

DCC011: Introdução a Banco de Dados

Rodrygo Santos

rodrygo@dcc.ufmg.br

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais

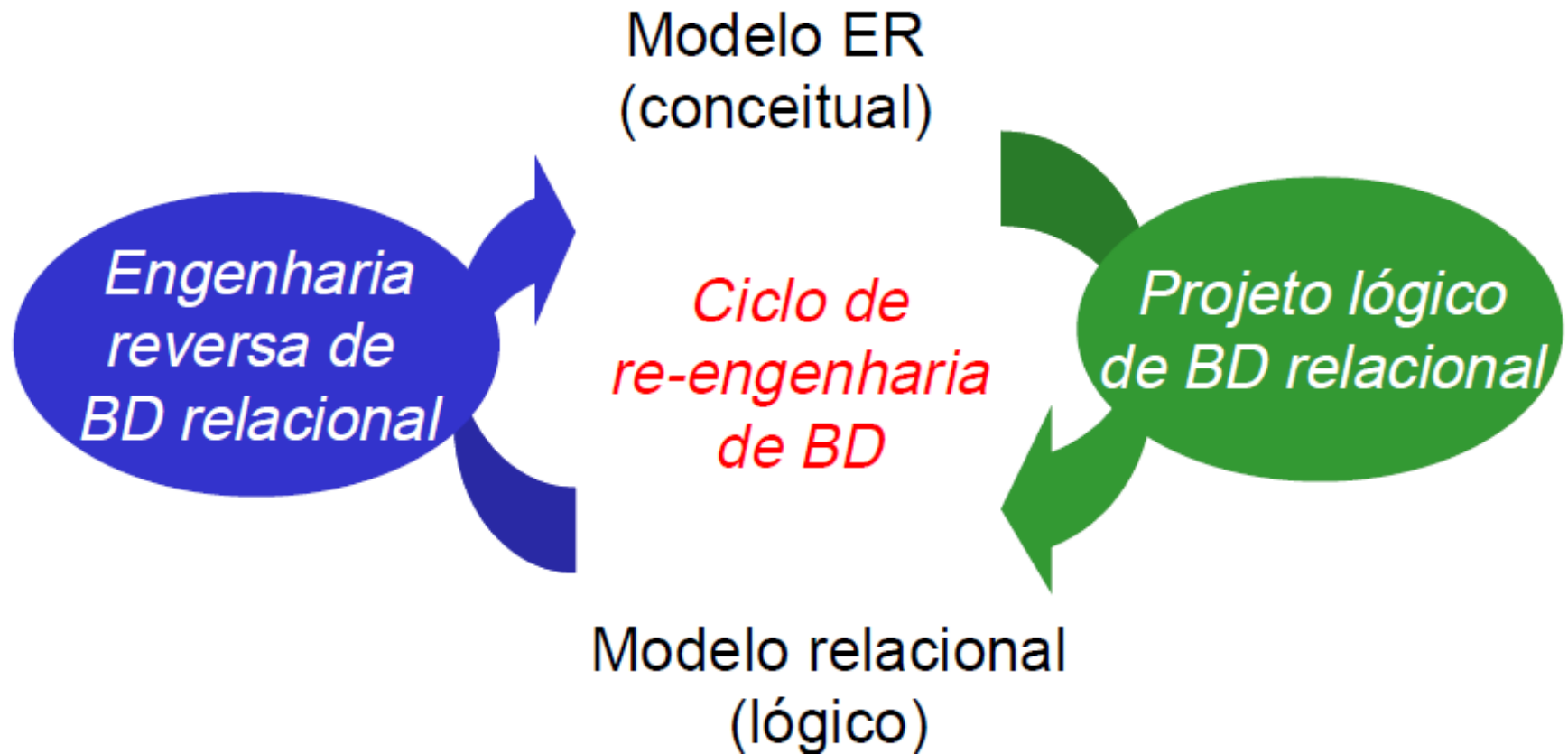
Mapeamento ER - Relacional

1. Transformações entre Modelos
2. Algoritmo de mapeamento ER → Relacional

1. Transformações entre Modelos

- Uma vez definido o modelo conceitual, o próximo passo é definir o modelo lógico
- Uma alternativa: mapear as construções do modelo conceitual para o lógico

Transformações entre Modelos



Transformações ER para Relacional

- Regras gerais
 - Aplicáveis à maioria dos casos
 - Há situações excepcionais
 - Por exigências da aplicação, outros mapeamentos são usados
 - Implementadas em ferramentas CASE
- Objetivos básicos
 - Bom desempenho
 - Simplificar o desenvolvimento

Regras gerais de tradução

- A. Evitar junções
- B. Diminuir o número de chaves
- C. Evitar campos opcionais

A. Evitar junções

- Junções
 - Operação para buscar dados de diversas linhas associadas pela igualdade de campos
 - Dados de empregados e seus respectivos departamentos
- 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

B. 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

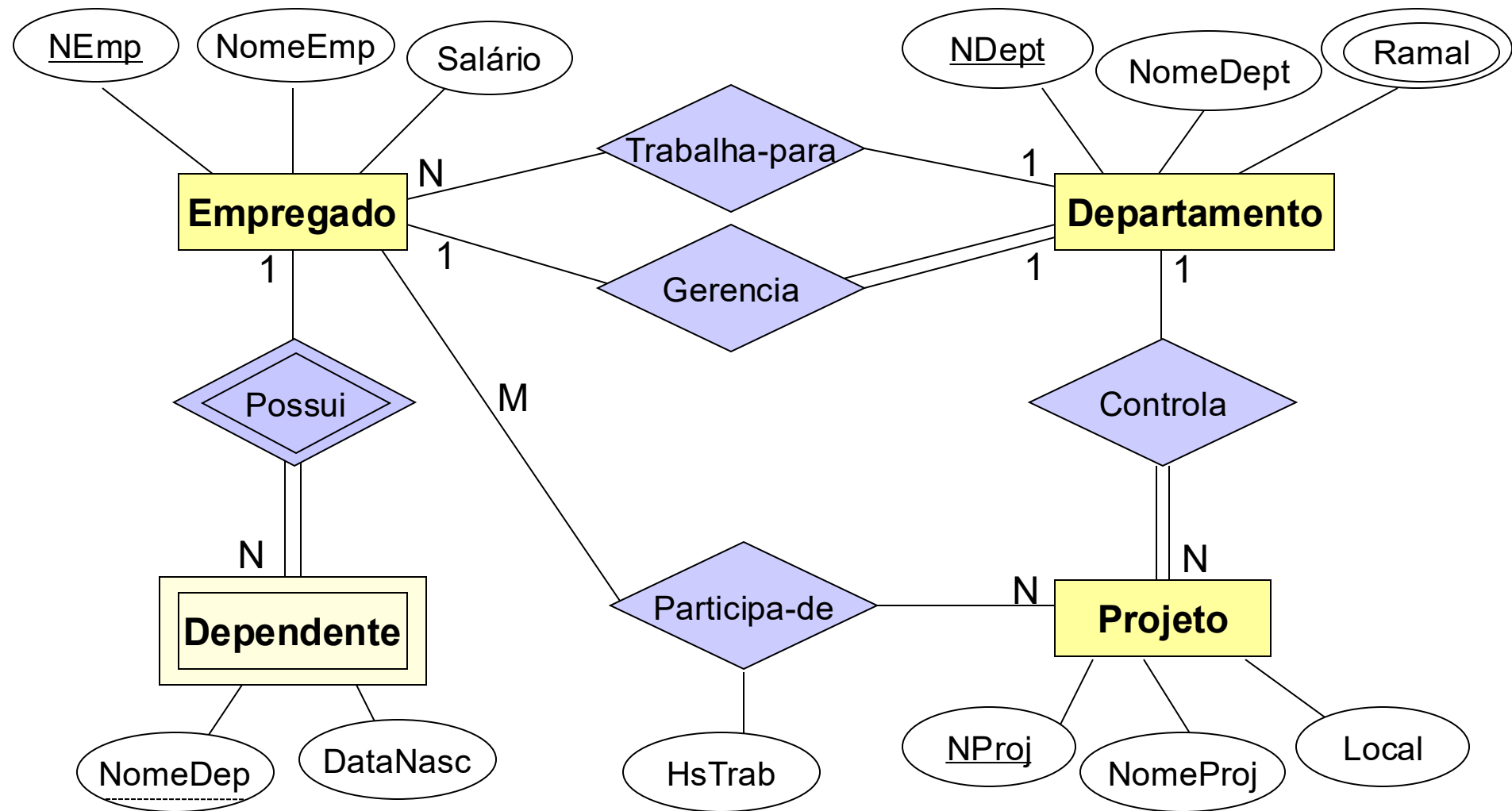
C. Campos opcionais

- 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 no desempenho
- EVITAR porque controle de campo opcional pode complicar programação
 - Verifica quais campos podem estar vazios

2. Algoritmo de Mapeamento Elmasri & Navathe

- a. Entidades regulares
- b. Atributos multivalorados
- c. Entidades fracas
- d. Relacionamentos
 - d.1 Relacionamentos binários 1:1
 - d.2 Relacionamentos binários 1:N
 - d.3 Relacionamentos binários N:M
 - d.4 Relacionamentos N-ários
- e. Hierarquias (especialização/generalização)

