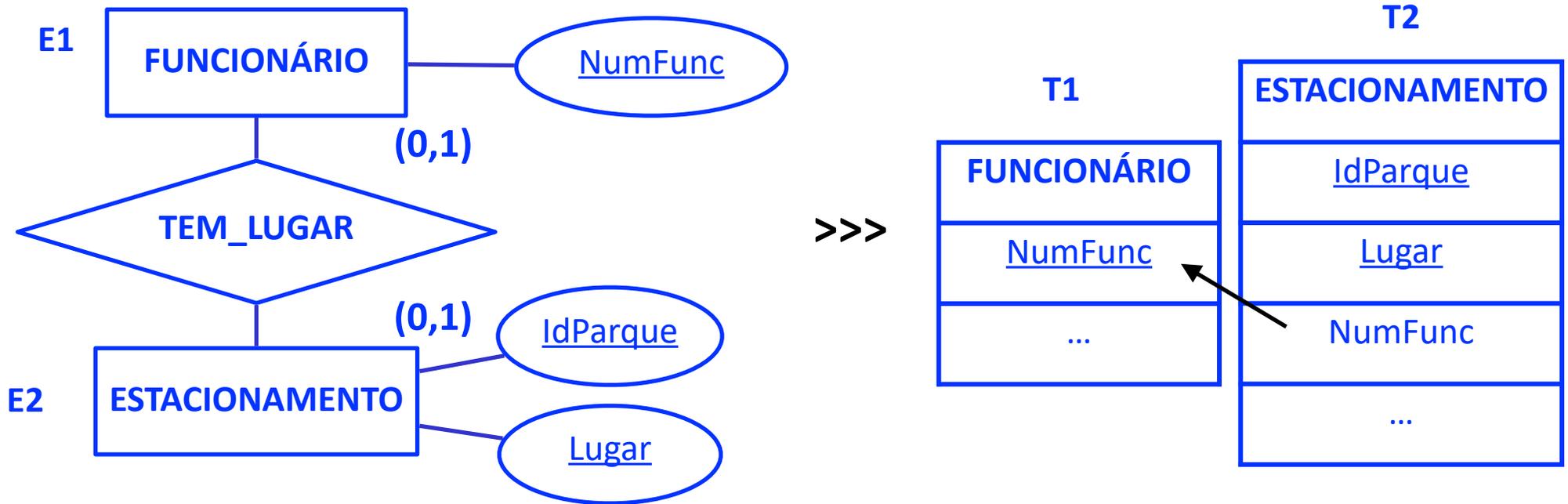


- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação parcial de E1, participação total de E2**

- Chave primária de T1 é adicionada como chave externa a T2

- Attrs mapeados em T2

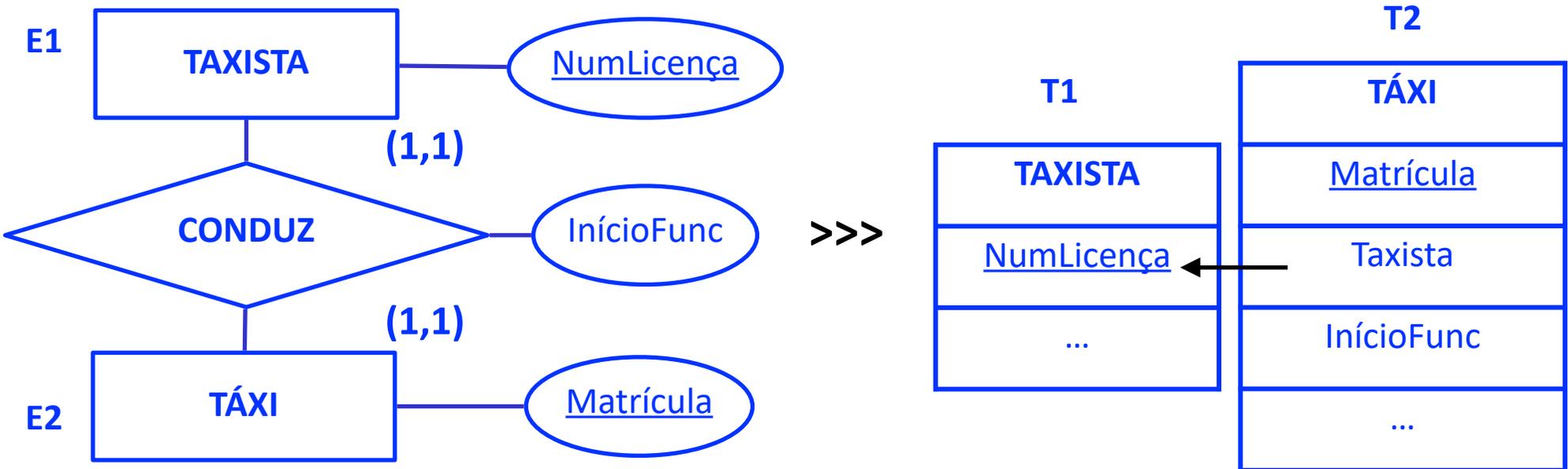
Relacionamentos 1:1 (cont.)



- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação parcial de ambas as entidades**

- Podemos usar estratégia semelhante à anterior. Em alternativa, uma tabela de “referências-cruzadas” poderá ser preferível se houverem poucas instâncias relacionadas.

