
Banco de Dados

Modelo ER

DCC-UFLA

Prof. Denilson Alves Pereira

denilsonpereira@dcc.ufla.br

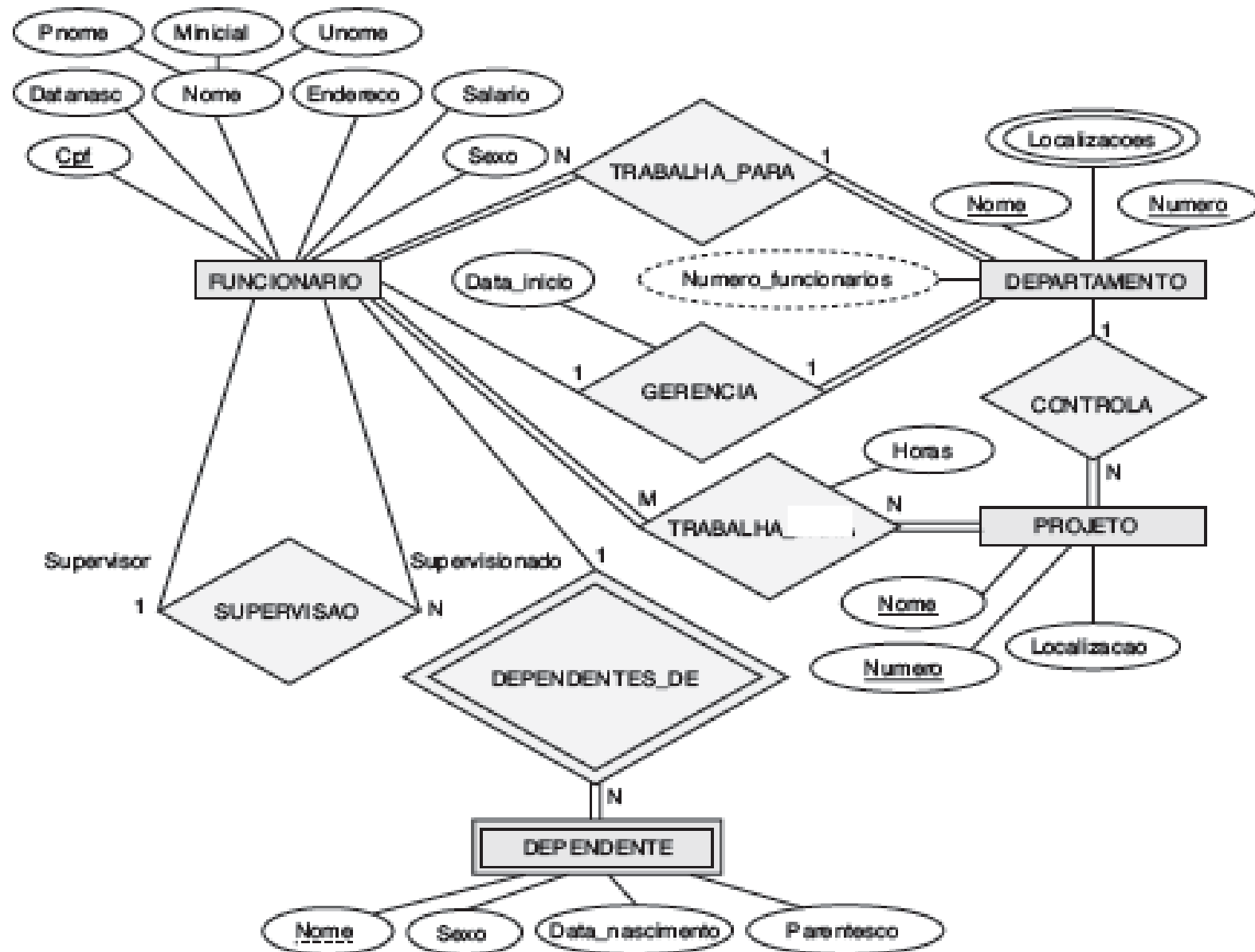
<http://lattes.cnpq.br/4120230814124499>



Modelo ER

- Metodologias de projeto de banco de dados incluem conceitos para especificar operações sobre os objetos do banco de dados.
- O ***Modelo de Entidades e Relacionamentos (ER)*** é um *modelo conceitual* usado para projeto de aplicações de banco de dados.
 - ◆ Independente de aspectos de implementação.
- É um modelo baseado na percepção do mundo real como conjuntos de objetos básicos chamados entidades e nos relacionamentos entre esses objetos.

Diagrama ER - Exemplo



Entidade

- **Entidade** é um objeto que existe no mundo real e é distinguível dos outros objetos.
 - ◆ O funcionário "João da Silva" com o CPF "890.123.456-00"
 - ◆ O projeto de nome "Inovax" e número "555"
- **Tipo Entidade** define uma coleção (ou classe) de entidades que têm os mesmos atributos.
 - ◆ Descreve o esquema para um conjunto de entidades.
 - ◆ Representada como um retângulo no diagrama. **Funcionário**
- **Conjunto de Entidades** é um conjunto com entidades de um mesmo tipo de entidade.
 - ◆ Ex: o conjunto de todos os funcionários pode ser definido como um conjunto de entidades Funcionário.

Entidade

Nome do tipo
de entidade:

FUNCIONARIO

EMPRESA

Nome, Idade, Salario

Nome, Matriz, Presidente

Nome dos
atributos:

f_1 •

(João Silva, 55, 80K)

f_2 •

(Fred Borges, 40, 30K)

f_3 •

(Juliana Campos, 25, 20K)

•
•
•

e_1 •

(Companhia Modelo, São Paulo,
João Silva)

e_2 •

(Rápido Informática, Diadema,
Roberto King)


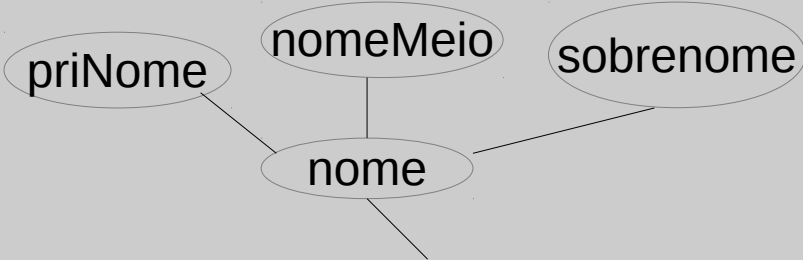
•
•
•

Conjunto de
entidade:
(Extensão)

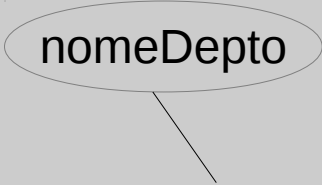

Atributo

- **Atributo** é uma propriedade que descreve uma característica particular de uma entidade.
 - ◆ Ex: Uma entidade funcionário tem os atributos nome, CPF, endereço, salário, sexo e data de nascimento com os valores “João da Silva”, “890.123.456-00”, “Contorno, 1900, Centro”, R\$ 1.500,00, Masculino, 20/10/1970, respectivamente.
- Tipos de atributos:
 - ◆ Simples ou compostos
 - ◆ Monovalorados ou multivalorados
 - ◆ Armazenados ou derivados
 - ◆ Complexos
 - ◆ Nulos
 - ◆ Chaves



Tipos de Atributos

Simple	Compostos
São indivisíveis (atômicos)	Podem ser divididos em subpartes menores
Ex: o atributo "salário" do tipo entidade <i>Funcionário</i>	Ex: o atributo "nome" do tipo entidade <i>Funcionário</i> pode ser dividido em "primeiro nome", "nome do meio" e "sobrenome"
	

Tipos de Atributos

Monovalorados	Multivalorados
Possuem um único valor para uma entidade particular	Podem ter um conjunto de valores para uma entidade particular
Ex: o atributo "nome" do tipo entidade <i>Departamento</i>	Ex: o atributo "localização" do tipo entidade <i>Departamento</i> (um departamento pode estar localizado em mais de um local)
	

Tipos de Atributos

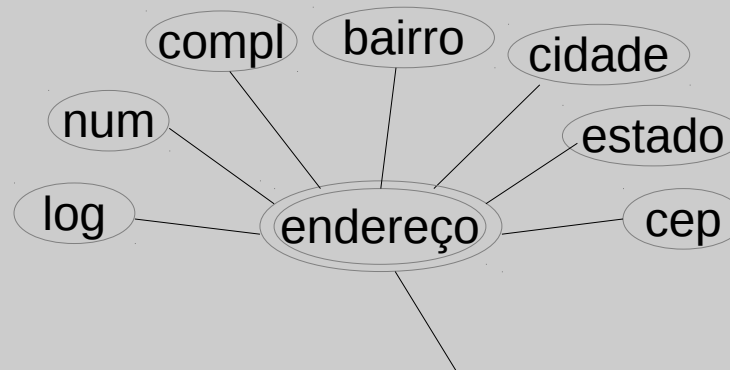
Armazenados	Derivados
Seus valores são fisicamente armazenados no banco de dados	Seus valores podem ser obtidos a partir dos atributos armazenados, e portanto, não precisam ser armazenados no banco de dados
Ex: o atributo "data de nascimento" do tipo entidade <i>Funcionário</i>	Ex: o atributo "idade" do tipo entidade <i>Funcionário</i> , o qual pode ser calculado a partir da data de nascimento
	

Tipos de Atributos

Complexos

São atributos compostos e multivalorados ao mesmo tempo

Ex: o atributo "endereço" de um tipo entidade *Funcionário* poderia ser composto por "logradouro", "no.", "complemento", "bairro", "cidade", "estado" e "cep". Um funcionário poderia ter mais de um endereço (comercial, residencial, de fim de semana etc.)