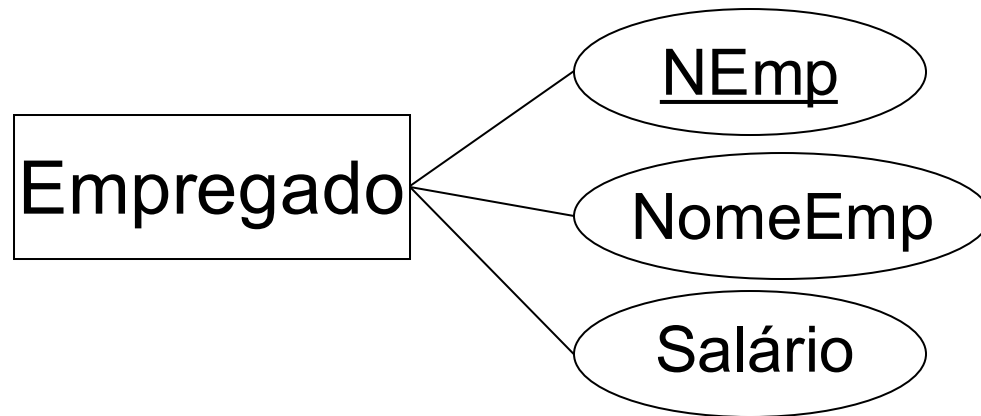
Exemplo de um Diagrama ER



a. Entidades regulares (sem atributos multivalorados)

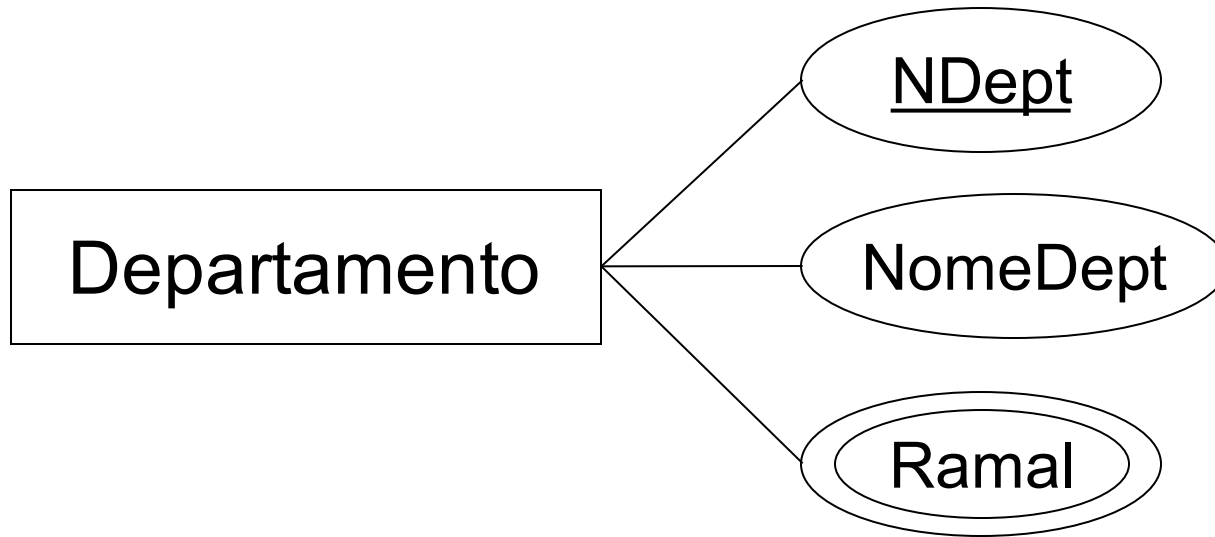
- Entidade regular $E \rightarrow$ Relação R
- Atributo em $E \rightarrow$ Coluna em R
- Atributo identificador em $E \rightarrow$ Chave primária em R

a. Entidades regulares (sem atributos multivalorados)



Empregado (NEmp, NomeEmp, Salário)

b. Atributos Multivalorados

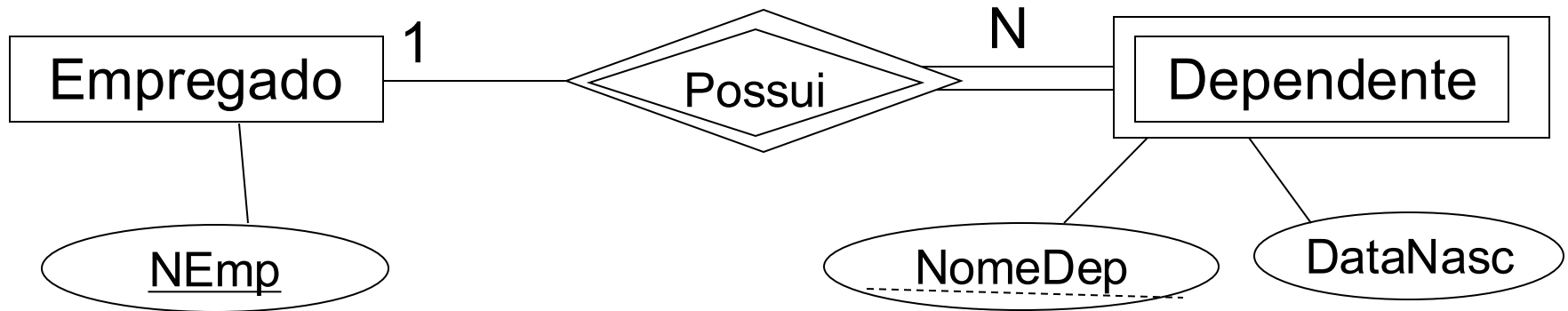


Departamento (NDept, NomeDept)

Ramal-Departamento (NDept, Ramal)

NDept referencia Departamento, por propagação

c. Entidade Fraca



Empregado (NEmp,...)

Dependente (NEmp, NomeDep, DataNasc)

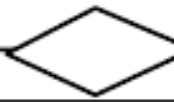
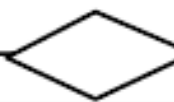
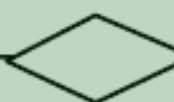
NEmp referencia Empregado, **por propagação**

d. Relacionamentos

- Tabela própria
- Adição de colunas a uma das tabelas
- Fusão de tabelas
- Alternativa depende da cardinalidade (máxima e mínima) do relacionamento
 - d.1 Relacionamentos binários 1:1
 - d.2 Relacionamentos binários 1:N
 - d.3 Relacionamentos binários N:M
 - d.4 Relacionamentos N-ários

d.1. Relacionamento binário

(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)		×	×	✓

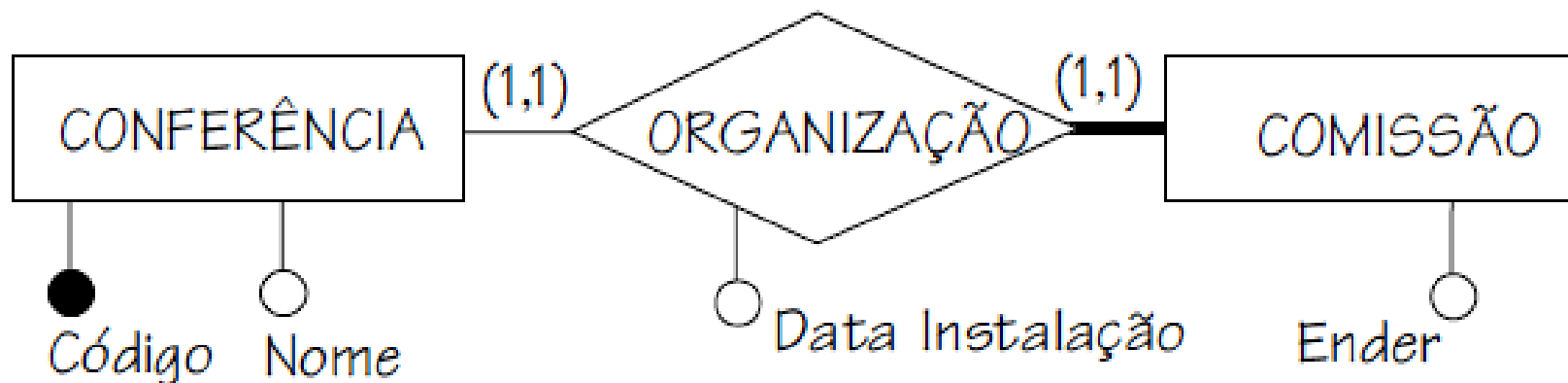
✓ Alternativa preferida

± Pode ser usada

× Não usar

Relacionamentos binários

(1,1) – (1,1) – ambas obrigatórias



Fusão de Tabelas

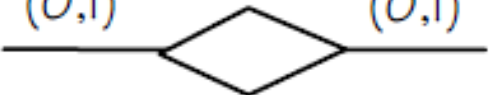
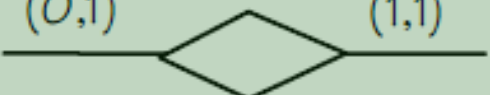
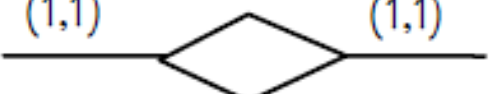
Conferência (CodConf, Nome, DataInstComOrg, EnderComOrg)

Relacionamentos binários

(1,1) – (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

d.1. Relacionamento binário (1:1)