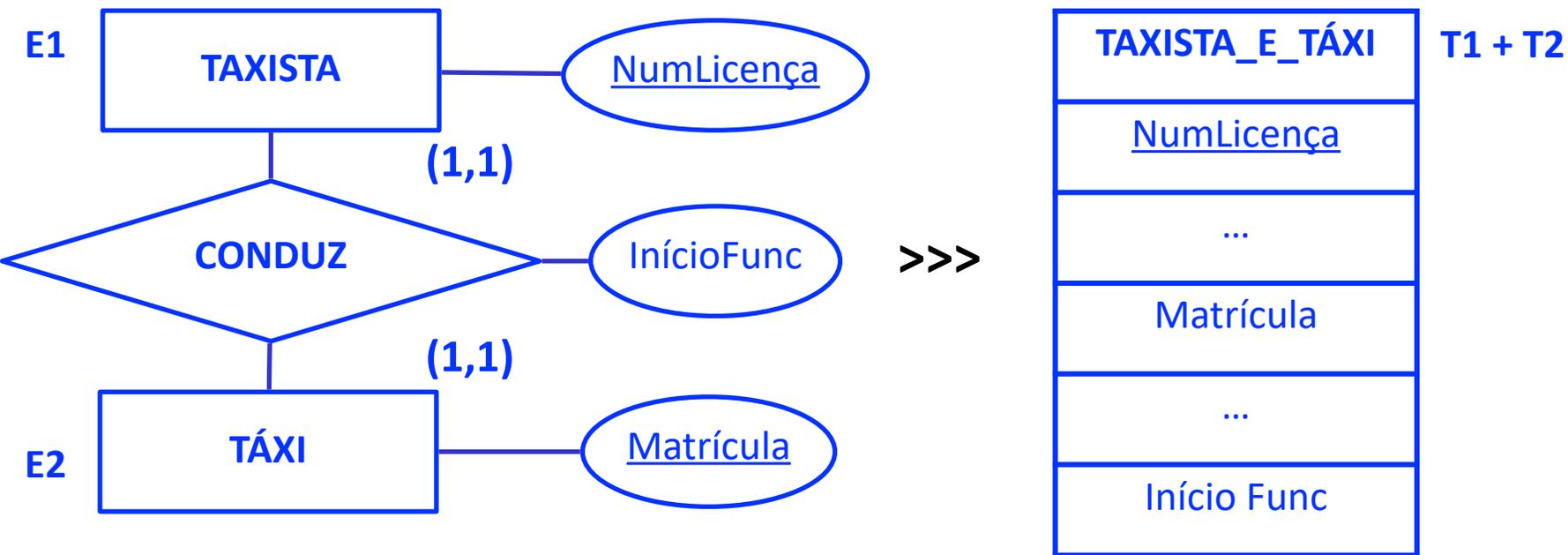
Relacionamentos 1:1 (cont.)



- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação total de ambas as entidades**

- **Opção 1:** Chave primária de T1 é adicionada como chave externa a T2 e Attrs mapeados em T2, ou vice-versa.

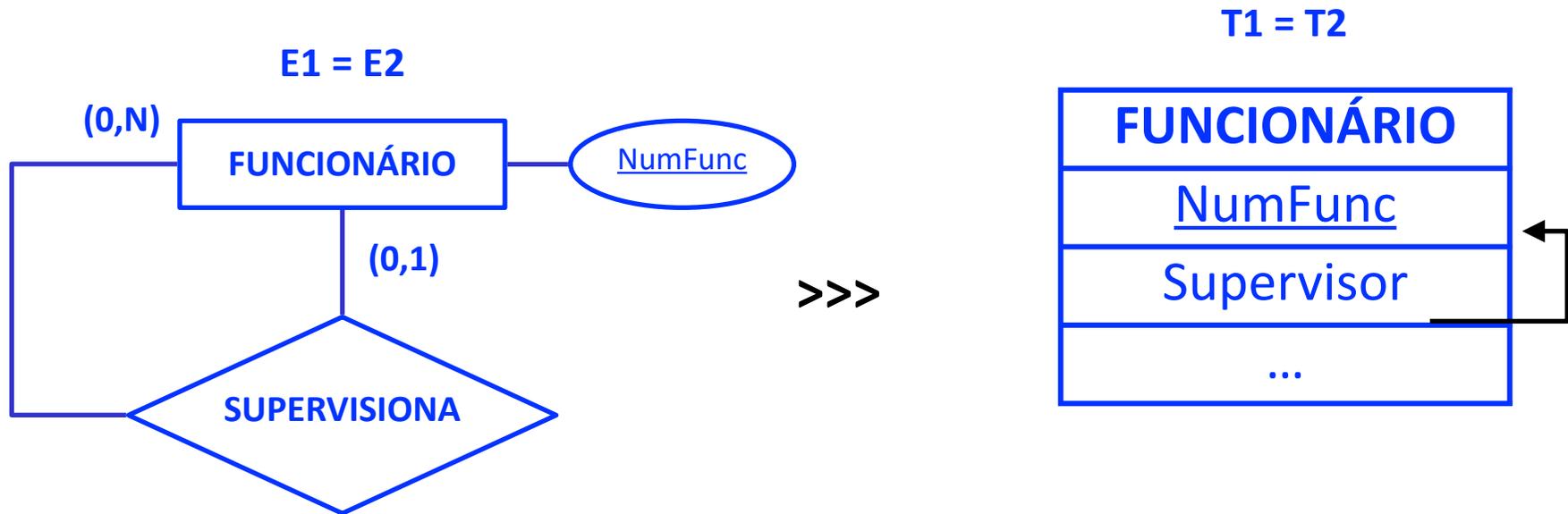
Relacionamentos 1:1 (cont.)



- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação total de ambas as entidades**

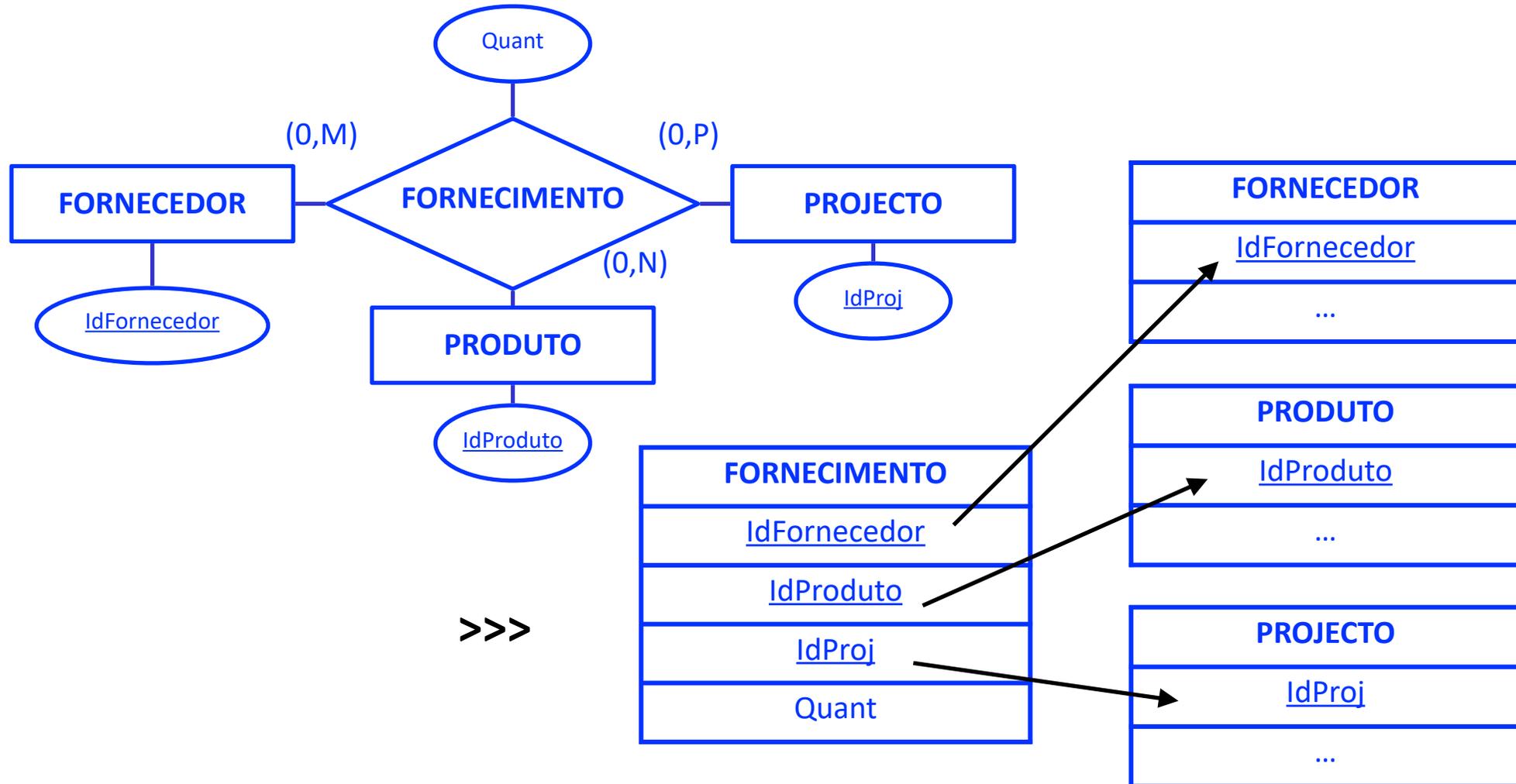
- **Opção 2:** Usar uma tabela só, abrangendo atributos de ambas as entidades e do relacionamento, definindo a chave de uma das entidades como chave primária. A opção é normalmente inadequada, pois define uma “multi-entidade híbrida”, que será mais sensível a mudanças no esquema conceptual e/ou relacional da BD.

