O Modelo Relacional

Bases de Dados (CC2005)
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Eduardo R. B. Marques — DCC/FCUP
— parcialmente adaptado de slides por Fernando Silva e Ricardo Rocha —

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 basedo na linguagem SQL (como veremos também depois).

Conceitos base

Conceito de relação

Atributos (colunas) →

Registos (linhas) {

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

- 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 SGBD baseado no modelo relacional.

Definições e notação genérica

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

ALUNO(NumMec, NumCC, Nome, Curso)

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

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

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

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

Atributos chave

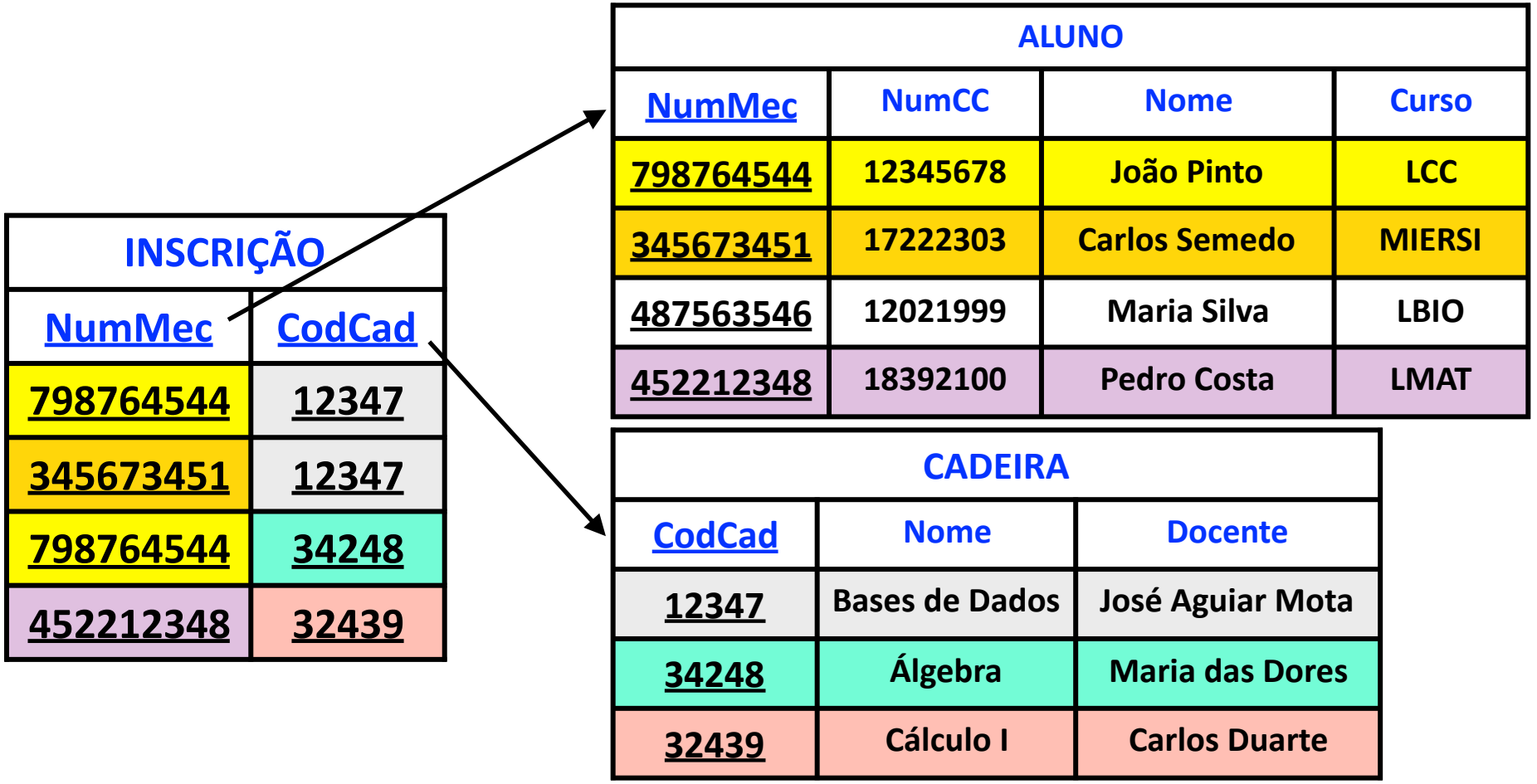
Analogamente ao modelo ER, uma **chave** para a tabela é um conjunto mínimo de atributos que permitem identificar de forma única um registo. Poderão haver várias chaves possíveis, sendo a chave mais adequada ao universo em causa escolhida como **a chave primária** (que aparece sublinhada).

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 menos adequada do que NumMec como chave primária.

