

Aula 017

Professoras:

Marta Mattoso

Vanessa Braganholo

Conteúdo:

Mapeamento ER - Relacional

Material da Aula

→ Material adaptado do original de Carlos Heuser, distribuído em conjunto com o livro Projeto de Banco de Dados, Editora Sagra Luzzatto, 2004.

Organização do Curso

- Conceitos Gerais
- SGBDs e Modelos de Dados
- Modelo ER
- Modelo Relacional
- Álgebra Relacional
- **Mapeamento ER-Relacional**
- SQL
- Normalização
- Evolução dos Modelos

Livro Texto

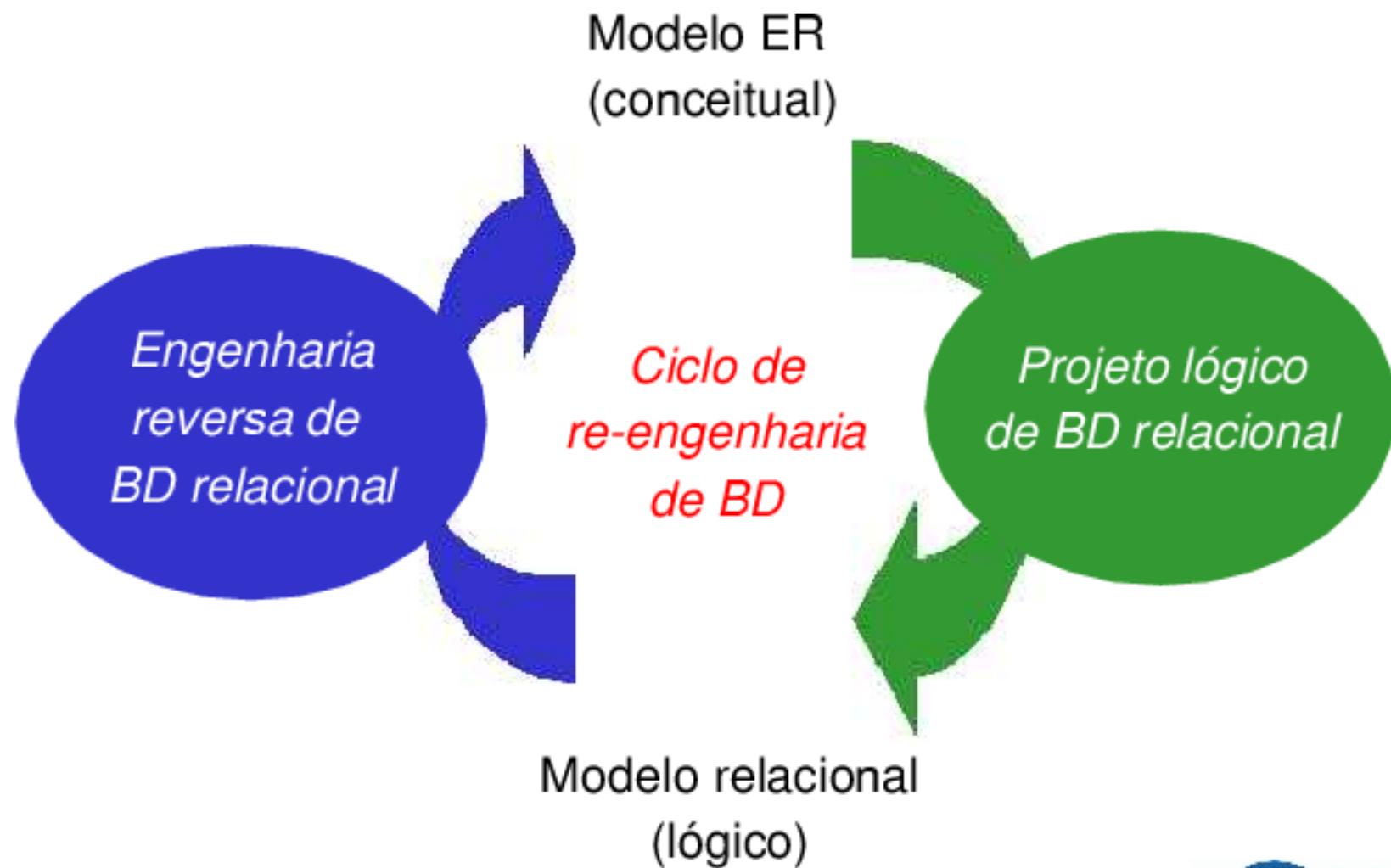
→ Referência para o livro texto Projeto de Banco de Dados

- Capítulo 5: Transformações entre modelos
 - Seção 5.1: Visão geral do projeto lógico
 - Seção 5.2: Transformação ER-Relacional (parcial)

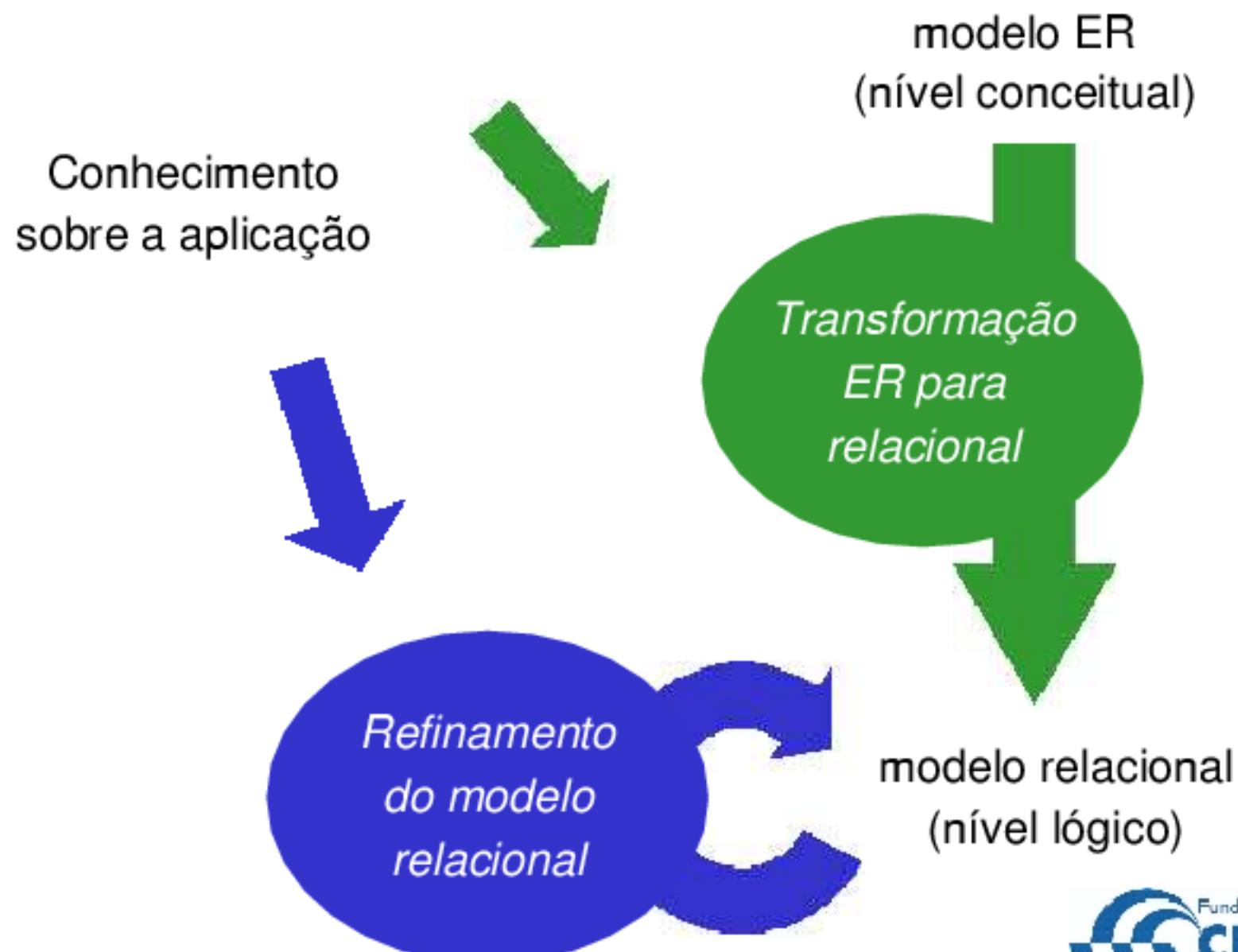
Organização da Aula

- ➡ Visão geral do projeto lógico
- ➡ Transformação ER-Relacional

Transformações entre modelos



Projeto lógico



Transformação ER para relacional

→ Regras gerais

- Aplicáveis à maioria dos casos
- Em alguns casos, por exigências da aplicação, outros mapeamentos podem ser usados
- Implementadas em ferramentas CASE

→ Objetivos básicos:

- Boa performance
- Simplificar o desenvolvimento

Regras gerais de tradução

- Evitar junções
- Diminuir o número de chaves
- Evitar campos opcionais

Junção

- ➔ Operação para buscar dados de diversas linhas associadas pela igualdade de campos
- ➔ Exemplo:
 - ➡ buscar os dados de um empregado e os dados de seu departamento (duas tabelas diferentes)

Evitar junções

- SGBD relacional normalmente armazena os dados de uma linha contiguamente em disco
- Junção envolve diversos acessos a disco
- Preferível
 - ter os dados necessários a uma consulta em uma única linha

Chave e índice

- Implementação eficiente do controle de chaves: SGBD usa um índice
 - Índices tendem a ocupar espaço considerável em disco
- Inserção e remoção de entradas em um índice
 - Podem exigir diversos acessos a disco

Diminuir o número de chaves

- Usar implementações com menos chaves
- Exemplo

Cliente (CodCliente,Nome,NomeContato,Endereço,Telefone)

ou

Cliente (CodCliente,Nome,NomeContato)

ClienteEnder (CodCliente,Endereço,Telefone)

CodCliente referencia Cliente

Campos opcionais

- Campo opcional = campo que pode assumir o valor VAZIO (NULL em SQL).
- SGBD relacional não desperdiça espaço pelo fato de campos de uma linha estarem vazios
- Campo opcional não tem influência na performance

Evitar campos opcionais

- Controle de campo opcional pode complicar programação
 - Verificar quais campos podem estar vazios, quando isto depende do tipo de linha
- Regra mais "fraca" que as precedentes

Para pensar...