Outros casos – relacionamentos recursivos



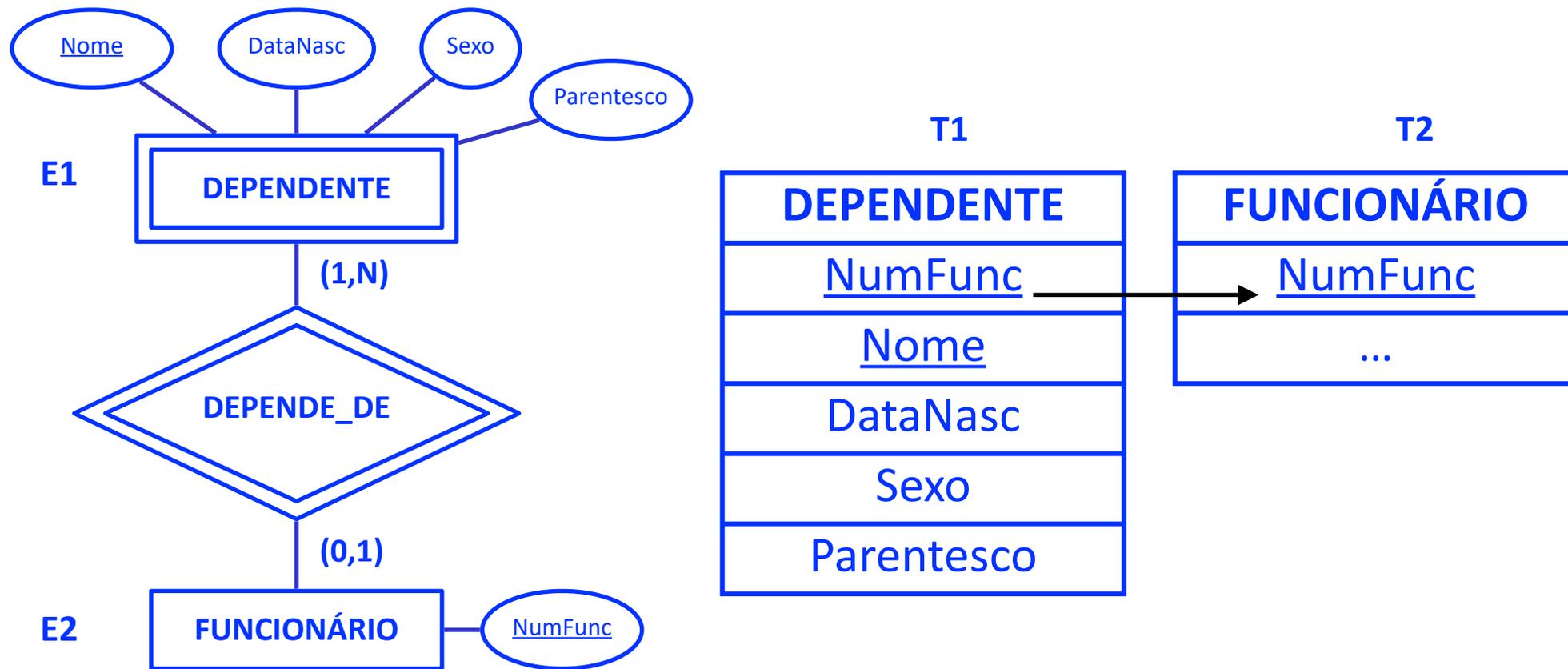
- Tratados como outros casos, mas afectando apenas uma tabela.
- Opção de tabela de “referência cruzada” também possível como em outros casos.

Outros casos – relacionamentos com grau > 2



- Cria-se tabela de referências cruzadas, com chave primária formada por chaves externas às tabelas correspondentes às entidades-tipo envolvidas.