Tipos de Atributos

Nulos

Podem receber um valor especial, chamado **nulo**. Nulo significa um valor desconhecido. É usado em situações onde um valor não se aplica ou não é conhecido

Ex: o atributo "apartamento" do endereço de um tipo entidade *Funcionário* seria nulo se ele morasse em uma casa

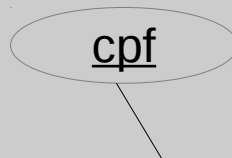
Não existe uma notação para atributos nulos no diagrama ER. Devem ser documentados no dicionário de dados.

Tipos de Atributos

Chave

Atributo chave (ou identificador) é um atributo cujos valores são distintos (únicos) para cada uma das entidades do conjunto de entidades

Ex: o atributo "CPF" do tipo entidade *Funcionário*

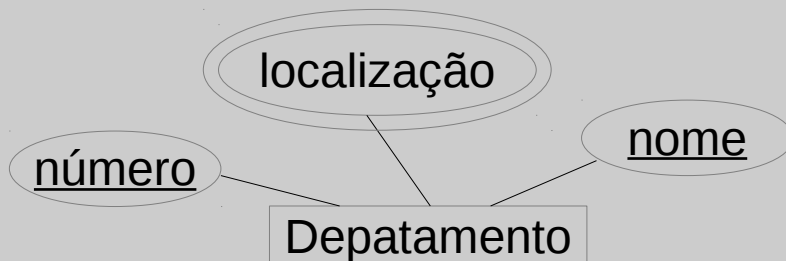


Tipos de Atributos

Vários atributos chave

Um tipo entidade pode ter mais de um atributo chave. Cada um, isoladamente, é uma chave

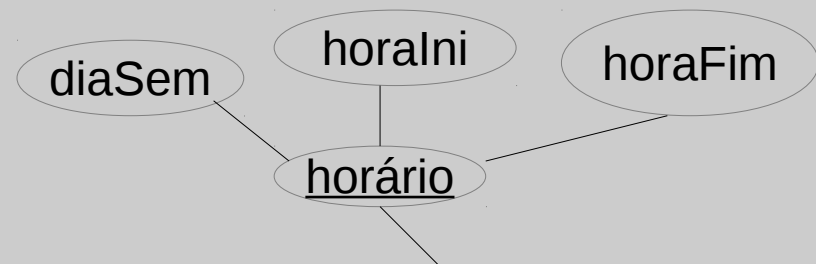
Ex: os atributos "número" e "nome" do tipo entidade *Departamento*



Chave composta

A chave de um tipo entidade pode ser formada por mais de um atributo. Neste caso, a combinação dos valores desses atributos é que é única para cada entidade

Ex: os atributos "dia da semana", "hora de início" e "hora de fim" de um tipo entidade *HorárioDeAula*



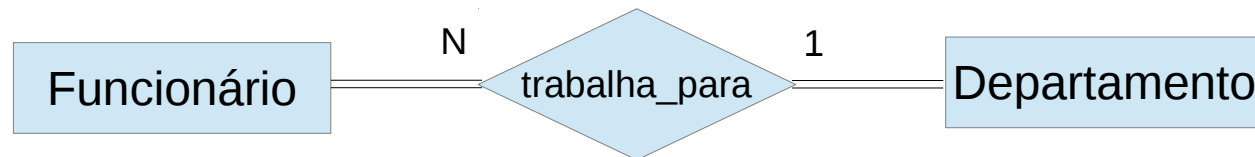
Conjunto de Valores dos Atributos

- Cada atributo de um tipo de entidade está associado com um **conjunto de valores** (ou **domínio** de valores), o qual especifica o conjunto de valores que pode ser atribuído aquele atributo para cada entidade individual.
 - ◆ Exemplos:
 - O conjunto de valores para o atributo "CPF" de um tipo entidade *Funcionário* pode ser definido como uma cadeia de 11 caracteres.
 - O atributo "salário" de um tipo entidade *Funcionário* pode ser definido como um número real maior ou igual a zero.
 - O atributo "sexo" de um tipo entidade *Funcionário* pode ser definido como 'M' ou 'F'.

Relacionamento

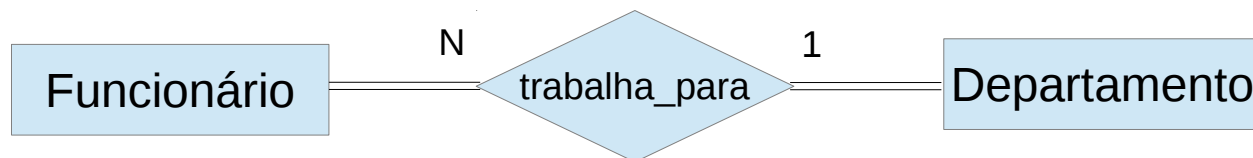
- **Relacionamento** é uma associação entre entidades que representa um fato do mundo real.
- **Tipo Relacionamento:** um tipo relacionamento **R** sobre **n** tipos entidades **E₁, E₂, ..., E_n** define um conjunto de associações entre entidades destes tipos.

Exemplo: o tipo relacionamento "trabalha_para" entre os tipos entidades *Funcionário* e *Departamento* associa cada funcionário com o departamento para o qual ele trabalha.

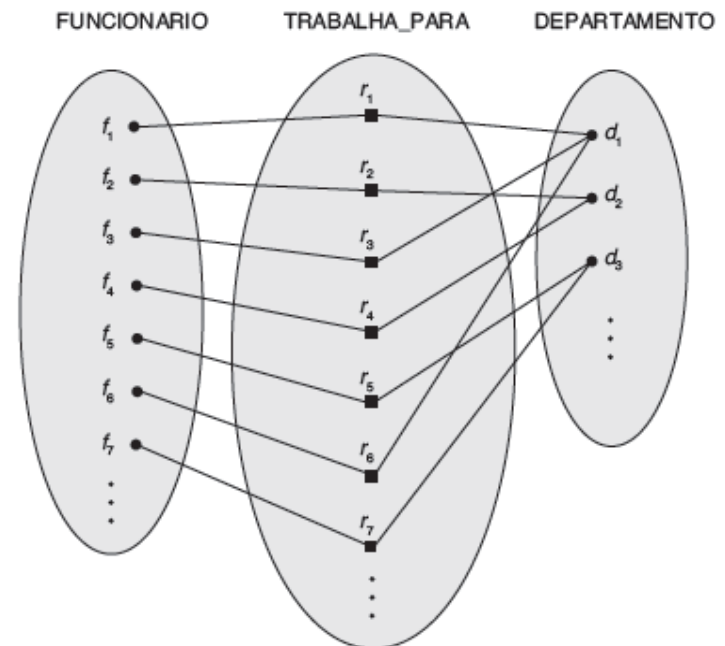


Relacionamento

- **Grau de um tipo relacionamento** é o número de tipos entidades participantes.
 - ◆ O tipo relacionamento "trabalha_para" é de grau 2 (**binário**)

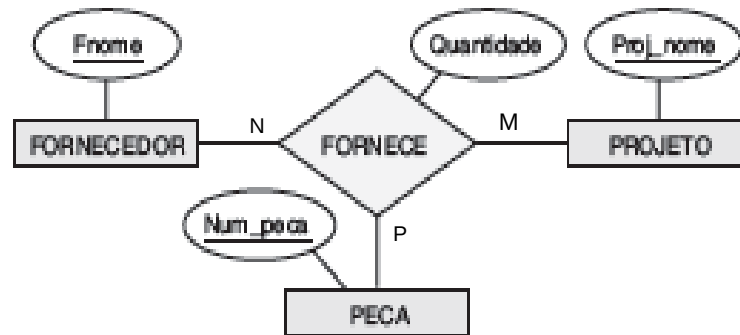


- ◆ Instâncias do relacionamento (r_i)

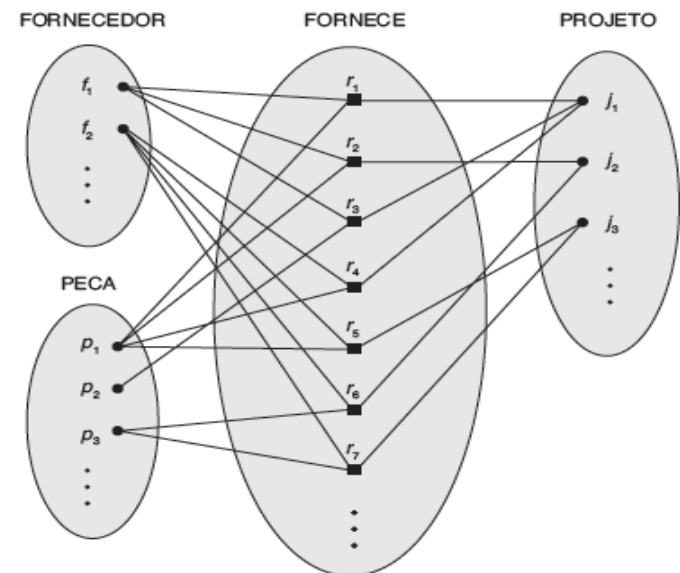


Relacionamento

- ❖ O tipo relacionamento entre "Fornecedor", "Peça" e "Projeto", onde fornecedores fornecem peças para os projetos, é de grau 3 (**ternário**).