→ Considere as seguintes alternativas de implementação de um banco de dados relacional:

Alternativa 1:

Aluno (CodAl,Nome,CodCurso,Endereco)

Alternativa 2:

Aluno (CodAl,Nome,CodCurso)

EnderecoAluno (CodAl,Endereco)

CodAl referencia Aluno

Em ambos casos está sendo representado um conjunto de alunos e informações (código, nome, código de curso, endereço) a ele referentes. Discuta à luz dos princípios que baseiam as regras de tradução de diagramas ER para modelo relacional, qual das duas alternativas é preferível.

Organização da Aula

- ➡ Visão geral do projeto lógico
- ➡ Transformação ER-Relacional

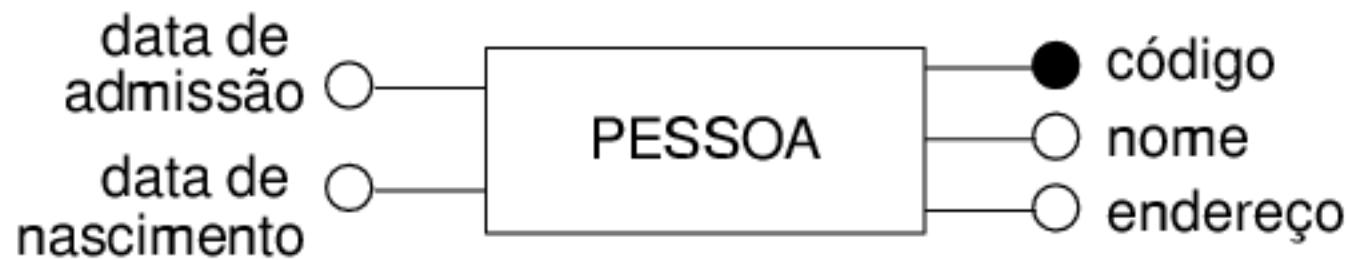
Passos da transformação ER para relacional

- ➡ Tradução inicial de entidades e respectivos atributos
- ➡ Tradução de relacionamentos e respectivos atributos
- ➡ Tradução de generalizações/especializações (próxima aula)

Implementação inicial de entidades

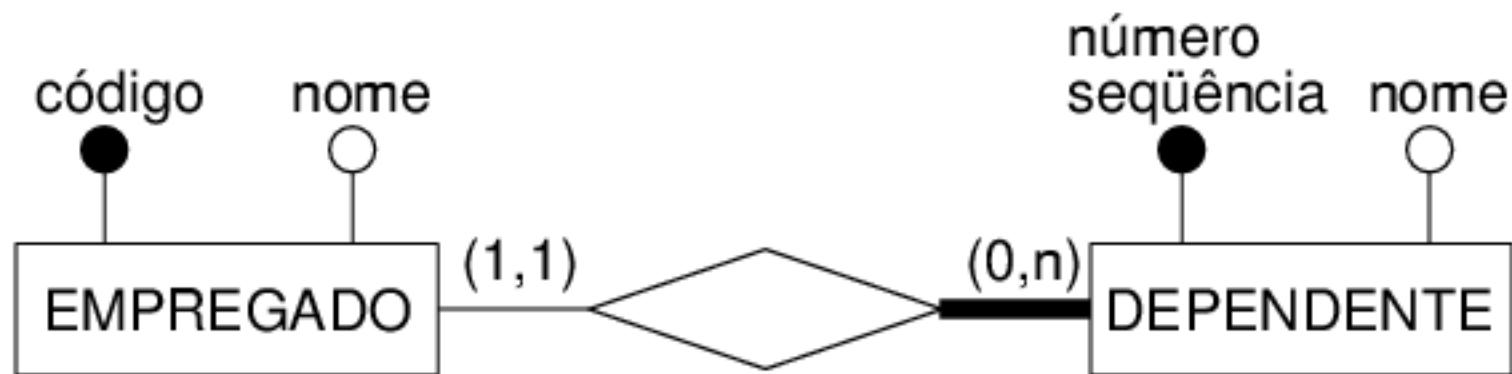
- Cada **entidade** é traduzida para uma **tabela**
- Cada **atributo** da entidade define uma **coluna** desta tabela
- Atributos **identificadores** da entidade correspondem a **chave primária** da tabela.
- Tradução **inicial**:
 - Regras que seguem podem fundir tabelas

Implementação de entidade: Exemplo



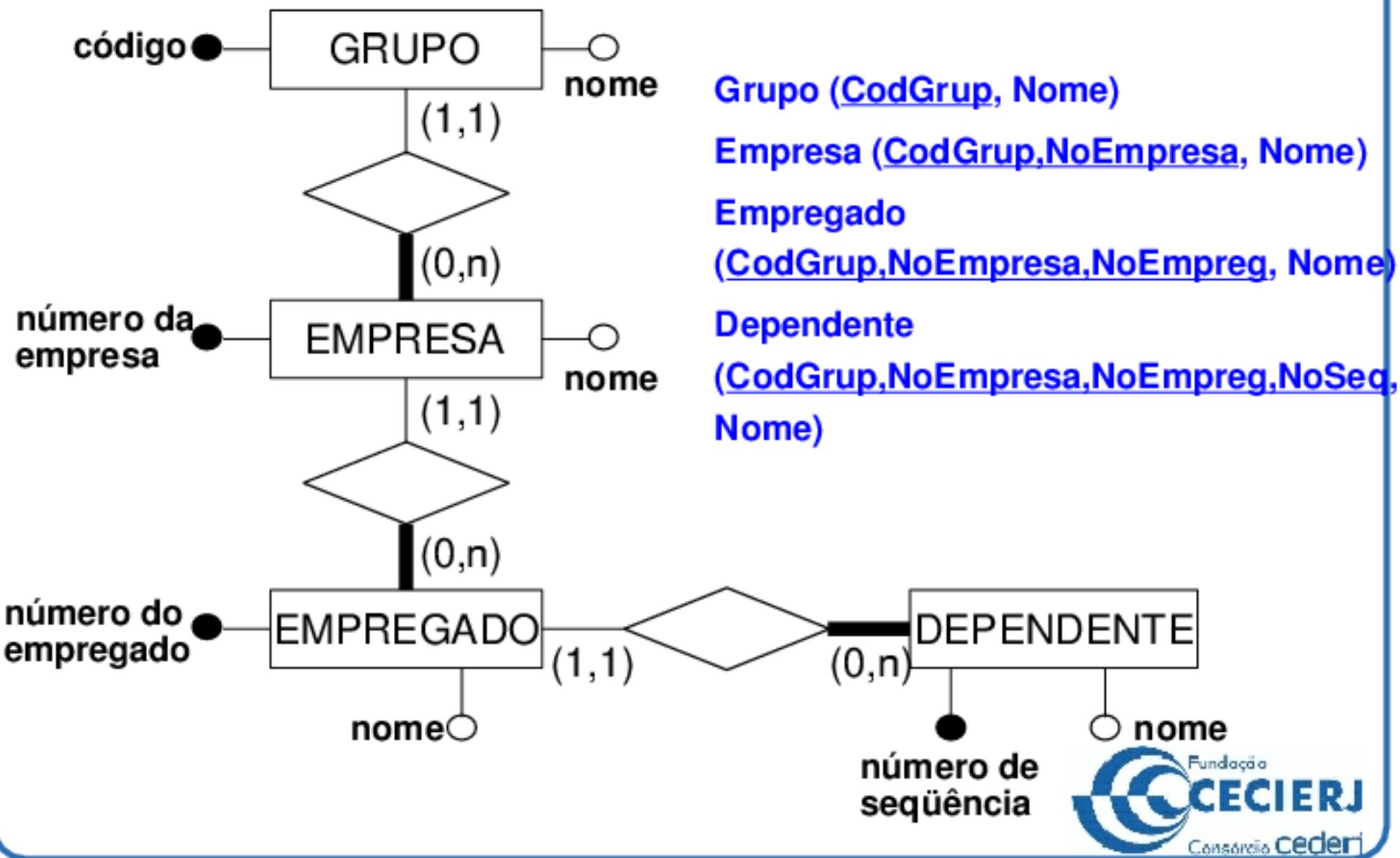
Pessoa (CodigoPess, Nome, Endereço, DataNasc, DataAdm)