Outros casos – entidades-tipo fracas



- Tratados de forma análoga a relações N:1 com participação total de E1
- Chave primária de T1 = chave parcial da entidade-tipo fraca + chave externa para T2

Caso de Estudo: BD para Biblioteca

Caso de estudo: BD para biblioteca

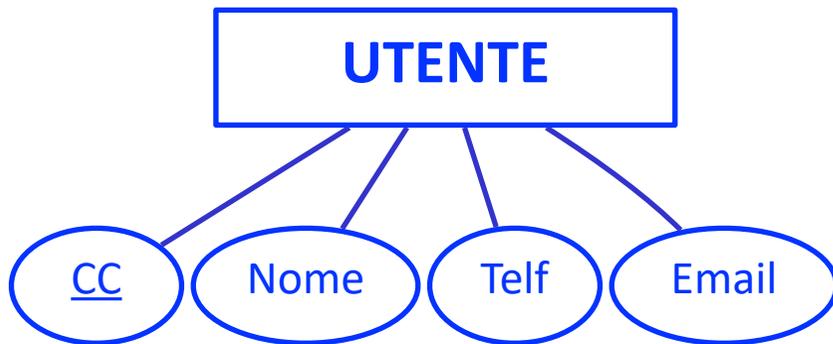
- Para pôr em perspectiva a modelação de uma BD, vamos considerar o universo de uma biblioteca como caso de estudo.
- Passo a passo, vamos considerar as seguintes fases e rever os conceitos associados:
 - Análise de requisitos e sua modelação ER
 - Mapeamento do modelo ER em modelo relacional
- Requisitos considerados para:
 - Utentes, livros, cópias de livros, empréstimo de livros, organização da arrumação de livros.

Utentes

■ Requisito:

- Cada utente da biblioteca é identificado de forma única pelo seu n.º CC, e tem também associado um nome, n.º telefónico e endereço email.

Entidade-tipo



Tabela

UTENTE
<u>CC</u>
Nome
Telf
Email

Utentes (cont.)

UTENTE

<u>CC</u>	Nome	Telf	Email
10583212	João Pinto	913 448 748	azulibranco@fcp.pt
12447555	Carlos Semedo	223 774 327	carlos@xpto.com
16348500	Maria Silva	939 939 939	maria @ silva.com
11983516	Pedro Costa	384 388 291	pc12345@xpto.com

Livros e cópias

■ Requisitos:

- Cada **livro** é caracterizado por código ISBN único, um título, **um ou mais nomes de autores**, ano de edição, editora, e quantidade de **cópias** possuídas pela biblioteca.
- Cada **cópia de um livro** tem de estar registada para efeitos de arrumação e empréstimos, e **está numerada (1,2,3, ...)** de forma única (apenas) para o mesmo livro.

Livros e cópias (cont.)

■ Modelação ER:

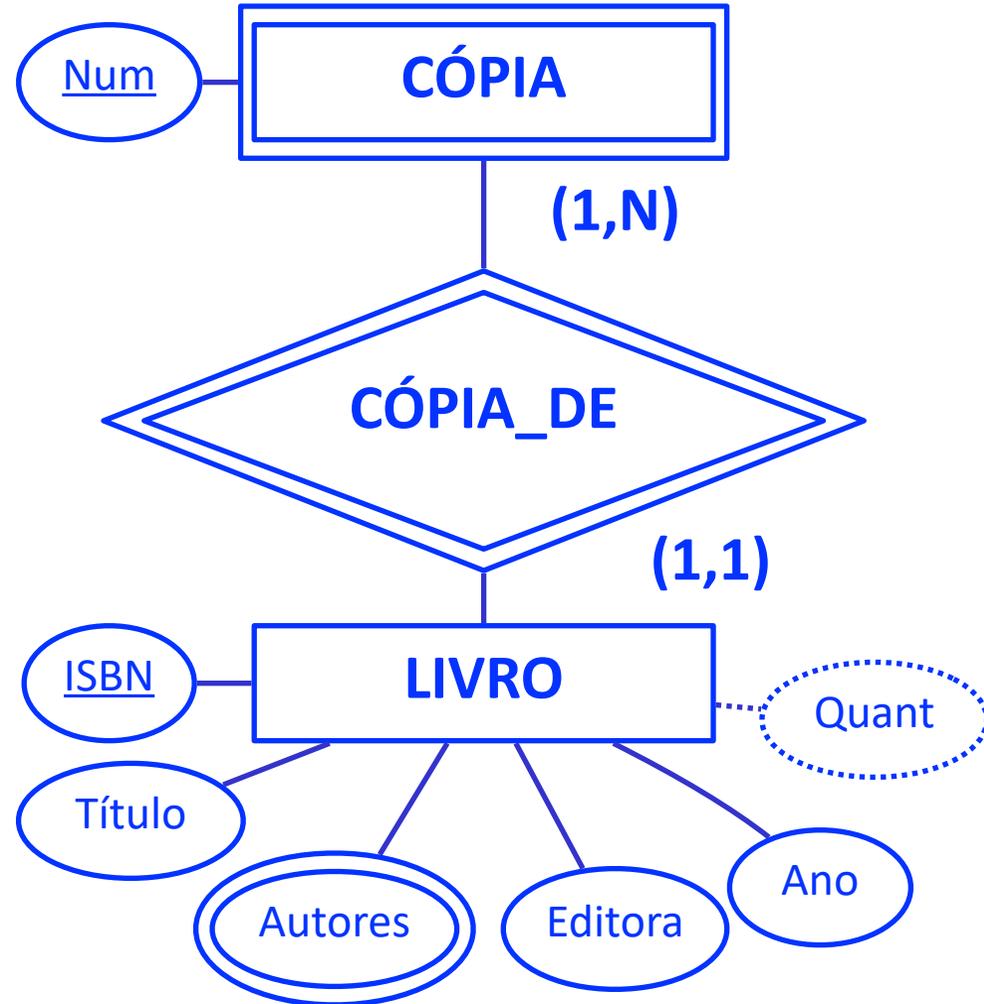
- **Livro** deve corresponder claramente a uma **entidade-tipo**.
- **Cópia** é uma **entidade-tipo fraca**, já que tem apenas uma **chave parcial** e é caracterizada existencialmente pela sua associação à **entidade-tipo identificadora** livro.

