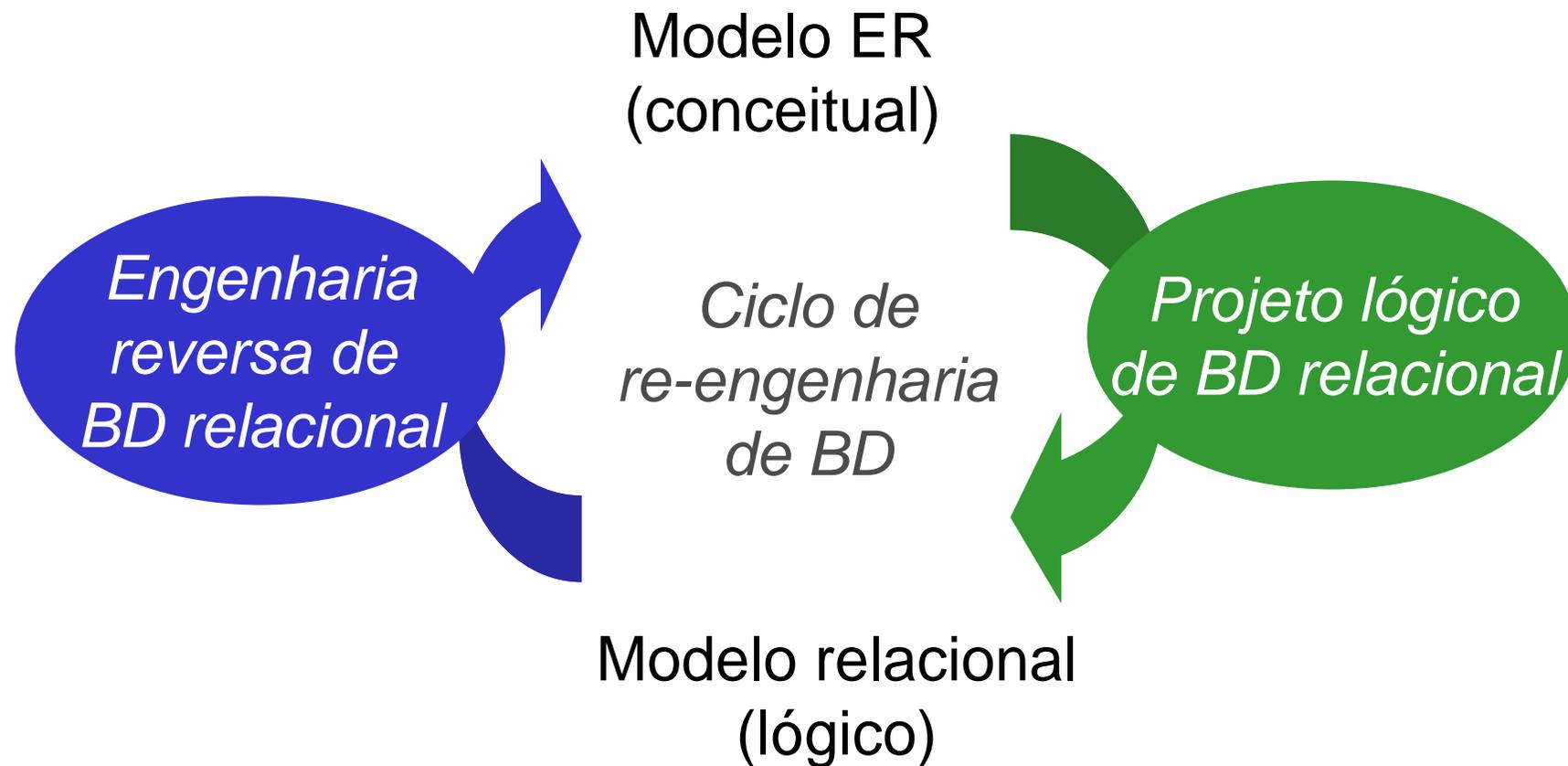


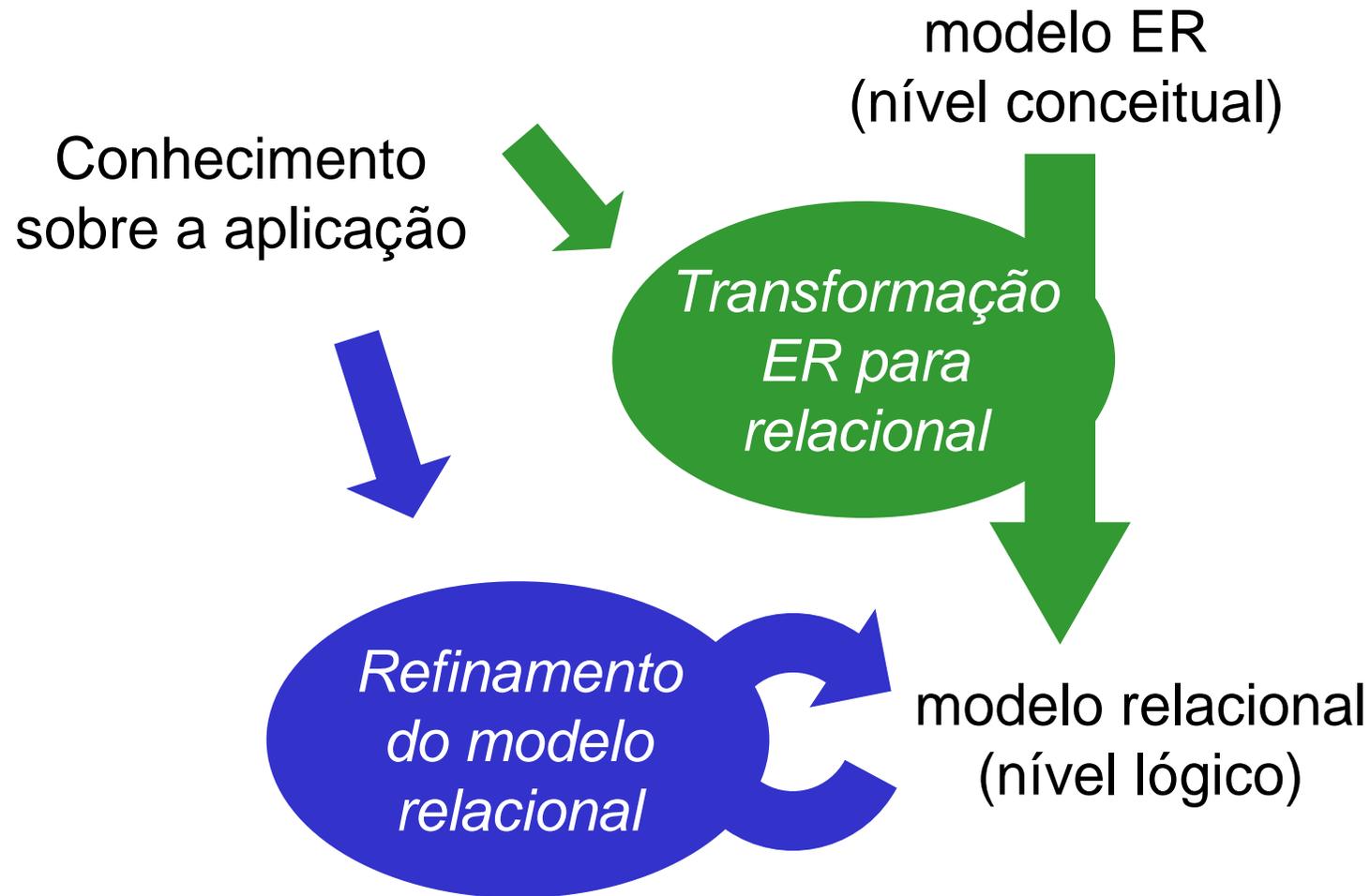
# *Transformações entre modelos*

## **Capítulo 5**

# Transformações entre modelos



# Projeto lógico



# *Transformação ER para relacional*

- **Regras gerais**
  - **Aplicáveis à maioria dos casos**
  - **Há situações**
    - **por exigências da aplicação, outros mapeamentos são 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 acesso 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 podem 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**

## Exercício 5.1

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.

# *Passos da transformação ER para relacional*

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

# *Implementação inicial de entidades*

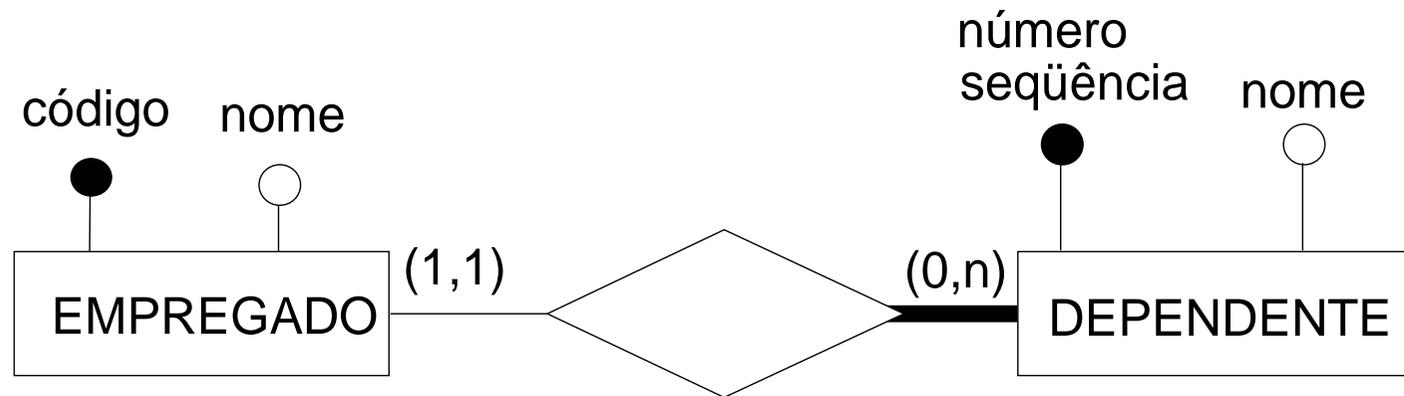
- **Cada entidade é traduzida para uma tabela**
- **Cada atributo da entidade define uma coluna desta tabela**
- **Atributos identificadores da entidade correspondem a chave primária da tabela.**
- **Tradução inicial:**
  - **Regras que seguem podem fundir tabelas**

# Implementação de entidade exemplo



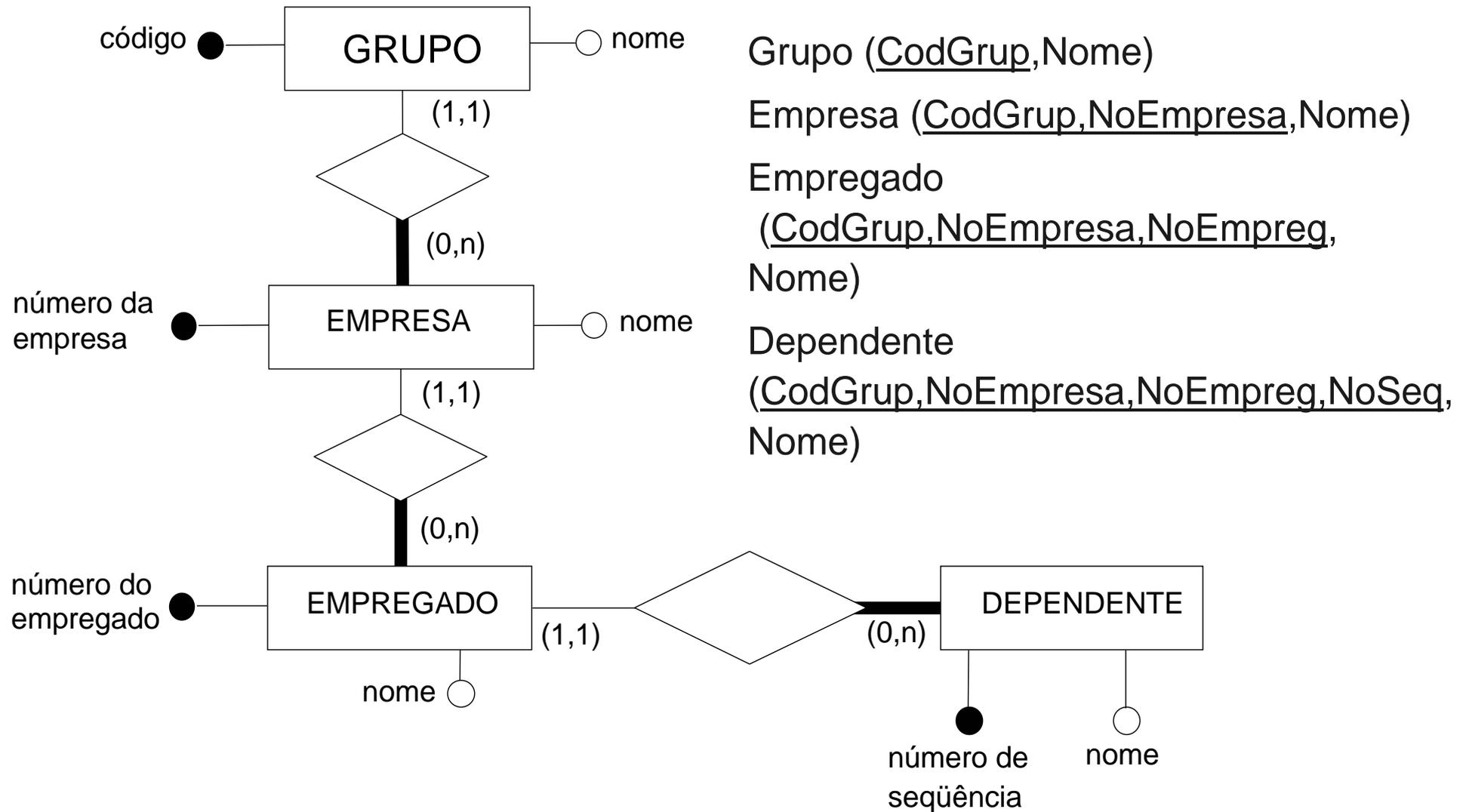
Pessoa (CodigoPess, Nome, Endereço, DataNasc, DataAdm)

# Tradução de entidade relacionamento identificador



Dependente (CodigoEmp, NoSeq, Nome)

# Relacionamento identificador recursão



Grupo (CodGrup, Nome)

Empresa (CodGrup, NoEmpresa, Nome)

Empregado  
(CodGrup, NoEmpresa, NoEmpreg,  
Nome)

Dependente  
(CodGrup, NoEmpresa, NoEmpreg, NoSeq,  
Nome)