Chaves externas



INSCRIÇÃO

NumMec	CodCad
798764544	12347
345673451	12347
798764544	34248
452212348	32439

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

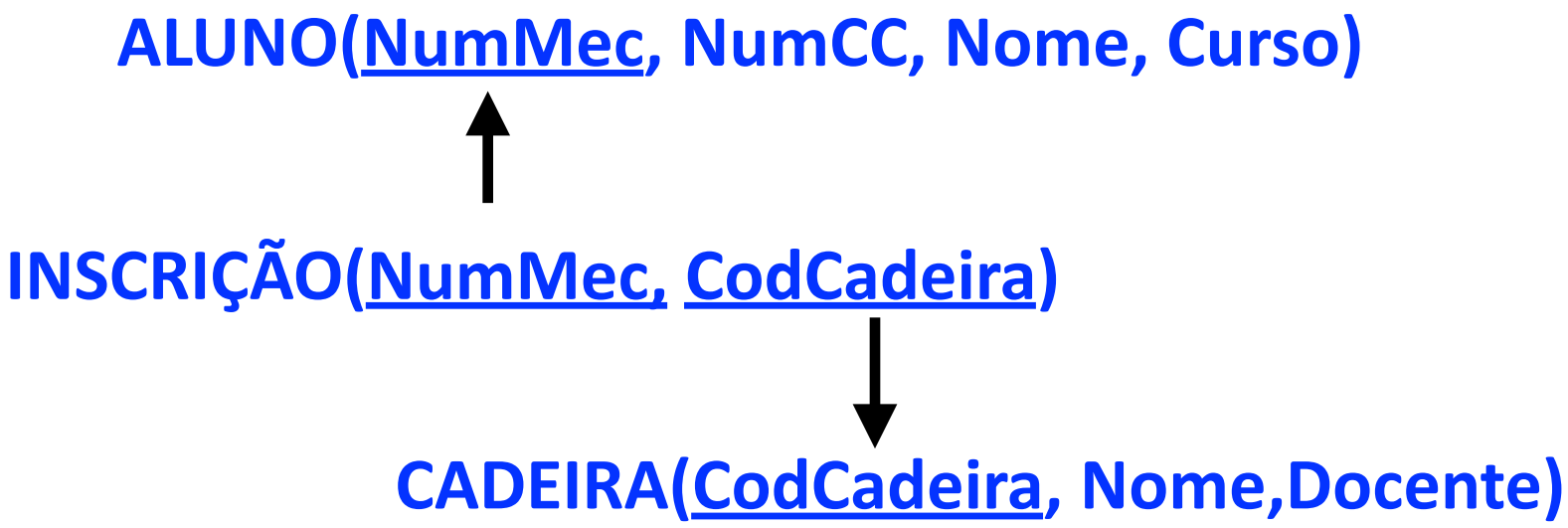
CADEIRA

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

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

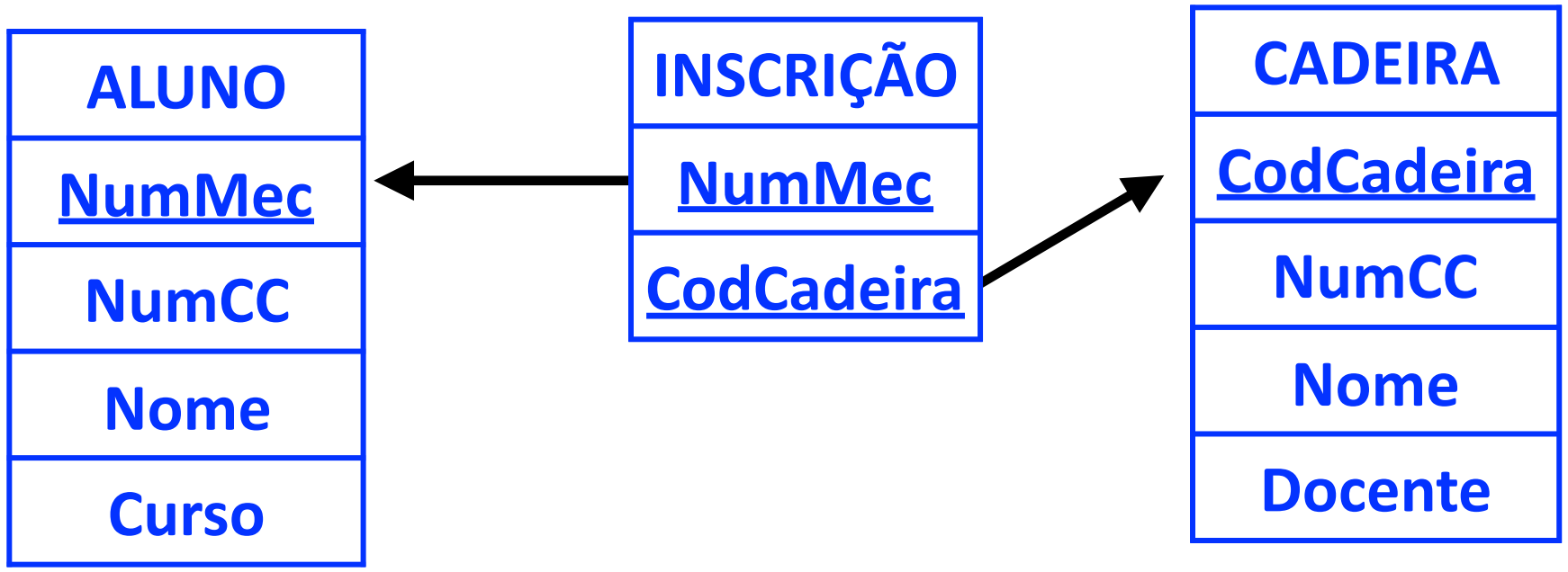
  [**Q:** consegue vislumbrar o modelo ER correspondente a esta BD?]

Representação de esquema



- 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	
NumMec	CodCad
111111111	12347
345673451	12347
798764544	34248
452212348	12347

?

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

?

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

?

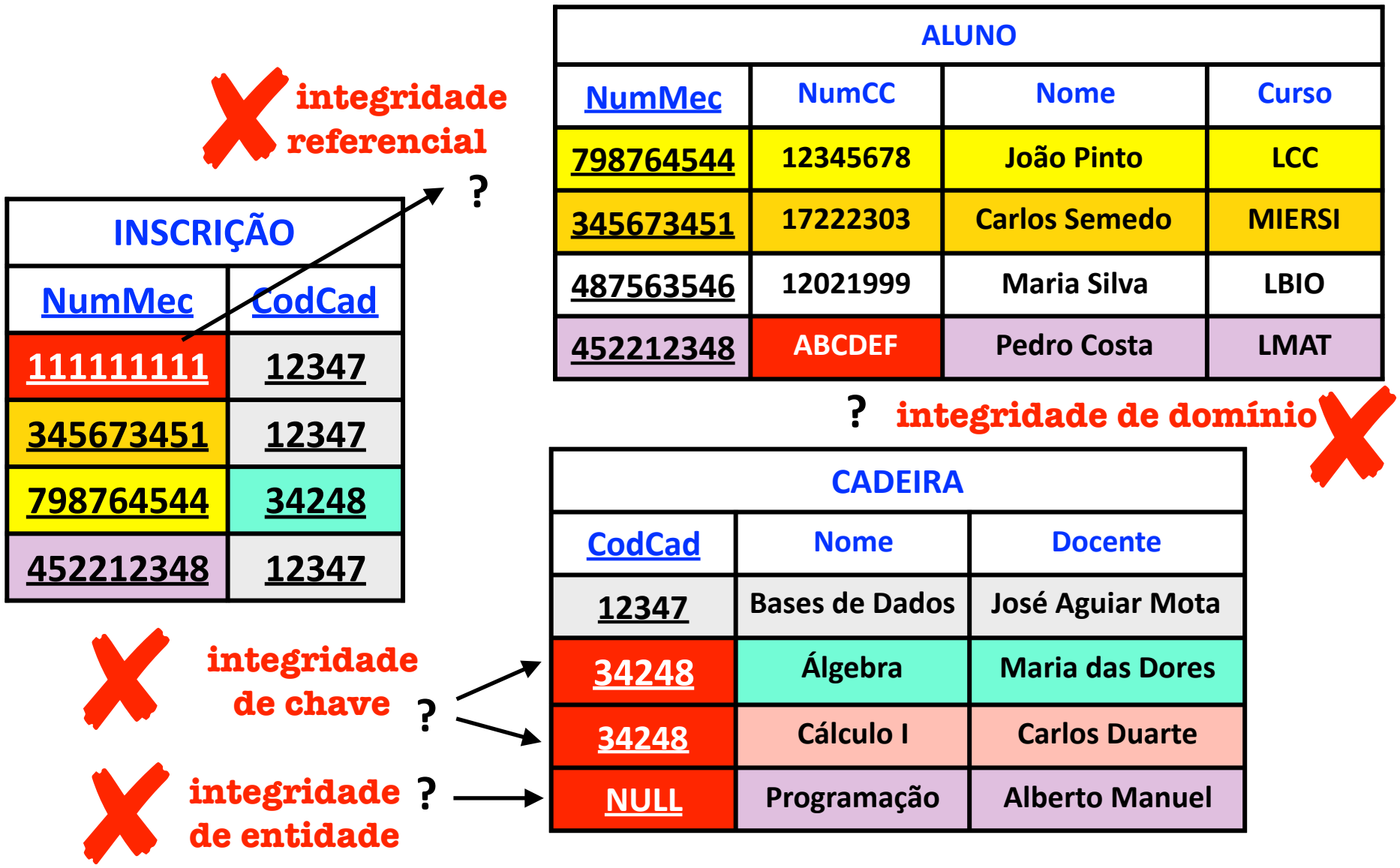
?