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

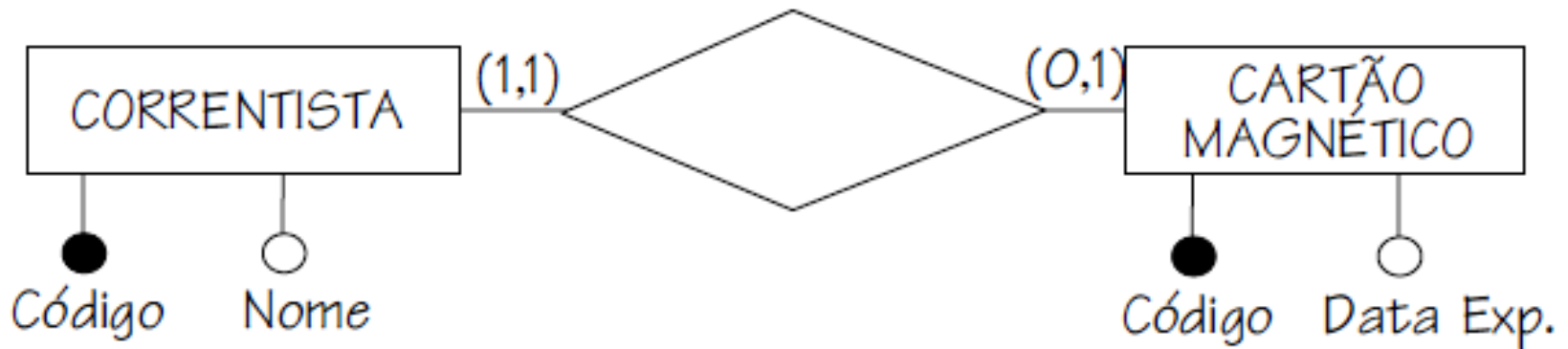
✓ Alternativa preferida

± Pode ser usada

× Não usar

Relacionamentos binários

(0,1) – (1,1) – opcional e obrigatória

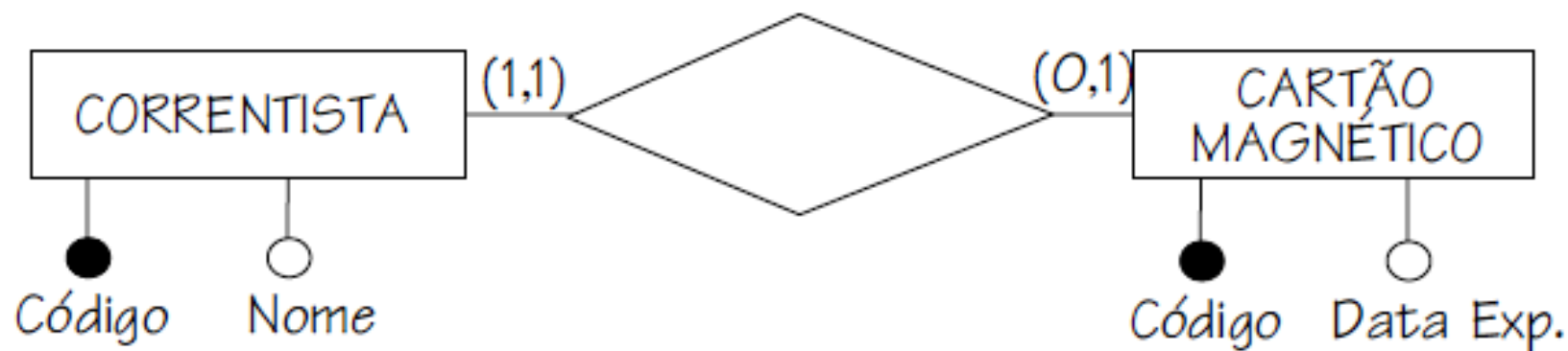


Fusão de Tabelas

Correntista (CodCorrent, Nome, CodCartao, DataExp)

Relacionamentos binários

(0,1) – (1,1) – opcional e obrigatória



Adição de Colunas

Correntista (CodCorrent, Nome)

Cartao (CodCartao, DataExp, CodCorrent)

CodCorrent referencia Correntista



Relacionamentos binários

(0,1) – (1,1) – opcional e obrigatória

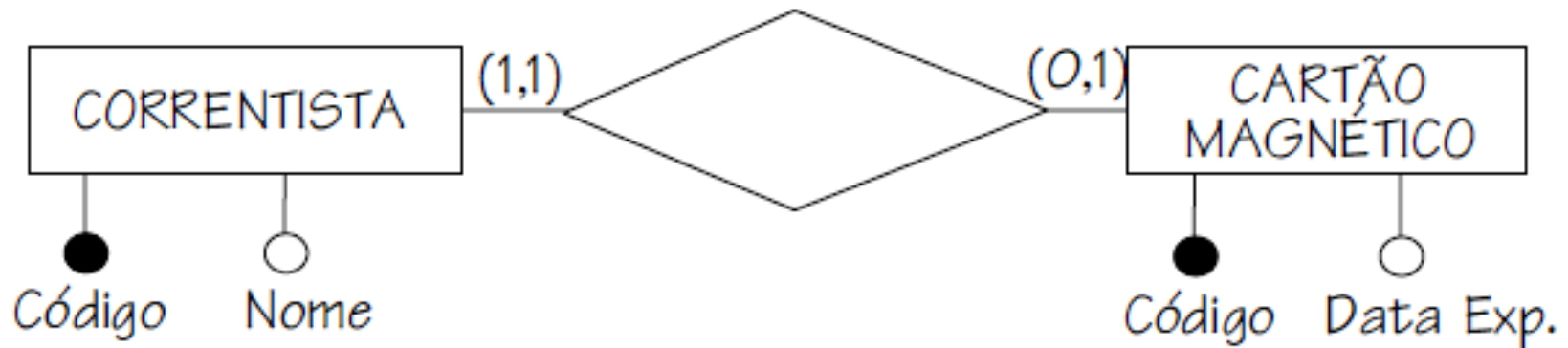


Tabela Própria

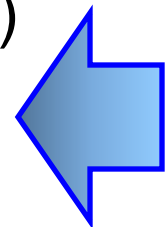
Correntista (CodCorrent, Nome)

Cartao (CodCartao, DataExp)

CartaoCorrentista (CodCartao, CodCorrent)

CodCorrent referencia Correntista

CodCartao referencia Cartao

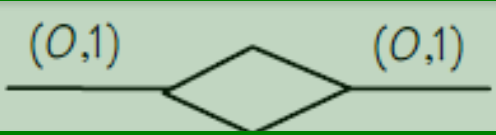

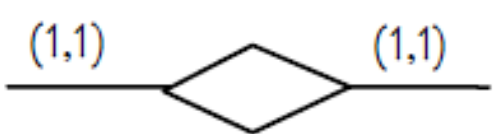


Relacionamentos binários

(0,1) – (1,1) – opcional e obrigatória

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

d.1. Relacionamento binário (1:1)

Tipo de relacionamento			Regra de implementação		
			Tabela própria	Adição coluna	Fusão tabelas
			±	✓	×
			×	±	✓
			×	×	✓

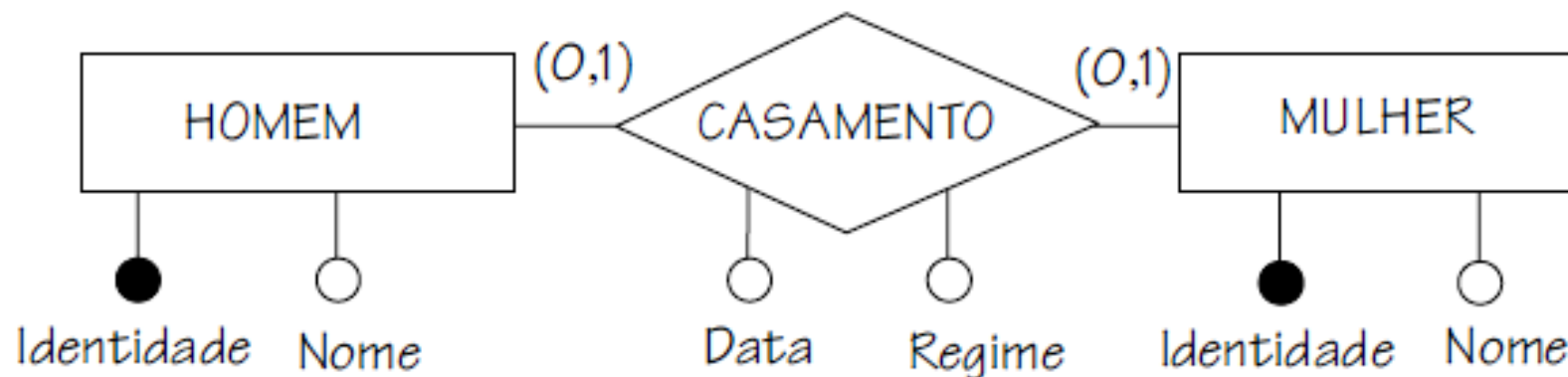
✓ Alternativa preferida

± Pode ser usada

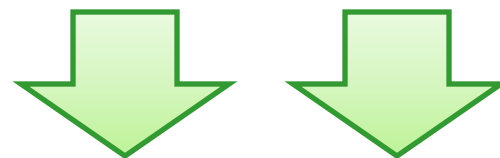
× Não usar

Relacionamentos binários

(0,1) – (0,1) – ambas opcionais



Adição de Colunas



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

IdentH referencia Homem

Homem (IdentH, Nome)