Tradução de entidade: relacionamento identificador



Dependente (CodigoEmp,NoSeq, Nome)

Relacionamento identificador: Recursão



Nomes de colunas

- Referenciados freqüentemente em programas e outras formas de texto em computador
- Para diminuir o trabalho de programadores
 - manter os nomes de colunas curtos.
- SGBD relacional
 - nome de uma coluna não pode conter brancos

Nomes de atributos e nomes de colunas

- Não transcrever os nomes de atributos para nomes de colunas.
- Nomes de atributos compostos de diversas palavras devem ser abreviados
- Nomes de colunas não necessitam conter o nome da tabela
 - Preferível usar o nome de coluna **Nome** a usar os nomes de coluna **NomePess** ou **NomePessoa**
 - SQL já exige muitas vezes forma
 - Pessoa.Nome

Nome da coluna chave primária

→ Chave primária

- pode aparecer em outras tabelas na forma de chave estrangeira

→ Recomendável

- nomes das colunas que compõem a chave primária
 - sufixados ou prefixados com o nome ou sigla da tabela na qual aparecem como chave primária
- Exemplo
 - [CodigoPess](#)

Passos da transformação ER para relacional

- Tradução inicial de entidades e respectivos atributos
- **Tradução de relacionamentos e respectivos atributos**
- Tradução de generalizações/especializações (próxima aula)

Implementação de relacionamento: alternativas

- Tabela própria
- Adição de colunas a uma das tabelas
- Fusão de tabelas
- Alternativa depende da cardinalidade (máxima e mínima do relacionamento)

Tabela própria



Engenheiro (CodEng, Nome)

Projeto (CodProj, Título)

Atuação (CodEng,CodProj, Função)

CodEng referencia Engenheiro

CodProj referencia Projeto

Adição de colunas



Departamento (CodDept, Nome)

Empregado (CodEmp, Nome, **CodDept**, DataLota)

CodDept referencia Departamento

Fusão de tabelas



Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

Implementação de relacionamentos 1:1

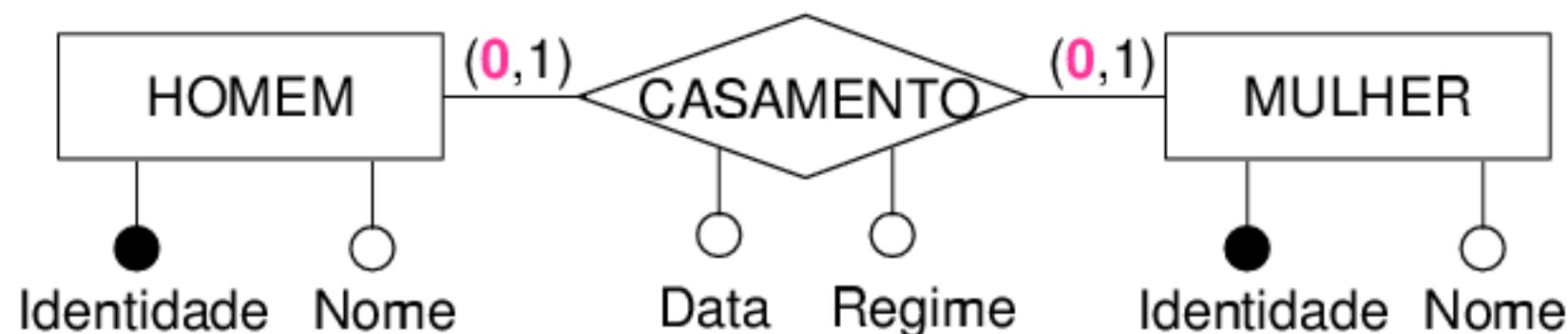
Tipo de Relacionamento	Regra de Implementação		
	Tabela Própria	Adição Coluna	Fusão Tabelas
(0,1) 	±	✓	✗
(0,1) 	✗	±	✓
(1,1) 	✗	✗	✓

✓ Alternativa preferida

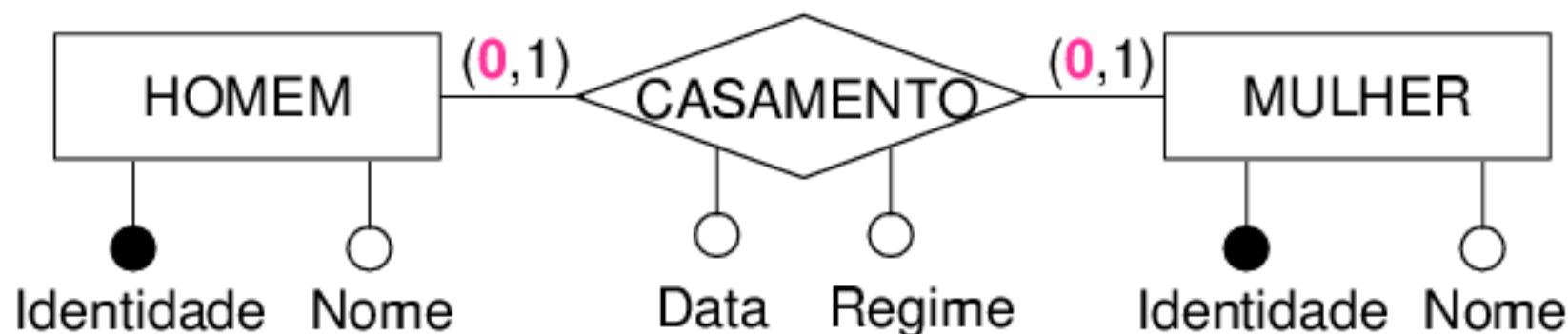
± Pode ser usada

✗ Não usar

1:1 - ambas entidades opcionais - exemplo



1:1 - ambas opcionais - Adição de colunas

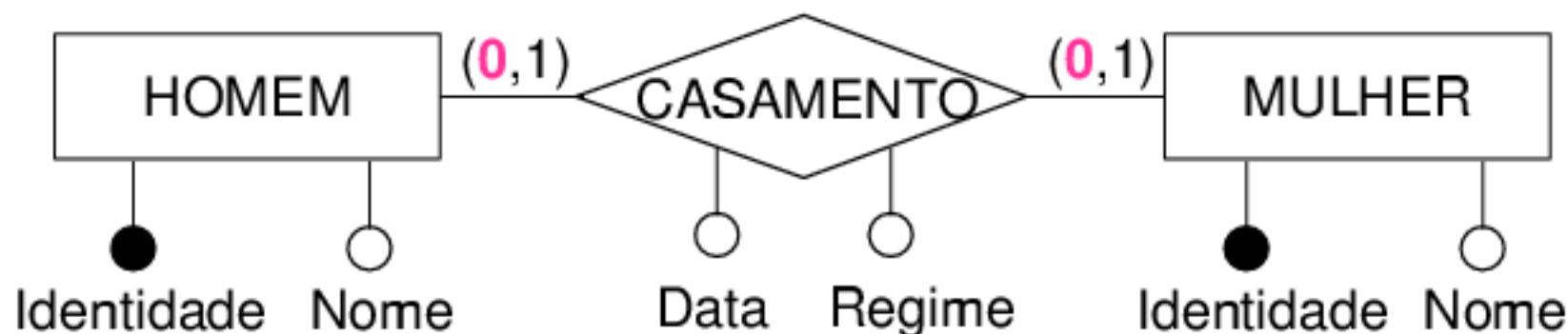


Mulher (IdentM, Nome, **IdentH**, Data, Regime)

IdentH referencia Homem

Homem (IdentH, Nome)

1:1 - ambas opcionais - Tabela própria



Mulher (IdentM, Nome)

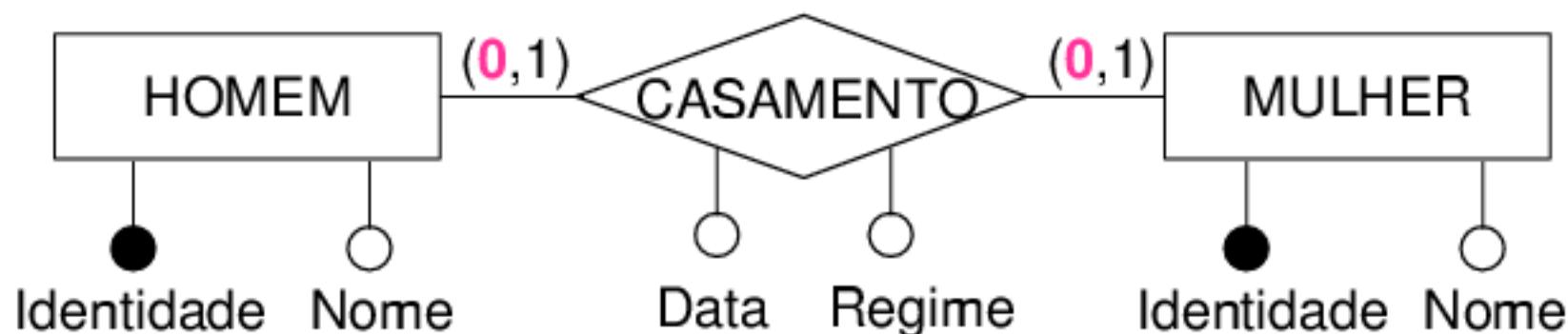
Homem (IdentH, Nome)