# *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**

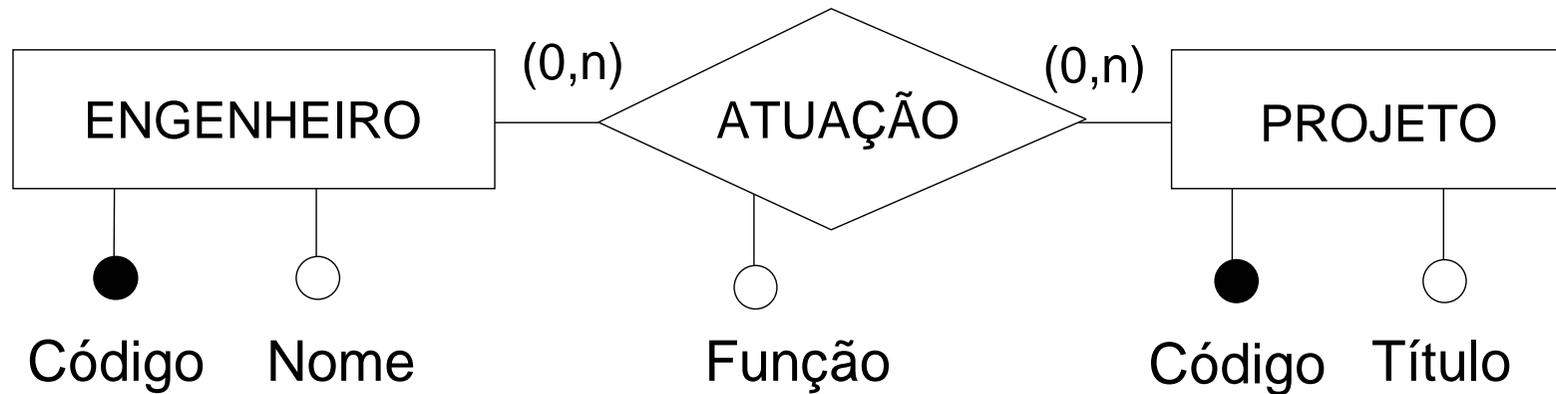
# *Nome da coluna chave primária*

- **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**

# *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)

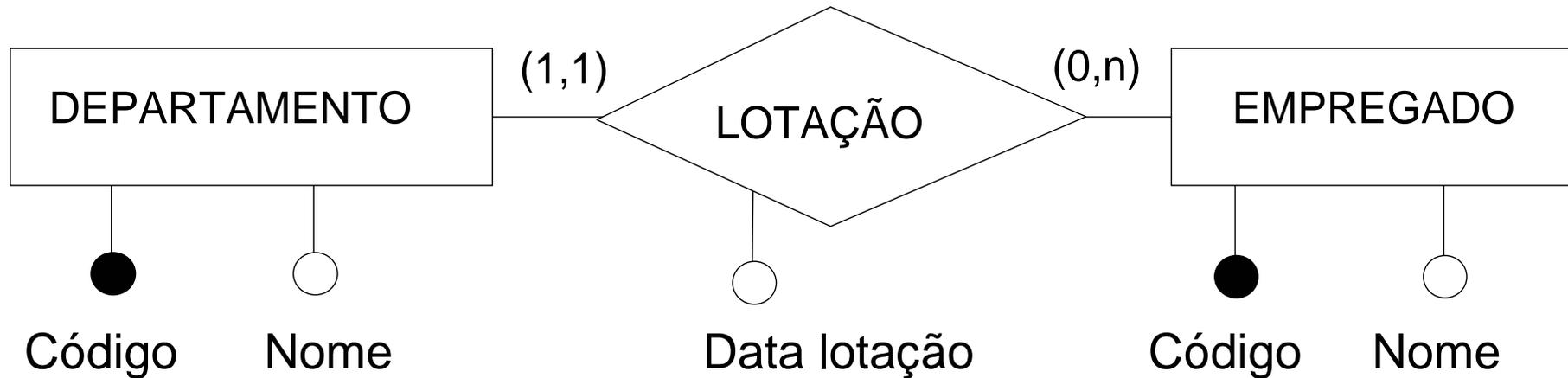
Projeto (CodProj, Título)

**Atuação (CodEng, CodProj, Função)**

**CodEng referencia Engenheiro**

**CodProj referencia Projeto**

# Adição de colunas

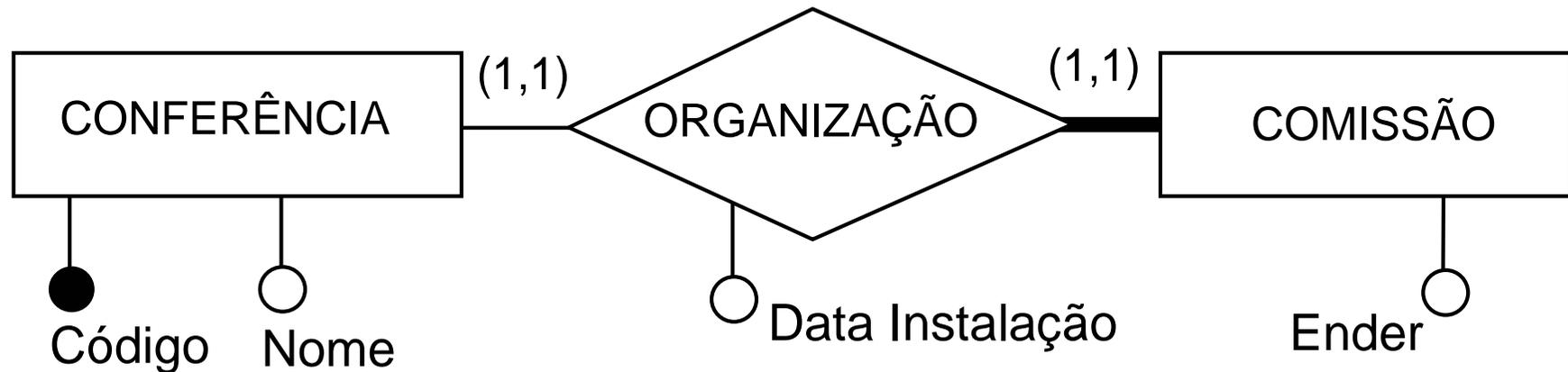


Departamento (CodDept, Nome)

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

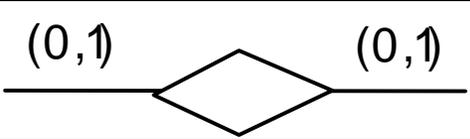
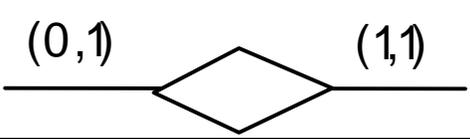
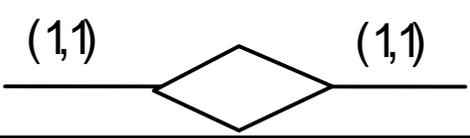
**CodDept referencia Departamento**

## *Fusão de tabelas*



Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

# Implementação de relacionamentos 1:1

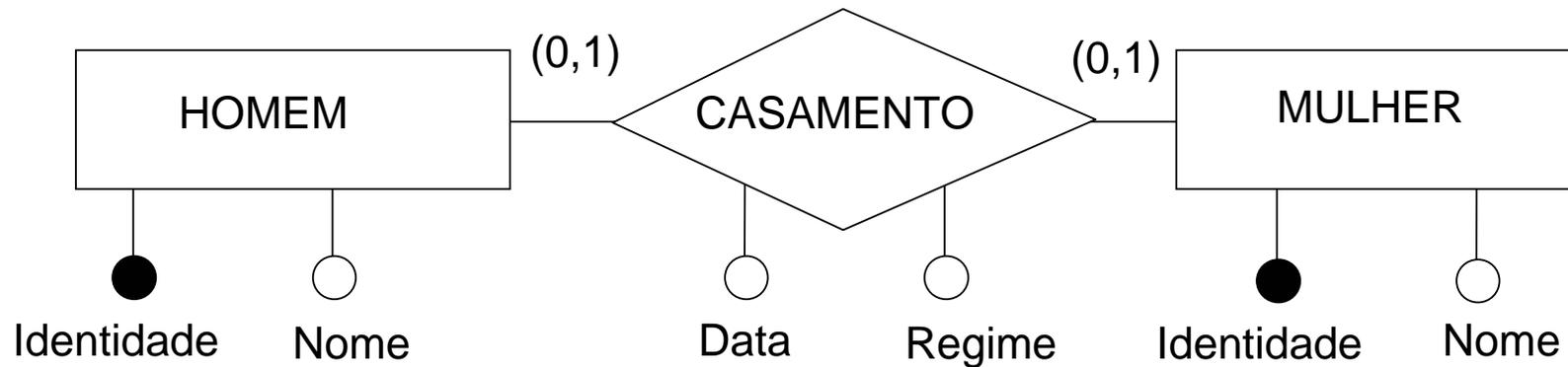
Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
(0,1)  (0,1)	±		
(0,1)  (1,1)		±	
(1,1)  (1,1)			

 Alternativa preferida

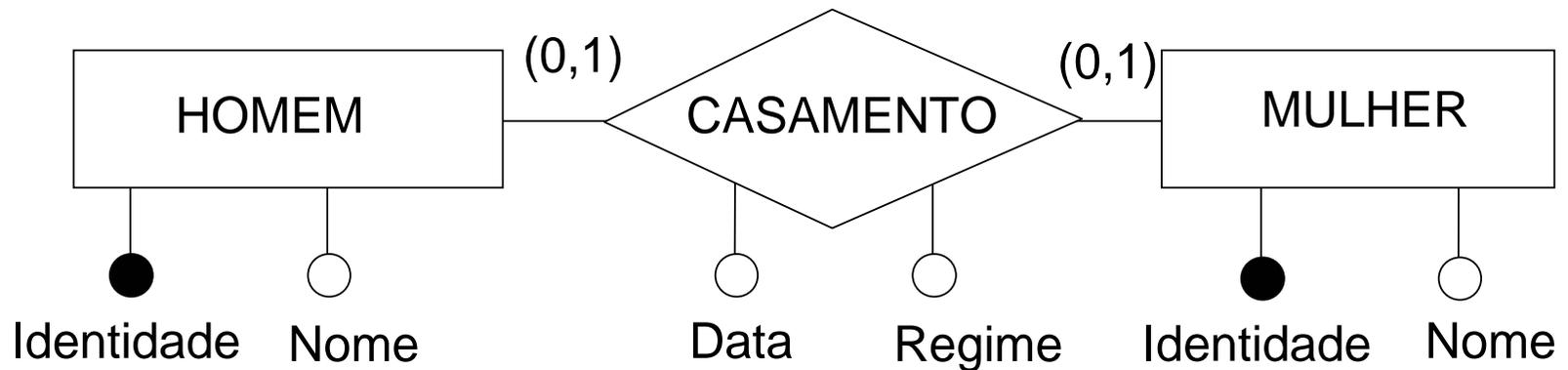
± Pode ser usada

 Não usar

# *1:1 - ambas entidades opcionais exemplo*



# 1:1 - ambas opcionais adição de colunas

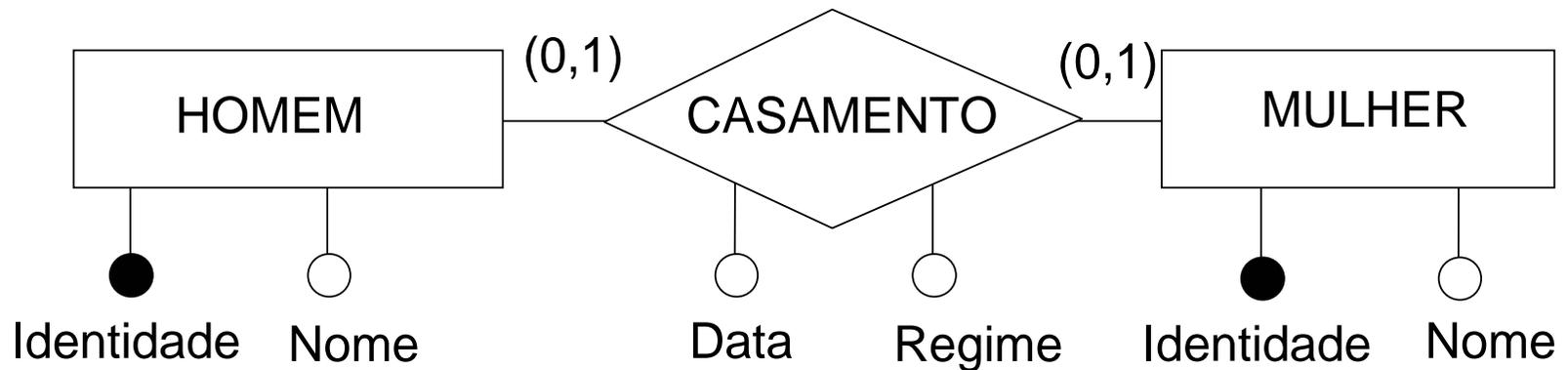


Mulher (IdentM, Nome, **IdentH**, Data, Regime)

**IdentH referencia Homem**

Homem (IdentH, Nome)

# 1:1 - ambas opcionais tabela própria



Mulher (IdentM, Nome)

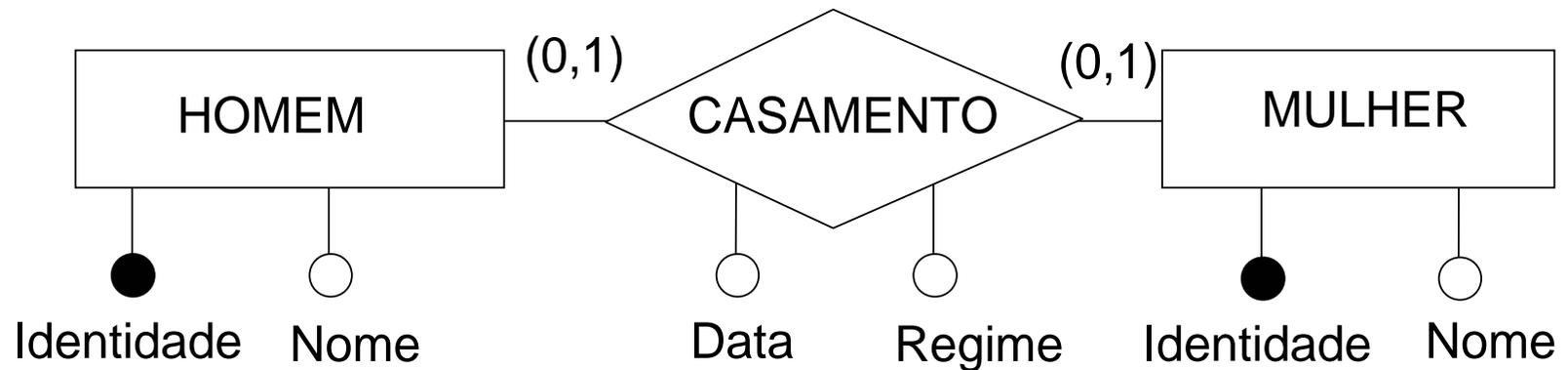
Homem (IdentH, Nome)

**Casamento (IdentM, IdentH, Data, Regime)**

**IdentM referencia Mulher**

**IdentH referencia Homem**

# 1:1 - ambas opcionais fusão de tabelas

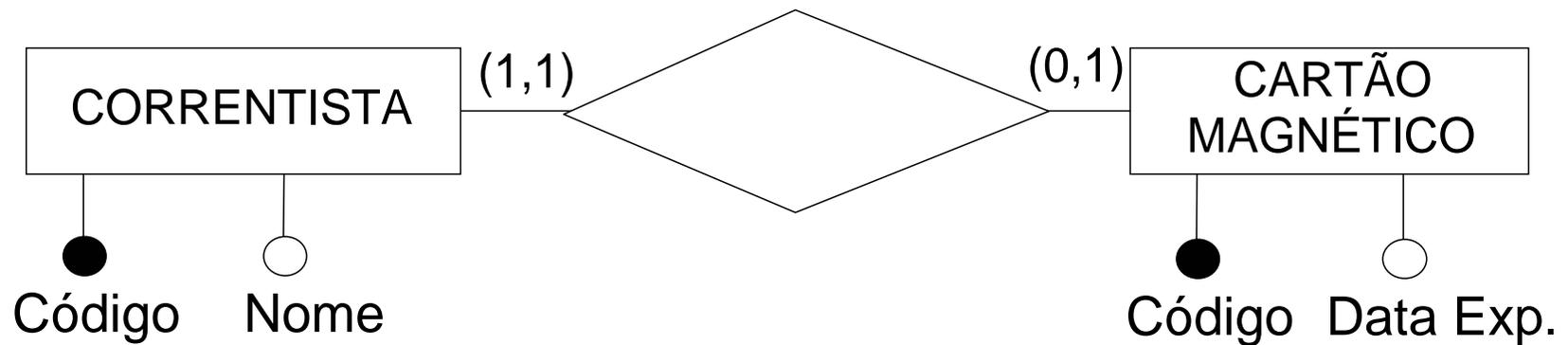


Casamento (IdentM, IdentH, Data, Regime, NomeH, NomeM)

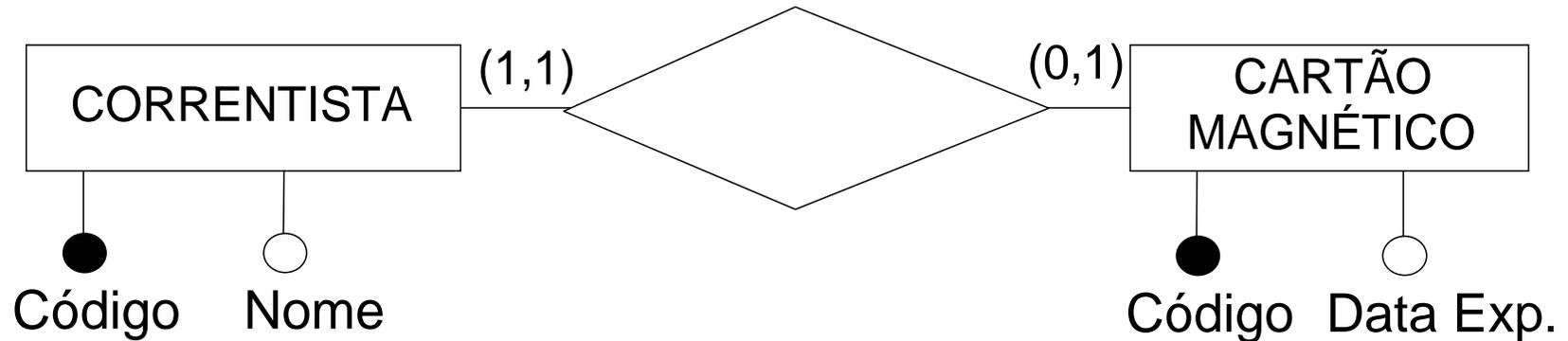
## ***1:1 - ambas opcionais discussão***