■ Mapeamento p/o modelo relacional

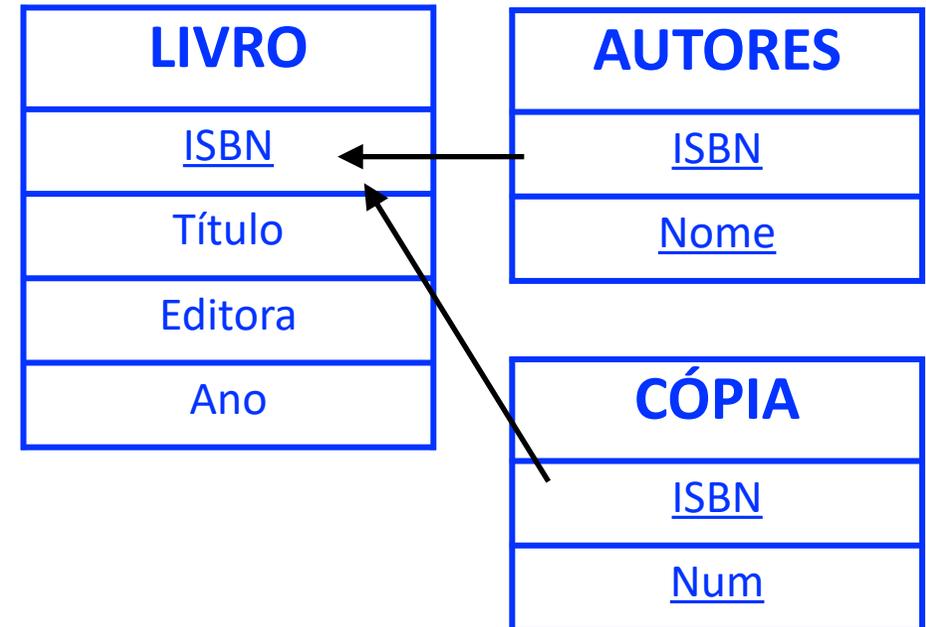
- São necessárias tabelas correspondentes a entidades-tipo livro e cópia de livro.
- ... e ainda uma tabela auxiliar para os **autores de um livro**, um **atributo multi-valor**.

Livros e cópias (cont.)

Modelo ER



Tabelas no modelo relacional



Livros e cópias (cont.)

LIVRO			
<u>ISBN</u>	<u>Título</u>	<u>Editora</u>	<u>Ano</u>
9789722709620	Os Lusíadas	INCM	1999
9789722526289	Sonetos	WOOK 11-17	2013
9780131103627	The C Programming Language, 2nd edition	Prentice Hall	1988

AUTOR	
<u>ISBN</u>	<u>Autor</u>
9789722709620	Luís de Camões
9789722526289	Luís de Camões
9780131103627	Brian W. Kernighan
9780131103627	Dennis M. Ritchie

CÓPIA	
<u>ISBN</u>	<u>Num</u>
9789722709620	1
9789722709620	2
9789722709620	3
9789722526289	1
9780131103627	1
9780131103627	2

Empréstimo de livros

■ Requisitos:

- Uma cópia de um livro pode ser emprestada a apenas um utente a dada altura. Um utente pode ter várias cópias de livros emprestadas.
- Cada empréstimo tem associada uma data.
- Muito simplistas os requisitos ... poderíamos considerar por ex. se é necessário guardar um histórico dos empréstimos, quando a cópia de um livro é devolvida por um utente, ...

■ Modelação ER:

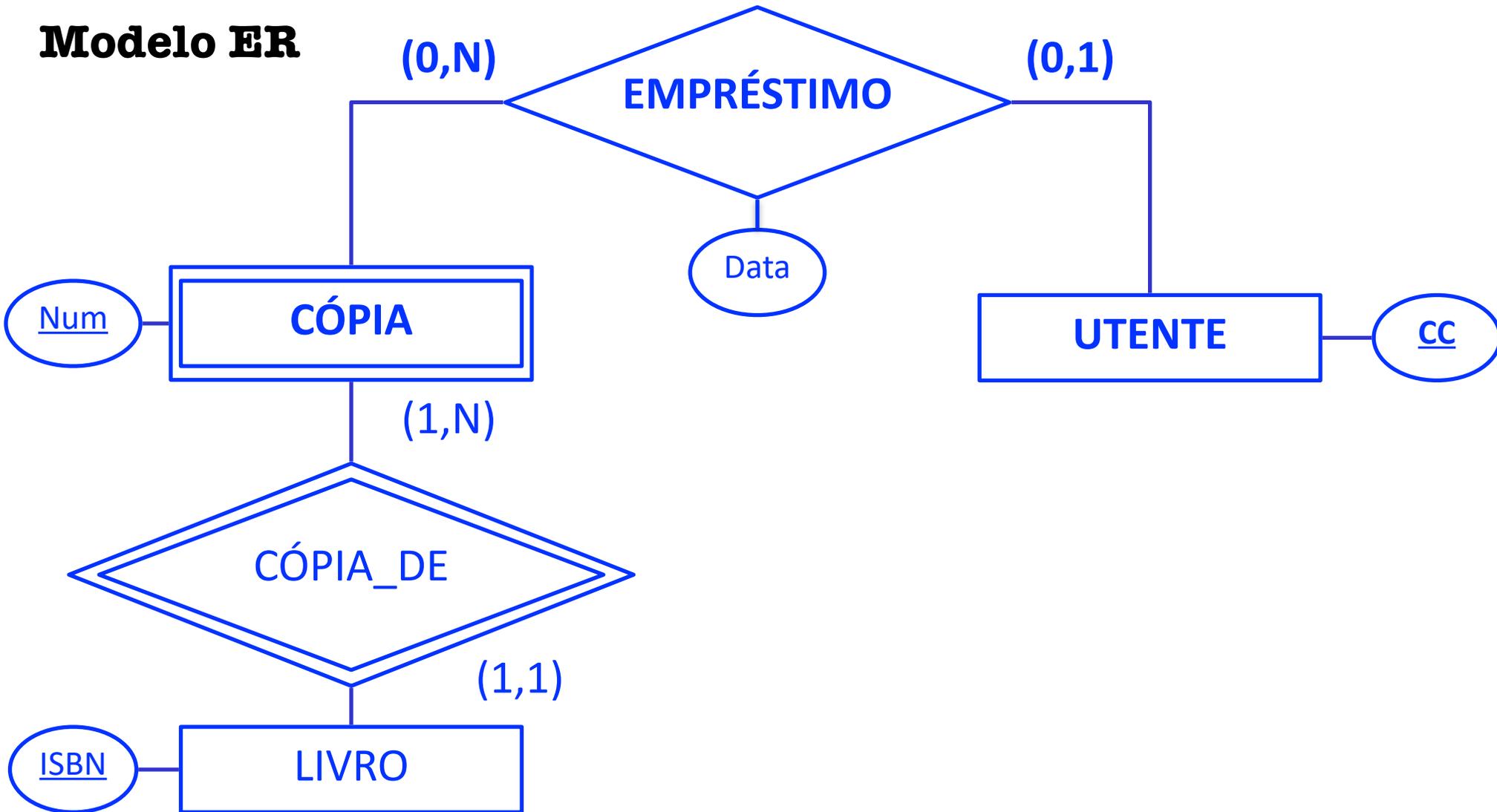
- Um relacionamento entre as entidades-tipo para utente e cópia de livro, com atributo p/data de empréstimo.

■ Mapeamento p/modelo relacional

- Campo extra na tabela de cópia (chave externa para utente) **OU** tabela de “referência cruzada” p/o relacionamento.