Casamento (IdentM, IdentH, Data, Regime)

IdentM referencia Mulher

IdentH referencia Homem

1:1 - ambas opcionais - Fusão de tabelas

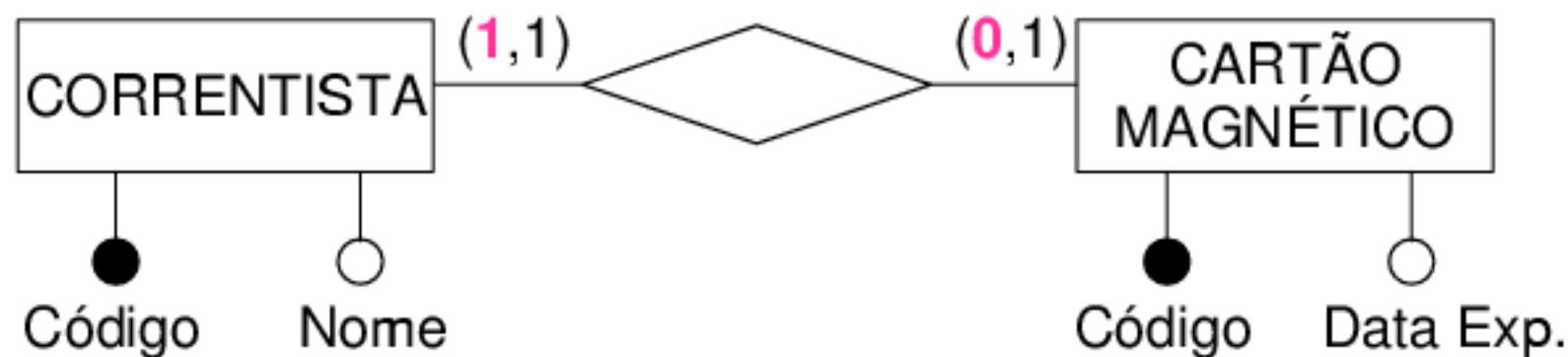


Casamento (IdentH, NomeH, Data, Regime, IdentM, NomeM)

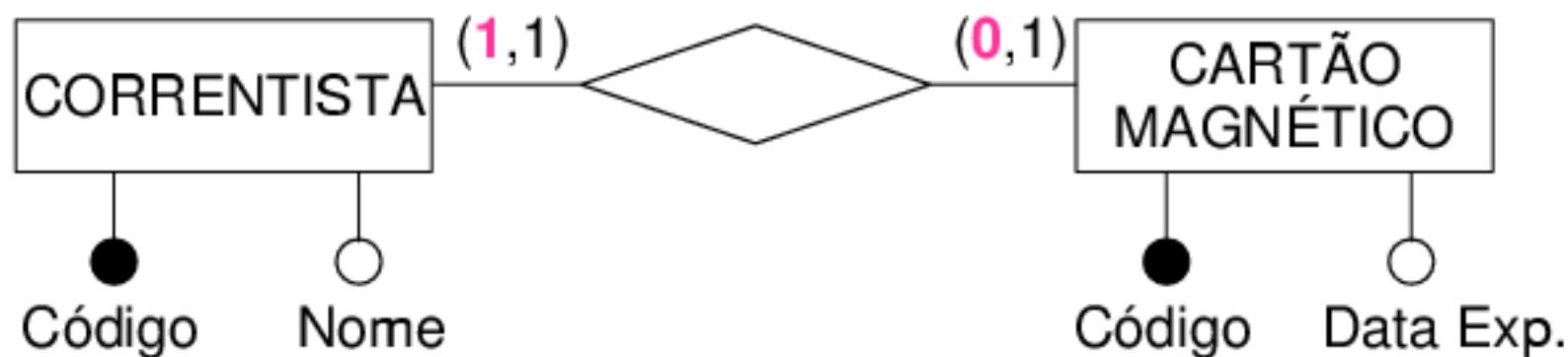
1:1 - ambas opcionais - discussão

- Solução por fusão de tabelas é inviável
 - Chave primária artificial
- Solução por adição de colunas melhor
 - Menor número de junções
 - Menor número de chaves
- Solução por tabela própria aceitável

1:1 - Uma entidade opcional, outra obrigatória

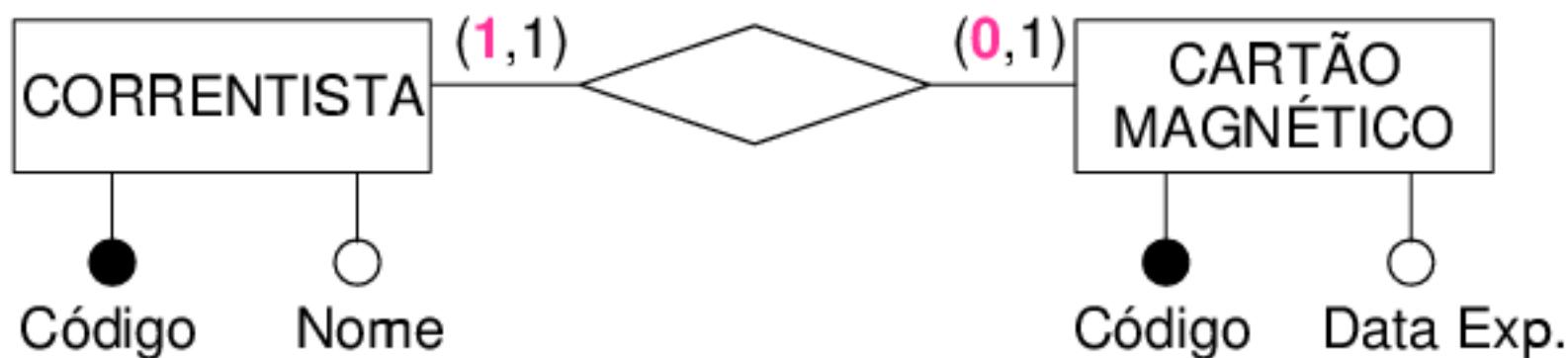


1:1 - opcional/obrigatória - Fusão de tabelas



Correntista (CodCorrent, Nome, CodCartão, DataExp)

1:1 - opcional/obrigatória - Adição de colunas

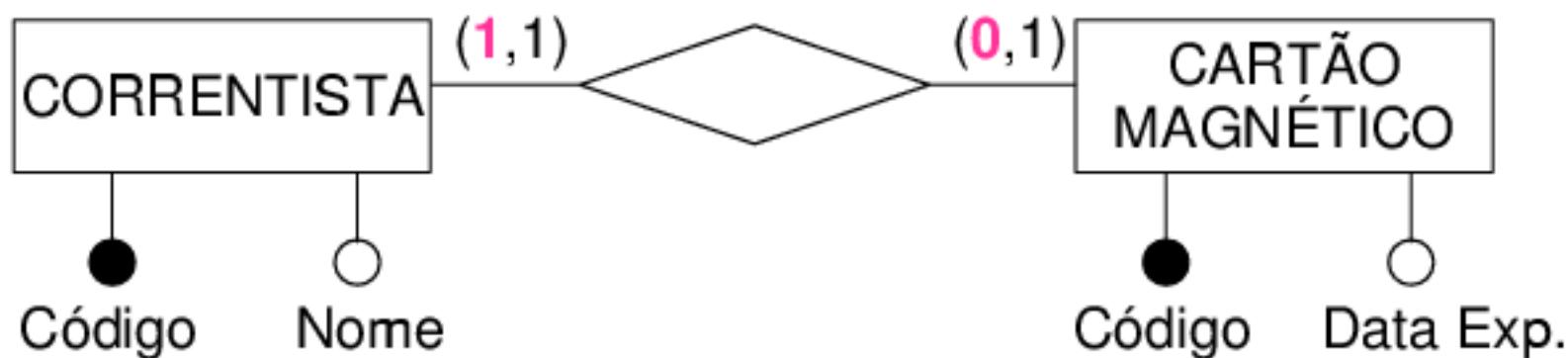


Correntista (CodCorrent, Nome)

Cartão(CodCartão, DataExp, **CodCorrent**)

CodCorrent referencia Correntista

1:1 - opcional/obrigatória - Tabela própria



Correntista (CodCorrent, Nome)

Cartão(CodCartão, DataExp)

CartãoCorrentista(CodCartão, CodCorrent)

CodCorrent referencia Correntista

CodCartão referencia Cartão

1:1 - opcional/obrigatória - discussão

- Solução por **tabela própria** é **pior que** a solução por **adição de colunas**
 - Maior número de junções
 - Maior número de índices
 - Nenhum tem problema de campos opcionais

1:1 - opcional/obrigatória - discussão

→ Adição de colunas versus fusão de tabelas

- Fusão de tabelas é melhor em termos de número de junções e número de chaves
- Adição de colunas é melhor em termos de campos opcionais
- Fusão de tabelas é considerada a melhor e adição de colunas é aceitável