- **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

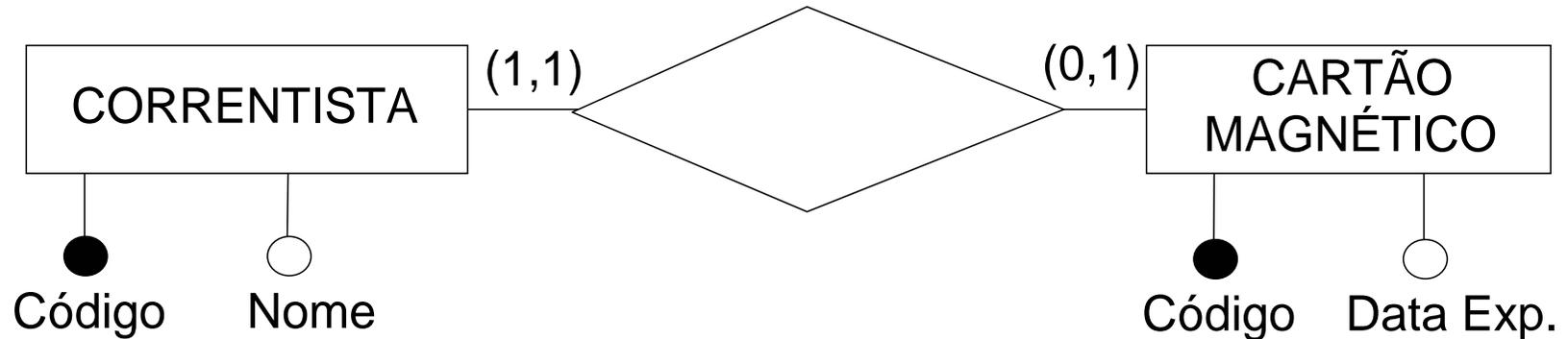


# 1:1 - opcional/obrigatória fusão de tabelas



Correntista (CodCorrent, Nome, CodCartão, DataExp)

# 1:1 - opcional/obrigatória adição de colunas

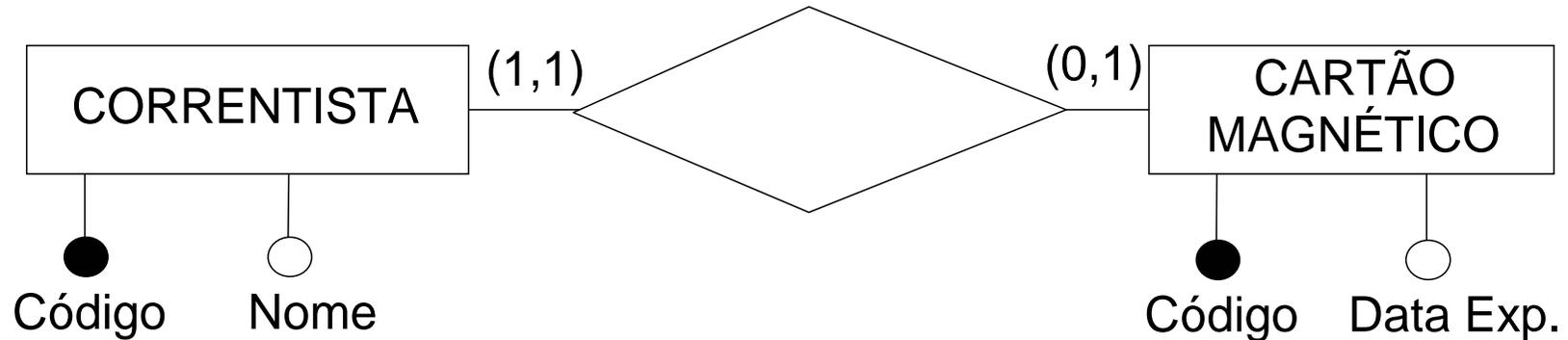


Correntista (CodCorrent, Nome)

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**

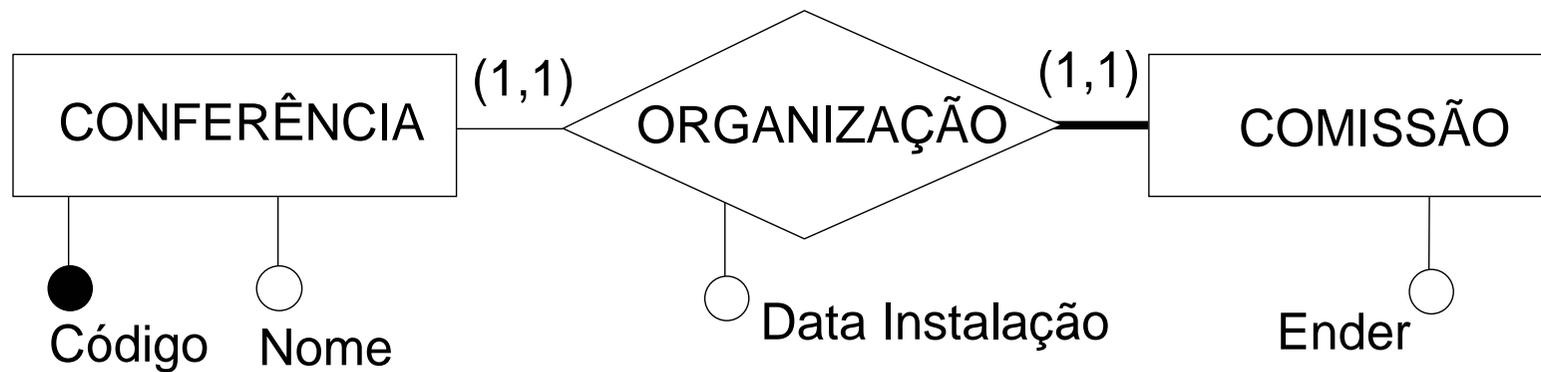
## ***1:1 - opcional/obrigatória discussão***

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

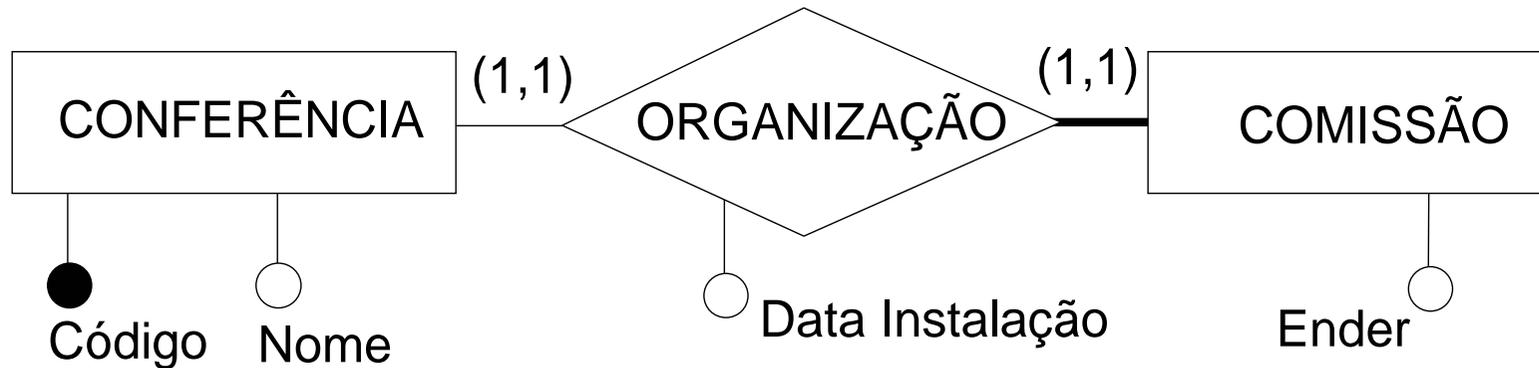
# ***1:1 - opcional/obrigatória discussão***

- **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 em 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

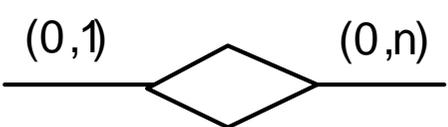
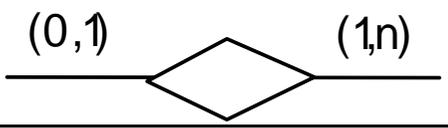
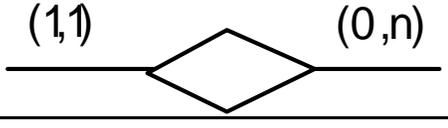
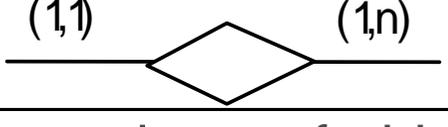


Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

## ***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,n)	±		
(0,1)  (1,n)	±		
(1,1)  (0,n)			
(1,1)  (1,n)			

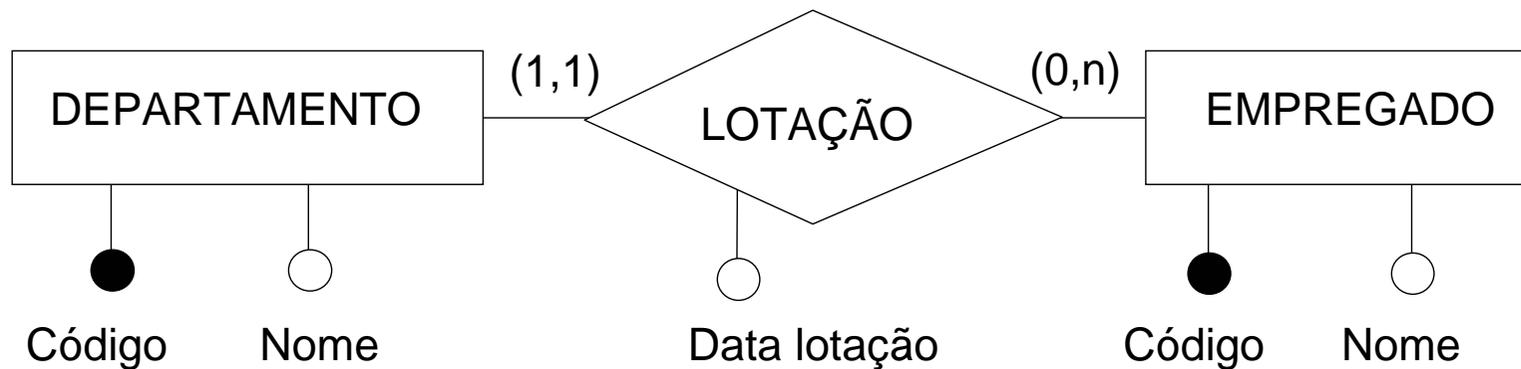
 Alternativa preferida

± Pode ser usada

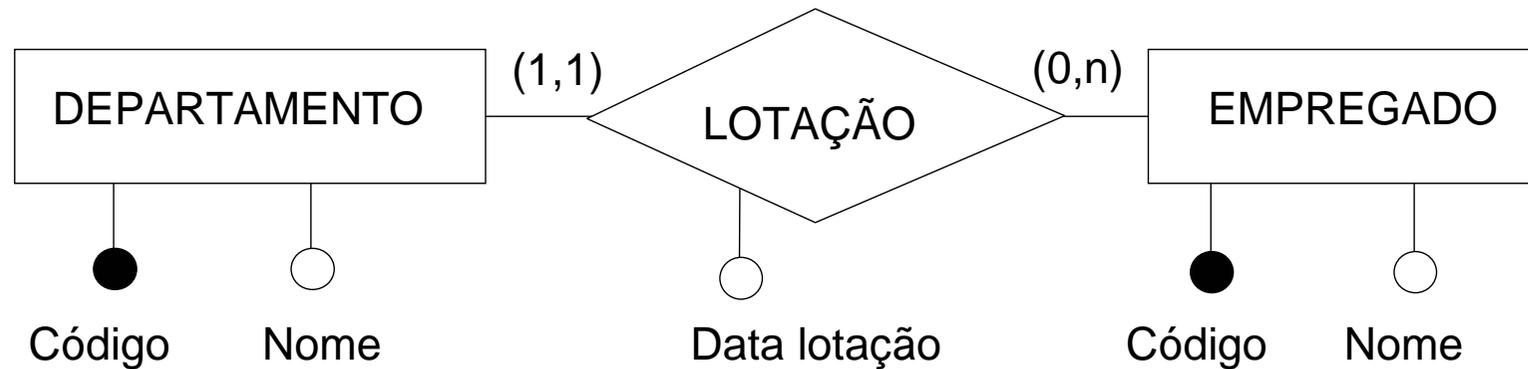
 Não usar

## 1:n - caso 1

- A entidade que tem cardinalidade máxima 1 é obrigatória



## 1:n - caso 1 adição de colunas

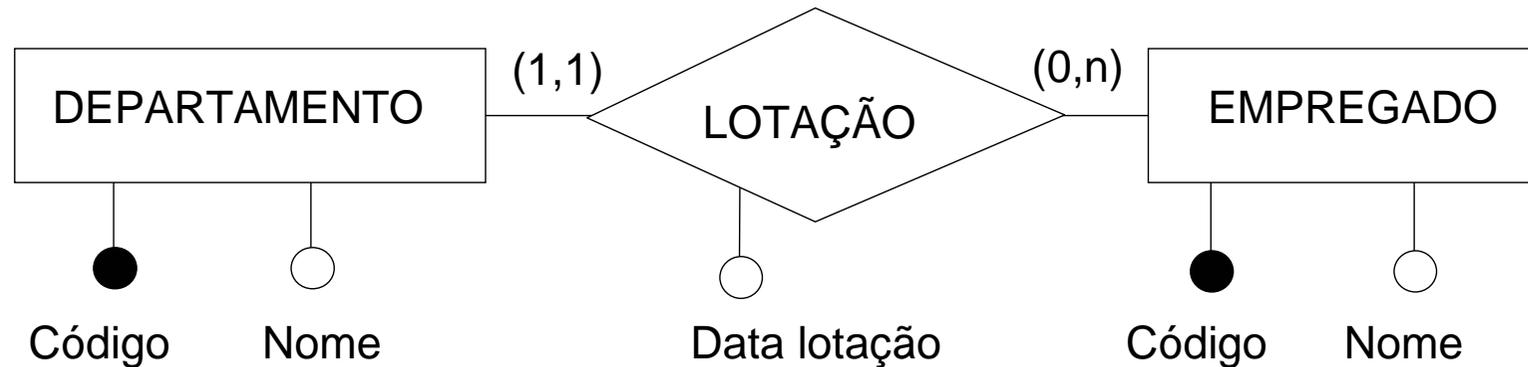


Departamento (CodDept, Nome)