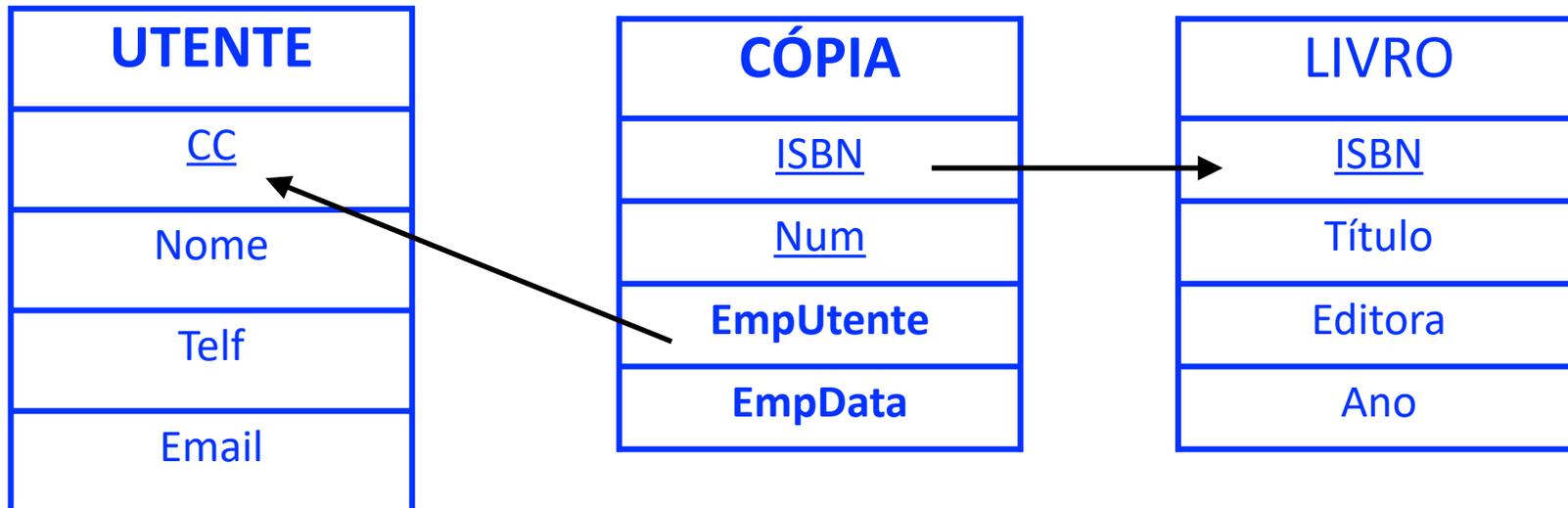
Empréstimo de livros (cont.)

Modelo ER



Empréstimo de livros (cont.)

Mapeamento p/modelo relacional – 1ª alternativa –



Empréstimo de livros (cont.)

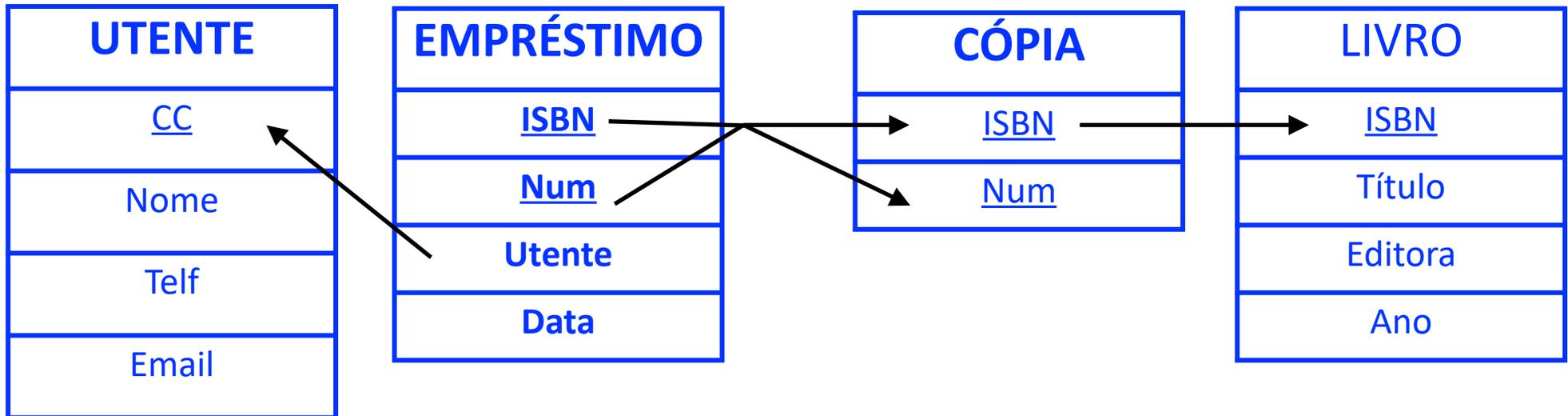
LIVRO			
<u>ISBN</u>	Título	Editora	Ano
9789722709620	Os Lusíadas	INCM	1999
9789722526289	Sonetos	WOOK 11-17	2013
9780131103627	The C Programming Language, 2nd edition	Prentice Hall	1988

UTENTE		
<u>CC</u>	Nome	...
10583212	João Pinto	...
12447555	Carlos Semedo	...
16348500	Maria Silva	...
11983516	Pedro Costa	...

CÓPIA			
<u>ISBN</u>	<u>Num</u>	EmpUtente	EmpData
9789722709620	1	10583212	17-01-2019
9789722709620	2	NULL	NULL
9789722709620	3	12447555	27-12-2018
9789722526289	1	12447555	28-12-2018
9780131103627	1	NULL	NULL
9780131103627	2	11983516	01-02-2019

Empréstimo de livros (cont.)

Mapeamento p/modelo relacional – 2ª alternativa –



Livros e cópias (cont.)

LIVRO			
<u>ISBN</u>	Título	Editora	Ano
9789722709620	Os Lusíadas	INCM	1999
9789722526289	Sonetos	WOOK 11-17	2013
9780131103627	The C Programming Language, 2nd edition	Prentice Hall	1988

CÓPIA	
<u>ISBN</u>	<u>Num</u>
9789722709620	1
9789722709620	2
9789722709620	3
9789722526289	1
9780131103627	1
9780131103627	2

UTENTE		
<u>CC</u>	Nome	...
10583212	João Pinto	...
12447555	Carlos Semedo	...
16348500	Maria Silva	...
11983516	Pedro Costa	...