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**
  - Os valores de um atributo fazem parte de um domínio do atributo.
- **Integridade de chave**
  - Dois registos da mesma tabela não podem ter valores iguais para uma chave, em particular a chave primária.
- **Integridade referencial**
  - O valor de atributo(s) que seja(m) chave externa deve referir-se a à chave primária da tabela a que a chave externa se refere.
- **Integridade de entidade**
  - O(s) valor(es) da chave primária não pode(m) ser NULL (sob pena de não conseguirmos identificar registos).

Violação de restrições de integridade



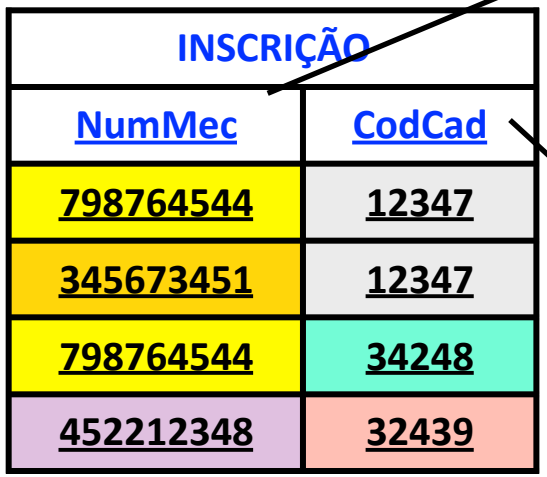
Estado e operações sobre uma base de dados

- **Esquema da BD = { Esquema de tabelas }**
- **Estado da BD = { Conteúdo das tabelas }**
- **O estado da BD é mutável** sendo normal considerar as seguintes operações nuclares:
  - **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 (a cobrir em próximas aulas).

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	
NumMec	CodCad
798764544	12347
345673451	12347
798764544	34248
452212348	32439

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

CADEIRA		
CodCad	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 integridades, assinalando o erro.
- SGBDs maduros normalmente suportam todos os tipos de restrições de integridade que consideramos (domínio, entidade, chave, referencial).
- Há no entanto excepções 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

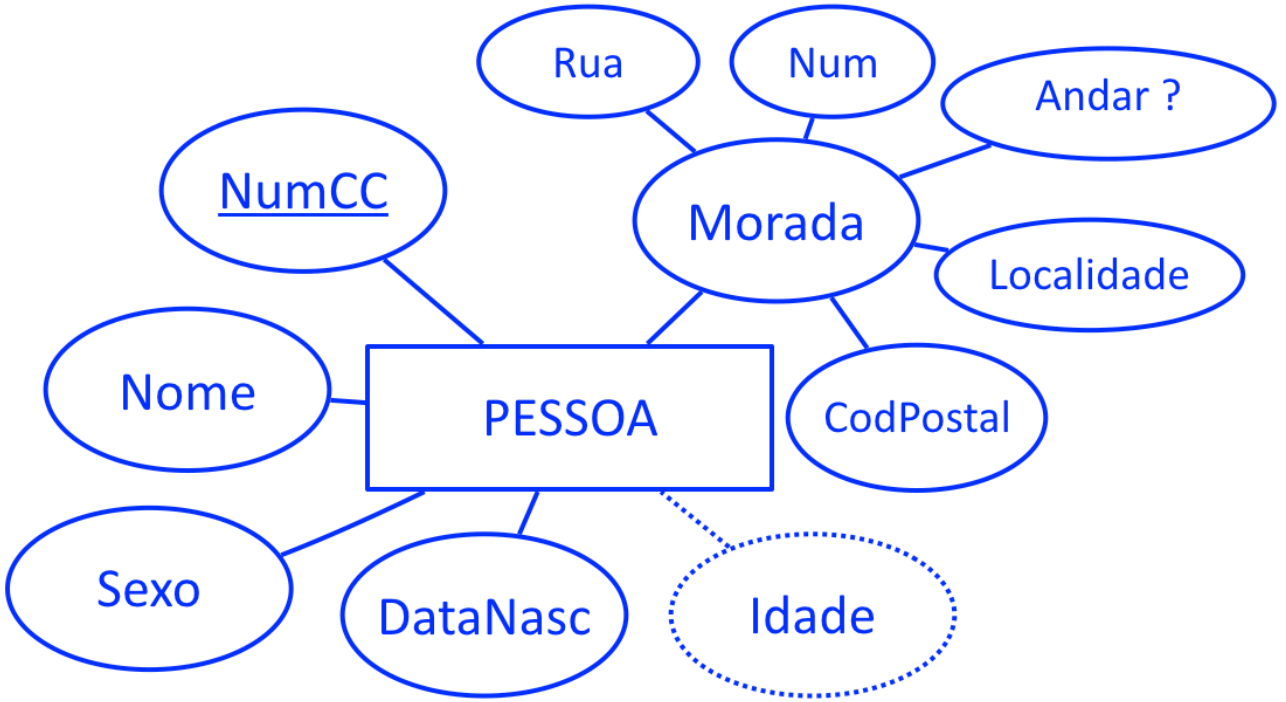
- Perspectiva geral
  - Entidades-tipo e relacionamentos são convertidos em tabelas.
- Mapeamento de entidades-tipo em tabelas
  - Passa pelo mapeamento directa de atributos, excepto no caso de atributos multi-valor em que precisamos de recorrer a “tabelas auxiliares”.
- Mapeamento de relacionamentos
  - Poderão resultar em uma tabela nova para o relacionamento ou na adição de atributos às tabelas correspondentes a entidade-tipo na forma de chaves externas e ainda considerando atributos do relacionamento.