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**

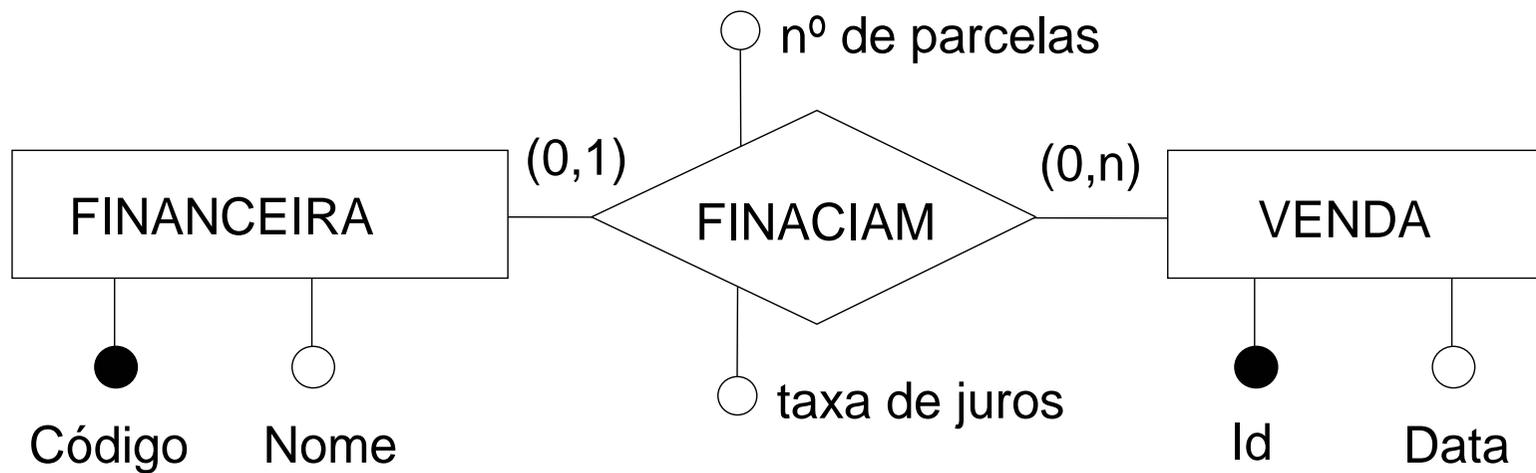
**CodEmp referencia Empregado**

# ***1:n - caso 1 discussão***

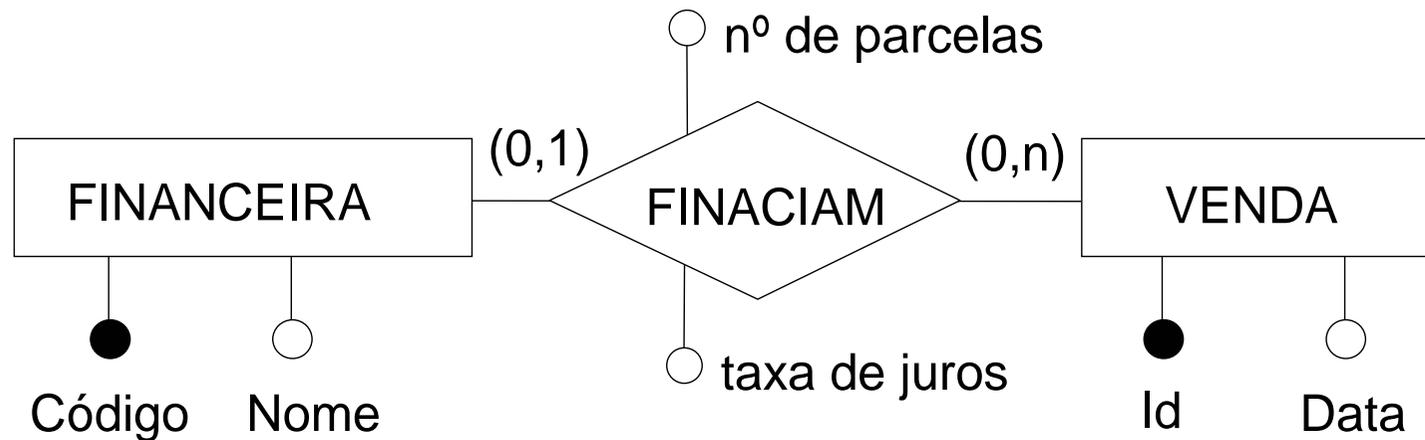
- **Fusão de tabelas**
  - Não se aplica
  - Implicaria em
    - redundância de dados de departamento, ou
    - tabela aninhada
- **Adição de colunas é melhor que tabela própria**
  - Menor número de chaves
  - Menor número de junções
  - Não há o problema de campos opcionais

## 1:n - caso 2

- A entidade que tem cardinalidade máxima 1 é opcional



## 1:n - caso 2 adição de colunas

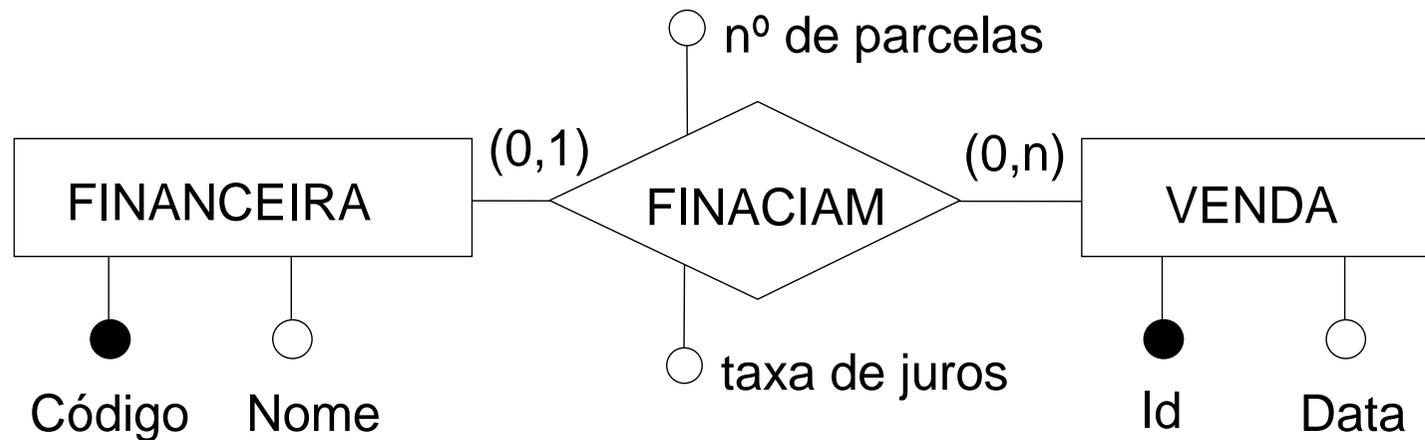


Financeira (CodFin, Nome)

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**

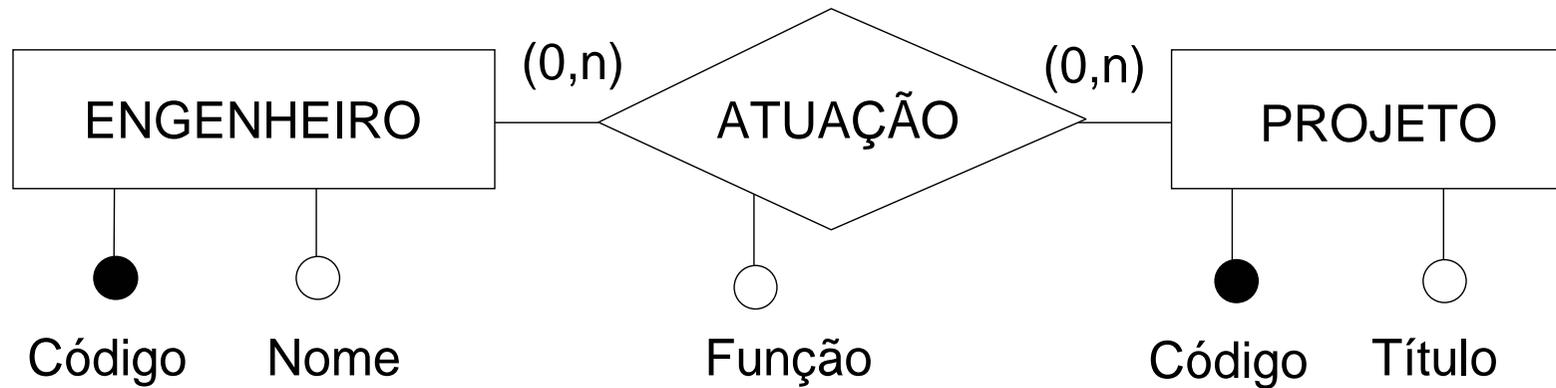
# Relacionamentos n:n

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
(0,n)  (0,n)			
(0,n)  (1,n)			
(1,n)  (1,n)			

Alternativa preferida

Não usar

# Relacionamentos n:n



Engenheiro (CodEng, Nome)

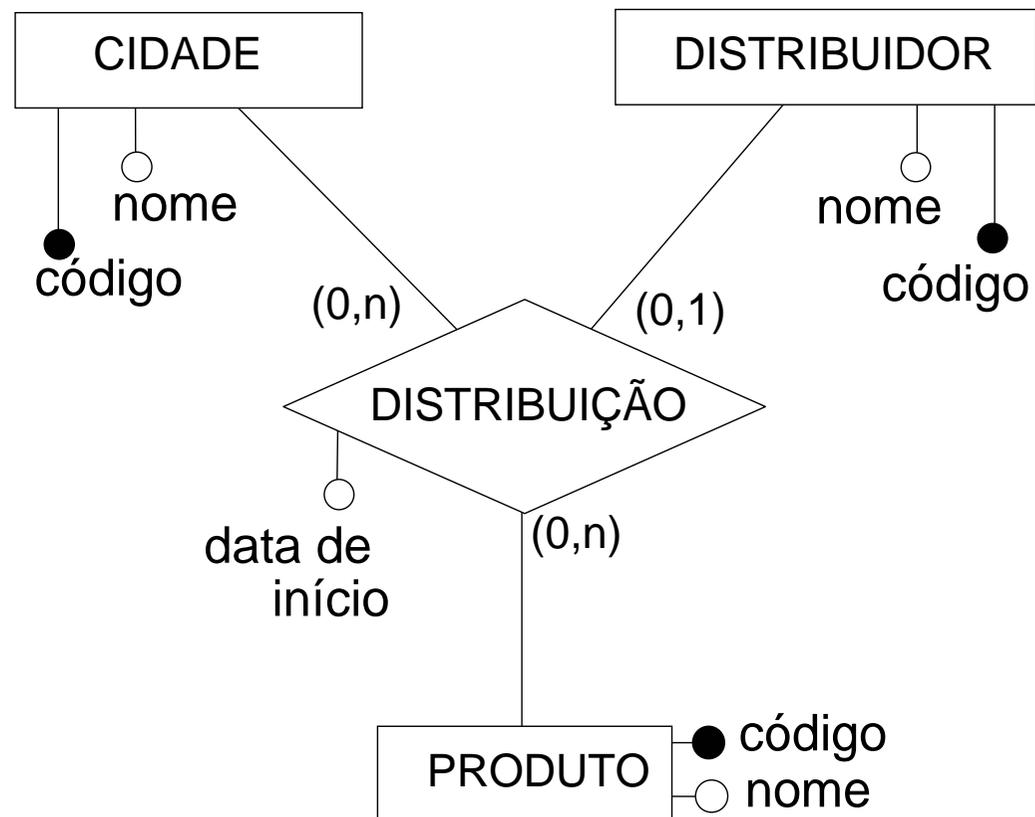
Projeto (CodProj, Título)

Atuação (CodEng, CodProj, Função)

**CodEng referencia Engenheiro**

**CodProj referencia Projeto**

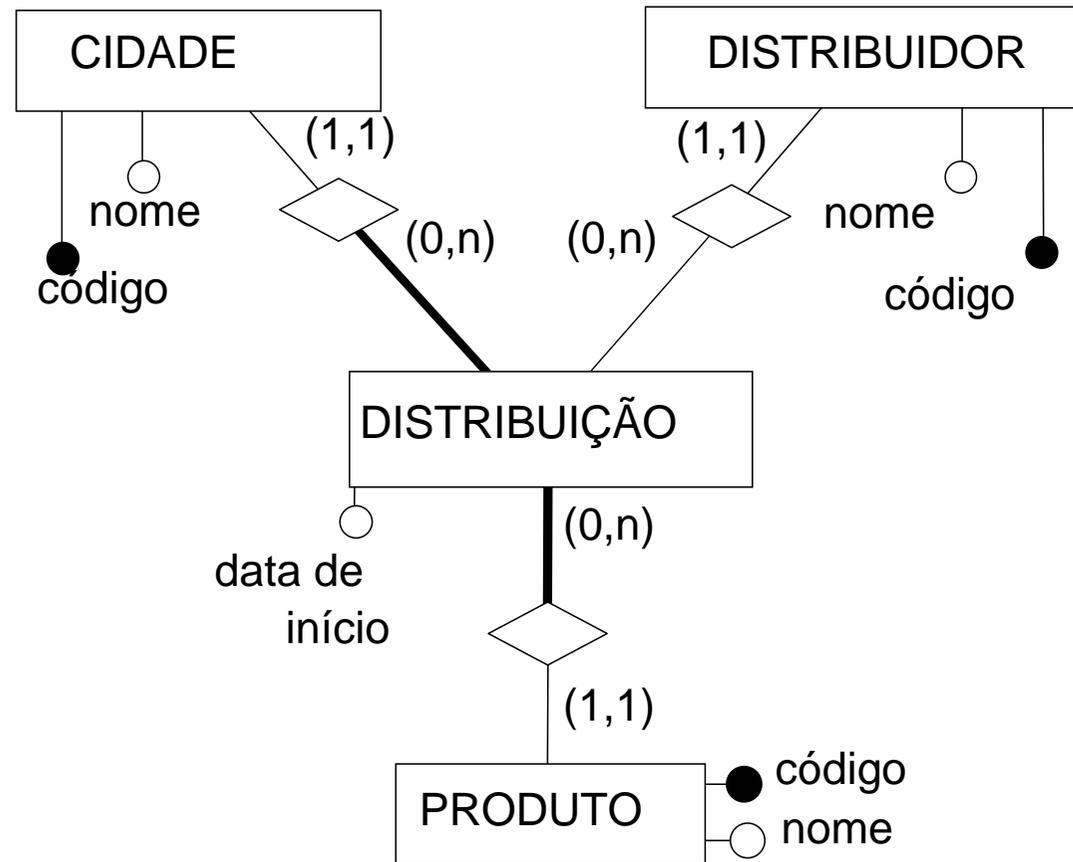
# Relacionamento grau > dois



## *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 grau>2*

Produto (CodProd,Nome)

Cidade (CodCid,Nome)

Distribuidor (CodDistr,Nome)

**Distribuição (CodProd,CodDistr,CodCid,DataInicio)**

**CodProd referencia Produto**

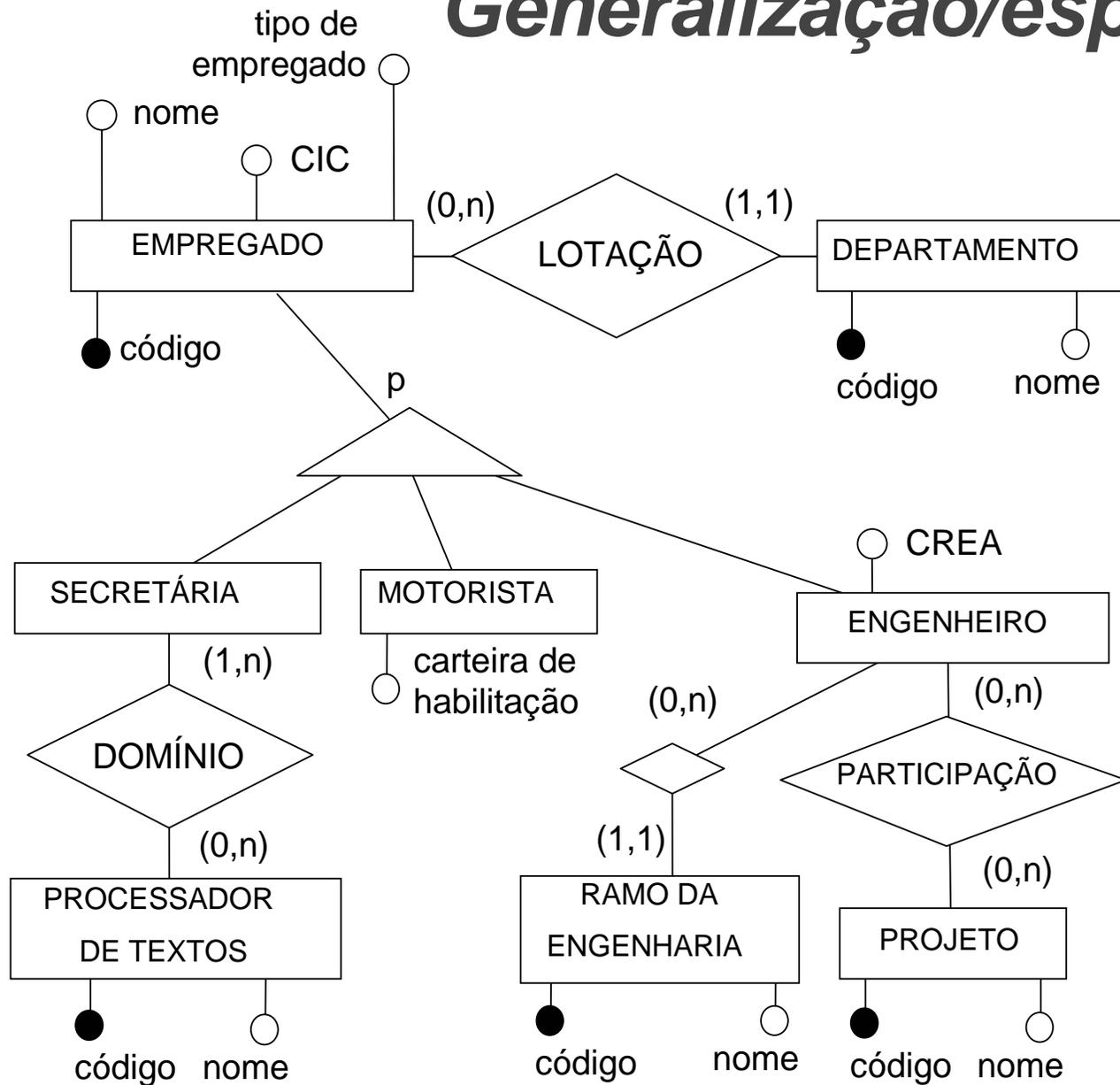
**CodDistr referencia Distribuidor**

**CodCid referencia Cidade**

# *Implementação de generalização/especialização*

- **Duas alternativas básicas**
  - uso de uma tabela para cada entidade
  - uso de uma única tabela para toda hierarquia
- **Outra alternativa (exótica)**
  - Subdivisão de entidade genérica