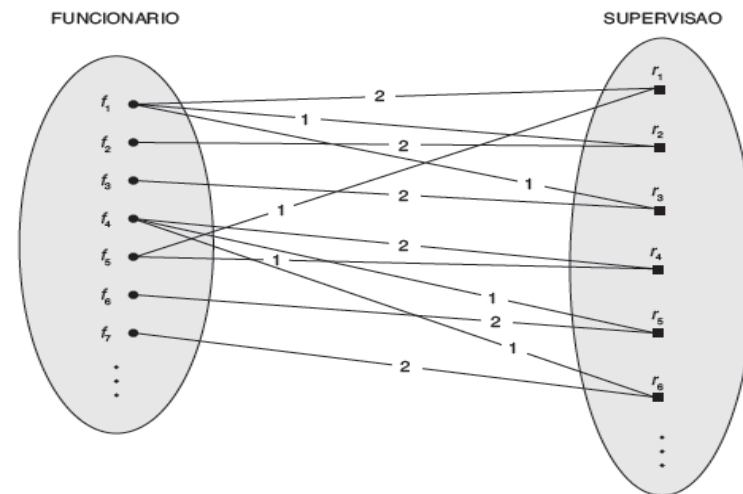
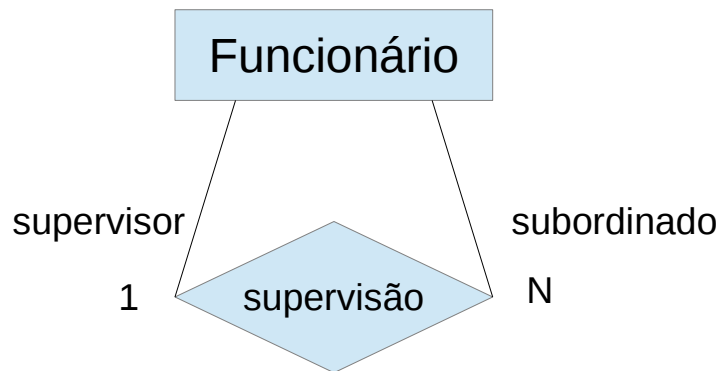
- ❖ Instâncias do relacionamento (r_i)



Relacionamentos Recursivos

- O mesmo tipo de entidade participa mais de uma vez em um tipo de relacionamento em funções diferentes.

■ Ex:

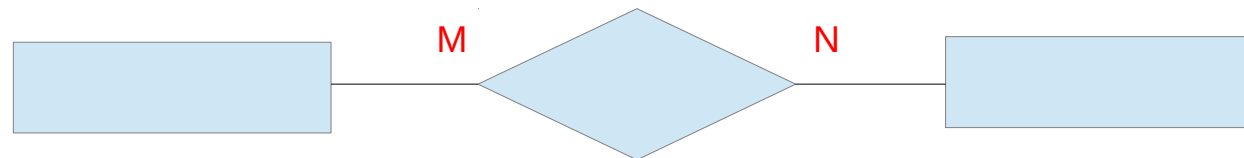
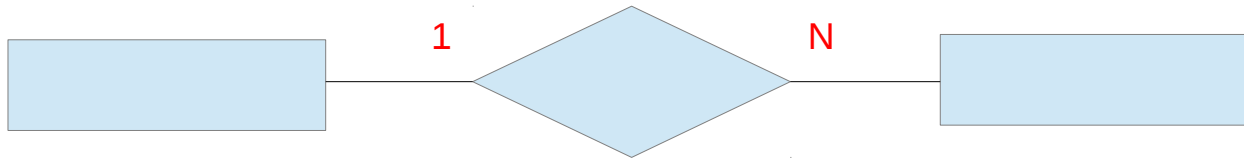
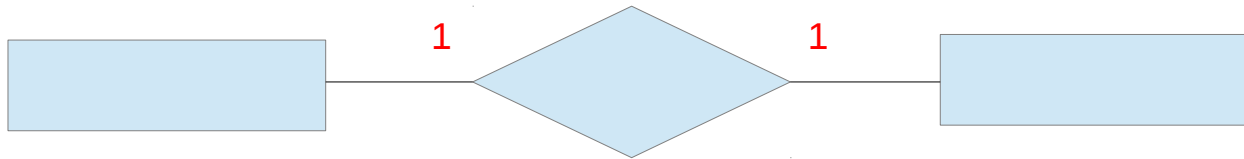


Um relacionamento recursivo SUPERVISAO entre FUNCIONARIO no papel de *supervisor* (1) e FUNCIONARIO no papel de *subordinado* (2).

- É necessário especificar o nome da função (papel) de cada entidade no relacionamento. Ex: supervisor e subordinado.

Restrição de Cardinalidade

- Especifica o número máximo de instâncias de um relacionamento que uma entidade pode participar.

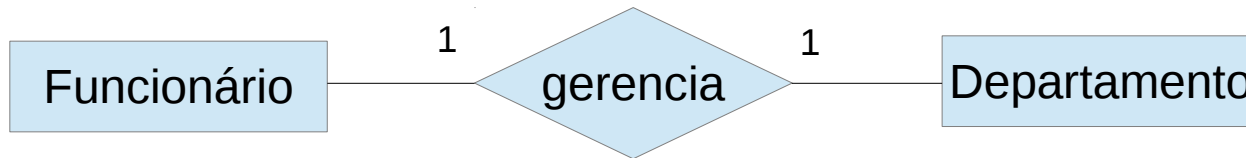


Restrição de Cardinalidade

- Para um tipo de relacionamento binário R entre os tipos entidades A e B, tem-se as seguintes relações de cardinalidade:
 - ♦ 1:1 (um-para-um) - uma entidade de A está associada a, no máximo, uma entidade de B, e uma entidade de B está associada a, no máximo, uma entidade de A.
 - ♦ 1:N (um-para-muitos) - uma entidade de A está associada a várias (zero ou mais) entidades de B, mas uma entidade de B está associada a, no máximo, uma entidade de A.
 - ♦ M:N (muitos-para-muitos) - uma entidade de A está associada a várias entidades de B, e uma entidade de B está associada a várias entidades de A.

Restrição de Cardinalidade

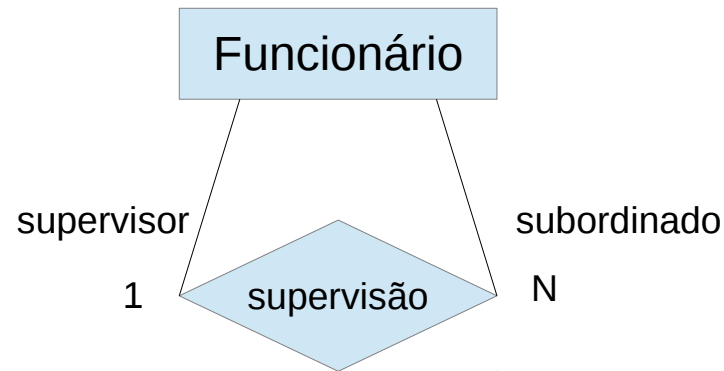
■ Relação de cardinalidade 1:1



- ◆ **Um** dado funcionário pode gerenciar, no máximo, **um** departamento.
- ◆ **Um** dado departamento pode ter, no máximo, **um** funcionário como gerente.

Restrição de Cardinalidade

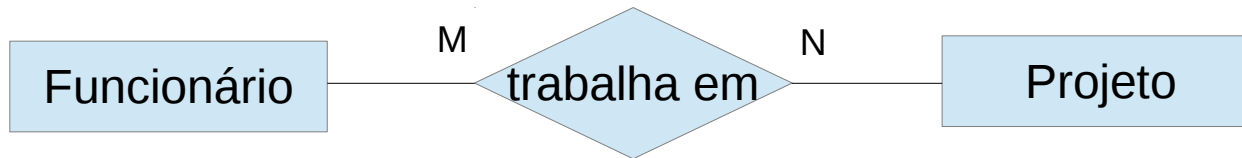
■ Relação de cardinalidade 1:N



- ♦ **Um** dado funcionário (no papel de supervisor) pode supervisionar **vários (zero ou mais)** funcionários (no papel de subordinado).
- ♦ **Um** dado funcionário (no papel de subordinado) pode ser supervisionado por, no máximo, **um** funcionário (no papel de supervisor).