Mapeamento de entidades-tipo

- **Entidade-tipo E >>> tabela T**
  - Nome de E **>>>** nome de T
  - Chave de E **>>>** chave primária de T
  - Atributos derivados de E não são mapeados.
  - Atributo simples de E **>>>** atributo simples de T
  - Atributos simples de atributo composto de E **>>>** atributos de T
  - Atributos opcionais levam simplesmente à inclusão de NULL no domínio do atributos.
  - Atributo multi-valor MV de E **>>>** tabela auxilar 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

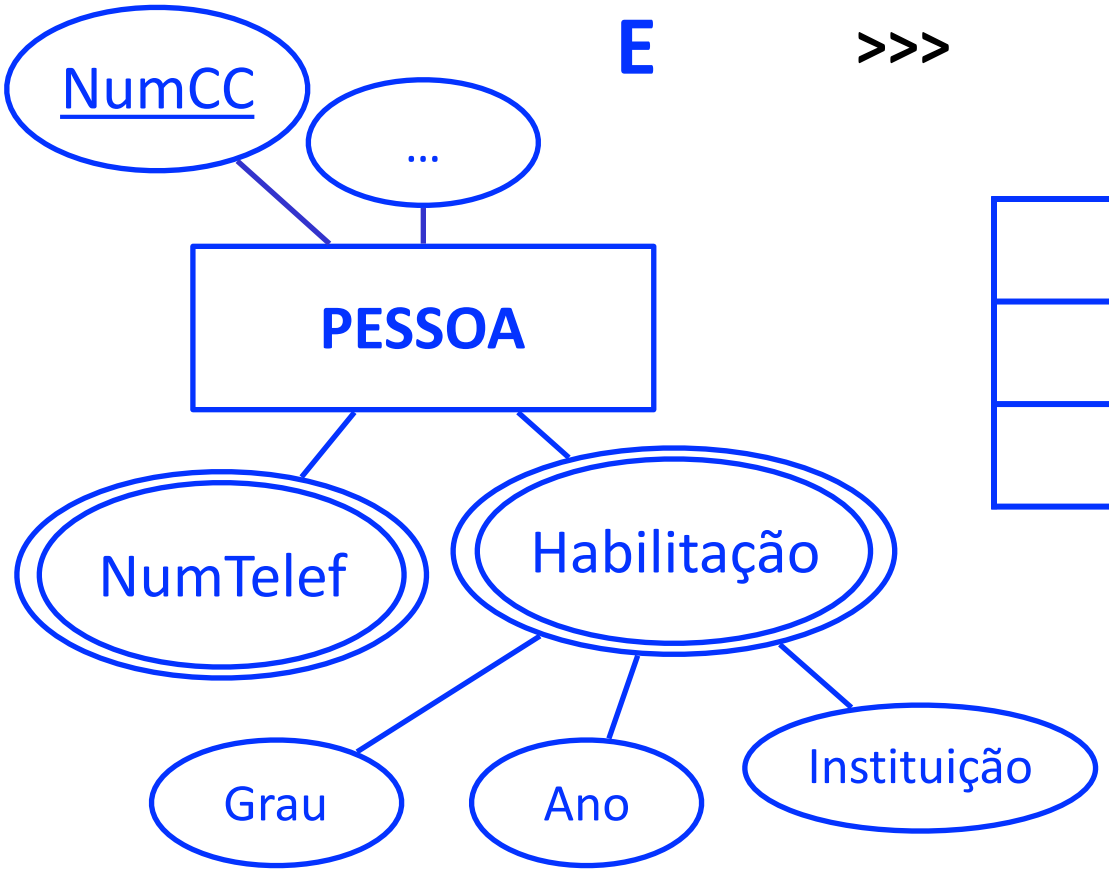
E >>> **T**

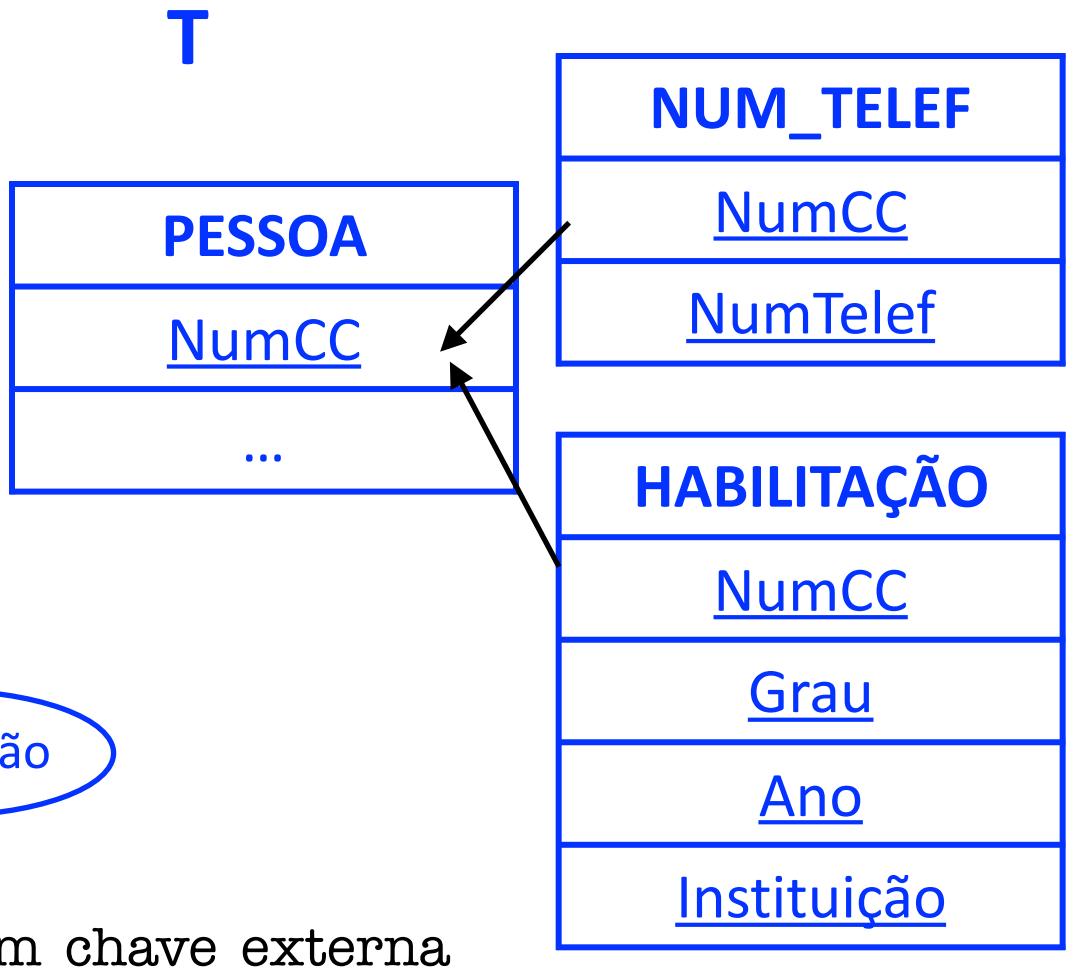


PESSOA
NumCC
Nome
Sexo
DataNasc
MRua
MNum
MAndar
MLocalidade
MCodPostal

Nota: não usaremos notação especial para atributos opcionais no modelo relacional. Está implícito no entanto que deveremos ter **NULL ∈ dom(MAndar)**