# Generalização/especialização exemplo



# ***Uma tabela por hierarquia***

- **Todas tabelas referentes às especializações são fundidas em uma única tabela**
- **Tabela contém:**
  - **Chave primária correspondente ao identificador da entidade mais genérica**
  - **Caso não exista, adicionar uma coluna Tipo**
  - **Uma coluna para cada atributo da entidade genérica**
  - **Colunas referentes aos relacionamentos dos quais participa a entidade genérica e que sejam implementados através da alternativa de adicionar colunas à tabela da entidade genérica**

*segue*

# *Uma tabela por hierarquia*

- **Tabela contém (continuação):**
  - **Uma coluna para cada atributo de cada entidade especializada (opcional)**
  - **Colunas referentes aos relacionamentos dos quais participa cada entidade especializada e que sejam implementados através da alternativa de adicionar colunas à tabela da entidade (campo opcional)**

# *Uma tabela por hierarquia*

Emp (CódigoEmp, Tipo, Nome, CIC, CodigoDept,  
CartHabil, CREA, CódigoRamo)

CódigoDept referencia Depto

CódigoRamo referencia Ramo

Depto (CódigoDept, Nome)

Ramo (CódigoRamo, Nome)

ProcessTexto (CódigoProc, Nome)

Domínio (CódigoEmp, CódigoProc)

CódigoEmp referencia Emp

CódigoProc referencia ProcessTexto

Projeto (CódigoProj, Nome)

Participação (CódigoEmp, CódigoProj)

CódigoEmp referencia Emp

CódigoProj referencia Projeto

## ***Uma tabela por entidade especializada***

- **Criar uma tabela para cada entidade que compõe a hierarquia**
- **Incluir a chave primária da tabela correspondente à entidade genérica, em cada tabela correspondente a uma entidade especializada**

# ***Uma tabela por entidade especializada***

Emp (CódigoEmp, Tipo, Nome, CIC, CódigoDept)

CódigoDept referencia Depto

Motorista(CódigoEmp, CartHabil)

CódigoEmp referencia Emp

Engenheiro(CódigoEmp, CREA, CódigoRamo)

CódigoEmp referencia Emp

CódigoRamo referencia Ramo

Depto (CódigoDept, Nome)

Ramo (CódigoRamo, Nome)

ProcessTexto (CódigoProc, Nome)

Domínio (CódigoEmp, CódigoProc)

CódigoEmp referencia Emp

Código Proc referencia ProcessTexto

Projeto (CódigoProj, Nome)

Participação (CódigoEmp, CódigoProj)

CódigoEmp referencia Engenheiro

CódigoProj referencia Projeto

## ***Vantagens da implementação com tabela única***

- **Dados referentes à entidade genérica + dados referentes às especializações**
  - em uma única linha
- **Minimiza junções**
- **Menor número de chaves**

# ***Vantagens da implementação com uma tabela por entidade especializada***

- **Colunas opcionais**
  - apenas aquelas referentes a atributos que podem ser vazios do ponto de vista da aplicação.

## ***Subdivisão da entidade genérica***

- **Uma tabela para cada entidade especializada que não possua outra especialização (entidade folha da árvore)**
- **Tabela contém**
  - **dados da entidade especializada +**
  - **dados da entidade genérica**

# ***Subdivisão da entidade genérica***

EmpOutros (CódigoEmp, Tipo, Nome, CIC, CódigoDept)

CódigoDept referencia Depto

Motorista(CódigoEmp, Nome, CIC, CódigoDept, CartHabil)

Engenheiro(CódigoEmp, Nome, CIC, CódigoDept, CREA, CódigoRamo)

CódigoRamo referencia Ramo

Depto (CódigoDept, Nome)

Ramo (CódigoRamo, Nome)

ProcessTexto (CódigoProc, Nome)

Domínio (CódigoEmp, CódigoProc)

Código Proc referencia ProcessTexto

Projeto (CódigoProj, Nome)

Participação (CódigoEmp, CódigoProj)

CódigoProj referencia Projeto

# *Subdivisão da entidade genérica*

- **Desvantagem:**
  - **Unicidade da chave primária**
    - não é garantida pelo SGBD
    - deve ser garantida pela aplicação
  - **Não há como especificar ao SGBD restrições de integridade referenciais, que façam referência ao conjunto de empregados como um todo**

# *Refinamento do modelo relacional*

- **Projeto (engenharia) em geral**
  - **compromisso entre o ideal e o realizável dentro das restrições de recursos impostas pelas prática**
- **Projeto de banco de dados**
  - **compromisso entre o**  
**ideal (regras de implementação)**  
**e o**  
**alcançável frente a limitações de performance**

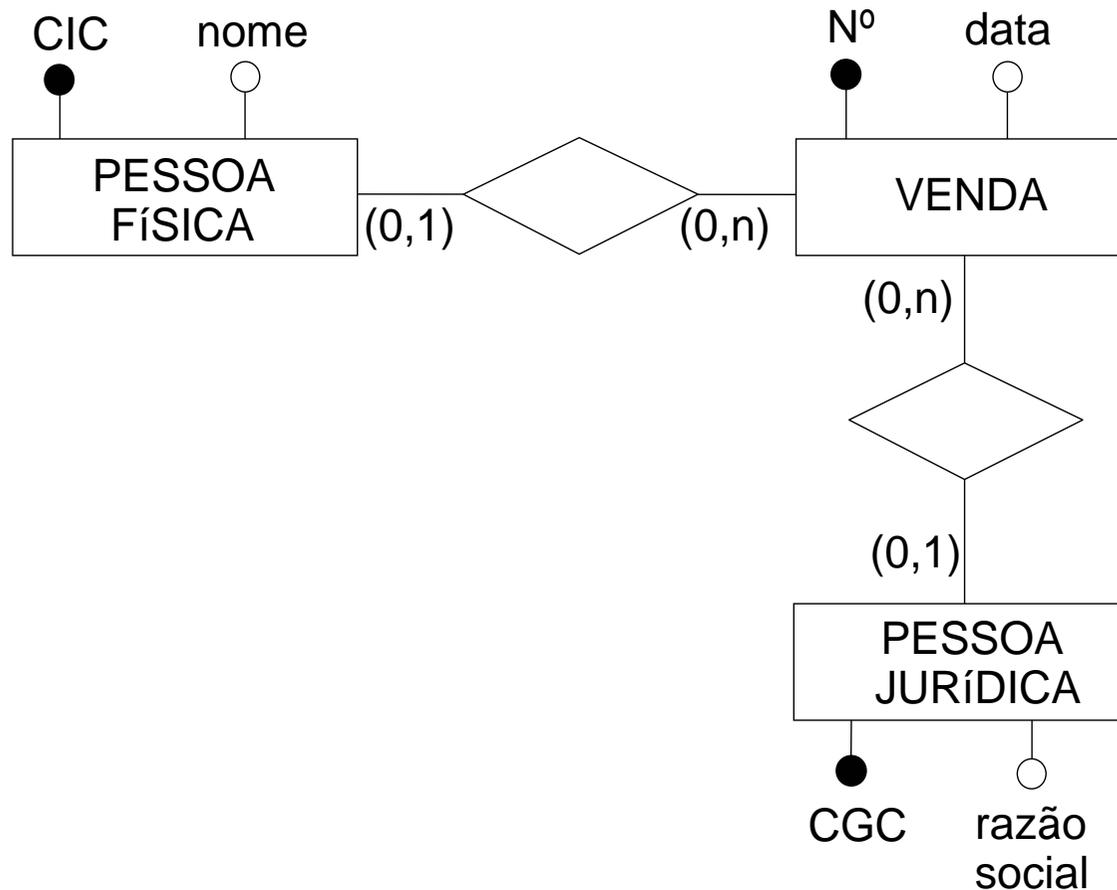
# *Refinamento do modelo relacional*

- **Algumas vezes**
  - **esquema de BD criado através do uso das regras acima não atende requisitos de performance impostos ao sistema**
- **Necessário buscar alternativa que resulte em melhor performance do sistema**
- **Alternativas somente devem ser tentadas em último caso**
  - **Do ponto de vista da programação são sempre piores**

# *Refinamento do modelo relacional*

- **Relacionamentos mutuamente exclusivos**
- **Simulação de atributos multivalorados**
- **Informações redundantes**

# Relacionamentos mutuamente exclusivos



# *Relacionamentos mutuamente exclusivos*

- **Implementação pelas regras**

PessFis(CIC,Nome)

PessJur(CGC,RazSoc)

Venda(No,data,CIC,CGC)

CIC referencia PessFis

CGC referencia PessJur

- **colunas CIC e CGC em Venda são especificadas como opcionais**

# *Relacionamentos mutuamente exclusivos*

- **Implementação alternativa**
  - **criar uma única coluna na qual aparece o CIC ou o CGC do comprador**

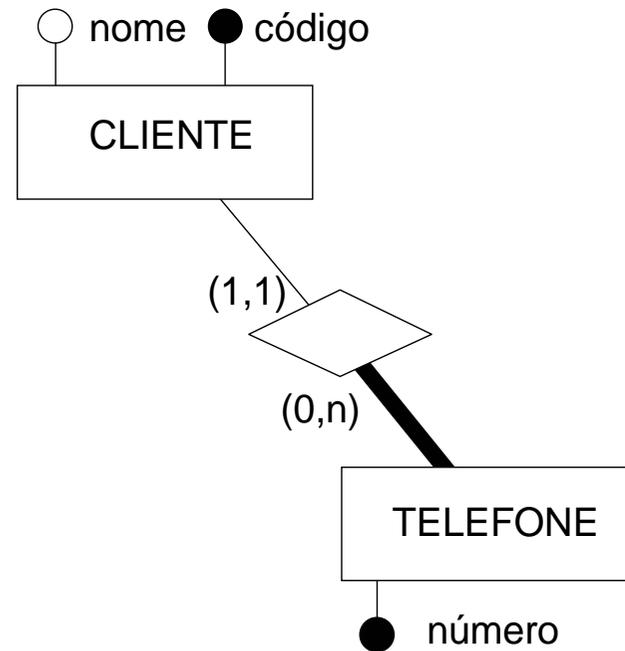
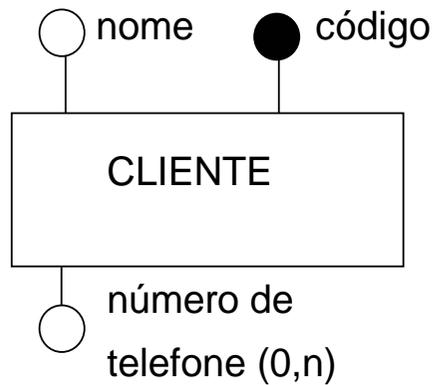
PessFis(CIC,Nome)

PessJur(CGC,RazSoc)

Venda(No,data,CIC/CGC,TipoCompr)

- **Desvantagem**
  - **Não é possível especificar ao SGBD que o campo CIC/CGC é chave estrangeira**
  - **não referencia uma única tabela**

# Tratamento de atributos multivalorados



# *Atributos multivalorados implementação padrão*

Cliente (CodCli,Nome)

Telefone (CodCli,Número)

CodCli referencia Cliente

# *Atributos multivalorados alternativa*

- **Condições de contorno:**
  - **Raros clientes possuem mais que dois telefones.**
  - **Quando isso ocorrer**
    - **é suficiente armazenarmos apenas dois números.**
  - **Não há consultas ao banco de dados usando o número de telefone como critério de seleção**
  - **Números de telefone são apenas exibidos ou impressos juntos às demais informações de cliente**

# *Simulação de atributos multivalorados*

- **Implementação “desnormalizada”**

**Cliente (CodCli,Nome,NumTel1,NumTel2)**

- **Simular uma coluna multi-valorada através da criação de diversas colunas NumTel sufixadas por um número**

# *Simulação de atributos multivalorados*

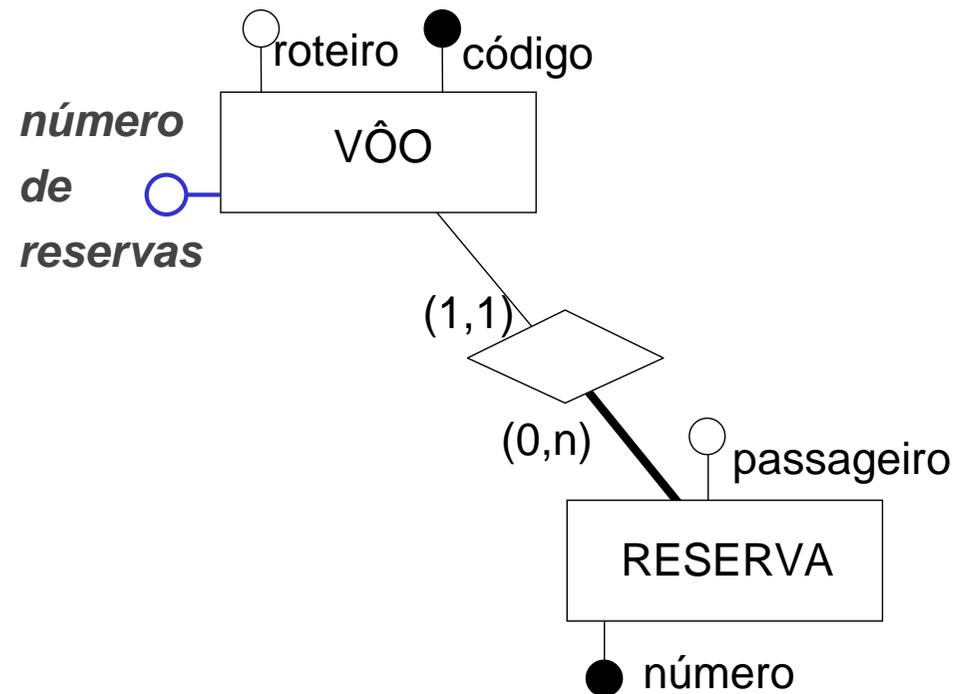
- **Permite que os telefones de um cliente sejam obtidos mais rapidamente**
- **Implica em menos espaço ocupado**
  - não é necessária chave primária da tabela Telefone
- **Inconveniente**
  - Consulta usando o número de telefone como critério de busca torna-se mais complicada
  - Manter os telefones "alinhados à esquerda" exige rotina complexa