1:1 - Ambas entidades tem participação obrigatória



1:1 - Ambas Obrigatórias - Fusão de tabelas



Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

1:1 - Ambas obrigatórias

→ Nenhuma das demais alternativas atende plenamente

→ Em ambas

- Entidades que participam do relacionamento seriam representadas através de duas tabelas distintas
- Estas tabelas teriam a mesma chave primária e relação um-para-um entre suas linhas
- Maior número de junções
- Maior número de chaves primárias

Relacionamentos 1:n

Tipo de Relacionamento	Regra de Implementação		
	Tabela Própria	Adição Coluna	Fusão Tabelas
(0,1) 	±	✓	✗
(0,1) 	±	✓	✗
(1,1) 	✗	✓	✗
(1,1) 	✗	✓	✗

✓ Alternativa preferida

✗ Não usar

± Pode ser usada

1:n - Caso 1

→ A entidade que tem cardinalidade máxima 1 é obrigatória



1:n - Caso 1 - Adição de colunas



Departamento (CodDept, Nome)

Empregado (CodEmp, Nome, CodDept, DataLota)

CodDept referencia Departamento

1:n - Caso 1 - Tabela própria



Departamento (CodDept, Nome)

Empregado (CodEmp, Nome)

Lotacao(CodEmp, CodDept, DataLota)

CodDept referencia Departamento

CodEmp referencia Empregado

1:n - Caso 1 - discussão

→ Fusão de tabelas

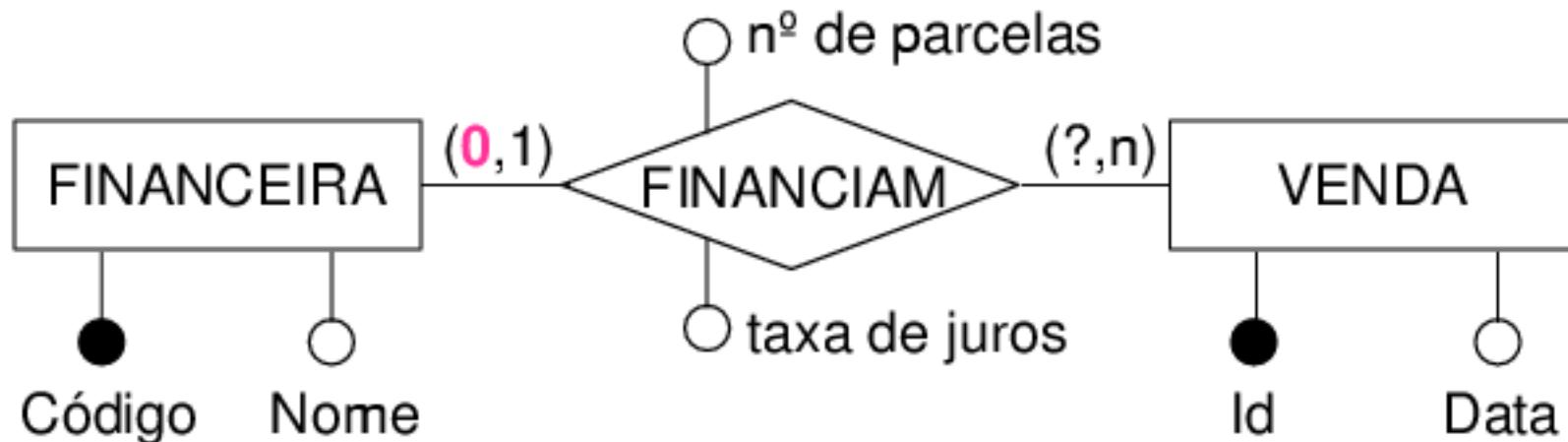
- Não se aplica
- Implicaria em
 - redundância de dados de departamento, ou
 - tabela aninhada

→ Adição de colunas é melhor que tabela própria

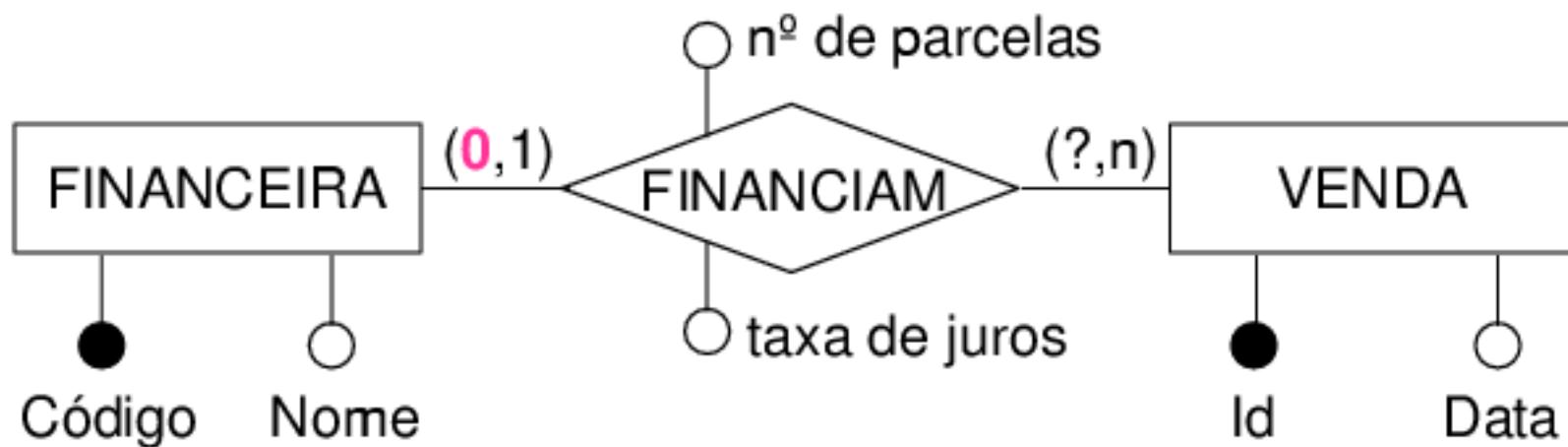
- Menor número de chaves
- Menor número de junções
- Não há o problema de campos opcionais

1:n - Caso 2

→ A entidade que tem cardinalidade máxima 1 é opcional



1:n - Caso 2 - Adição de colunas

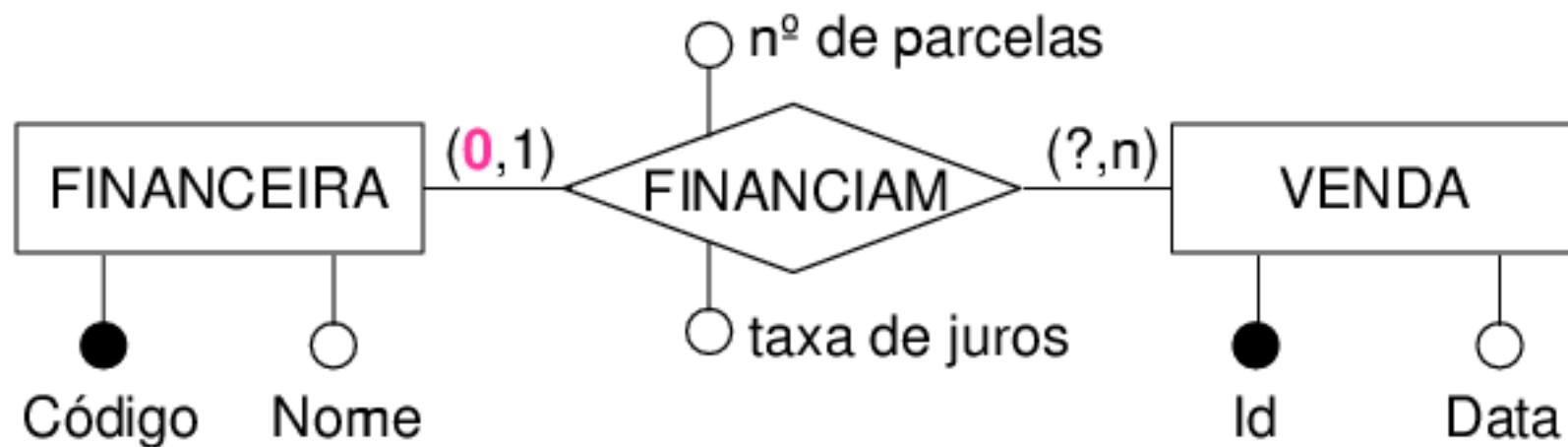


Financeira (CodFin, Nome)

Venda (IdVend, Data, **CodFin**, **NoParc**, **TxJuros**)

CodFin referencia Financeira

1:n - Caso 2 - Tabela própria



Financeira (CodFin,Nome)

Venda (IdVend,Data)

Fianciam (IdVend,CodFin,NoParc,TxJuros)

IdVend referencia Venda

CodFin referencia Financeira

1:n - caso 2 - discussão

- Implementação por tabela própria também é aceitável
 - É melhor em relação a campos opcionais
 - Perde em relação a junções e número de chaves

Relacionamentos n:n

Tipo de Relacionamento	Regra de Implementação		
	Tabela Própria	Adição Coluna	Fusão Tabelas
(0,n) 	✓	✗	✗
(0,n) 	✓	✗	✗
(1,n) 	✓	✗	✗

✓ Alternativa preferida

✗ Não usar

Relacionamentos n:n



Engenheiro (CodEng, Nome)