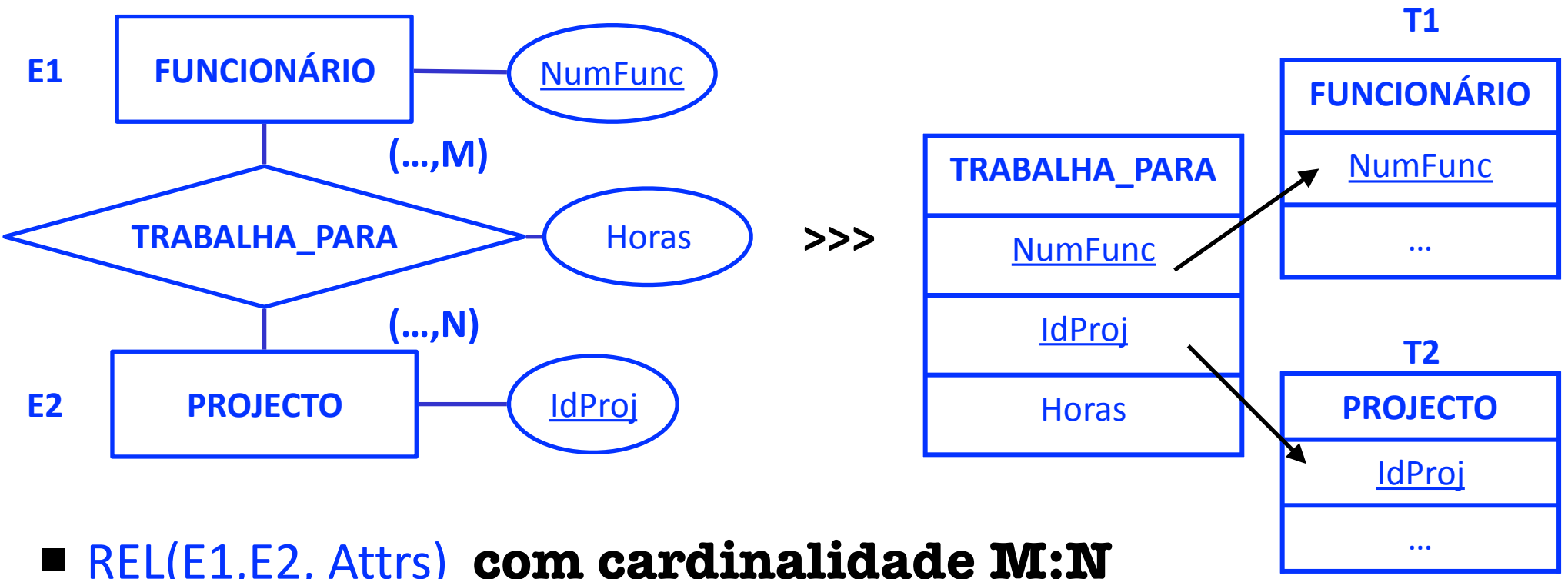
Entidades-tipo — atributos multivalor





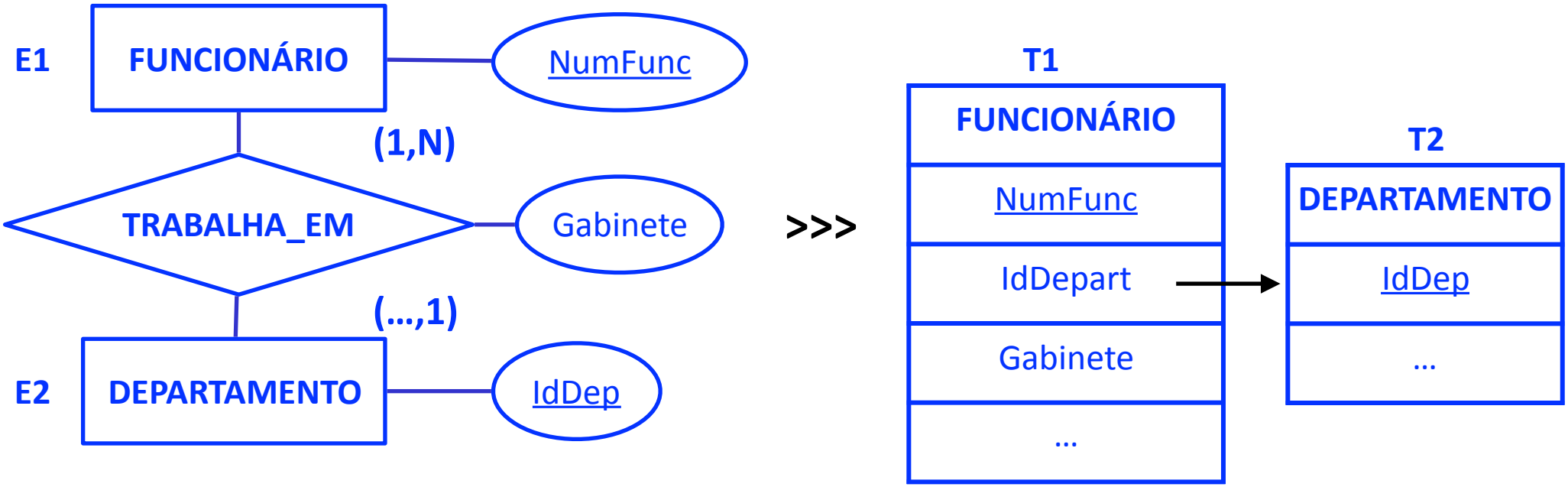
É criada nova tabela "auxilar" com chave externa referenciando a chave primária de T. Chave externa e restantes atributos definem a chave de MV.