Relacionamentos binários

(0,1) – (0,1) – ambas opcionais

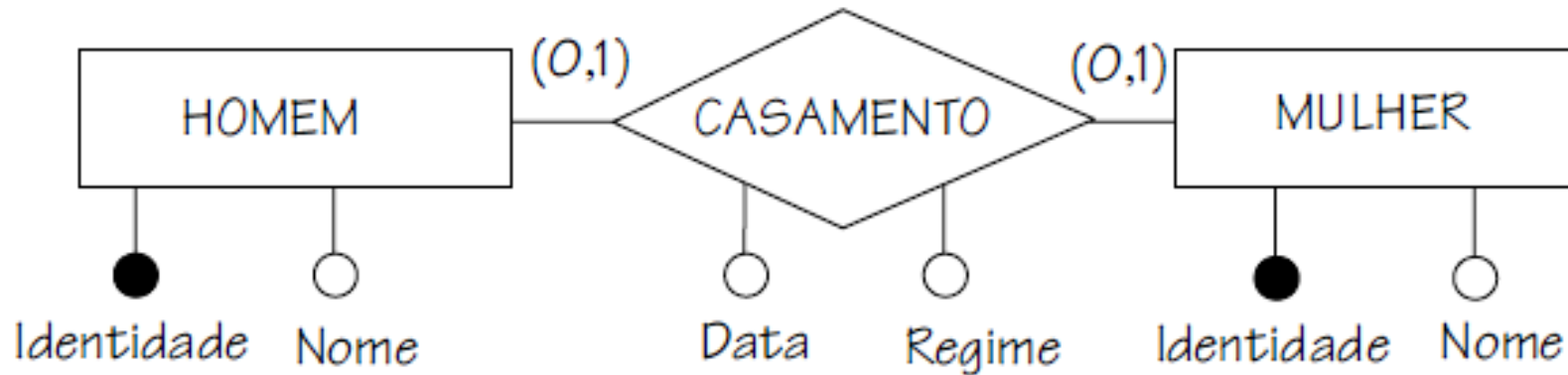


Tabela Própria

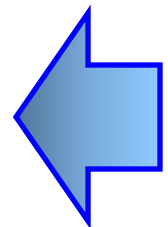
Mulher (IdentM, Nome)

Homem (IdentH, Nome)

Casamento (IdentM, IdentH, Data, Regime)

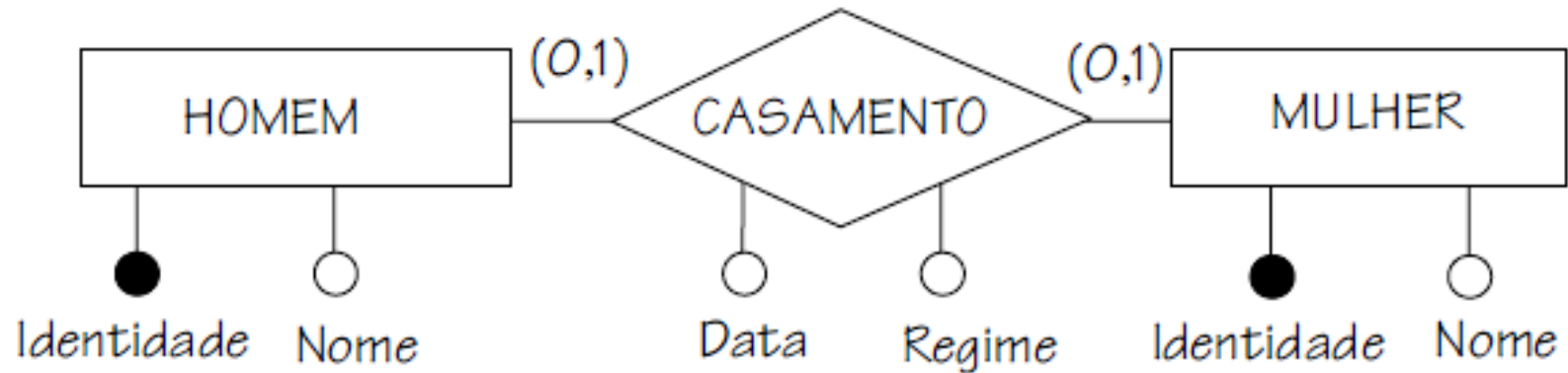
IdentM referencia Mulher

IdentH referencia Homem



Relacionamentos binários

(0,1) – (0,1) – ambas opcionais



Fusão de Tabelas

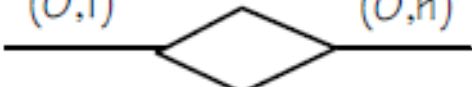
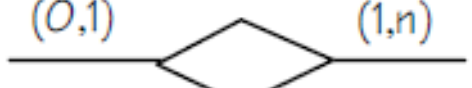
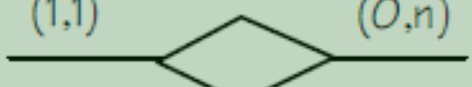
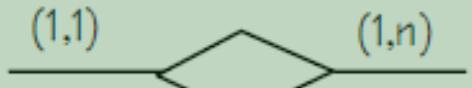
Casamento (IdentM, IdentH, Data, Regime, NomeH, NomeM)

Relacionamentos binários

(0,1) – (0,1) – ambas opcionais

- Solução por adição de colunas melhor
 - Menor número de junções
 - Menor número de chaves
- Solução por tabela própria aceitável
 - Maior número de junções
 - Chave primária não modela cardinalidade 1:1
- Solução por fusão de tabelas é inviável
 - Chave primária não garante cardinalidade 1:1
 - Modelagem de participação parcial comprometida

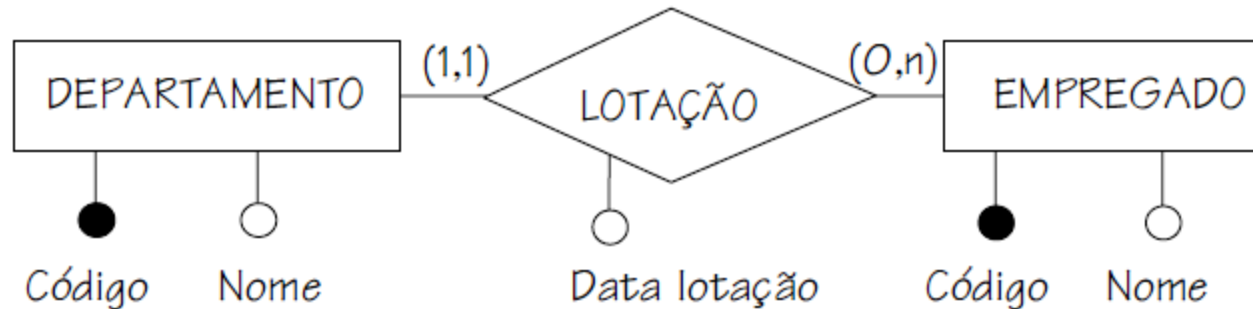
d.2. Relacionamentos binários (1:N)

Tipo de relacionamento		Regra de implementação		
		Tabela própria	Adição coluna	Fusão tabelas
“1” opc.	(O,1)  (O,n)	±	✓	×
	(O,1)  (1,n)	±	✓	×
“1” obr.	(1,1)  (O,n)	×	✓	×
	(1,1)  (1,n)	×	✓	×

✓ Alternativa preferida ± Pode ser usada
 × Não usar

Relacionamentos binários

(1,1) – (0,N) – obrigatória e opcional

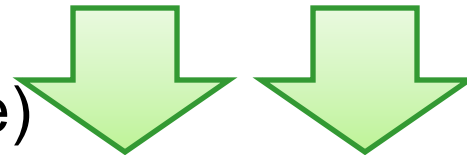


Adição de Colunas

Departamento (CodDept, Nome)

Empregado (CodEmp, Nome, CodDept, DataLota)

CodDept referencia Departamento



Relacionamentos binários

(1,1) – (0,N) – obrigatória e opcional

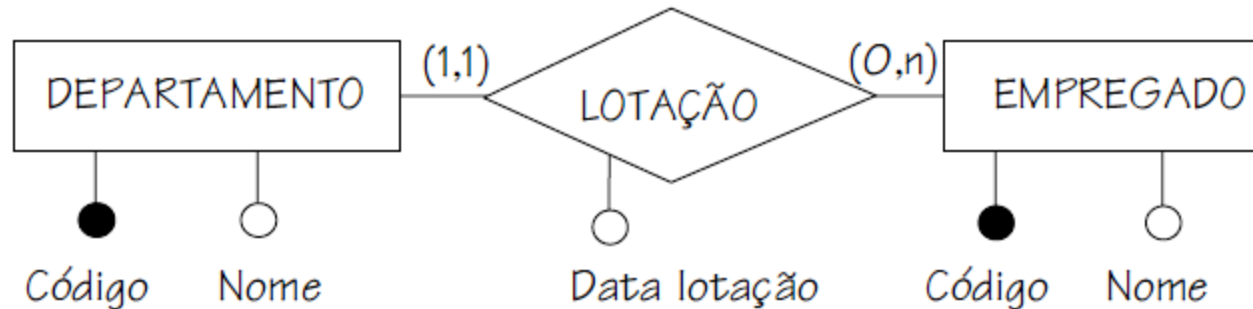


Tabela Própria

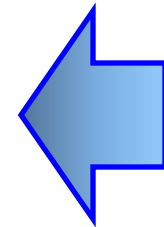
Departamento (CodDept, Nome)

Empregado (CodEmp, Nome)

Lotacao (CodEmp, CodDept, DataLota)

CodDept referencia Departamento

CodEmp referencia Empregado

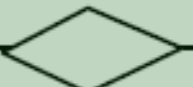





Relacionamentos binários

(1,1) – (0,N) – obrigatória e opcional

- Adição de colunas é melhor que tabela própria
 - Menor número de chaves
 - Menor número de junções
 - Não há problema de campos opcionais
- Fusão de Tabelas
 - Não se aplica
 - Implicaria em
 - Redundância de dados de departamento ou
 - Tabela aninhada