Restrição de Cardinalidade

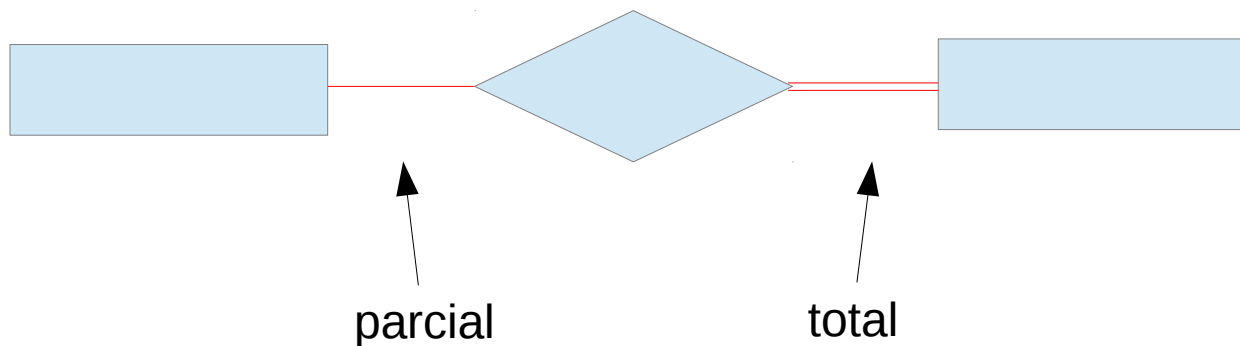
■ Relação de cardinalidade M:N



- ♦ **Um** dado funcionário pode trabalhar em **vários (zero ou mais)** projetos.
- ♦ **Um** dado projeto pode ter **vários (zero ou mais)** funcionários trabalhando nele.

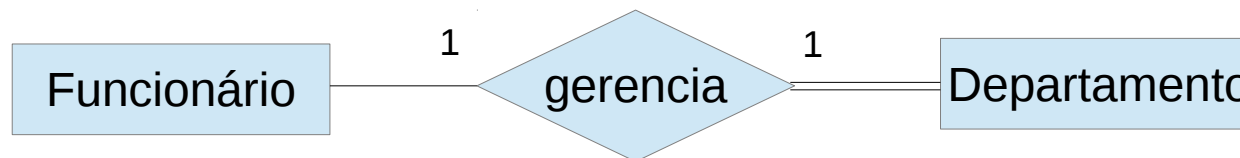
Restrição de Participação

- Especifica se a existência de uma entidade depende de sua associação a outra entidade através de um relacionamento.
- Determina a cardinalidade mínima: zero (participação parcial) ou um (participação total).



Restrição de Participação

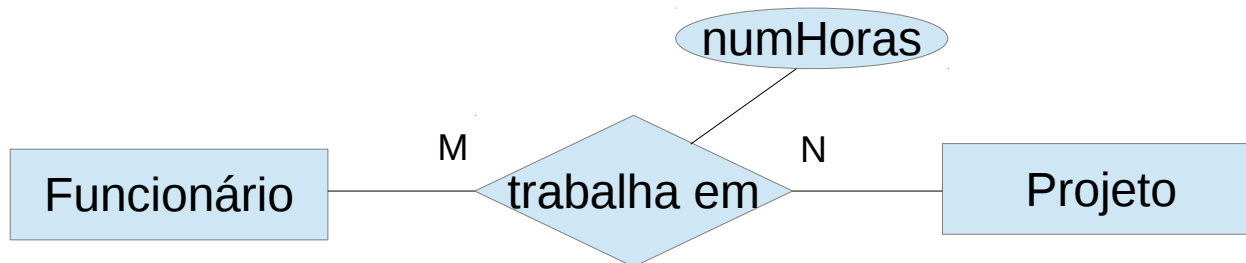
- A participação pode ser:
 - ◆ **Total** (dependência de existência): todas as instâncias ***devem*** participar do relacionamento.
 - ◆ **Parcial**: as instâncias ***podem*** participar do relacionamento.



- Total do lado Departamento (linhas duplas): todo departamento tem obrigatoriamente um gerente.
- Parcial do lado Funcionário (linhas simples): nem todo funcionário é gerente de departamento.

Atributos de Tipos Relacionamento

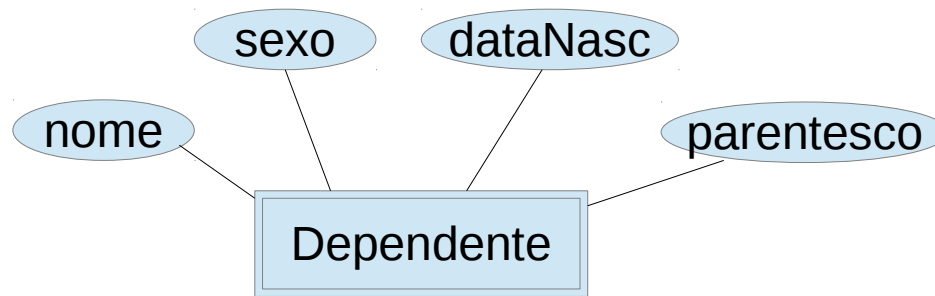
- Tipos relacionamentos podem ter atributos, similares àqueles dos tipos entidades.
- O atributo faz sentido quando as entidades se relacionam. Ele não é típico de nenhum tipo entidade participar.



- NumHoras indica o número de horas por semana que cada funcionário trabalha em cada projeto.
- NumHoras não é atributo de Funcionário e nem de Projeto. Ele só faz sentido quando se toma um par de entidades funcionário-projeto.

Tipo Entidade Fraca

- É um tipo entidade que não possui atributos chave (não tem identificação própria).

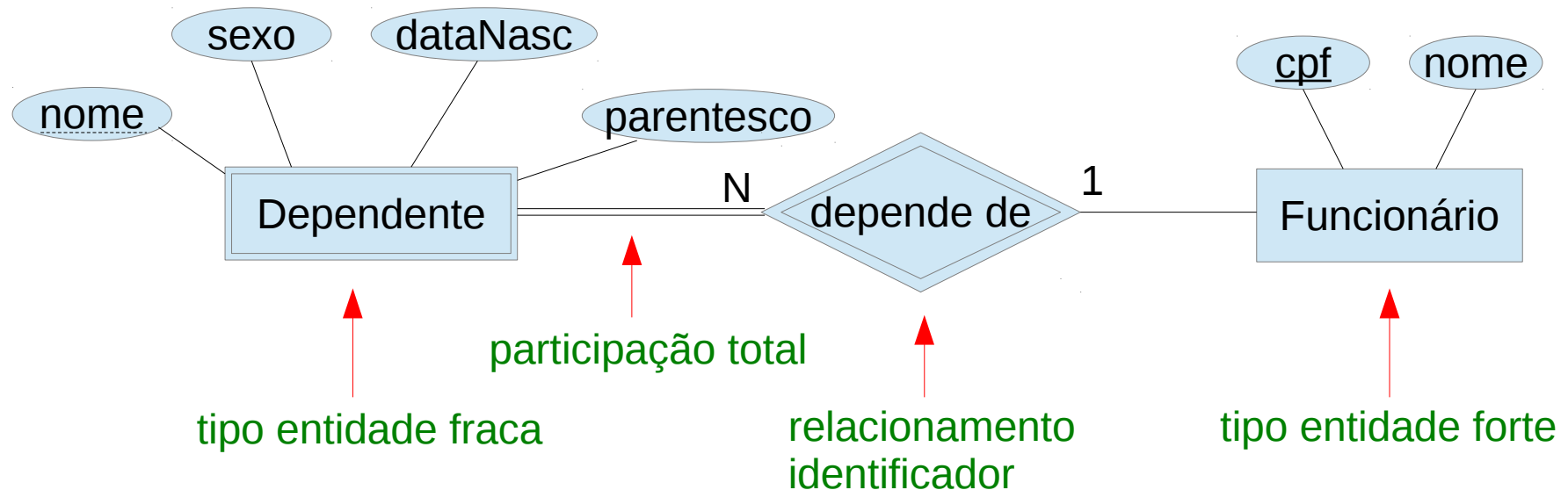


Nenhum dos atributos (ou combinação deles) identifica unicamente as entidades em Dependente.

- Representado no diagrama pelo retângulo duplo.

Tipo Entidade Fraca

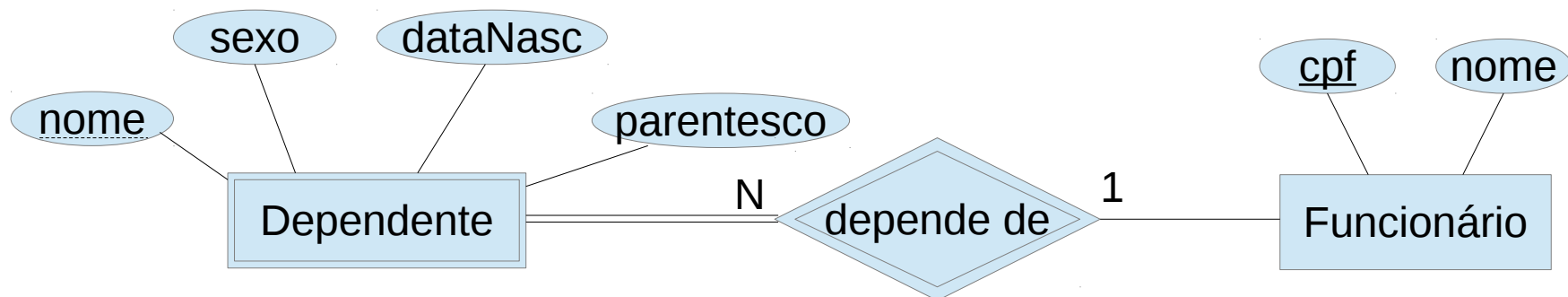
- Está sempre associado a pelo menos um **tipo entidade forte** através de um **relacionamento identificador** (representado no diagrama pelo losângulo duplo).