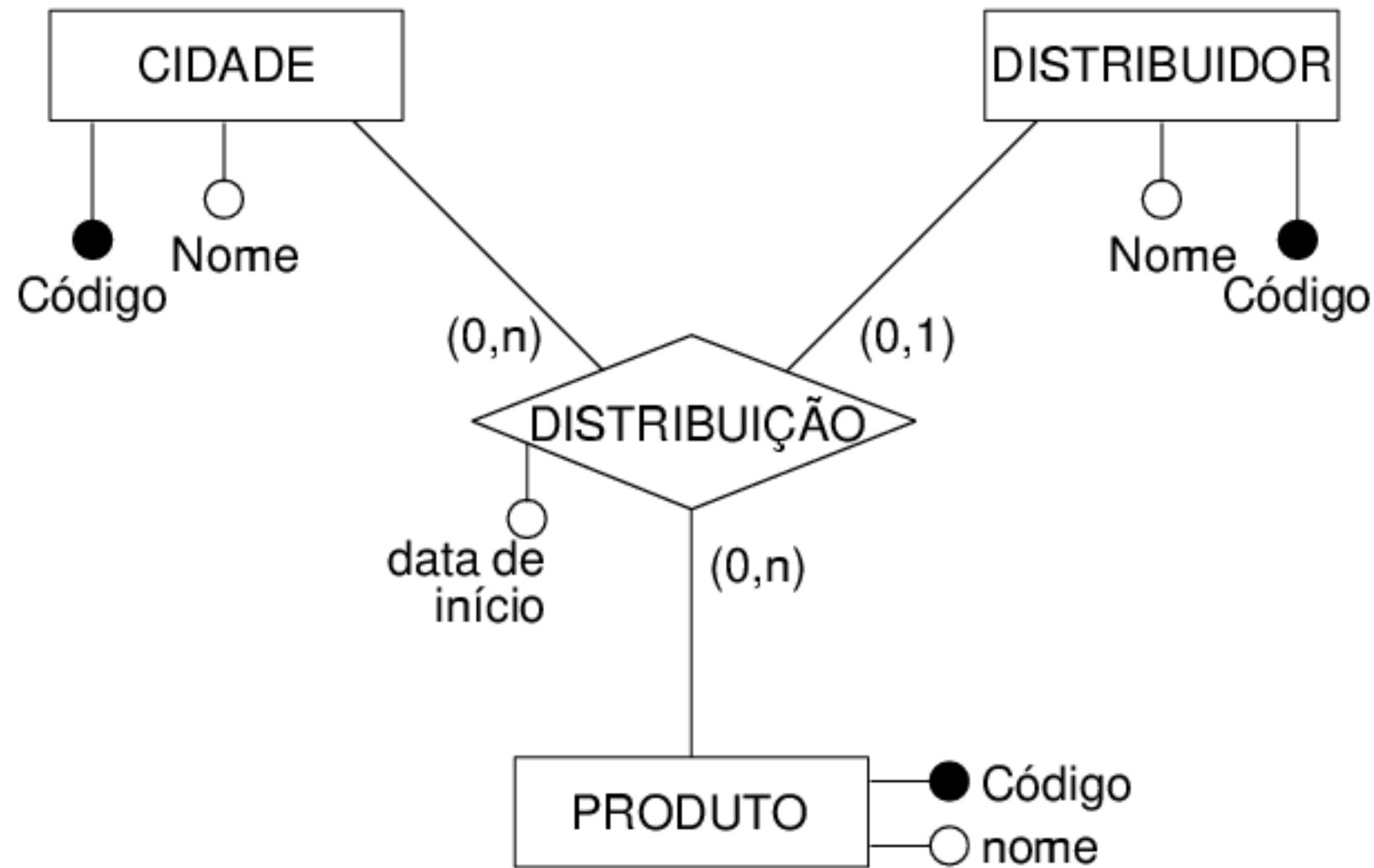
Projeto (CodProj, Título)

Atuação (CodEng,CodProj, Função)