Mapeamento de relacionamentos M:N



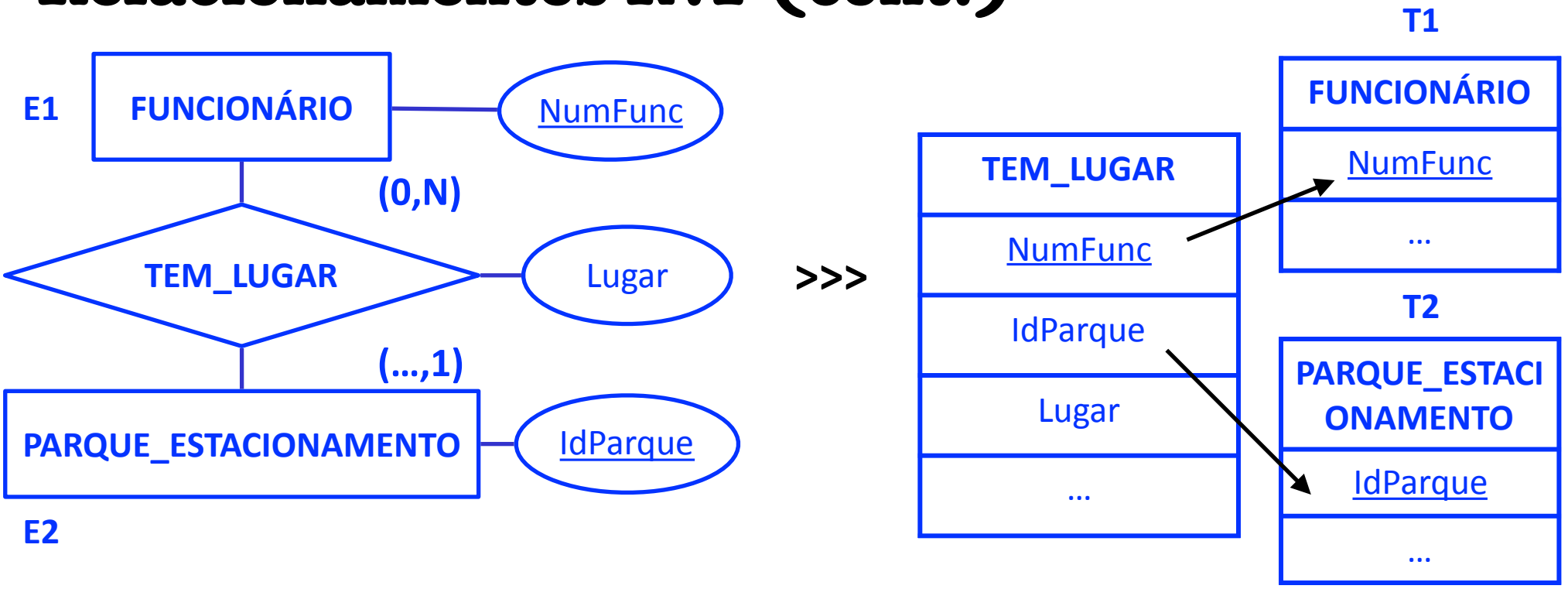
- REL(E1,E2, Attrs) **com cardinalidade M:N**
  - Criar tabela de "referência-cruzadas" específica a REL.
  - Chave primária de REL = Chave primária de T1 + Chave primária de T2 (ambas chaves externas)
  - Attrs mapeados na tabela para REL

Relacionamentos N:1



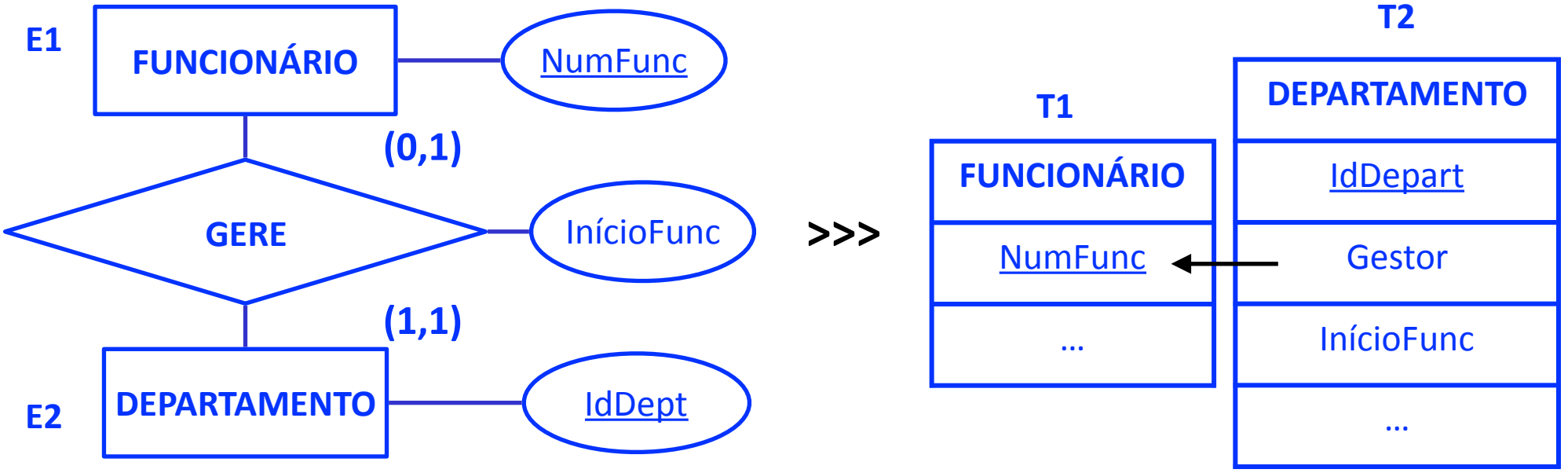
- REL(E1,E2, Attrs) **com cardinalidade N:1 e participação total de E1**
  - chave externa em T1 para a chave primária de T2
  - Attrs mapeados em T1

Relacionamentos N:1 (cont.)



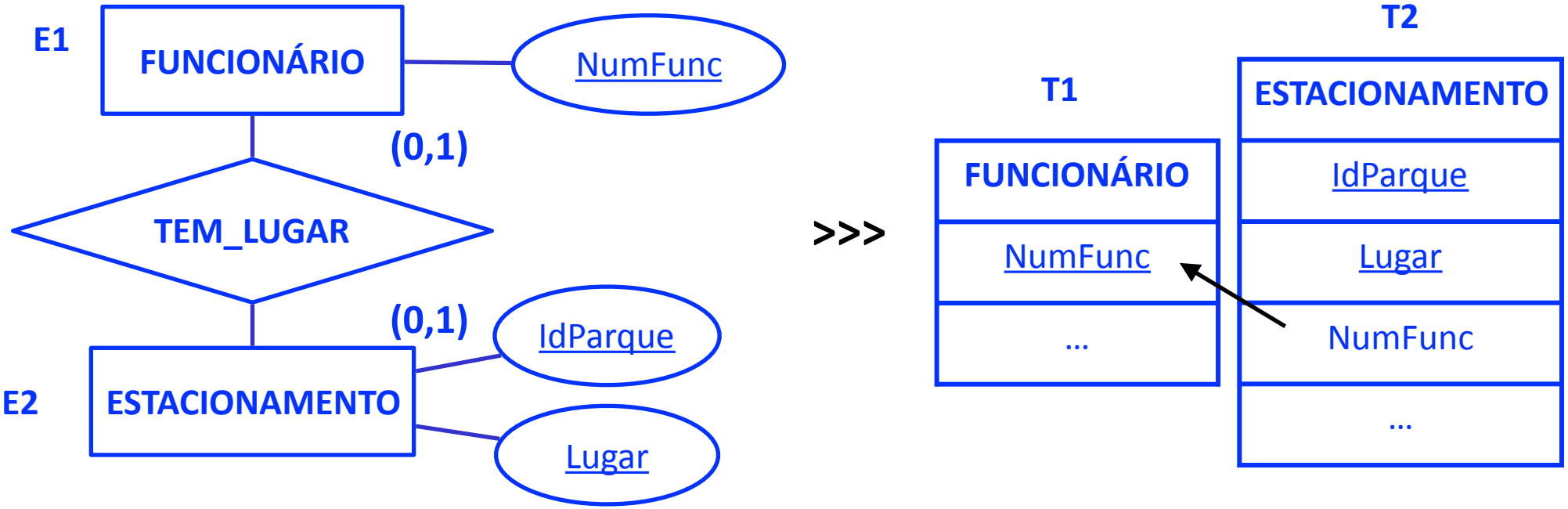
- **REL(E1,E2, Attrs) c/cardinalidade N:1 e participação parcial de E1**
  - Criar tabela específica a REL incorporando Attrs
  - Chave primária de T1 é chave primária na tabela e chave externa.
  - Chave primária de T2 é chave externa (apenas).
  - Estratégia anterior também válida, mas onerosa se apenas algumas instâncias se envolverem em REL levando a demasiados valor NULL.