# *Informações redundantes*

- **Exemplo:**
  - atributos que resultam de uma operação que envolve diversas entidades do banco de dados
  - valor destes atributos
    - deve ser obtido com frequência ou
    - serve frequentemente como critério de busca de informações no banco de dados
- **Pode ser mais eficiente (performance global do sistema)**
  - armazenar redundantemente o atributo derivado

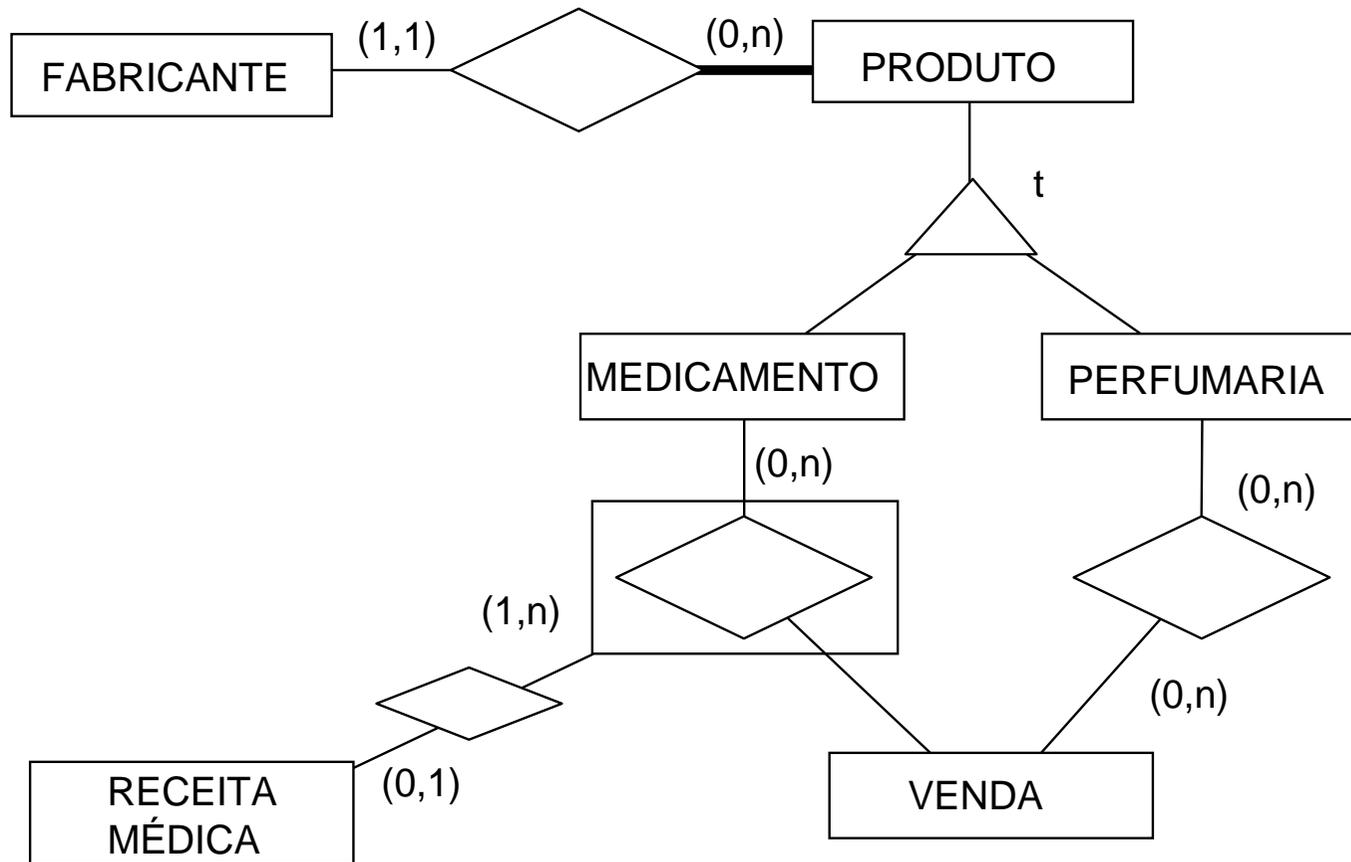
# Informações redundantes

## Exemplo



# Exercício 5.2

## DER



## ***Exercício 5.2*** ***atributos***

Produto (Número, NomeComercial, TipoEmbalagem,  
Quantidade, PreçoUnitário)

Fabricante (CGC, Nome, Endereço)

Medicamento(Tarja, Fórmula)

Perfumaria(Tipo)

Venda(Data, NúmeroNota, NomeCliente, CidadeCliente)

PerfumariaVenda(Quantidade, Imposto)

MedicamentoReceitaVenda(Quantidade, Imposto)

ReceitaMédica(CRM, Número, Data)

## ***Exercício 5.2*** ***solução***

Produto (CGC, NúmeroProd, NomeComercial, TipoEmbalagem,  
Quantidade, PreçoUnitário, TipoProd, Tarja,Fórmula, Tipo)  
CGC referencia Fabricante

Fabricante (CGC,Nome,Endereço)

Venda(Data,NúmeroNota,NomeCliente,CidadeCliente)

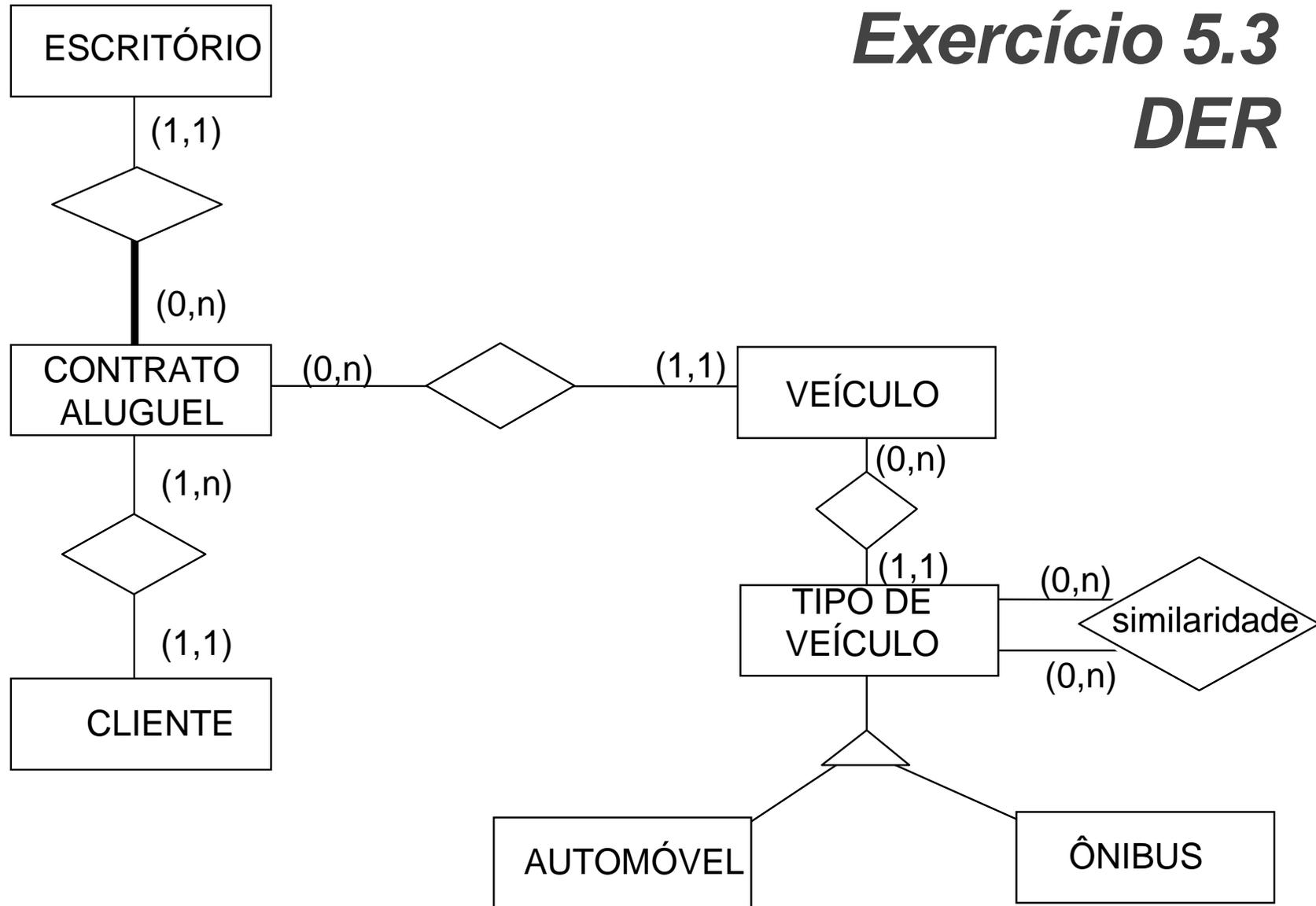
PerfumariaVenda(CGC, NúmeroProd, NúmeroNota,Quantidade,Imposto)  
(CGC,NúmeroProd) referencia Produto  
NúmeroNota referencia Venda

MedicamentoVenda(CGC, NúmeroProd, NúmeroNota,  
Quantidade,Imposto, CRM, Número)  
(CGC,NúmeroProd) referencia Produto  
NúmeroNota referencia Venda  
(CRM, Número) referencia ReceitaMédica

ReceitaMédica(CRM,Número,Data)

# Exercício 5.3

## DER



## ***Exercício 5.3*** ***atributos***

Escritório (Número, Local)

Cliente (NúmeroCartMotorista, EstadoCartMotorista,  
Nome, Endereço, Telefone)

Contrato aluguel (Número, Data, Duração)

Veículo (Número, DataPróximaManutenção, Placa)

Tipo de Veículo (Código, Nome, ArCondicionado)

Automóvel (NúmeroPortas, DireçãoHidráulica, CâmbioAutomático, Rádio)

Ônibus (NúmeroPassageiros, Leito, Sanitário)

## **Exercício 5.3**

### ***solução***

Escritório (NúmeroEsqr, Local)

Contrato aluguel (NúmeroEsqr, NúmeroContr, Data, Duração,  
NúmeroVeic, NúmeroCarMotorista, EstadoCarMotorista)

NúmeroEsqr referencia Escritório

NúmeroVeic referencia Veículo

Cliente (NúmeroCartMotorista, EstadoCartMotorista, Nome, Endereço, Telefone)

Veículo (Número, DataPróximaManutenção, Placa, CódigoTipo)

CódigoTipo referencia TipoVeiculo

TipoVeículo (CódigoTipo, Nome, ArCondicionado)

Similaridade (CódigoTipo, CódigoTipoSimilar)

CódigoTipo referencia TipoVeículo

CódigoTipoSimilar referencia TipoVeículo

Automóvel (CódigoTipo, NúmeroPortas, DireçãoHidráulica,  
CâmbioAutomático, Rádio)

CódigoTipo referencia TipoVeículo

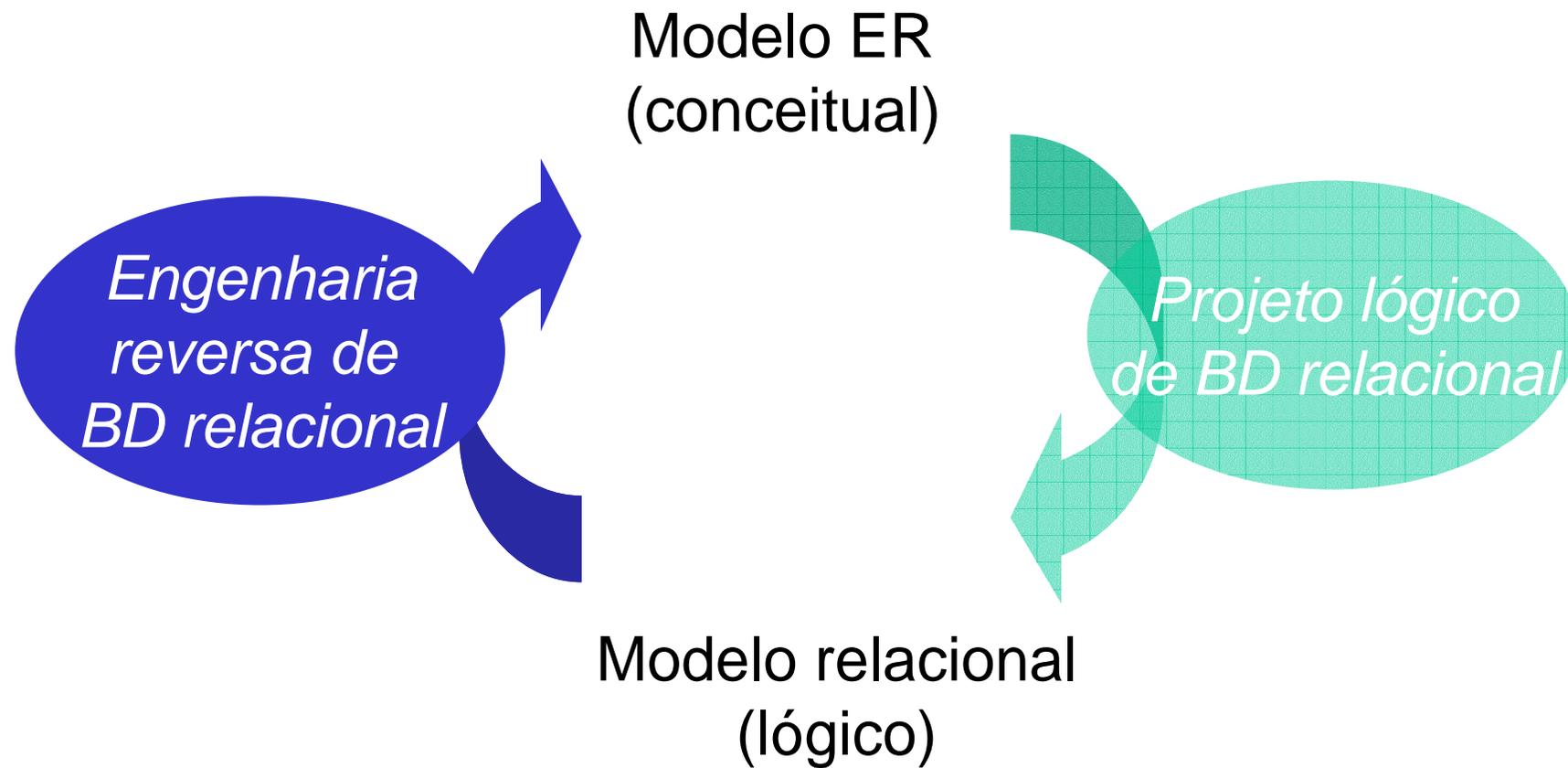
Ônibus (CódigoTipo, NúmeroPassageiros, Leito, Sanitário)

CódigoTipo referencia TipoVeículo

# *Engenharia reversa de modelos relacionais*

- **Engenharia reversa**
  - parte de modelo de implementação
  - obtém modelo de especificação (modelo conceitual)

# *Engenharia reversa*



# *Engenharia reversa de modelo relacional*

- **Passos:**
  - **Identificação da construção ER correspondente a cada tabela**
  - **Definição de relacionamentos 1:n e 1:1**
  - **Definição de atributos**
  - **Definição de identificadores de entidades e relacionamentos**

# *Esquema relacional para engenharia reversa*

Disciplina (CodDisc, NomeDisc)

Curso (CodCr, NomeCr)

Curric (CodCr, CodDisc, Obr/Opc)

CodCr referencia Curso

CodDisc referencia Disciplina

Sala (CodPr, CodSl, Capacidade)

CodPr referencia Prédio

Prédio (CodPr, Endereço)

Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)

CodDisc referencia Disciplina

(CodPr, CodSl) referencia Sala

Laboratório ( CodPr, CodSl, Equipam)

(CodPr, CodSl) referencia Sala

# *Identificação da construção ER correspondente a cada tabela*

- **Uma tabela pode corresponder a:**
  - uma entidade
  - um relacionamento n:n
  - uma entidade especializada
- **Fator determinante**
  - composição de sua chave primária

## *Tipos de tabelas para identificação de construção ER*

Composição da chave primária	Construção ER correspondente
Múltiplas chaves estrangeiras	Relacionamento n:n
Toda chave primária é uma chave estrangeira	Especialização
Demais casos	Entidade