CodEng referencia Engenheiro

CodProj referencia Projeto

Relacionamento grau > dois

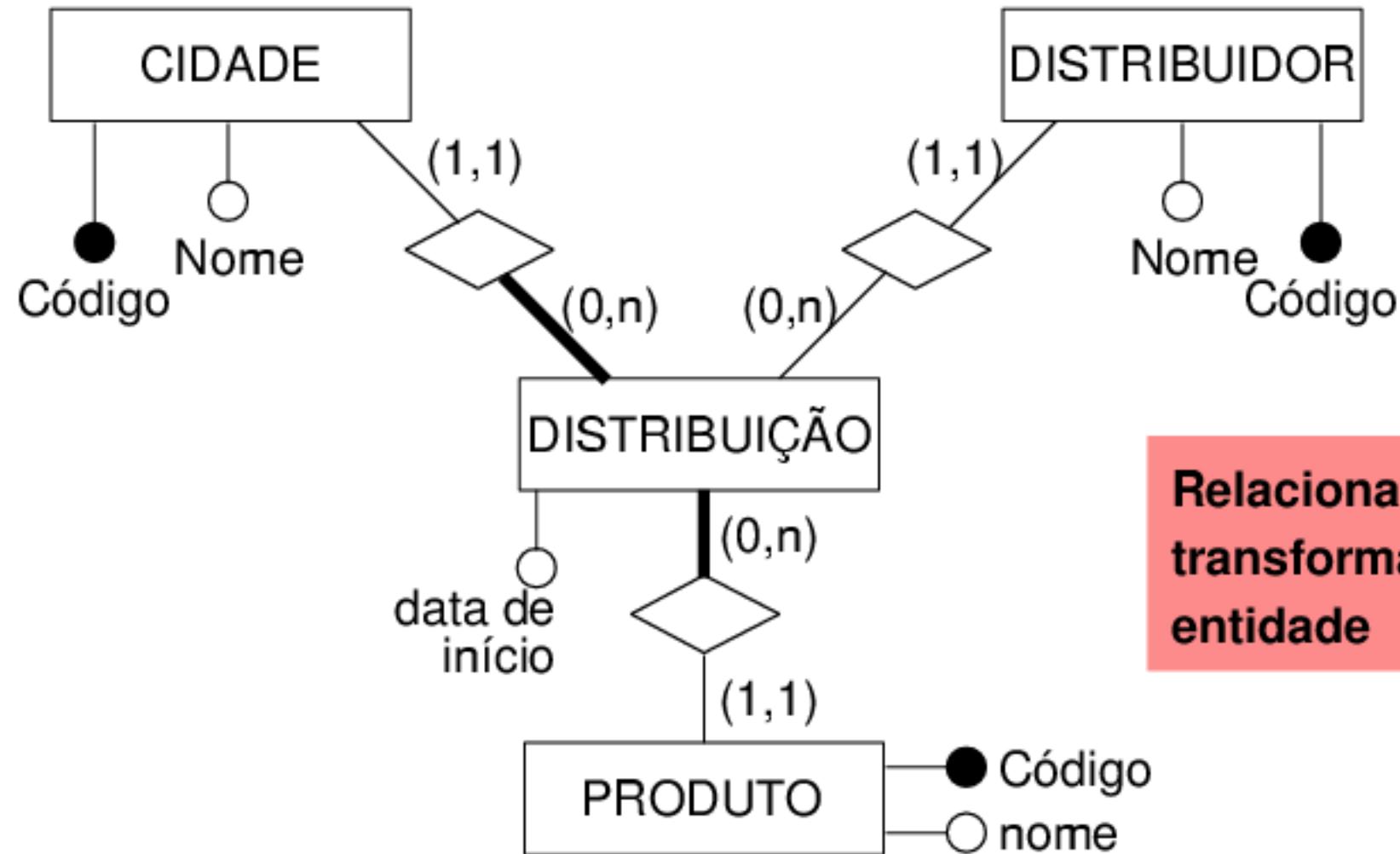


Relacionamento grau>2

→ Não são definidas regras específicas

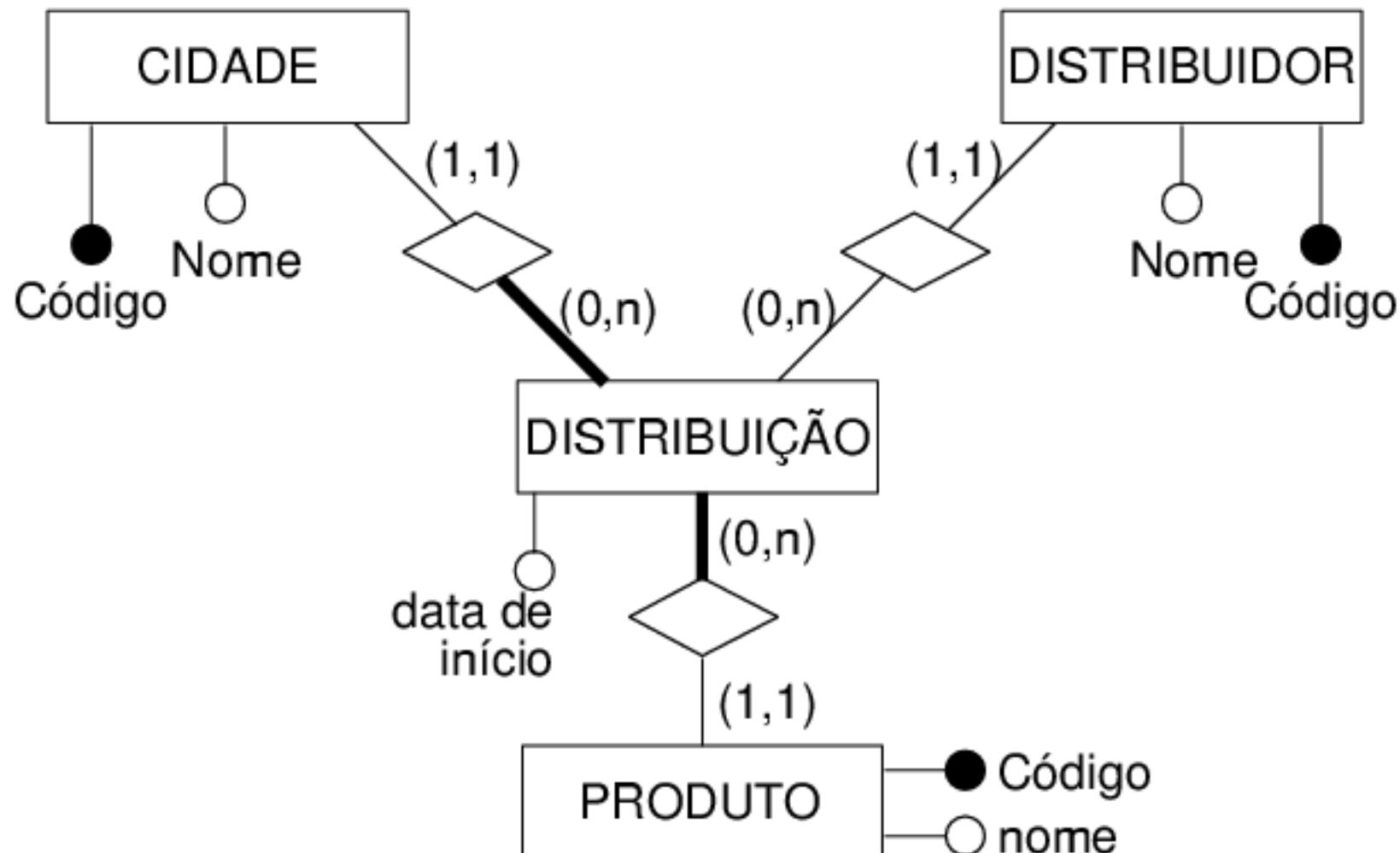
- O relacionamento é transformado em uma entidade
- São aplicadas regras de implementação relacionamentos binários