Relacionamentos 1:1



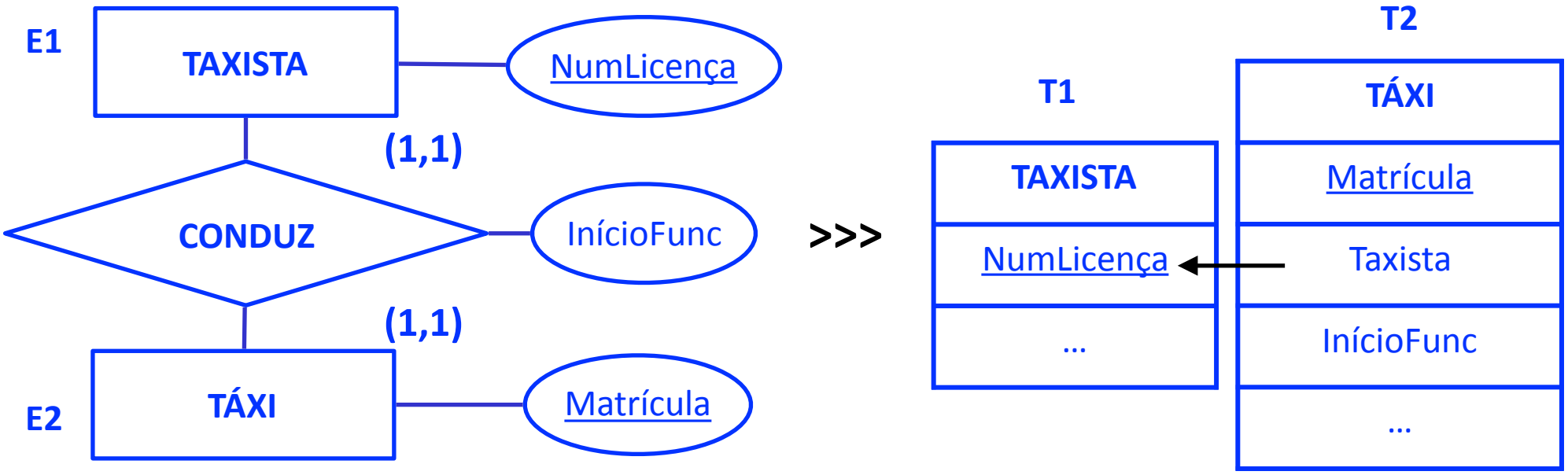
- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação parcial de E1, participação total de E2**
  - Chave primária de T1 é adicionada como chave externa a T2
  - Attrs mapeados em T2

Relacionamentos 1:1 (cont.)



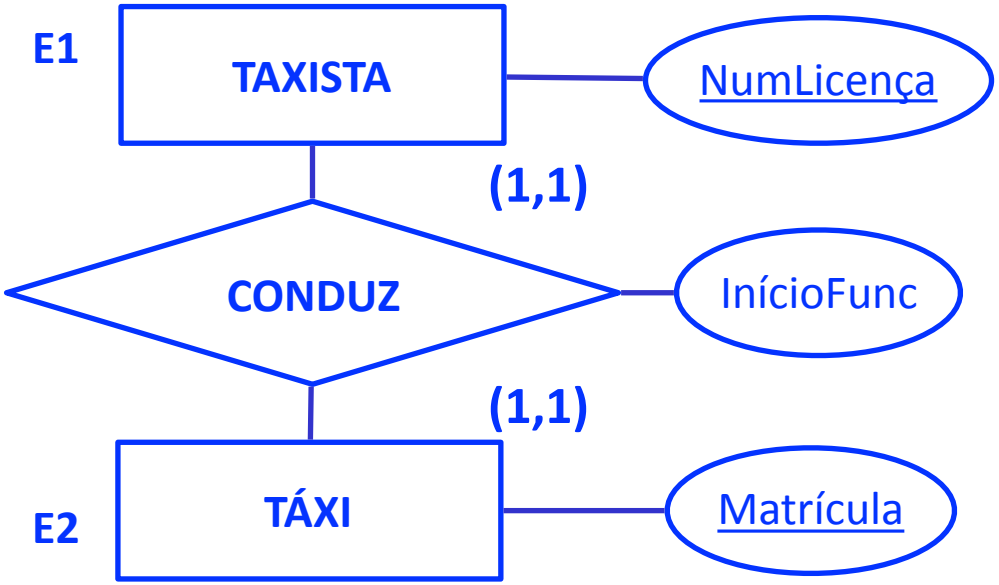
- **REL(E1,E2, Attrs) c/cardinalidade 1:1, participação parcial de ambas as entidades**
  - Podemos usar estratégia semelhante à anterior. Em alternativa, uma tabela de "referências-cruzadas" poderá ser preferível se houverem poucas instâncias relacionadas.

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.