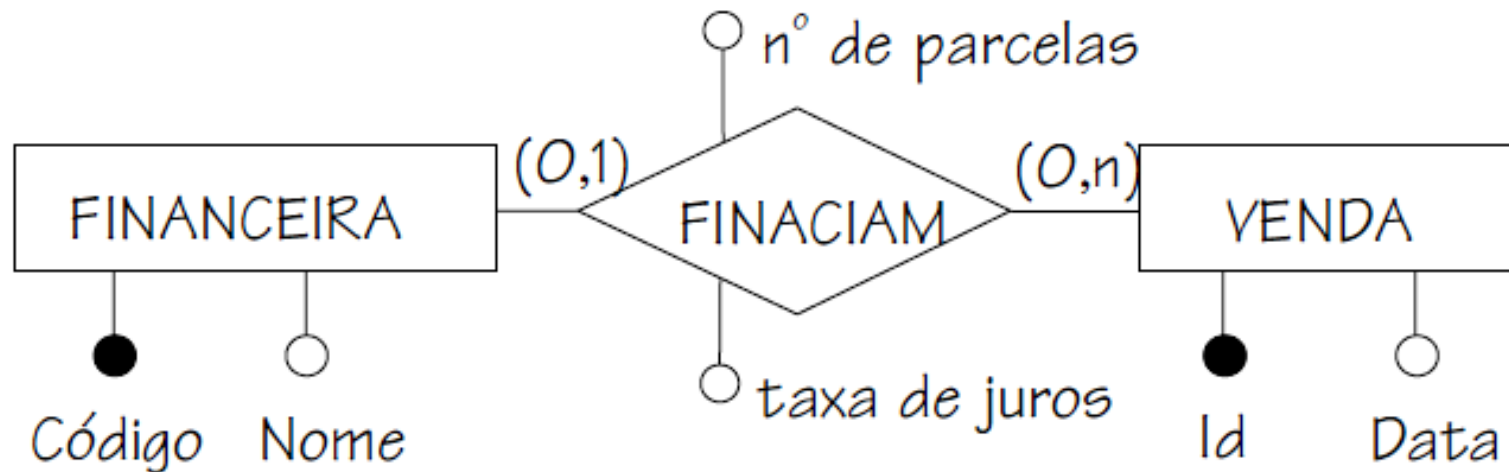
d.2. Relacionamentos binários (1:N)

Tipo de relacionamento				Regra de implementação		
				Tabela própria	Adição coluna	Fusão tabelas
“1” opc.	(O,1)		(O,n)	±	✓	×
	(O,1)		(1,n)	±	✓	×
“1” obr.	(1,1)		(O,n)	×	✓	×
	(1,1)		(1,n)	×	✓	×

✓ Alternativa preferida ± Pode ser usada
 × Não usar

Relacionamentos binários

(0,1) – (0,N) – ambas opcionais



Adição de Colunas

Financeira (CodFin, Nome)

Venda (IdVenda, Data, CodFin, NoParc, TxJuros)

CodFin referencia Financeira

Relacionamentos binários

(0,1) – (0,N) – ambas opcionais

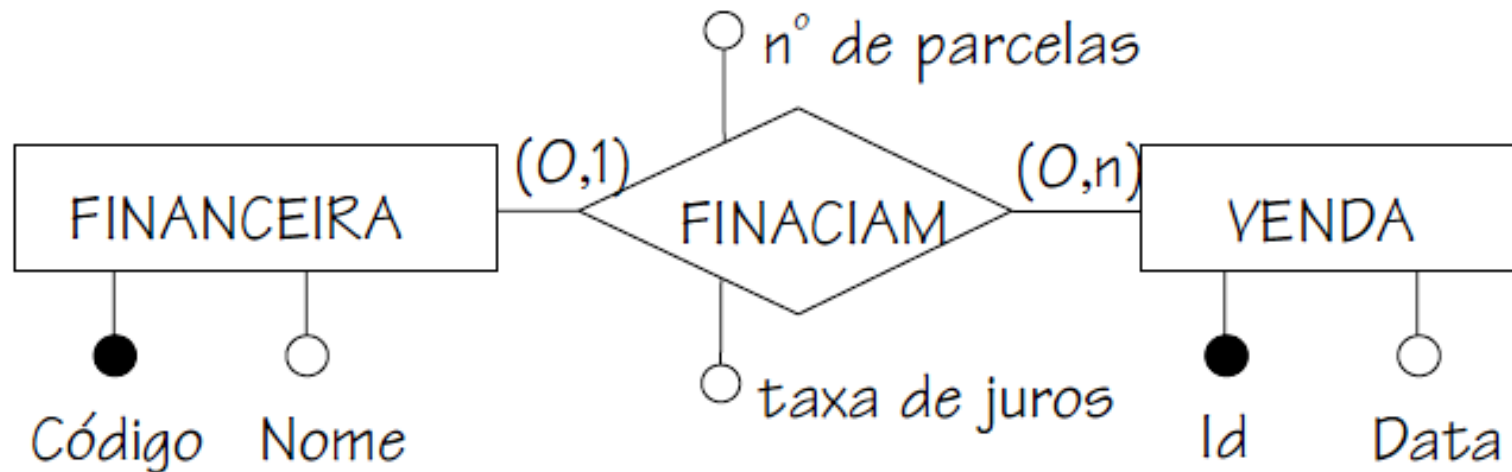


Tabela Própria

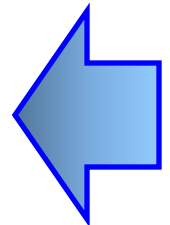
Financeira (CodFin, Nome)

Venda (IdVenda, Data)

Financiam (IdVenda, CodFin, NoParc, TxJuros)

IdVenda referencia Venda

CodFin referencia Financeira

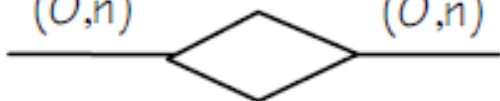
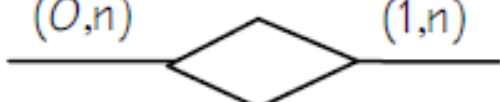
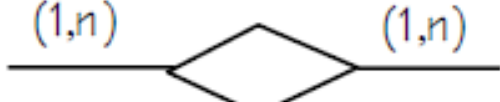


Relacionamentos binários

(0,1) – (0,N) – ambas opcionais

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

d.3 Relacionamento binário (N:M)

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

d.3 Relacionamento binário

(N:M)

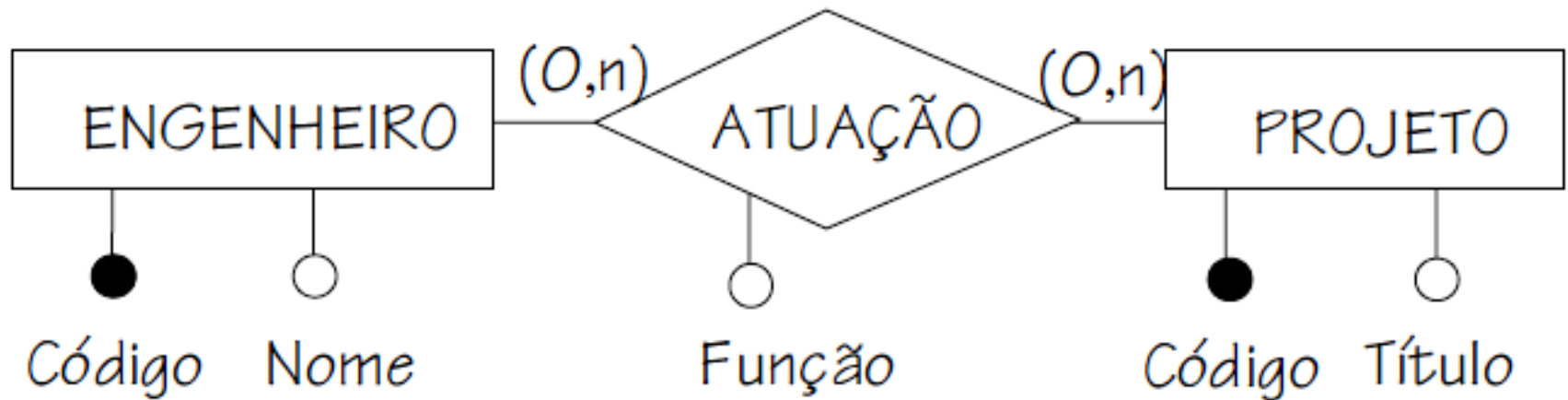


Tabela Própria

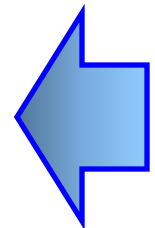
Engenheiro (CodEng, Nome)

Projeto (CodProj, Título)

Atuacao (CodEng, CodProj, Funcao)