EMPRÉSTIMO			
<u>ISBN</u>	<u>Num</u>	Utente	Data
9789722709620	1	10583212	17-01-2019
9789722709620	3	12447555	27-12-2018
9789722526289	1	12447555	28-12-2018
9780131103627	2	11983516	01-02-2019

Secções, estantes e prateleiras

■ Requisitos:

- A biblioteca está organizada em **secções** (ex. p/Livros Técnicos, Literatura, ...), identificadas por um código único e uma descrição.
- Cada secção tem **estantes**, tendo cada estante associado também um código único. **As estantes são amovíveis, podendo mudar entre secções.**
- **Cada estante está dividida em prateleiras (fixas à estante!).** Cada **prateleira** tem associada um **número único no contexto de uma estante – poderá no entanto haver estantes com numeração semelhante.**

Secções, estantes e prateleiras (cont.)

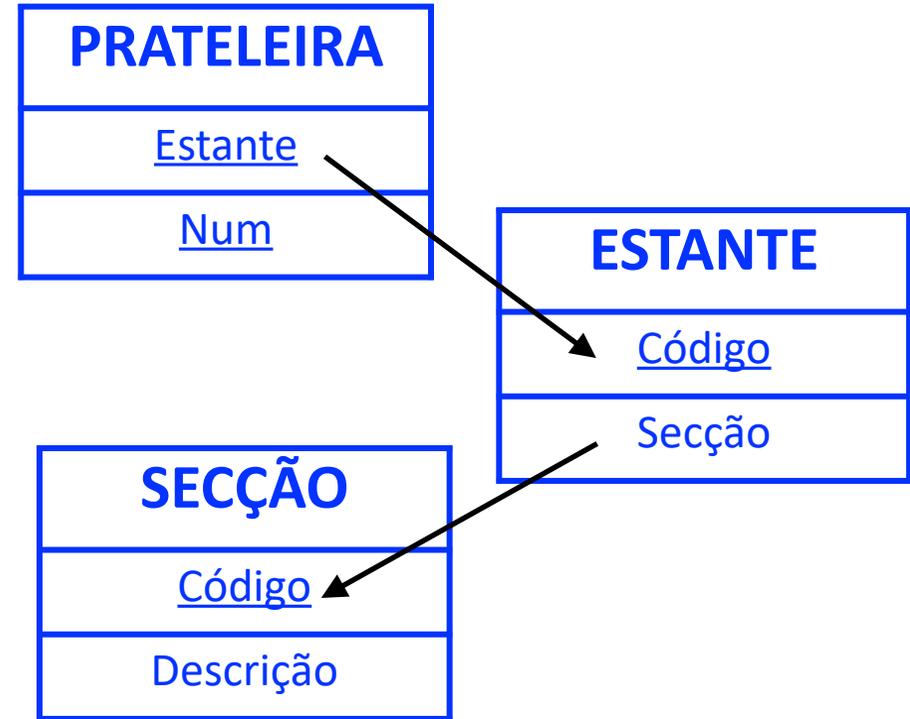
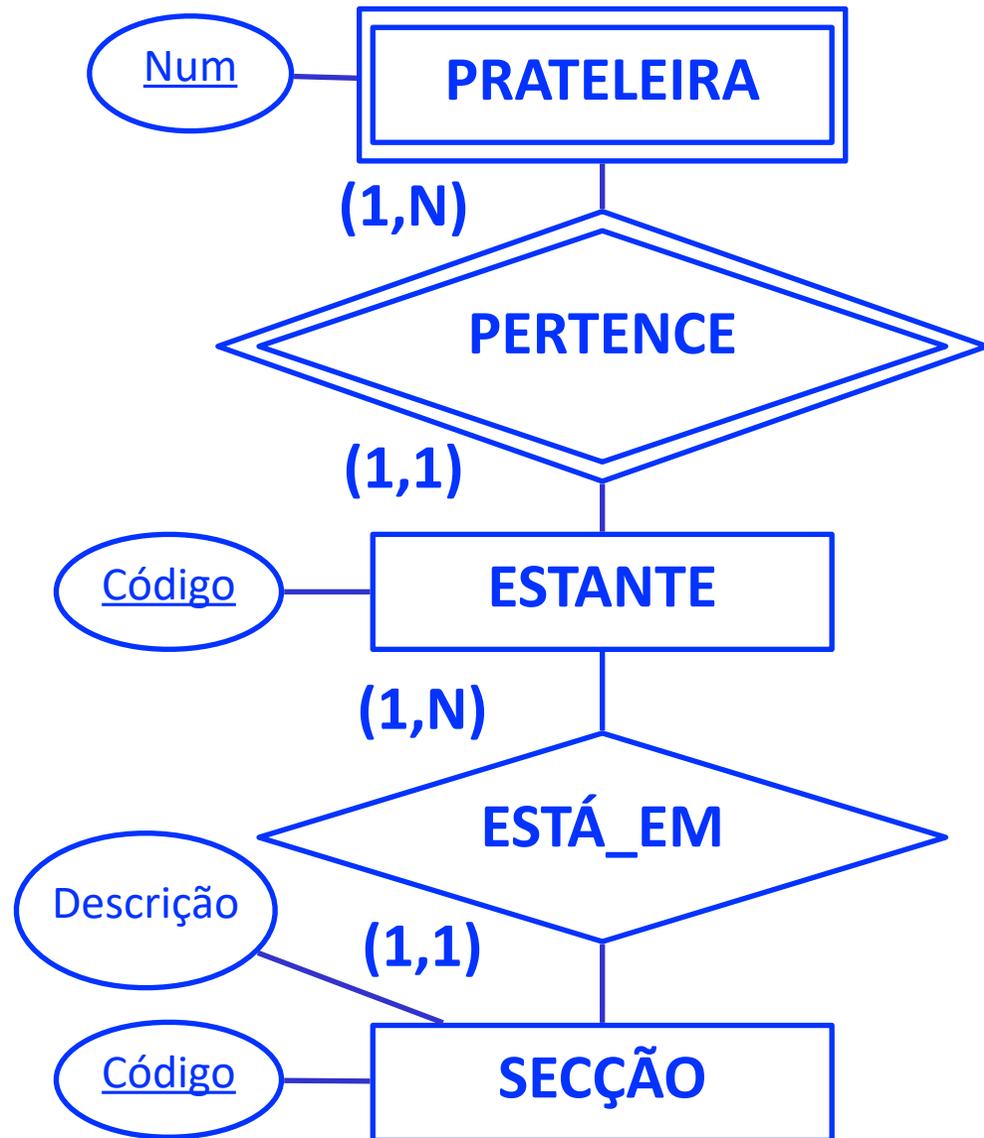
■ Modelação ER

- **3 Entidades-tipo:** 1) secção; 2) estante; 3) prateleira.
- **2 Relacionamentos:** 1) associação prateleira-estante; 2) associação estante-secção.
- A entidade-tipo para prateleira é **fraca**, como no caso de cópia de um livro (chave parcial, dependência existencial).
- Em contraste a estante tem chave associada e a associação a secções é mutável.