# *Construções identificadas*

Disciplina (CodDisc, NomeDisc) **entidade**

Curso (CodCr, NomeCr) **entidade**

Curric (CodCr, CodDisc, Obr/Opc) **relacionamento n:n**

CodCr referencia Curso

CodDisc referencia Disciplina

Sala (CodPr, CodSl, Capacidade) **entidade**

CodPr referencia Prédio

Prédio (CodPr, Endereço) **entidade**

Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl) **entidade**

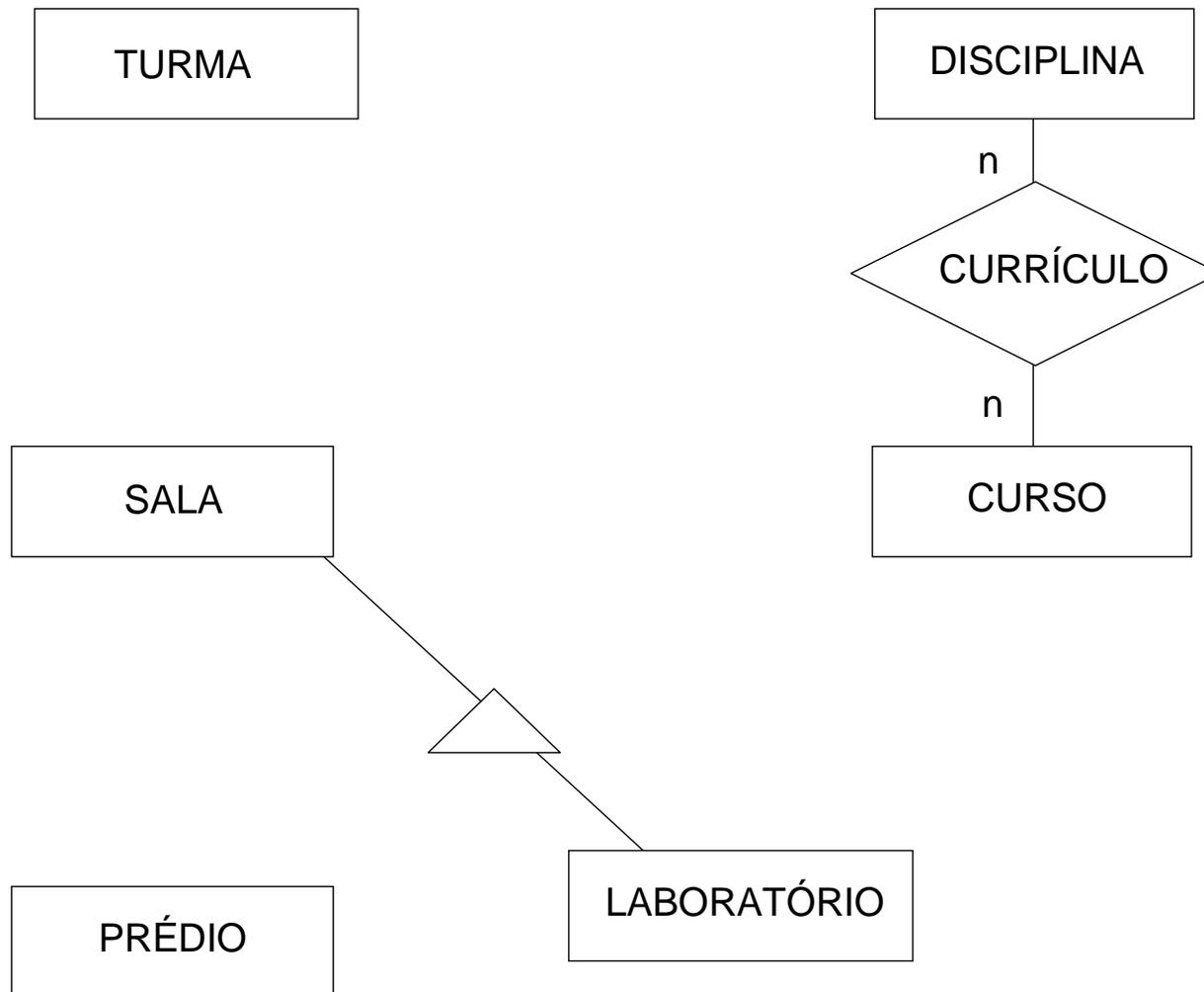
CodDisc referencia Disciplina

(CodPr, CodSl) referencia Sala

Laboratório ( CodPr, CodSl, Equipam) **especialização**

(CodPr, CodSl) referencia Sala

# Construções identificadas



## *Identificação de relacionamentos 1:n ou 1:1*

- **Chave estrangeira que não se enquadra nas regras acima**
  - **representa**
    - relacionamento 1:n
    - ou**
    - relacionamento 1:1
- **Esquema não informa se é 1:1 ou 1:n**

# Identificação de relacionamentos 1:n ou 1:1

Disciplina (CodDisc, NomeDisc)

Curso (CodCr, NomeCr)

Curric (CodCr, CodDisc, Obr/Opc)

CodCr referencia Curso

CodDisc referencia Disciplina

Sala (CodPr, CodSl, Capacidade)

CodPr referencia Prédio

Prédio (CodPr, Endereço)

Turma (Anosem, CodDisc, SiglaTur, Capacidade, CodPr, CodSl)

CodDisc referencia Disciplina

(CodPr, CodSl) referencia Sala

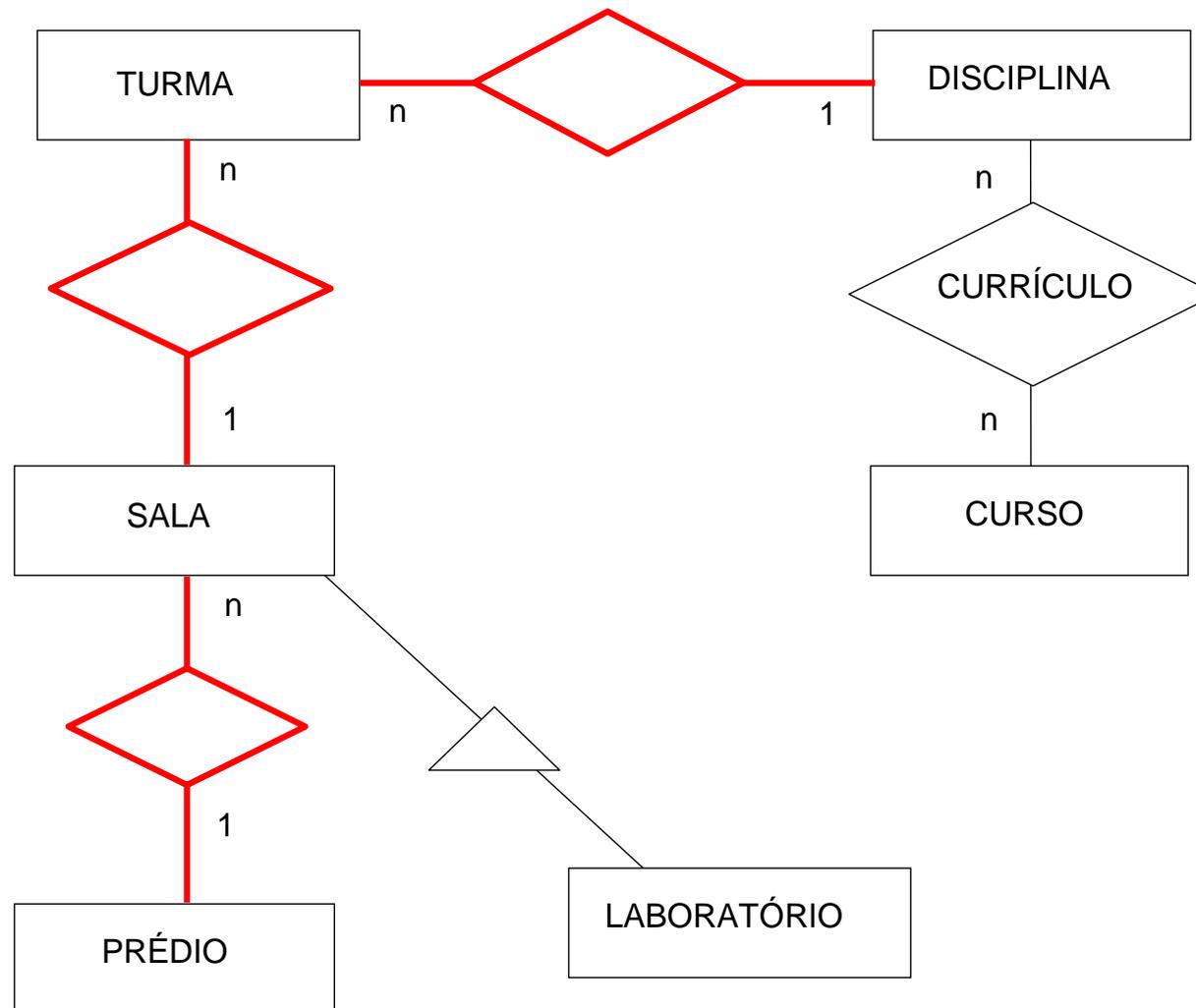
Laboratório ( CodPr, CodSl, Equipam)

(CodPr, CodSl) referencia Sala

Chaves estrangeiras por tratar indicadas em vermelho



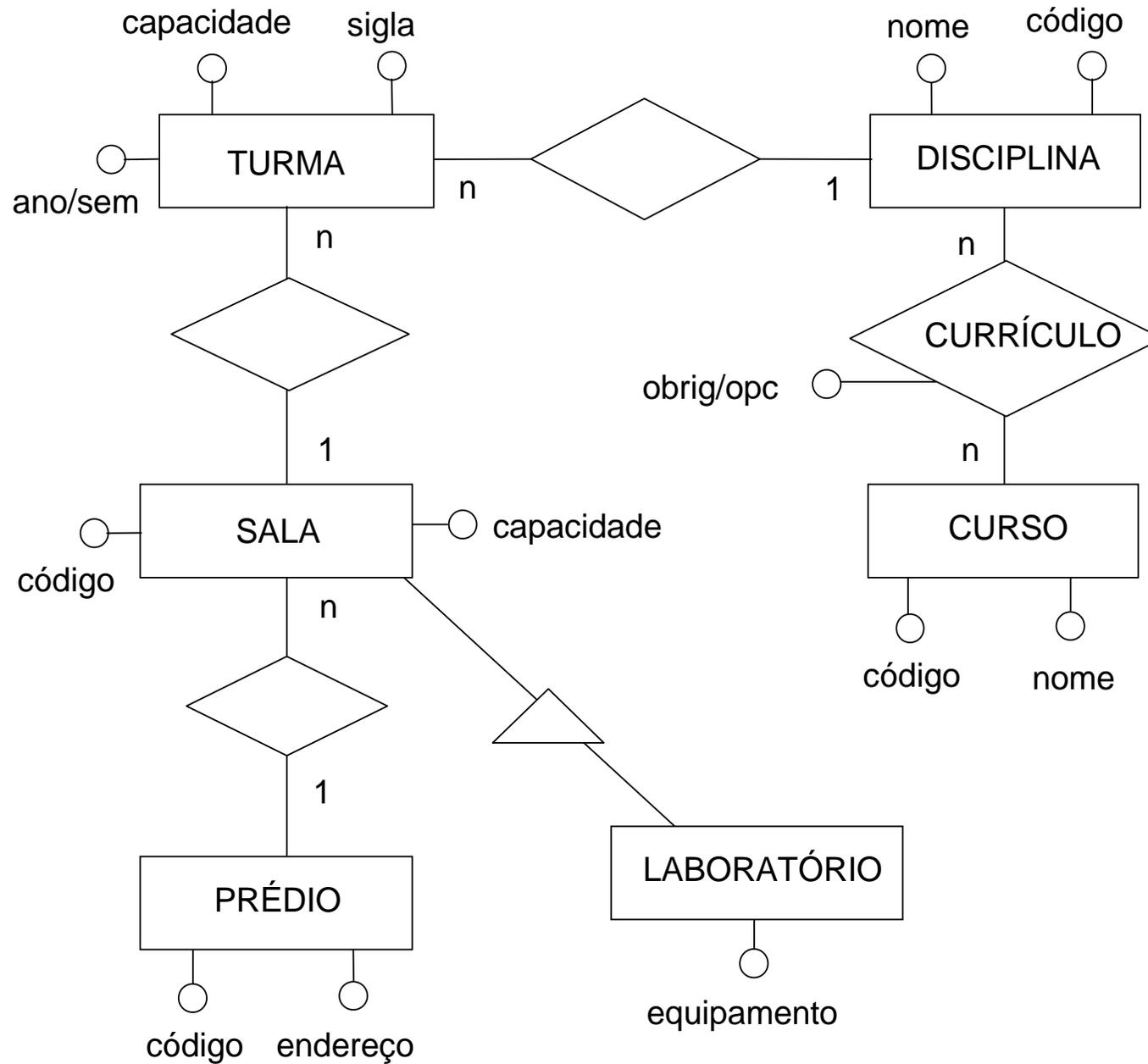
# Identificação de relacionamentos 1:n ou 1:1



# *Definição de atributos*

- **Cada coluna não chave estrangeira é**
  - **um atributo na entidade/relacionamento correspondente à tabela**
- **As colunas chave estrangeira não correspondem a atributos**
  - **correspondem a relacionamentos**
  - **já foram tratadas nas etapas anteriores**

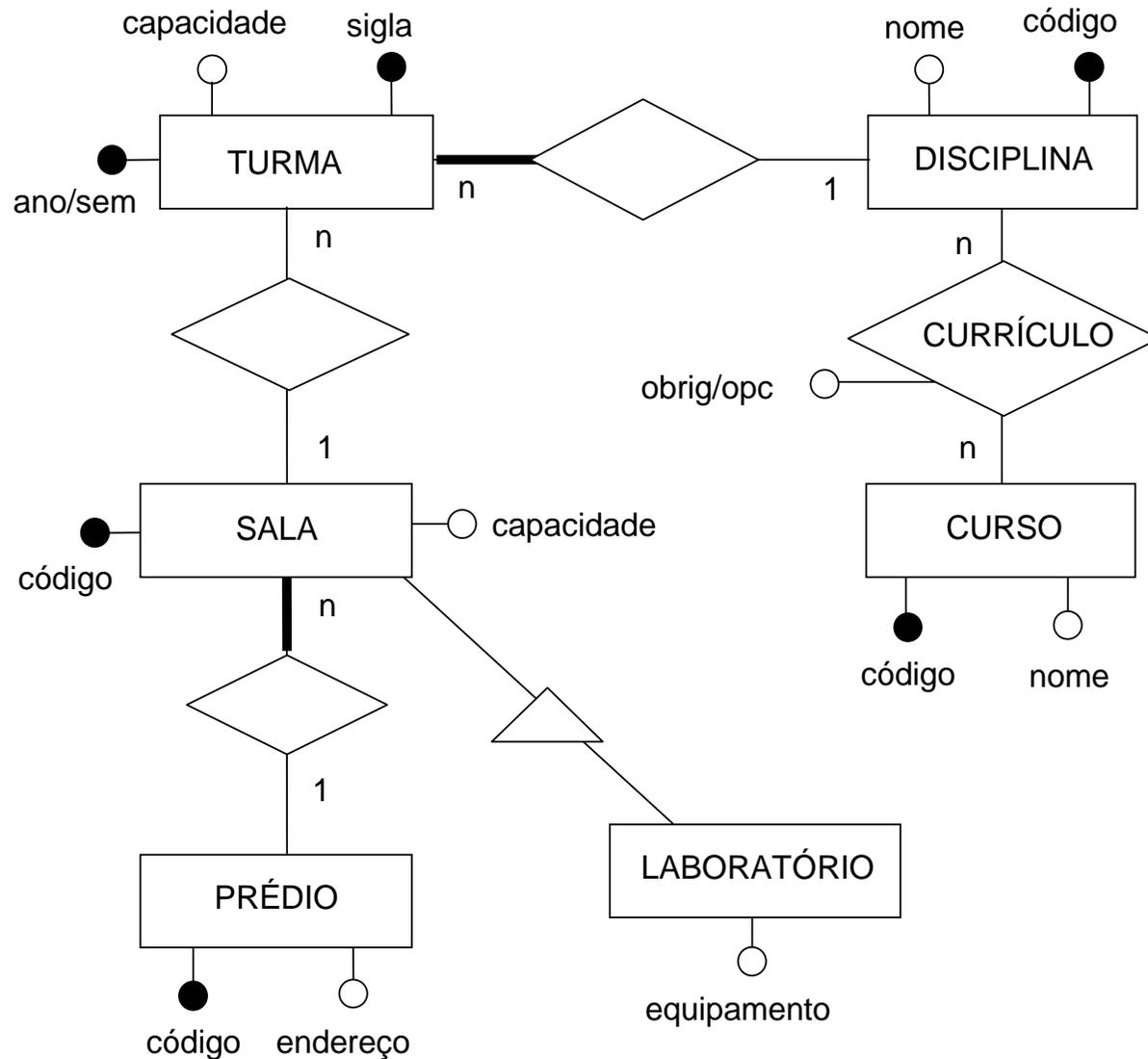
# Definição de atributos



## *Definição de identificadores de entidades*

- **Coluna da chave primária que não é chave estrangeira**
  - **corresponde a um atributo identificador da entidade ou relacionamento.**
- **Coluna da chave primária que é chave estrangeira**
  - **corresponde a um relacionamento identificador da entidade**