CodEng referencia Engenheiro

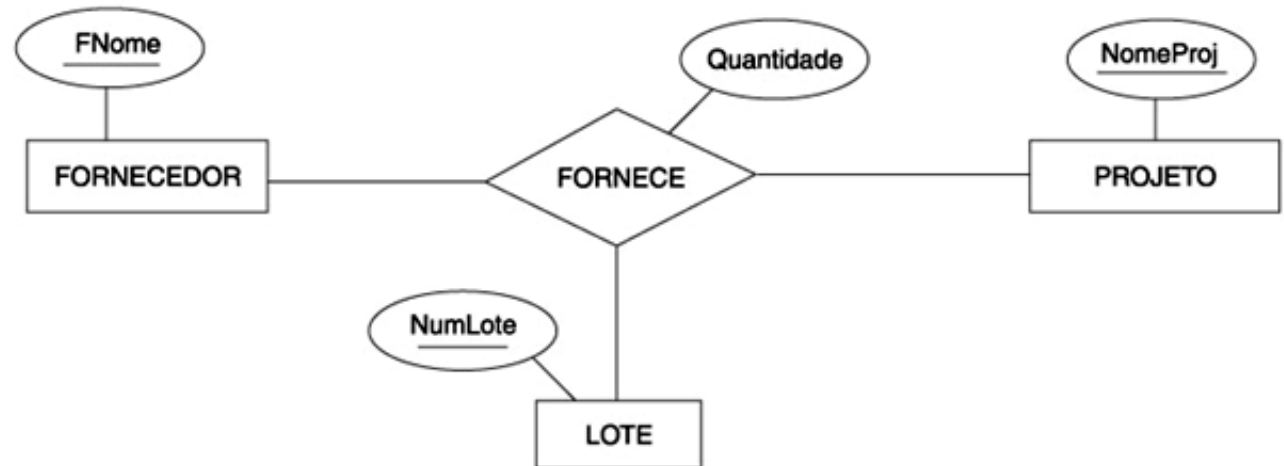
CodProj referencia Projeto



d.4. Relacionamento N-ario

- Não são definidas regras específicas
 - O relacionamento é transformado em uma entidade
 - São aplicadas regras de implementação de relacionamentos binários
- Nova entidade Rel
 - Colunas = chaves primárias das tabelas relacionadas

Relacionamento N-ario



FORNECEDOR

<u>FNOME</u>	...
--------------	-----

PROJETO

<u>NOMEPROJ</u>	...
-----------------	-----

LOTE

<u>NUMLOTE</u>	...
----------------	-----

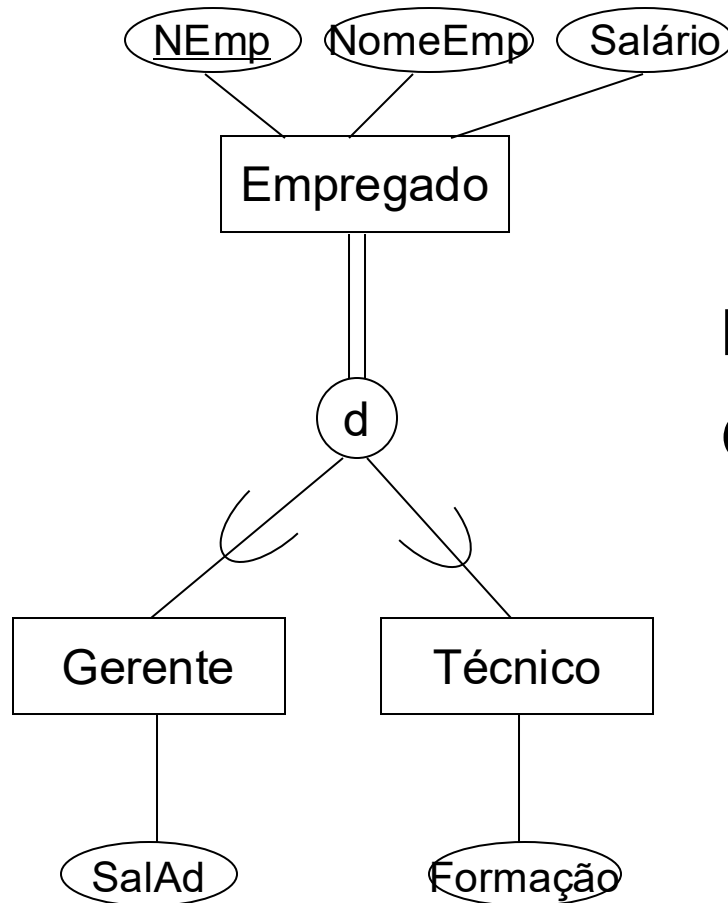
FORNECE

<u>FNOME</u>	<u>NOMEPROJ</u>	<u>NUMLOTE</u>	QUANTIDADE
--------------	-----------------	----------------	------------

e. Hierarquias

- Geralmente quatro opções
 - e.1. Relações : superclasse e subclasses
 - e.2. Relações : subclasses
 - e.3. Relação única
 - e.4. Relação única : atributos tipo

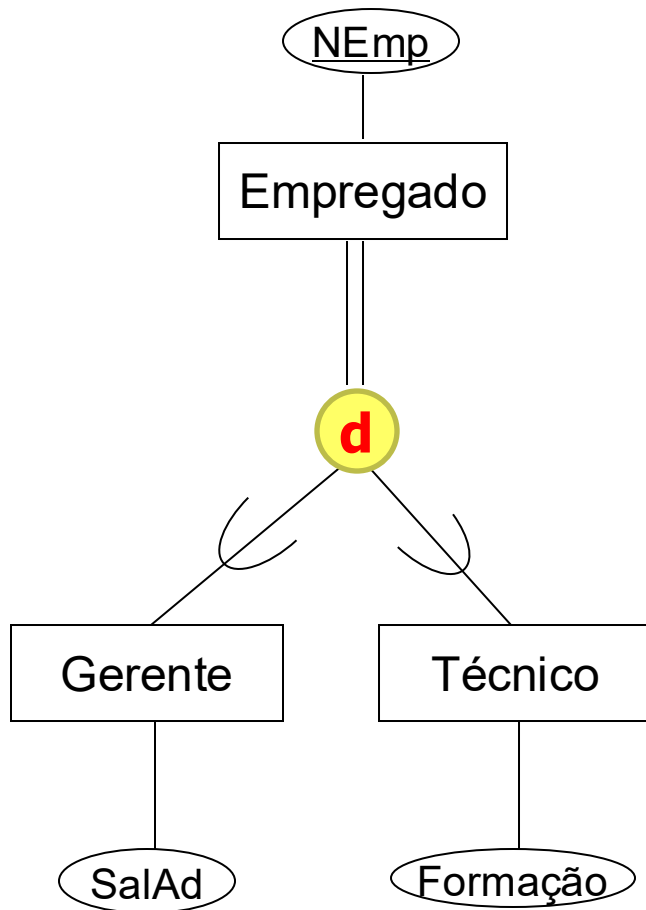
Hierarquias



$\text{Empregado} = \text{Gerente} \cup \text{Técnico}$

$\text{Gerente} \cap \text{Técnico} = \emptyset$

e.1. Relações superclasse+subclasses



Empregado (NEmp, ...)

Gerente (NEmp, SalAd)

Técnico (NEmp, Formação)

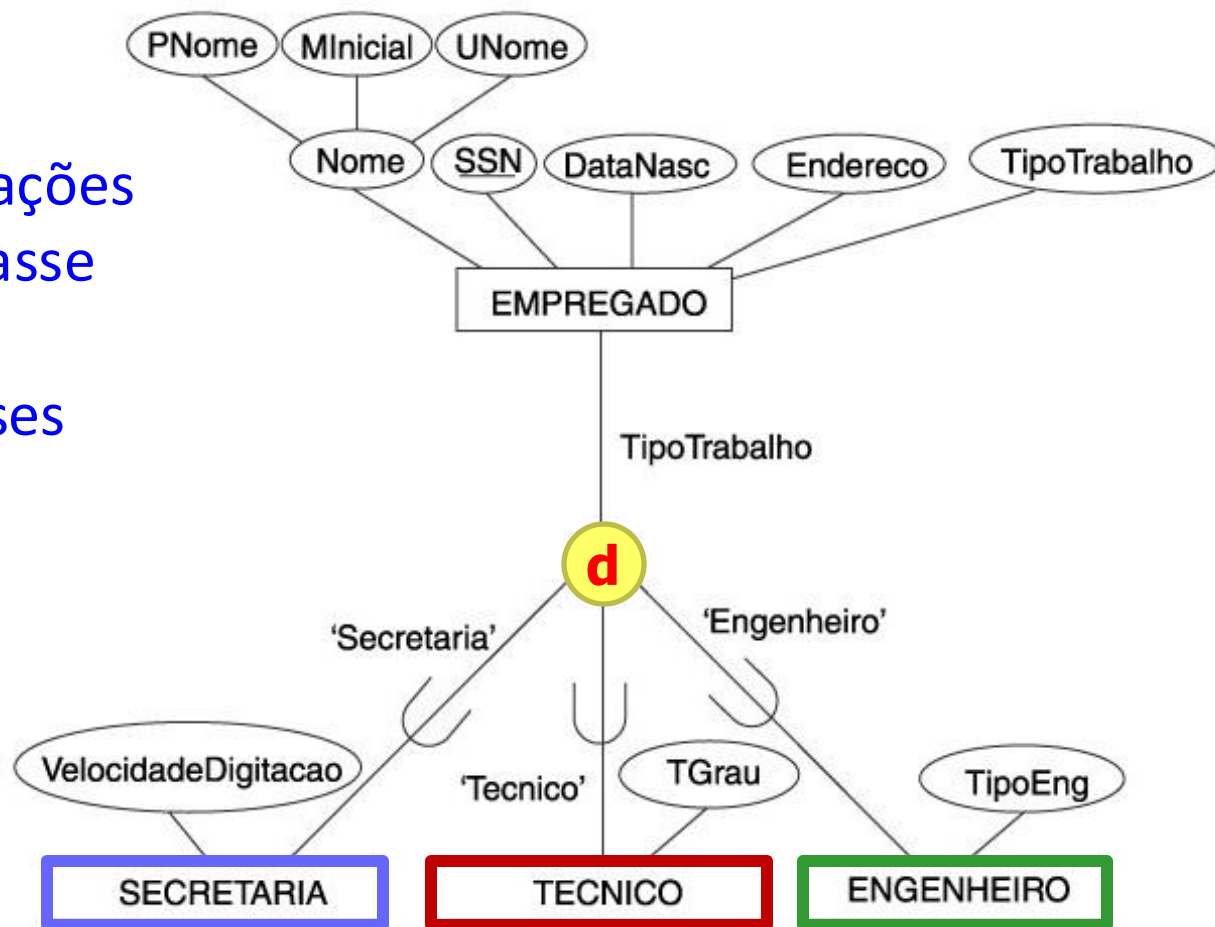
Gerente [NEmp] \xrightarrow{p} Empregado [NEmp]