- Sempre tem uma restrição de participação total (dependência de existência) em relação ao relacionamento identificador.

Tipo Entidade Fraca

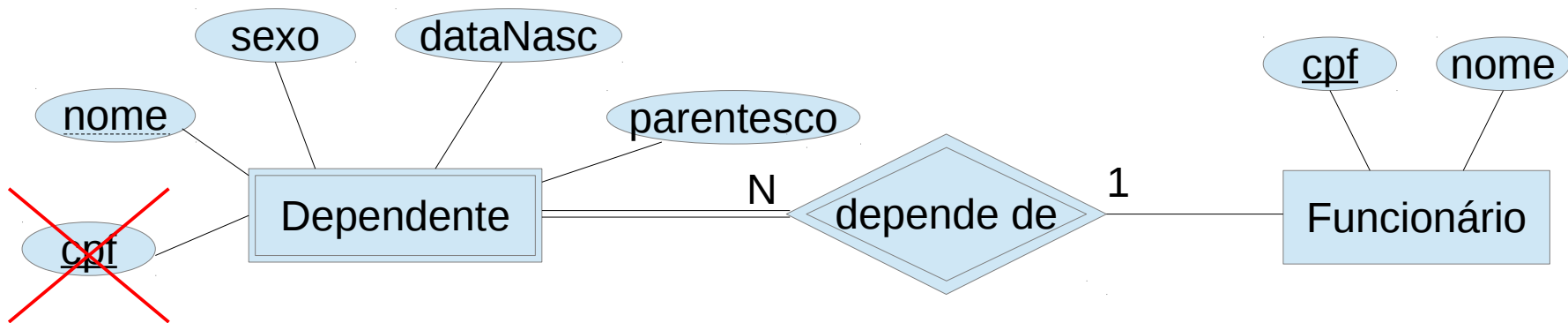
- Sua chave é formada pela combinação de uma chave do tipo entidade forte + uma **chave parcial** própria do tipo entidade fraca.
 - ◆ Chave parcial é representada com um sublinhado tracejado



- O atributo "nome" é uma chave parcial em Dependente
- Um dependente é identificado pelo seu nome + o cpf do funcionário do qual ele depende (essa combinação é única)

Tipo Entidade Fraca

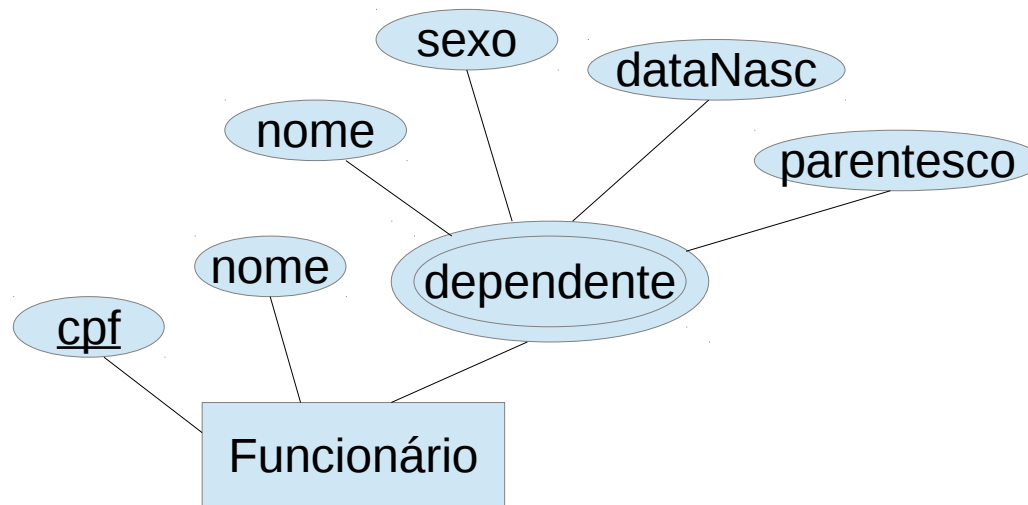
- **Não** adicione uma chave do tipo entidade forte como atributo do tipo entidade fraca. O relacionamento identificador já indica de onde vem a chave.



- "cpf" não é atributo de Dependente

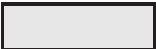
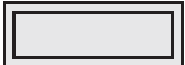
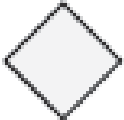

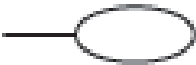
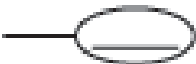



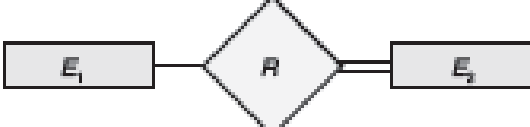

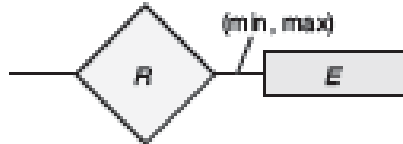
Tipo Entidade Fraca

- Algumas vezes, um tipo entidade fraca pode ser modelado usando atributo complexo (composto e multivalorado)



- ◆ Essa solução não deve ser usada se dependente tiver relacionamentos com outras entidades

Resumo da Notação para Diagramas ER

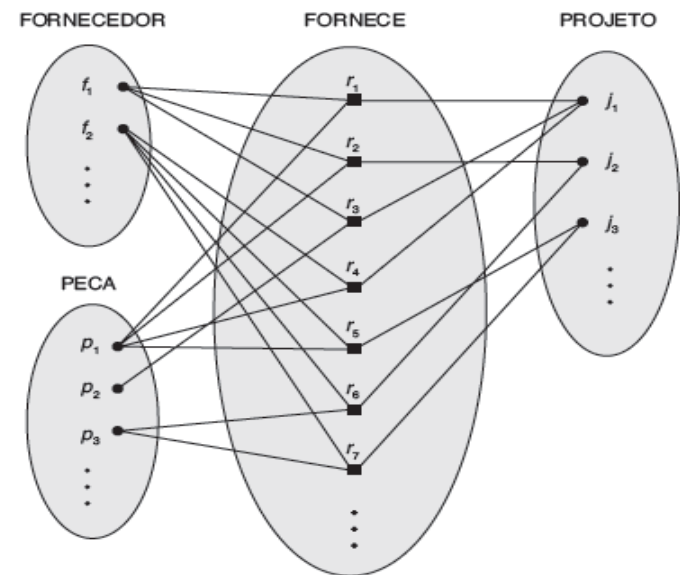
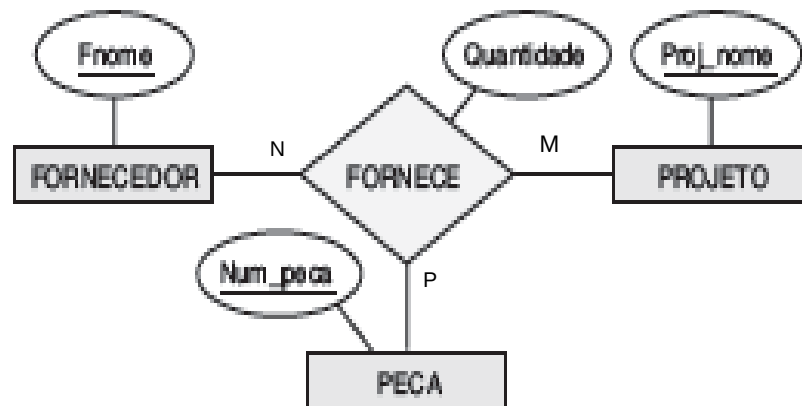
Símbolo	Significado
	Entidade
	Entidade fraca
	Relacionamento
	Relacionamento de identificação
	Atributo
	Atributo-chave
	Atributo multivalorado
	Atributo composto
	Atributo derivado
	Participação total de E_2 em R
	Razão de cardinalidade 1: N para $E_1 : E_2$ em R
	Restrição estrutural (min, max) na participação de E em R

Diretrizes para Escolha de Nomes

- Escolha nomes que transmitam os significados conectados às diferentes construções no esquema.
- Use substantivos para os nomes de tipos entidade.
- Use verbos para os nomes de tipos relacionamento.
- Escolha nomes para os tipos relacionamento binários de forma que possam ser lidos no diagrama ER da esquerda para a direita e de cima para baixo.