Relacionamento grau > 2

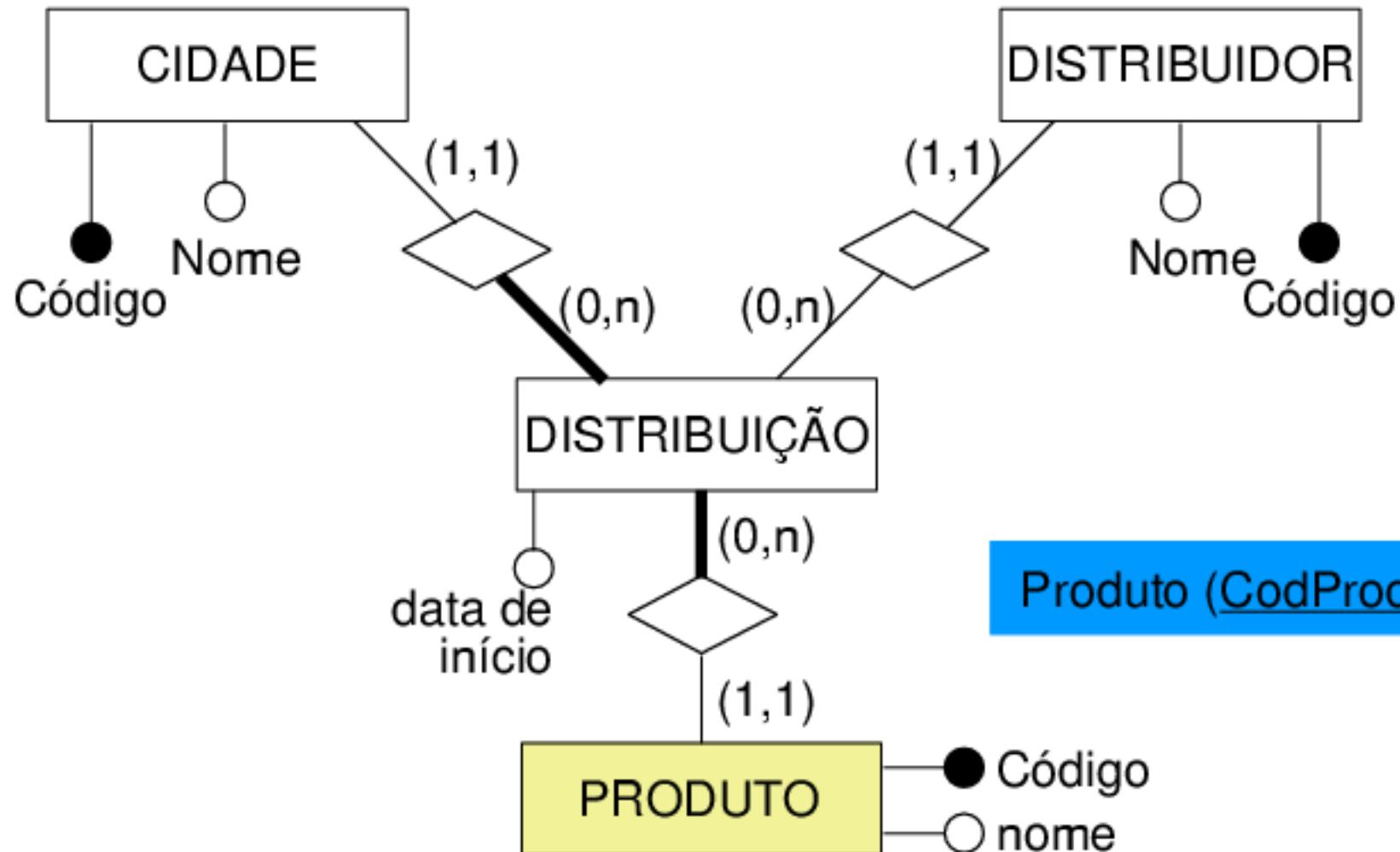


Relacionamento foi
transformado em
entidade

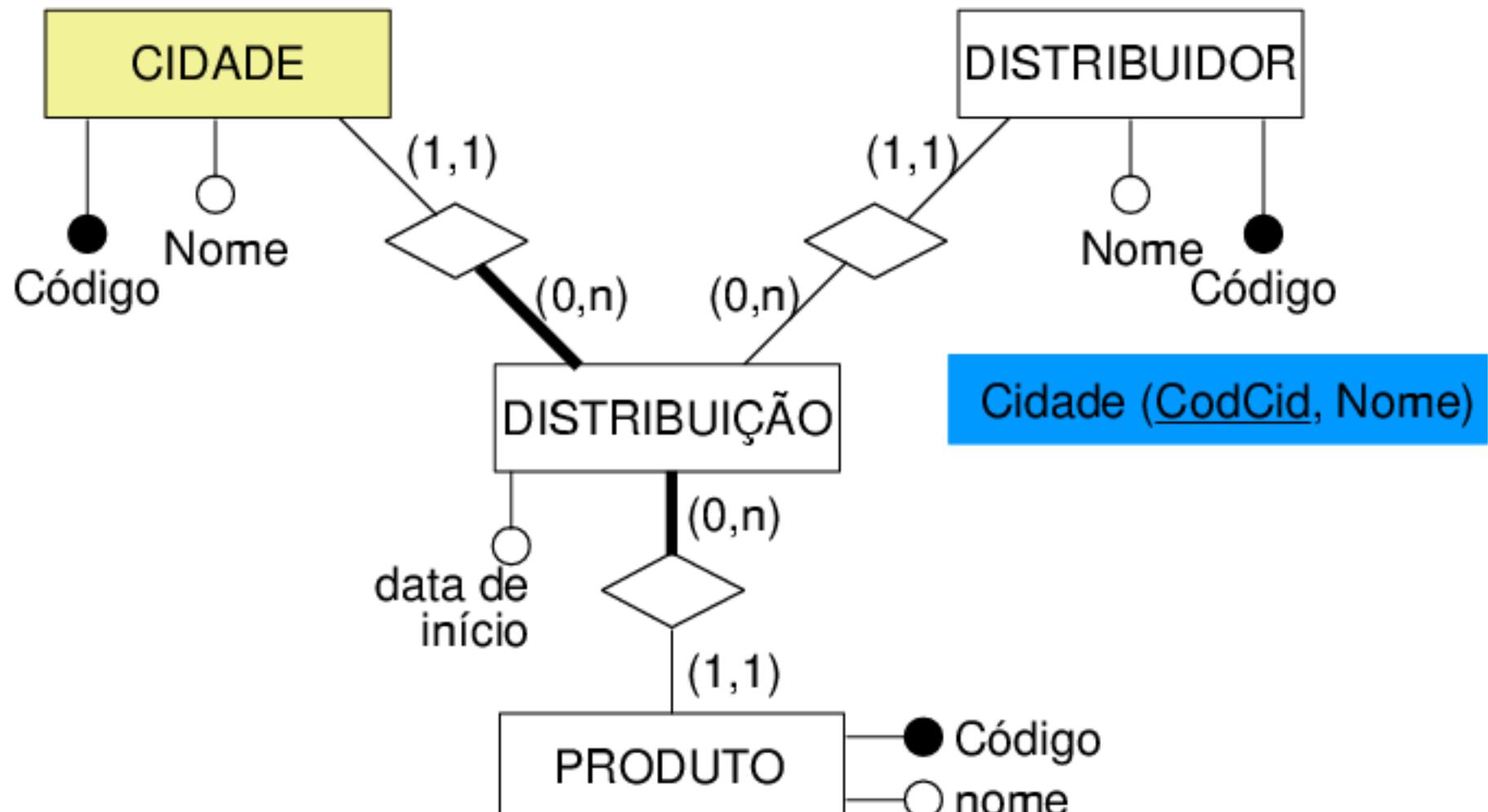
Relacionamento grau >2



Relacionamento grau > 2



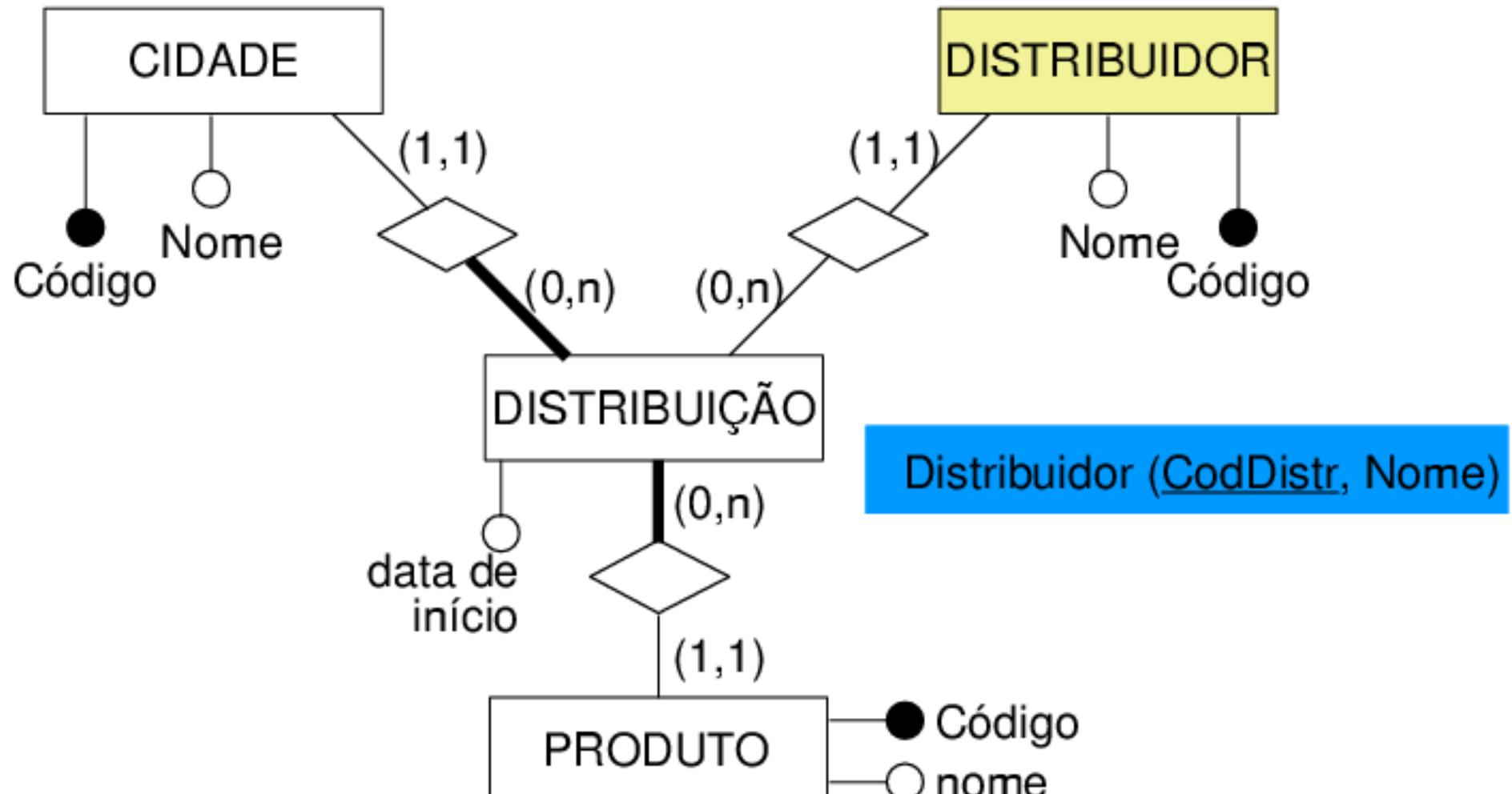
Relacionamento grau > 2



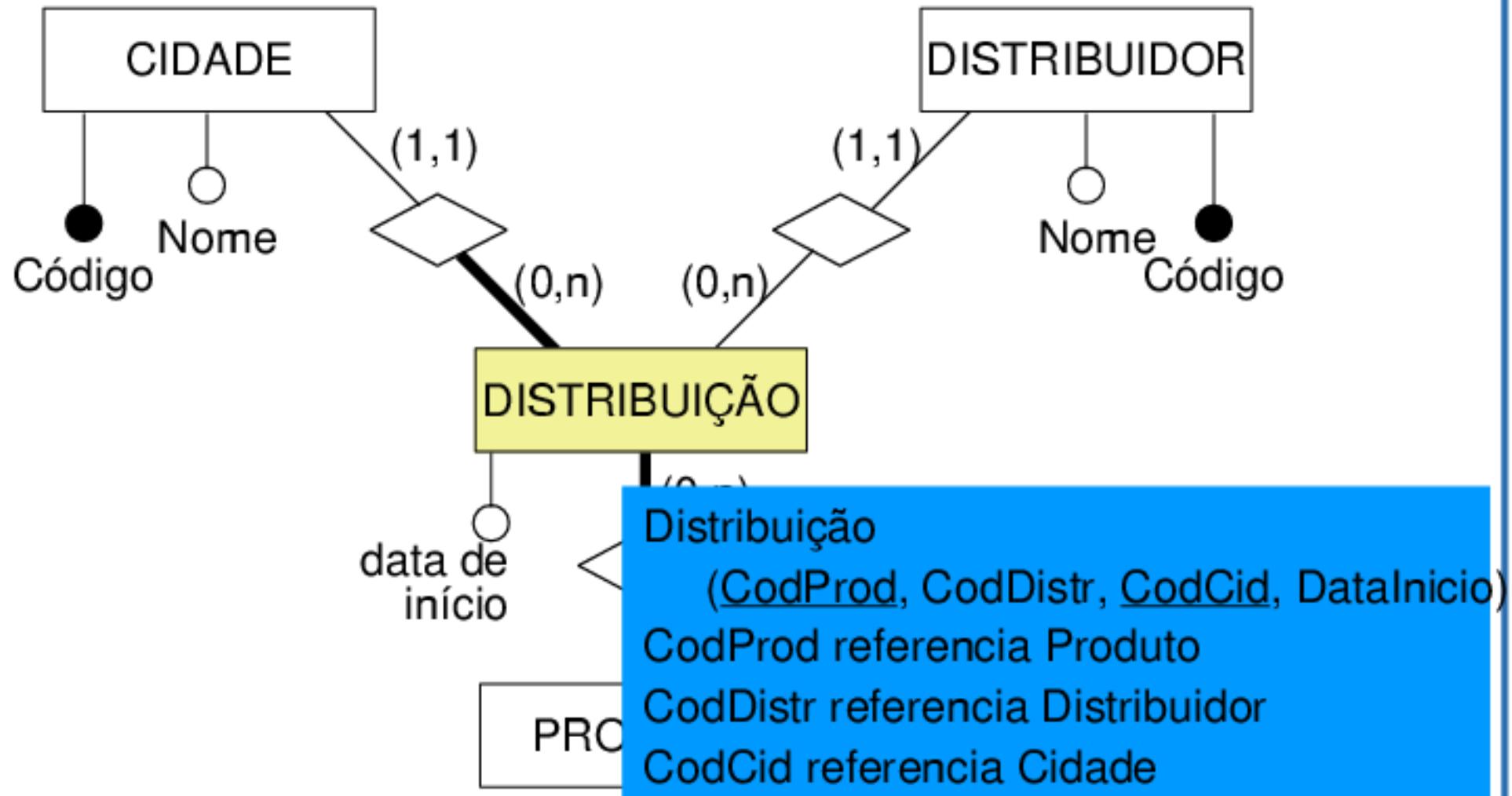
Cidade (CodCid, Nome)



Relacionamento grau > 2



Relacionamento grau >2



Na próxima aula...

→ Passos da transformação ER para relacional

- Tradução inicial de entidades e respectivos atributos
- Tradução de relacionamentos e respectivos atributos
- Tradução de generalizações/especializações**