Técnico [NEmp] \xrightarrow{p} Empregado [NEmp]

$$\pi_{NEmp}(\text{Gerente}) \cap \pi_{NEmp}(\text{Tecnico}) = \emptyset$$

$$\pi_{NEmp}(\text{Gerente}) \cup \pi_{NEmp}(\text{Tecnico}) = \pi_{NEmp}(\text{Empregado})$$

e.1. Relações
superclasse
+
subclasses



EMPREGADO

<u>SSN</u>	PNome	MInicial	UNome	DataNasc	Endereco	TipoTrabalho
------------	-------	----------	-------	----------	----------	--------------

SECRETARIA

<u>SSN</u>	VelocidadeDigitacao
------------	---------------------

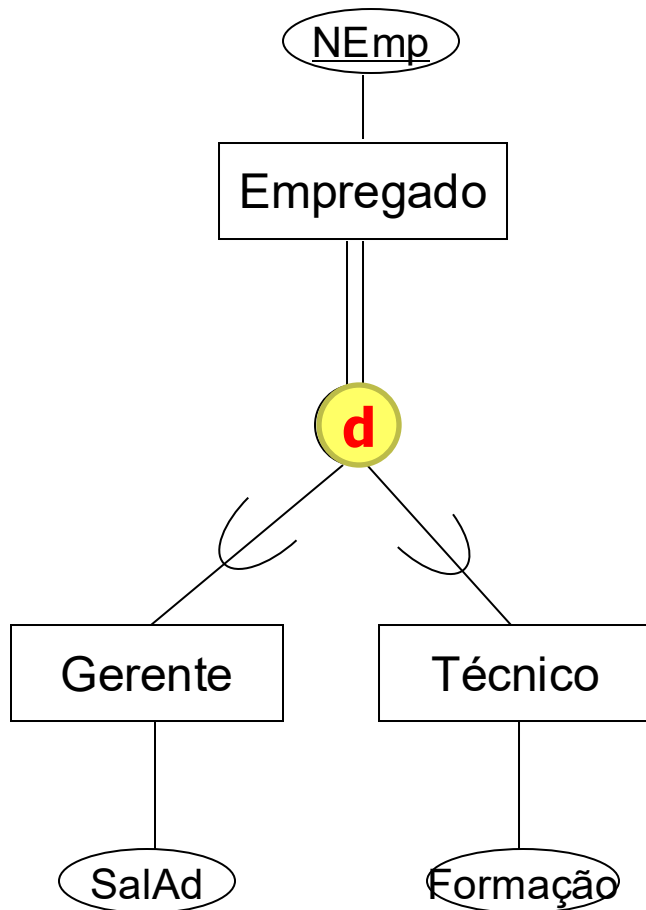
TECNICO

<u>SSN</u>	TGrau
------------	-------

ENGENHEIRO

<u>SSN</u>	TipoEng
------------	---------

e.2. Relações subclasses



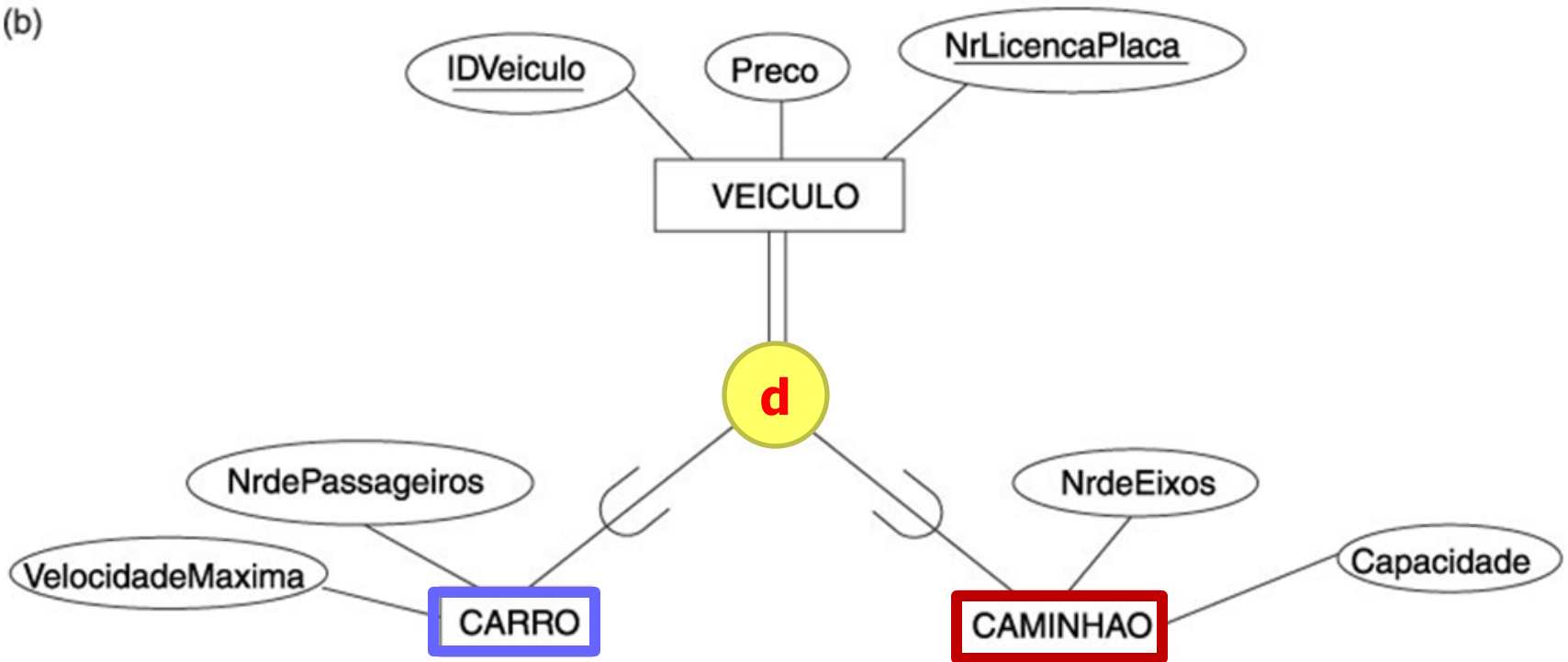
Gerente (NEmp, ..., SalAd)

Técnico (NEmp, ..., Formação)

$$\pi_{NEmp}(\text{Gerente}) \cap \pi_{NEmp}(\text{Tecnico}) = \emptyset$$

$$\pi_{NEmp}(\text{Gerente}) \cup \pi_{NEmp}(\text{Tecnico}) = \pi_{NEmp}(\text{Empregado})$$

(b)



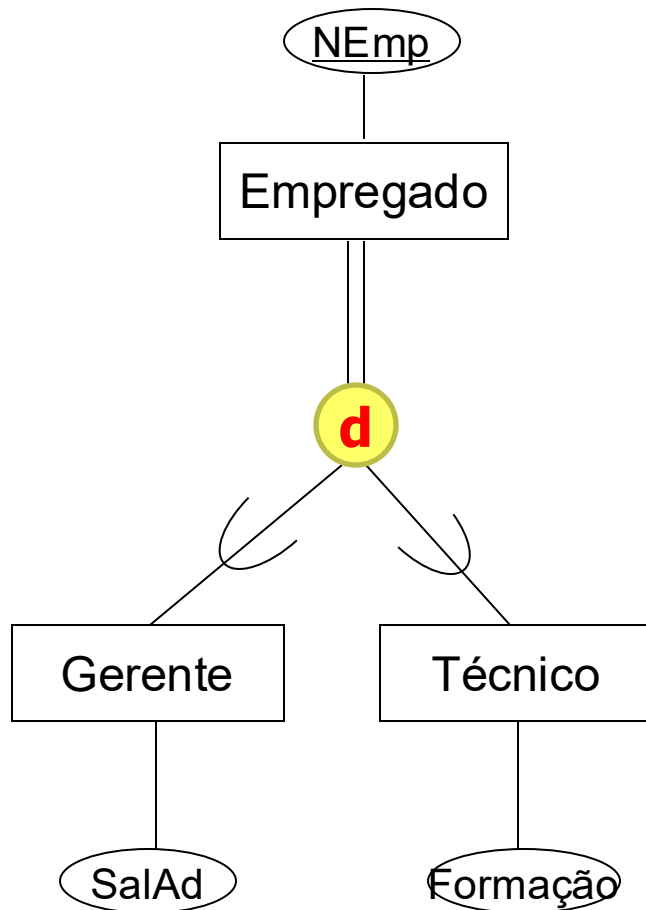
CARRO

<u>IdVeiculo</u>	NrLicencaPlaca	Preco	VelocidadeMax	NrDePassageiros
------------------	----------------	-------	---------------	-----------------

CAMINHAO

<u>IdVeiculo</u>	NrLicencaPlaca	Preco	NrDeEixos	Capacidade
------------------	----------------	-------	-----------	------------

e.3. Relação única



Empregado (NEmp, ..., SalAd, Formação)