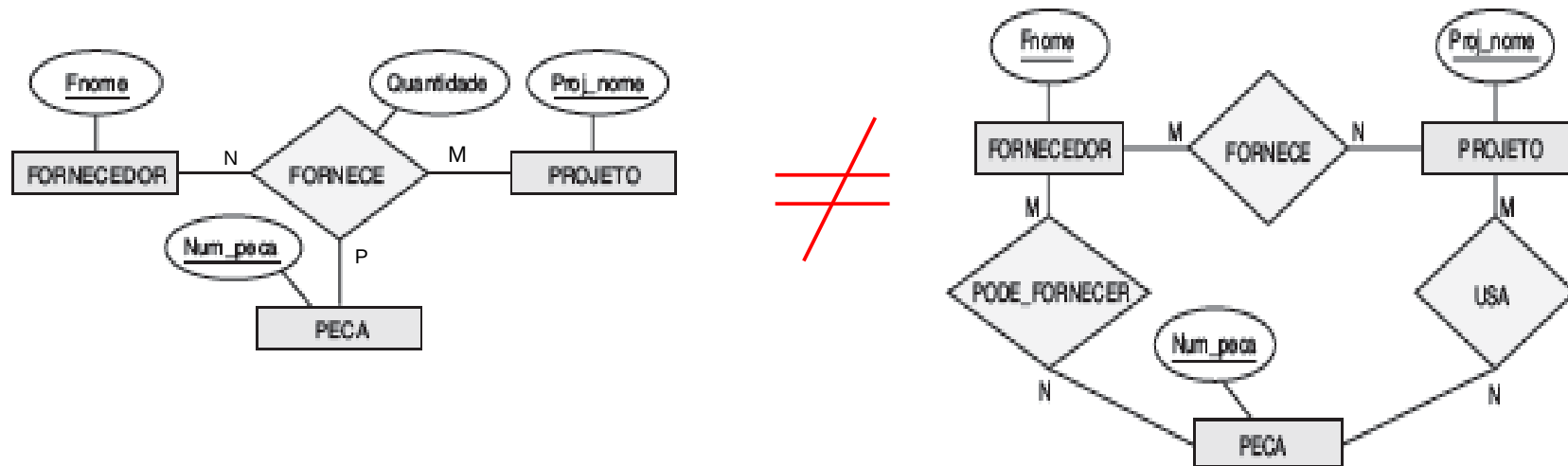
Relacionamentos Ternários

- Um tipo relacionamento é ternário quando possui 3 tipos entidade participantes.
- Exemplo: um relacionamento entre os tipos entidade Fornecedor, Peça e Projeto, o qual indica o fornecedor que forneceu cada peça a cada projeto.



Relacionamentos Ternários

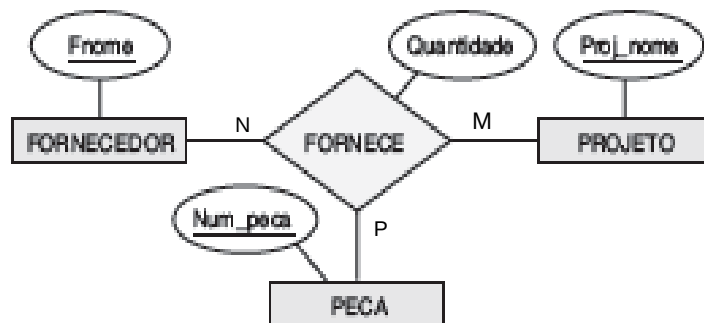
- Como mapear um tipo relacionamento ternário para um conjunto de tipos relacionamento binários?



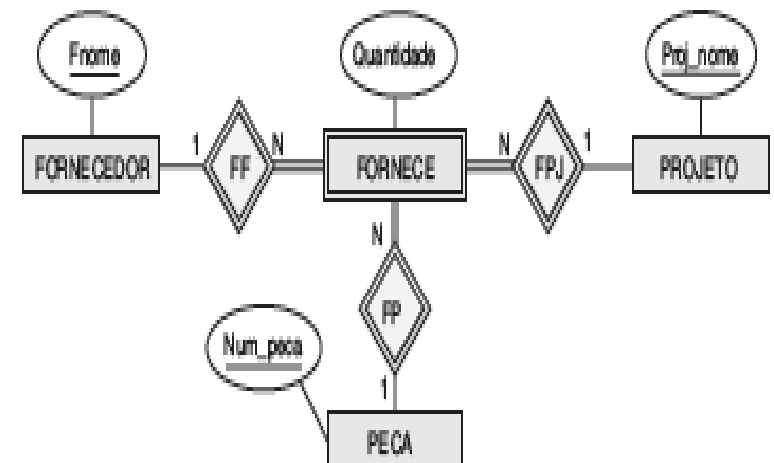
- Os diagramas acima **NÃO** são equivalentes. Por que?
- Em geral, um tipo relacionamento ternário representa mais informação do que três tipos relacionamento binários.

Relacionamentos Ternários

- Como mapear um tipo relacionamento ternário para um conjunto de tipos relacionamento binários?
- ◆ Solução: criar um tipo de entidade fraca sem chaves parciais e com três relacionamentos identificadores. Os três tipos entidade participantes formam juntas os tipos entidade fortes.



=



Notações Alternativas para Diagramas ER

- Existem muitas notações diagramáticas alternativas para exibir diagramas ER.
 - ◆ O próximo slide apresenta algumas delas.
 - ◆ O diagrama de classes da Unified Modeling Language (UML) é outra alternativa, principalmente para projetos orientados a objetos.

Notações Alternativas para Diagramas ER

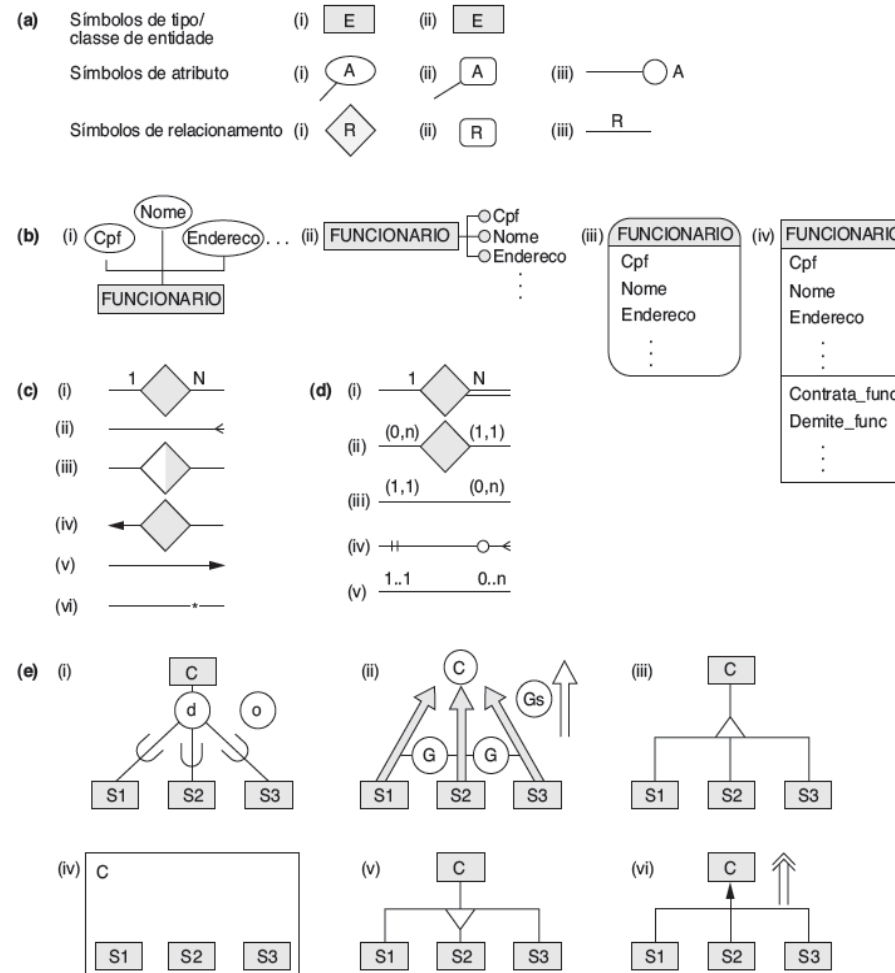
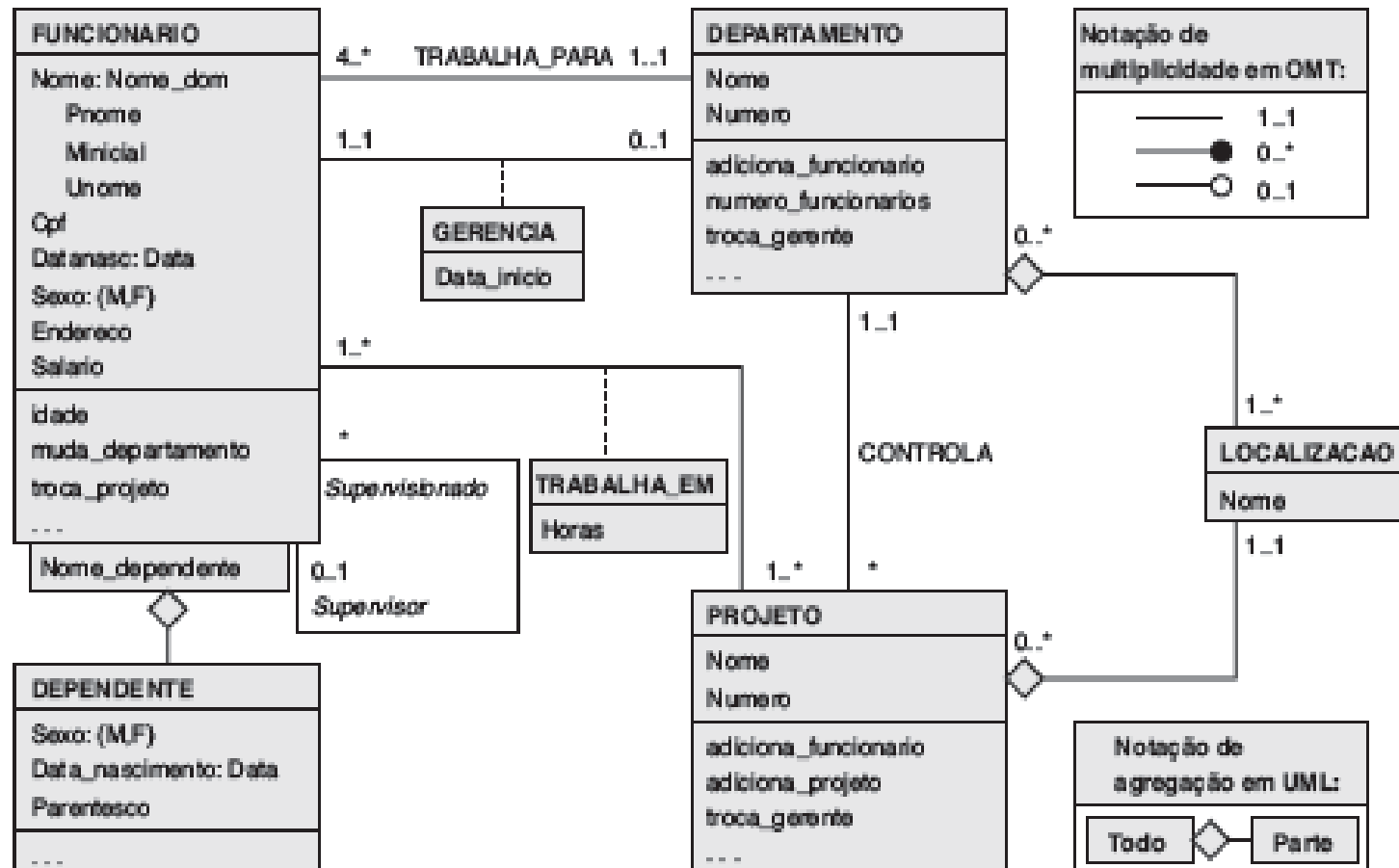