# Definição de identificadores de entidades



# Exercício

## 5.4

Produto (CodigoTipoProd,NumeroProd,DescricaoProd,PreçoProd)  
CodigoTipoProd referencia TipoProd

Similaridade (CodigoTipoProd,NumeroProd,  
CodigoTipoProdSim,NumeroProdSim)  
(CodigoTipoProd,NumeroProd) referencia Produto  
(CodigoTipoProdsim,NumeroProdSim) referencia Produto

TipoProd (CodigoTipoProd,DescricaoTipoProd)

Venda (NúmeroNF,DataVenda,CodReg,CodEmp)  
(CodigoReg) referencia Registradora  
(CodEmp) referencia Empregado

ItemVenda (NúmeroNF,CodigoTipoProd,NumeroProd, Qtdeltem,Preçoltem)  
(NúmeroNF) referencia Venda  
(CodigoTipoProd,NumeroProd) referencia Produto

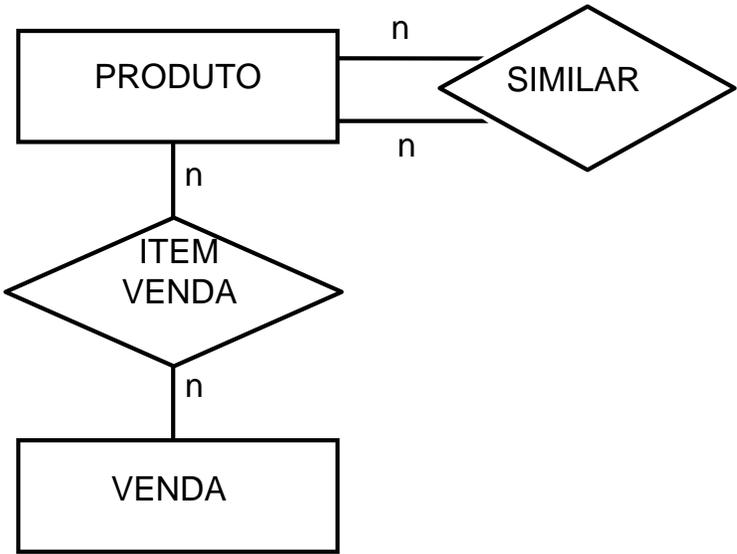
Registradora (CodReg, SaldoReg)

Empregado (CodEmp, NomeEmp, SenhaEmp)

TIPO DE  
PRODUTO

# Exercício 5.4

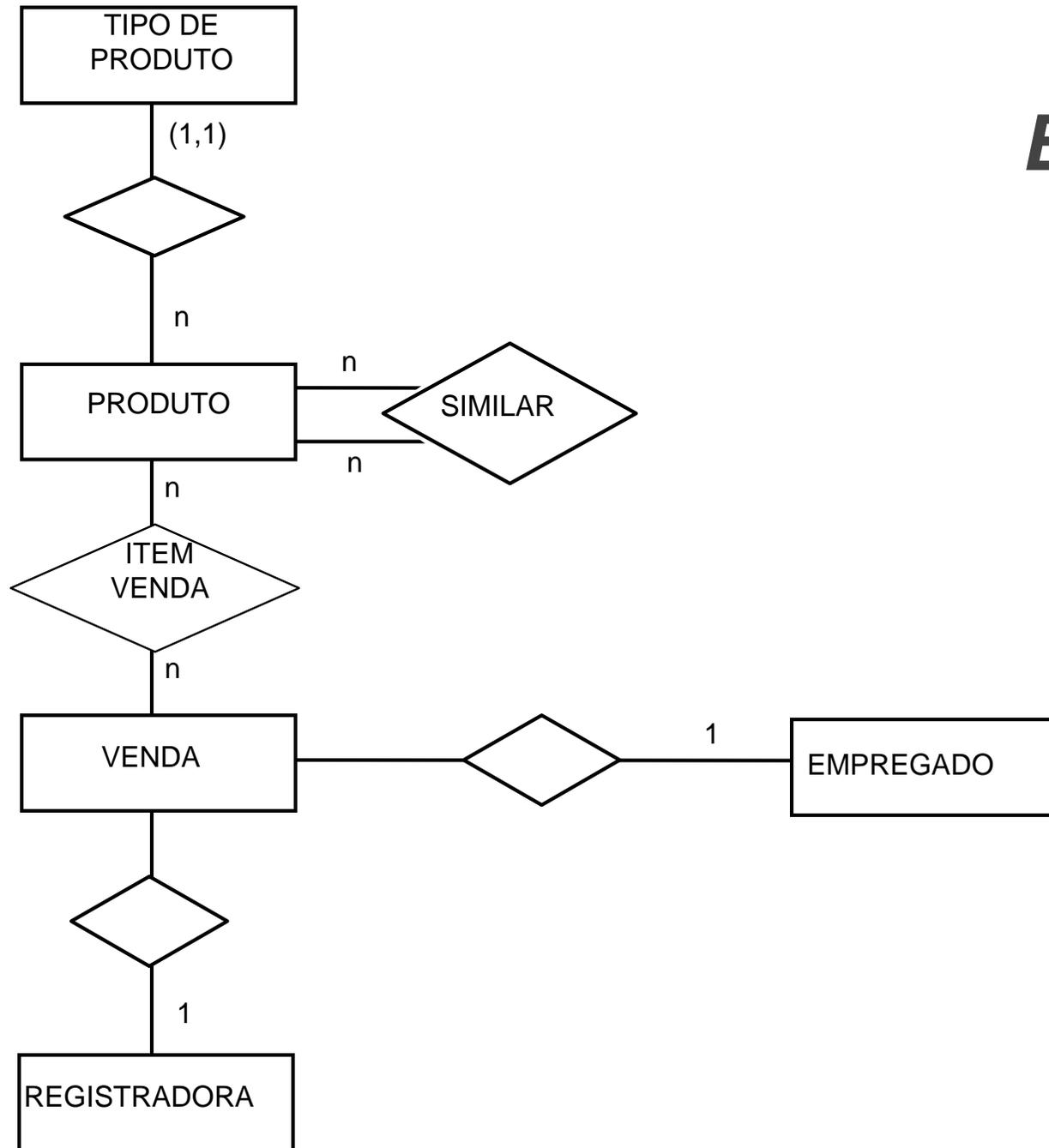
## passo 1



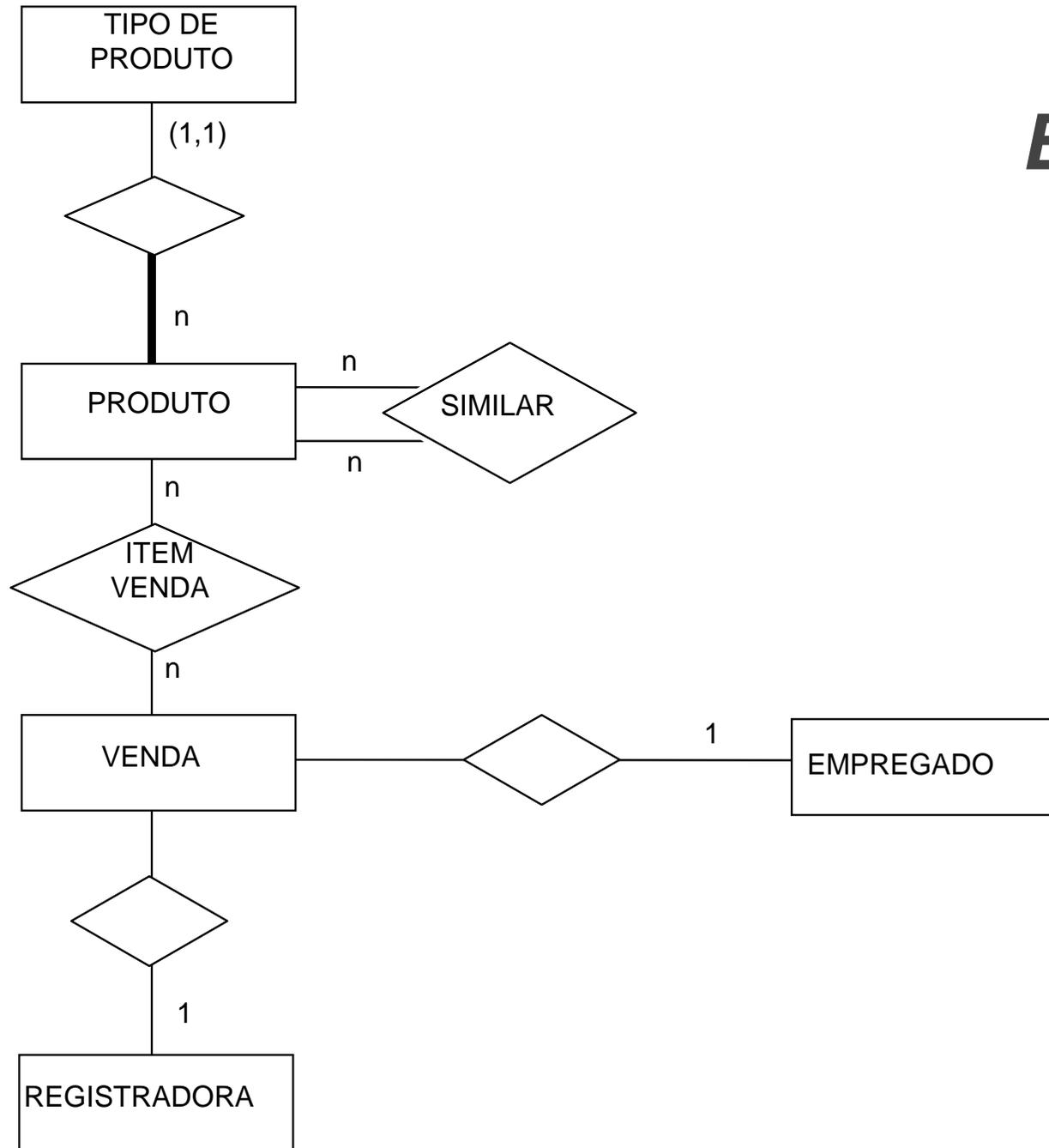
EMPREGADO

REGISTRADORA

## Exercício 5.4 passo 2



## Exercício 5.4 DER final



## ***Exercício 5.4*** ***atributos***

Produto (NumeroProd,DescricaoProd,PreçoProd)

TipoProd (CodigoTipoProd,DescricaoTipoProd)

Venda (NúmeroNF,DataVenda)

ItemVenda (QtdeItem,PreçoItem)

Registradora (CodReg, SaldoReg)

Empregado (CodEmp, NomeEmp, SenhaEmp)

## ***Exercício 5.5***

Pessoa (PessID, PessNome, NascLocID, DataNasc, FalecLocID,  
DataFalec, ProfID, FilhoCasamID, Sexo)

NascLocID referencia Local

FalecLocID referencia Local

ProfID referencia Profiss

FilhoCasamID referencia Casam

Local (LocID, Cidade, País)

Profiss (ProfID, ProfNome)

Casam (CasamID, MaridoPessID, EsposaPessID,  
DataCasam, CasamLocID)

MaridoPessID referencia Pessoa

EsposaPessID referencia Pessoa

CasamLocID referencia Local

# ***Exercício 5.5***

## ***passo 1***

LOCAL

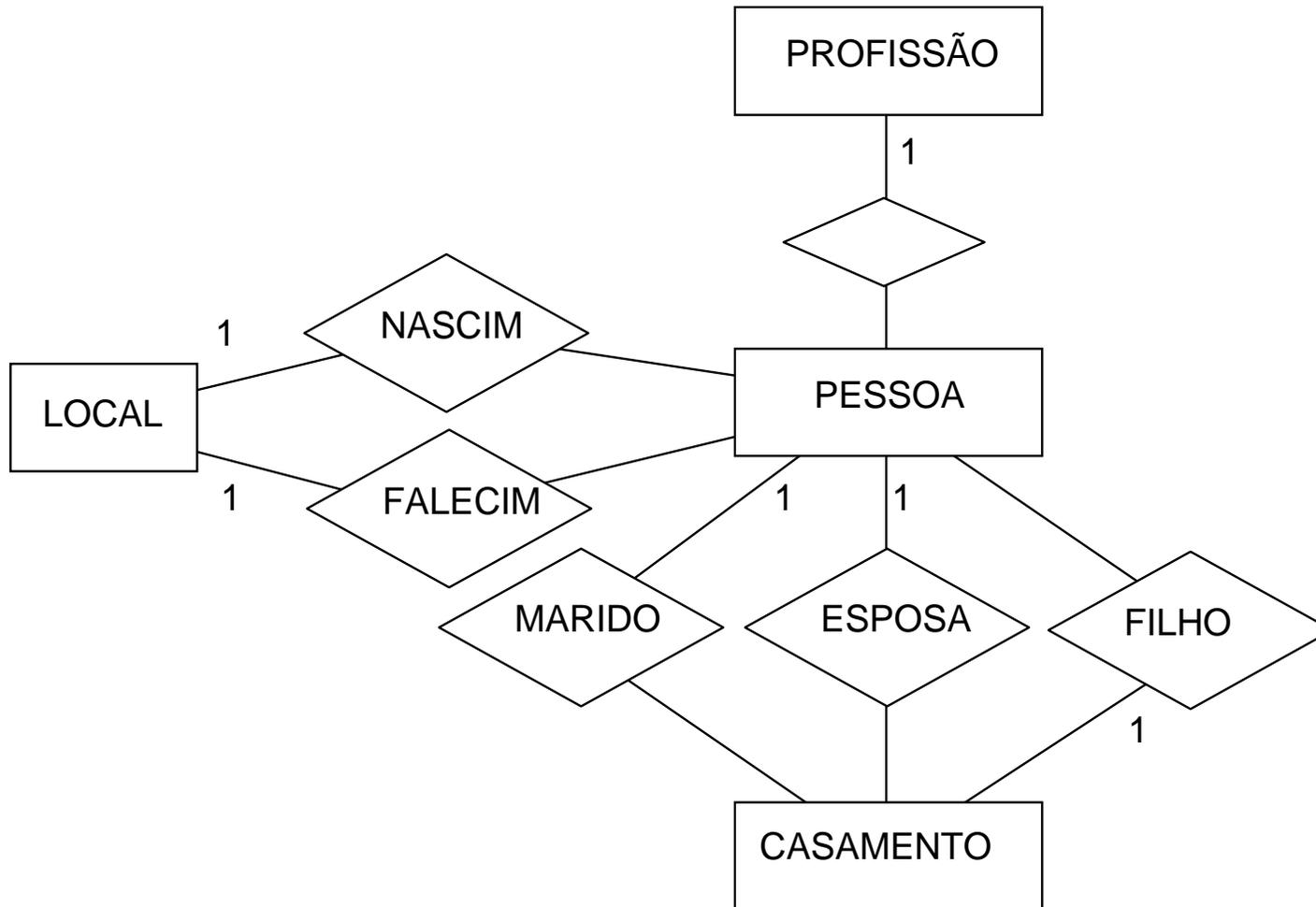
PROFISSÃO

PESSOA

CASAMENTO

# Exercício 5.5

## passo 2



## ***Exercício 5.5*** ***atributos***

Pessoa (PessID, PessNome, DataNasc, DataFalec, Sexo)

Local (LocID, Cidade, País)

Profissão (ProfID, ProfNome)

Casamento (CasamID, DataCasam)