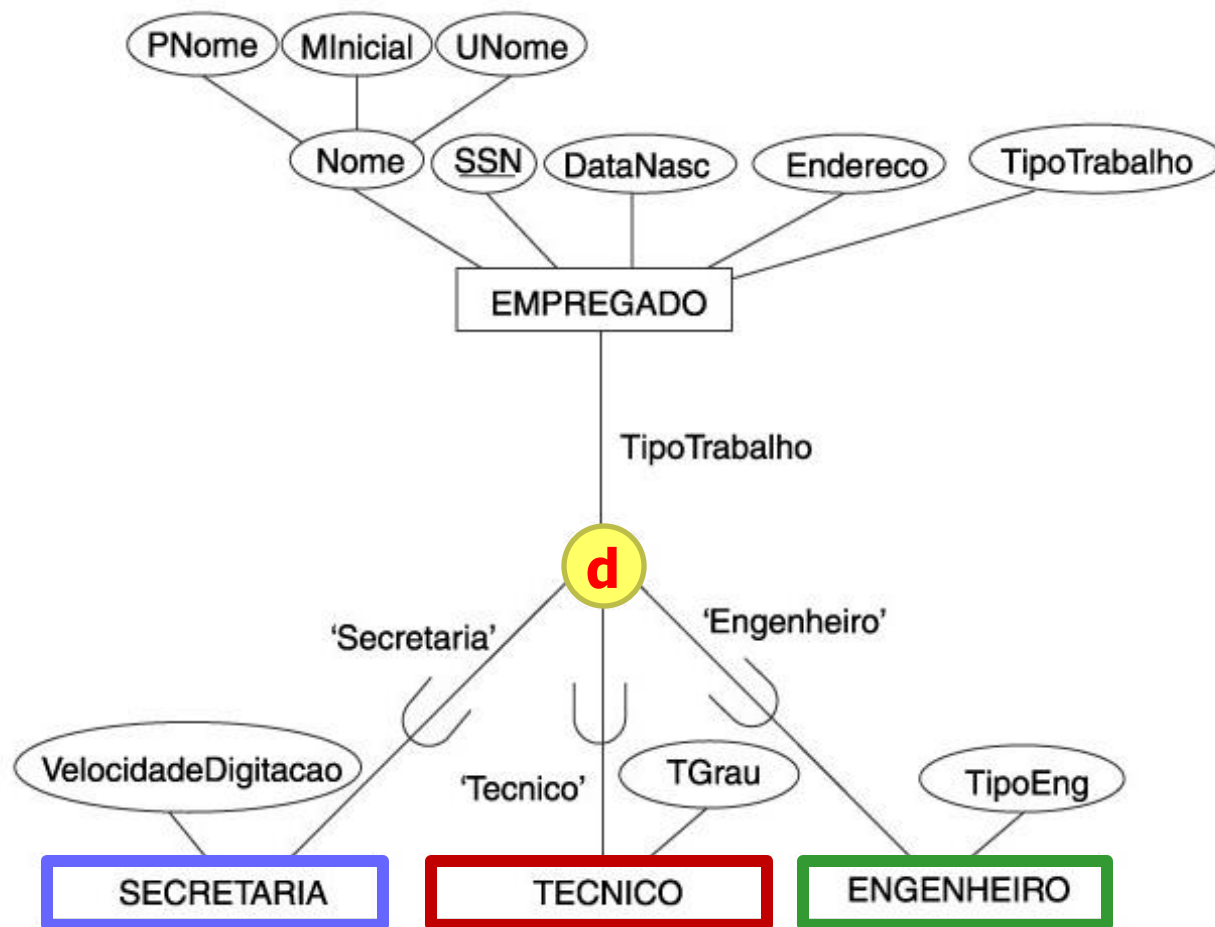
$$\pi_{NEmp}(\sigma_{SalAd \neq \text{nulo}}(\text{Empregado})) \cap$$

$$\pi_{NEmp}(\sigma_{Formação \neq \text{nulo}}(\text{Empregado})) = \emptyset$$

$$\pi_{NEmp}(\sigma_{SalAd \neq \text{nulo}}(\text{Empregado})) \cup$$

$$\pi_{NEmp}(\sigma_{Formação \neq \text{nulo}}(\text{Empregado})) =$$

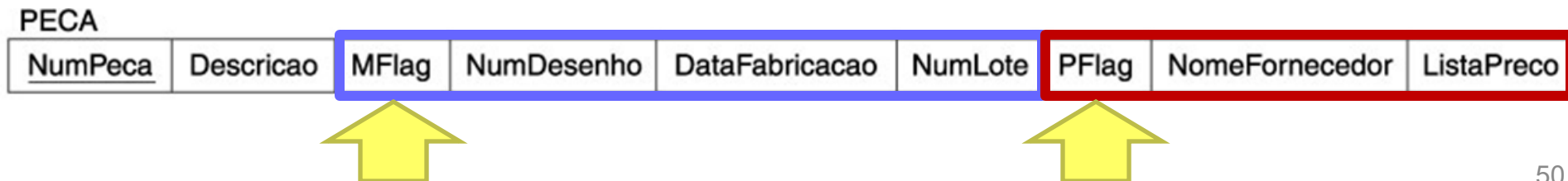
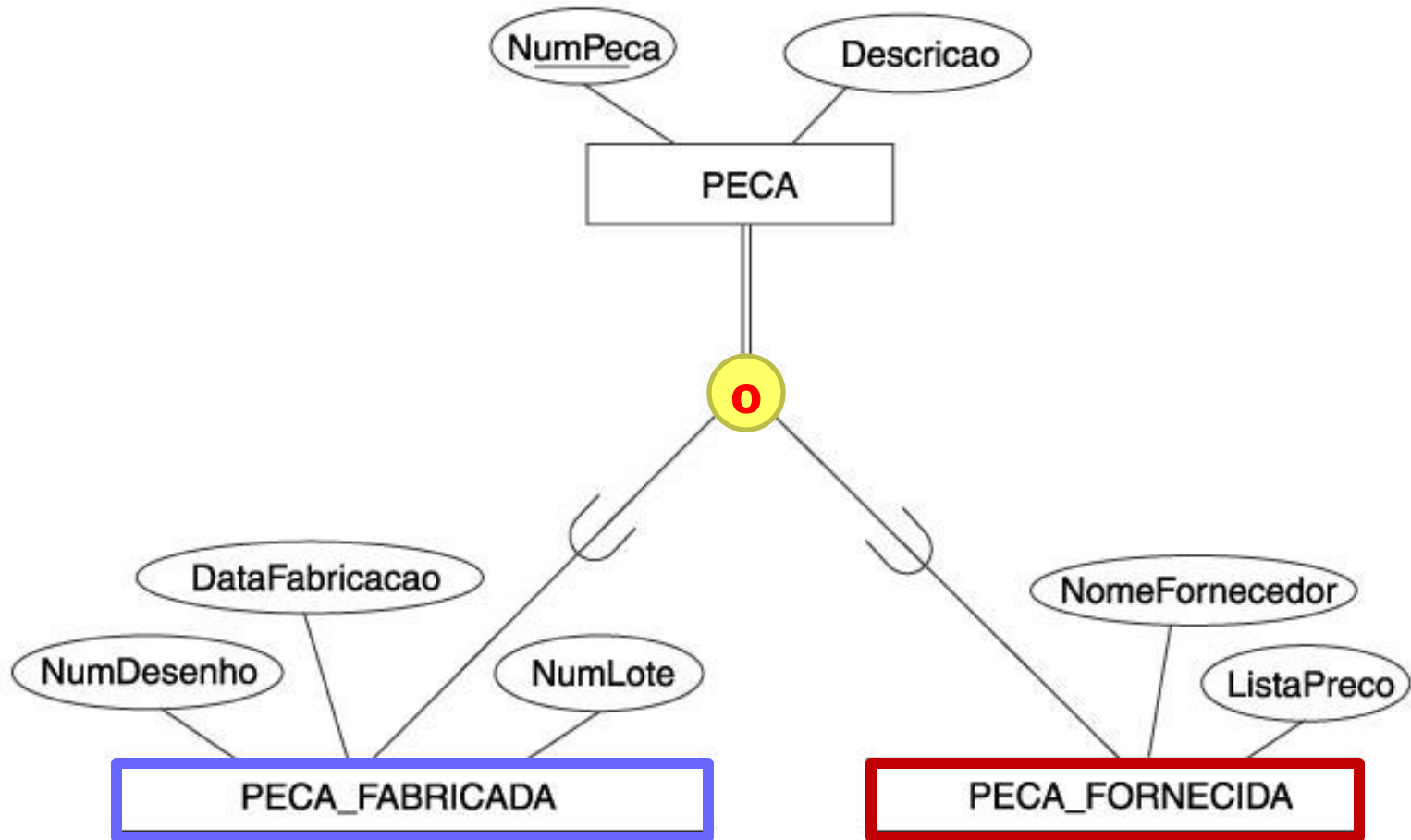
$$\pi_{NEmp}(\text{Empregado})$$



EMPREGADO

<u>SSN</u>	PNome	MInicial	UNome	DataNasc	Endereco	TipoTrabalho	VelocidadeDigitacao	TGrau	TipoEng
------------	-------	----------	-------	----------	----------	--------------	---------------------	-------	---------

e.4. Relação única: atributos tipo



Recomendações

- e.1. Relações : superclasse e subclasses
 - Funciona para total/partial + disjoint/overlapping
- e.2. Relações : subclasses
 - Funciona somente para total + disjoint
 - Precisa de OUTER UNION (ou FULL OUTER JOIN) para obter todas as instâncias da superclasse
- e.3. Relação única (disjoint)
- e.4. Relação única + tipos (overlapping)
 - Trade-off esparsidade vs. eficiência