Figura A.1

Notações alternativas. (a) Símbolos para tipo/classe, atributo e relacionamento de entidade. (b) Exibindo atributos. (c) Exibindo razões de cardinalidade. (d) Diversas notações (min, max). (e) Notações para exibir especialização/generalização.

Exemplo de um Diagrama de Classes em UML



Modelo ER Estendido (EER)

Modelo ER Estendido (EER)

- O Modelo ER original foi desenvolvido por Peter Chen e publicado em 1976.
- Com o passar do tempo, foi surgindo a necessidade de se modelar dados mais complexos.
- Surgiu, então, uma extensão ao ER básico denominada Modelo ER Estendido (EER).
- Novos conceitos a serem abordados:
 - ◆ Agregação
 - ◆ Herança
 - ◆ Tipo União (ou Categoria)

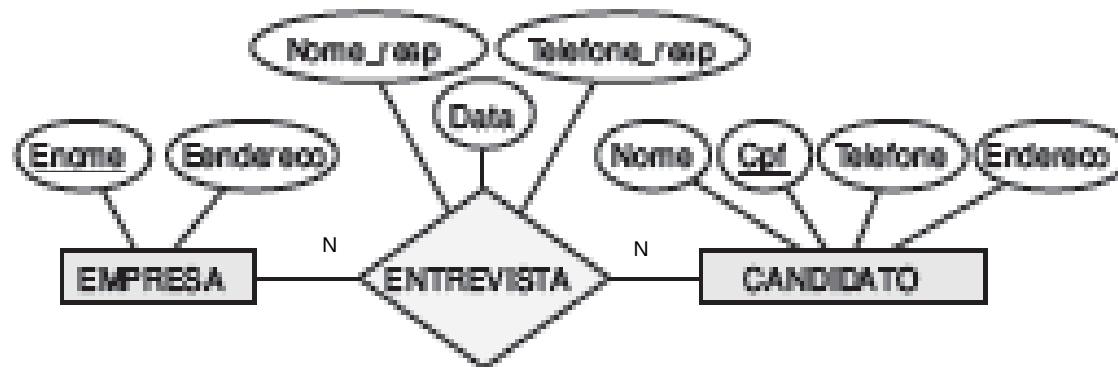
Agregação

- É uma abstração usada para construir objetos compostos a partir de seus objetos componentes.
 - ◆ Representa relacionamentos "todo-parte". O "todo" (objeto composto) é constituído de "partes" (objetos componentes).
 - ◆ A exclusão de um objeto composto ("todo") resulta na exclusão de todos os seus objetos componentes ("partes").

Agregação

Exemplo:

- ♦ O diagrama abaixo representa um banco de dados de entrevistas para emprego de uma agência de RH.

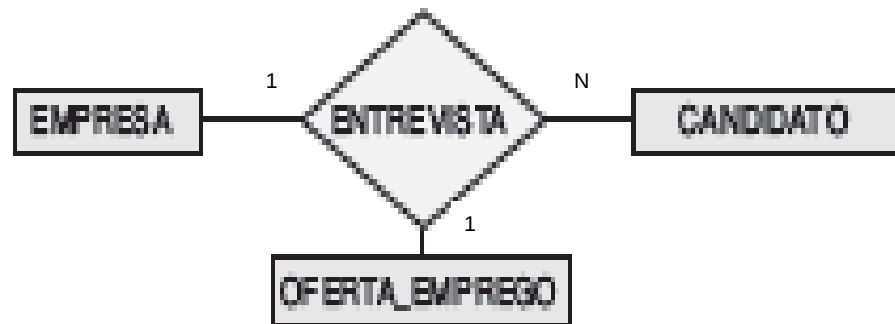


- ♦ Suponha que algumas entrevistas resultem em oferta de emprego, e outras não.
- ♦ Como representar este fato?

Agregação

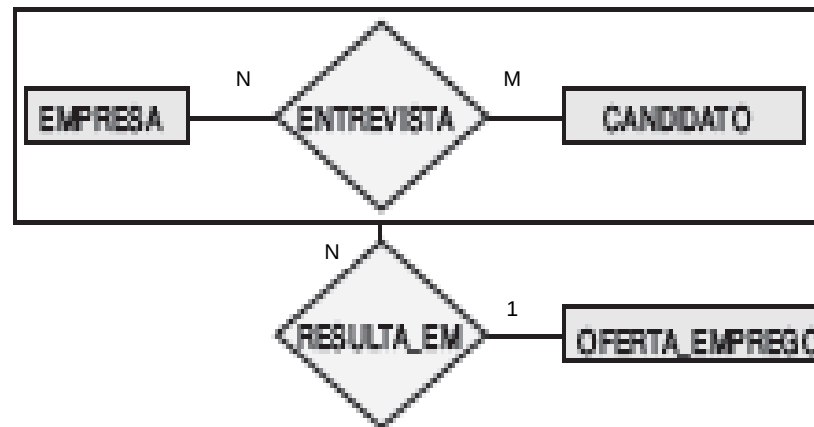
- **Opção 1:** usar um tipo relacionamento ternário

==> INCORRETO pois requer que cada instância do relacionamento entrevista tenha uma oferta de emprego.



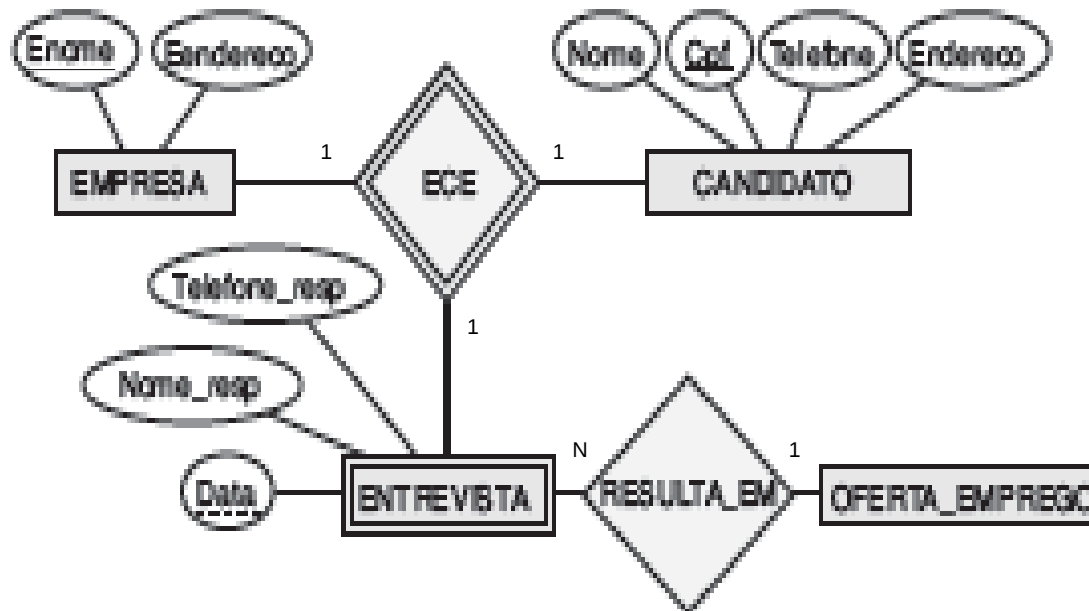
Agregação

- **Opção 2:** criar uma classe agregada de nível mais alto composta por Empresa, Candidato e Entrevista e relacioná-la a Oferta de Emprego
- ◆ **Não faz parte da notação do diagrama ER,** mas é usada por algumas ferramentas de modelagem.