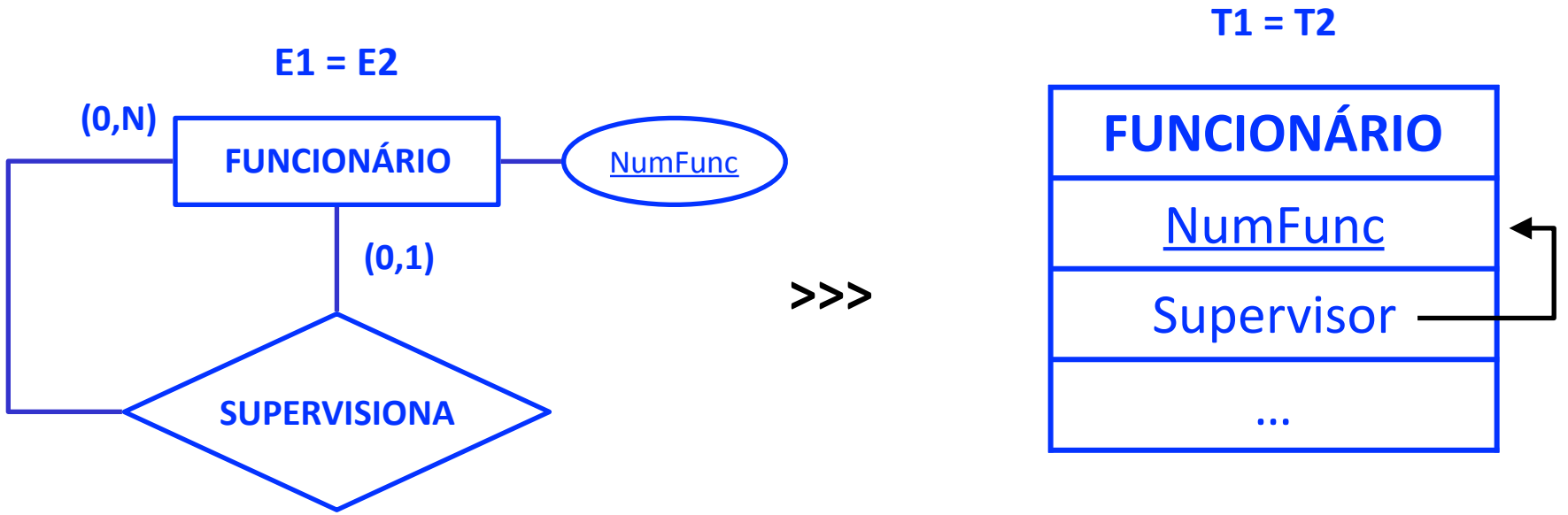
Relacionamentos 1:1 (cont.)



>>>

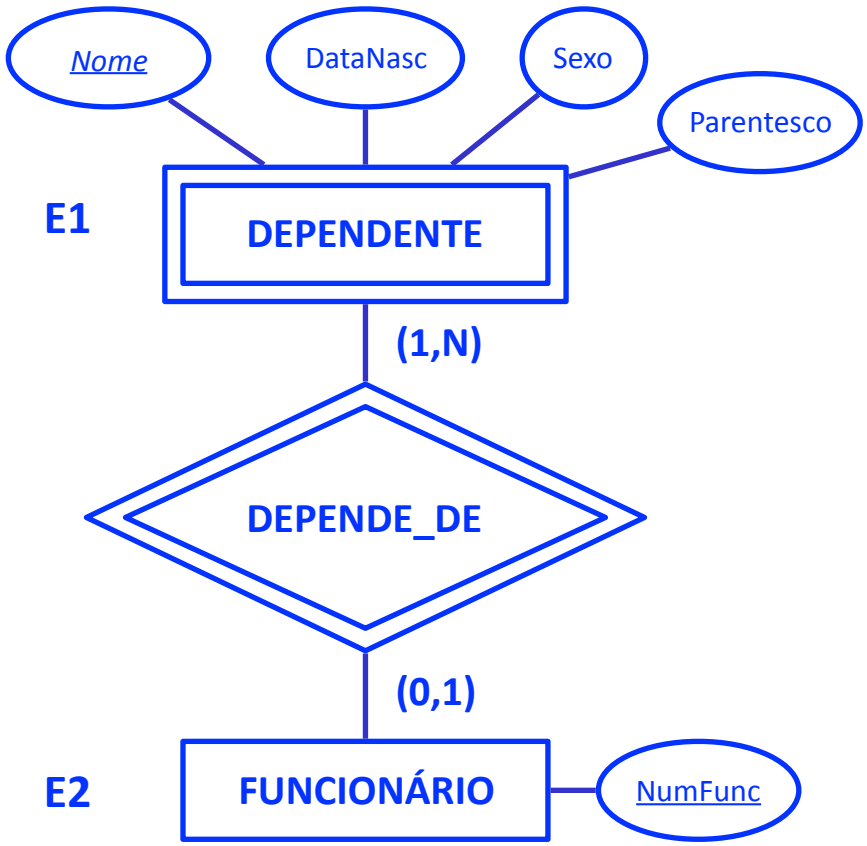
- REL(E1,E2, Attrs) **c/cardinalidade 1:1, participação total de ambas as entidades**
  - **Opção 2:** Usar uma tabela só, abrangendo atributos de ambas as entidades e do relacionamento, definindo a chave de uma das entidades como chave primária. A opção é normalmente inadequada, pois define uma "multi-entidade híbrida", que será mais sensível a mudanças no esquema conceptual e/ou relacional da BD.

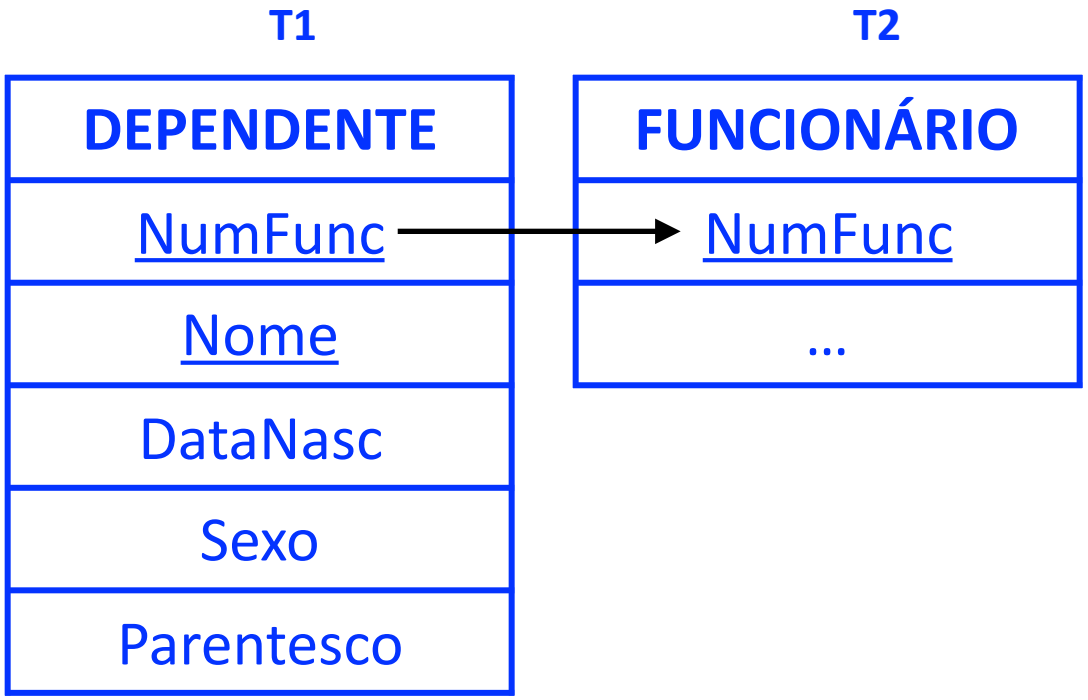
Outros casos — relacionamentos recursivos



- Tratado como outros casos, mas afectando apenas uma tabela.
- Opção de tabela de "referência cruzada" também possível como em outros casos.

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 frace + chave externa para T2