■ Mapeamento p/modelo relacional

- Teremos 3 tabelas em correspondência a cada uma das entidades.

Secções, estantes e prateleiras (cont.)



Secções, estantes e prateleiras (cont.)

SECÇÃO	
<u>Código</u>	<u>Descrição</u>
L	Literatura
BD	Banda Desenhada
T	Livros Técnicos

ESTANTE	
<u>Código</u>	<u>Secção</u>
E12	L
E13	L
E14	BD
E99	T

PRATELEIRA	
<u>Estante</u>	<u>Num</u>
E12	1
E12	2
E12	3
E13	1
E13	2
E13	3
E14	1
E14	2
E99	1
E99	2
E99	3

Arrumação dos livros

■ Requisito:

- As cópias de um livro são tipicamente arrumadas na mesma prateleira de uma estante.
- No entanto, por razões de espaço, cópias diferentes do mesmo livro podem ser arrumadas em prateleiras da mesma estante ou até estantes na mesma secção.
- Poderão haver prateleiras vazias.

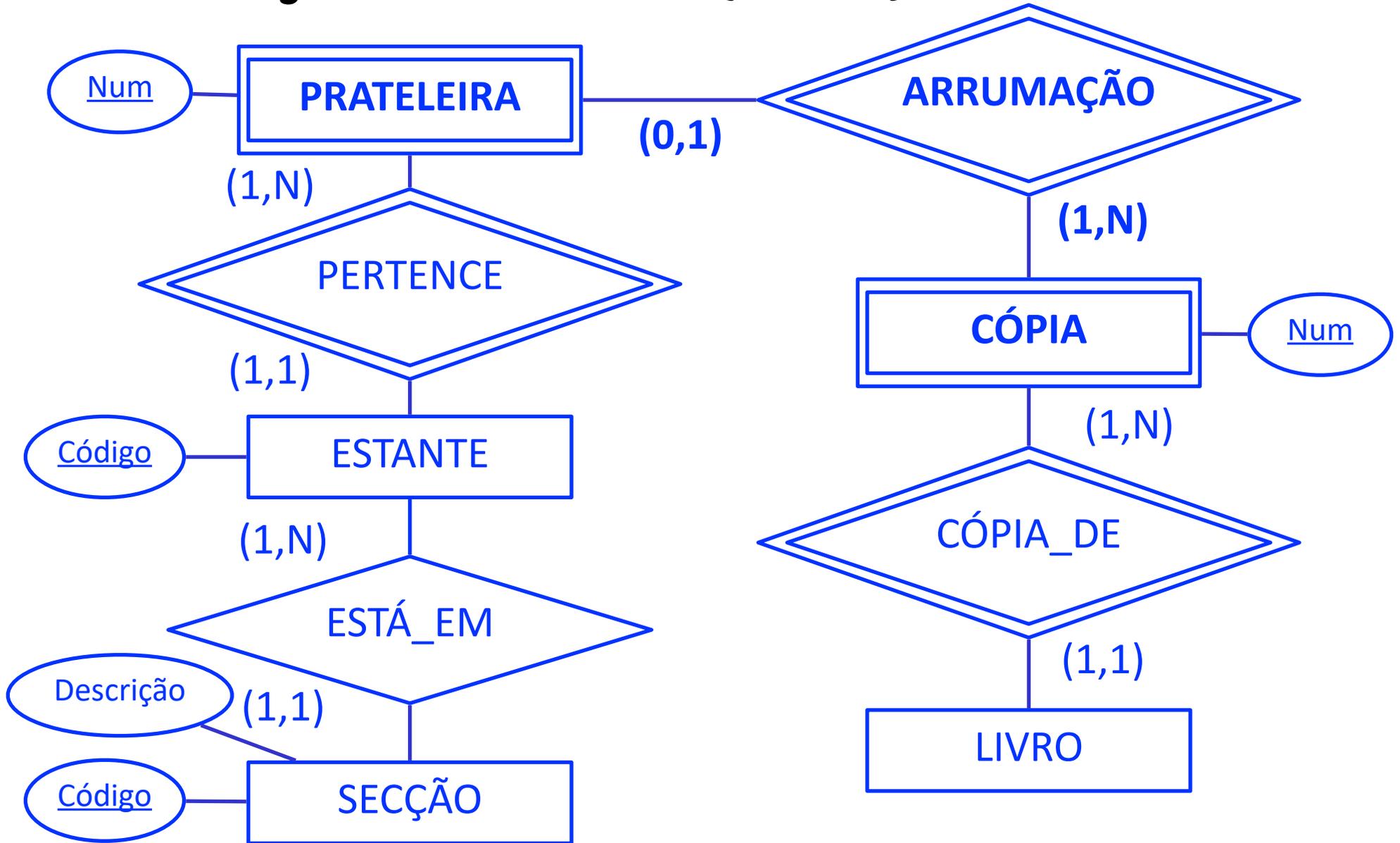
■ Modelação ER:

- **Um relacionamento entre as entidades-tipo para cópia e prateleira** (curiosamente as duas fracas), não entre livro e prateleira ou entre livro e estante.

■ Mapeamento p/modelo relacional

- Implica atributos adicionais na tabela para cópia de livro.

Arrumação dos livros (cont.)



Arrumação dos livros (cont.)

SECÇÃO	
<u>Código</u>	Descrição
L	Literatura
BD	Banda Desenhada
T	Livros Técnicos

ESTANTE	
<u>Código</u>	Secção
E12	L
E13	L
E14	BD
E99	T

vazias

PRATELEIRA	
<u>Código</u>	<u>Num</u>
E12	1
E12	2
E12	3
E13	1
E13	1
E13	3
E14	1
E14	2
E99	1
E99	2
E99	3

CÓPIA					
<u>ISBN</u>	<u>Num</u>	Estante	Prateleira	EmpCC	EmpData
9789722709620	1	E12	1	10583212	17-01-2019
9789722709620	2	E12	1	NULL	NULL
9789722709620	3	E12	2	12447555	27-12-2018
9789722526289	1	E12	2	12447555	28-12-2018
9780131103627	1	E99	2	NULL	NULL
9780131103627	2	E99	2	11983516	01-02-2019