Agregação

- **Opção 3:** criar um tipo entidade fraca Entrevista
 - ◆ Opção correta para diagramas ER



Herança

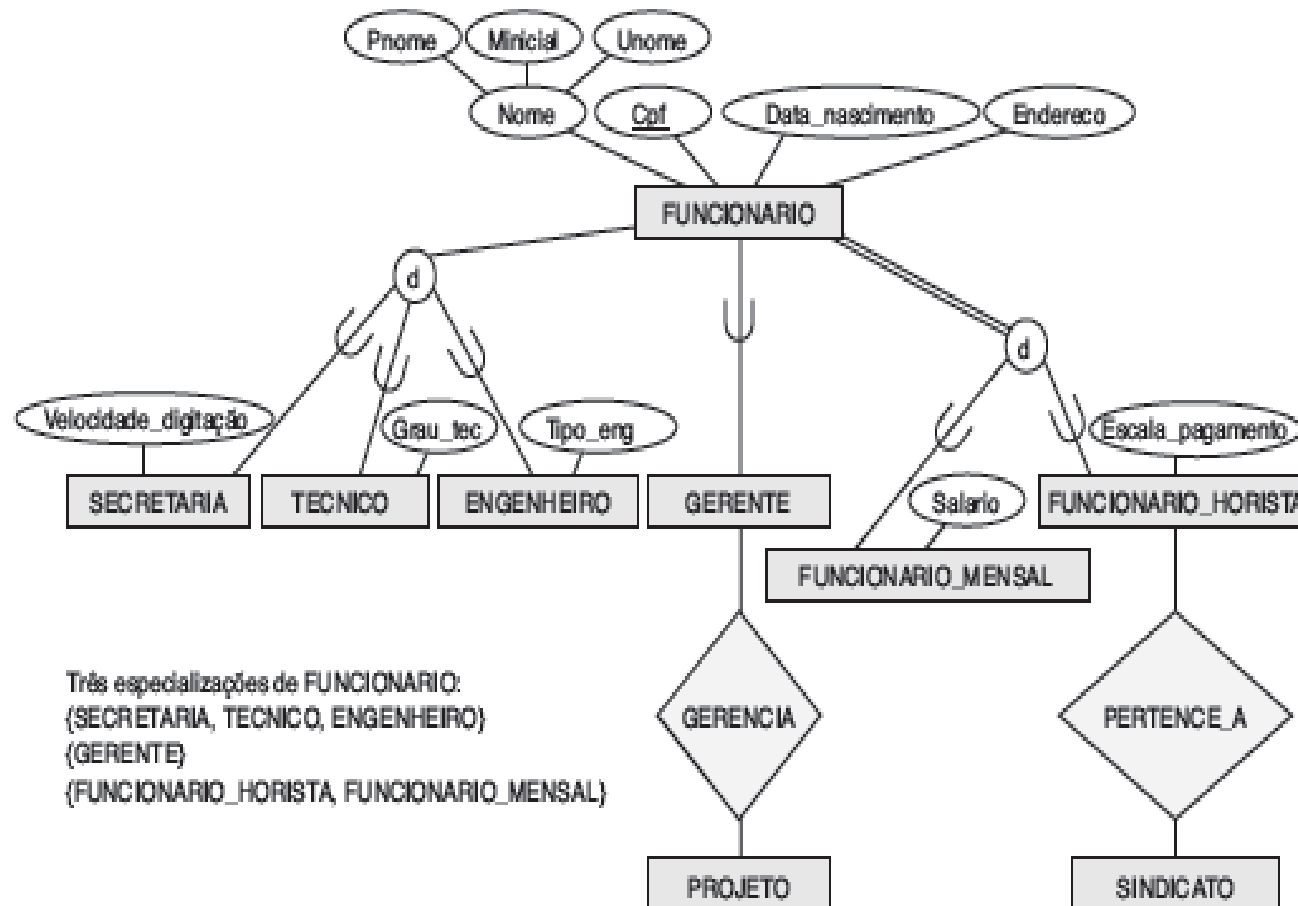
- Em muitos casos, um tipo entidade tem vários subconjuntos de entidades que são significativos para a aplicação.
 - ◆ Exemplo: as entidades de um tipo entidade **Funcionário** podem ser agrupadas em **Secretária**, **Engenheiro**, **Gerente**, **Funcionário_assalariado**, **Funcionário_horista** etc.
- Cada um dos subconjuntos é chamado de **subclasse** do tipo entidade Funcionário, e o tipo entidade Funcionário é chamado de **superclasse**.
- Uma entidade não pode existir no banco de dados como membro somente de uma subclasse, ela deve também ser membro da superclasse.

Herança

- Não é necessário que toda entidade em uma superclasse seja membro de alguma subclasse.
- Como uma entidade na subclasse representa a mesma entidade no mundo real da superclasse, então ela possui valores de seus atributos específicos bem como valores de seus atributos como um membro da superclasse.
 - ◆ Uma entidade que pertence a uma subclasse herda todos os atributos da superclasse.
- A entidade também herda todos os relacionamentos dos quais a superclasse participa.
 - ◆ E pode ter seus próprios relacionamentos como subclasse.

Especialização

- É o processo de definir um conjunto de subclasses de um tipo entidade (superclasse).



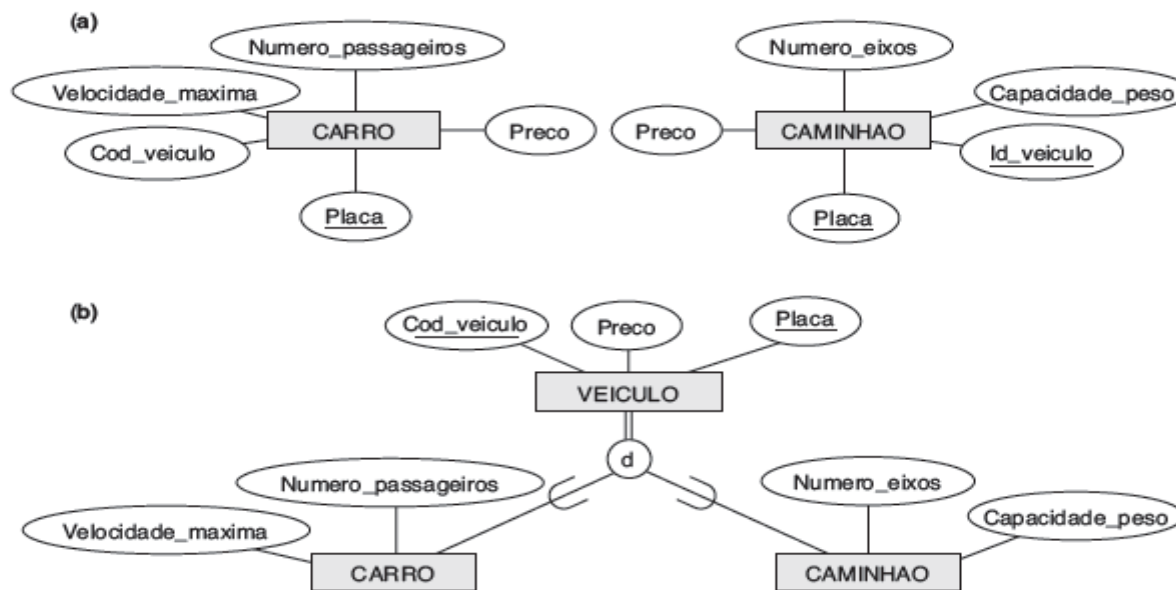
de cima
pra baixo
(top-down)

Especialização

- Detalhes do exemplo anterior:
 - ◆ O conjunto de subclasses {Secretária, Técnico e Engenheiro} é uma especialização de Funcionário que distingue as entidades baseado no tipo de trabalho.
 - ◆ O conjunto de subclasses {Mensalista e Horista} é uma especialização de Funcionário que distingue as entidades baseado na forma de pagamento.
 - ◆ Somente as entidades da subclasse Engenheiro possuem o atributo *tipo_eng* (tipo de engenheiro). Todas as entidades em {Secretário, Técnico, Engenheiro, Gerente, Assalariado e Horista} possuem os atributos *cpf* e *nome*.
 - ◆ Somente as entidades da subclasse Gerente podem participar do tipo relacionamento *gerencia*.

Generalização

- É o processo inverso da Especialização.
- As diferenças entre dois ou mais tipos entidades são suprimidas, suas características comuns são identificadas, e é feita uma generalização em uma única superclasse da qual os tipos originais são subclasses especiais.



de baixo
pra cima
(bottom-up)

Figura 8.3

Generalização. (a) Dois tipos de entidade, CARRO e CAMINHAO. (b) Generalizando CARRO e CAMINHAO na superclasse VEICULO.

Restrições sobre Especialização/Generalização

■ Restrição de Disjunção:

- ◆ **Disjunção:** uma entidade pode ser membro de, no máximo, uma das subclasses da especialização.
 - Representada pela letra **d** dentro do círculo da disjunção.
- ◆ **Sobreposição:** uma entidade pode ser membro de mais de uma subclasse da especialização.
 - Representada pela letra **o** dentro do círculo da disjunção.

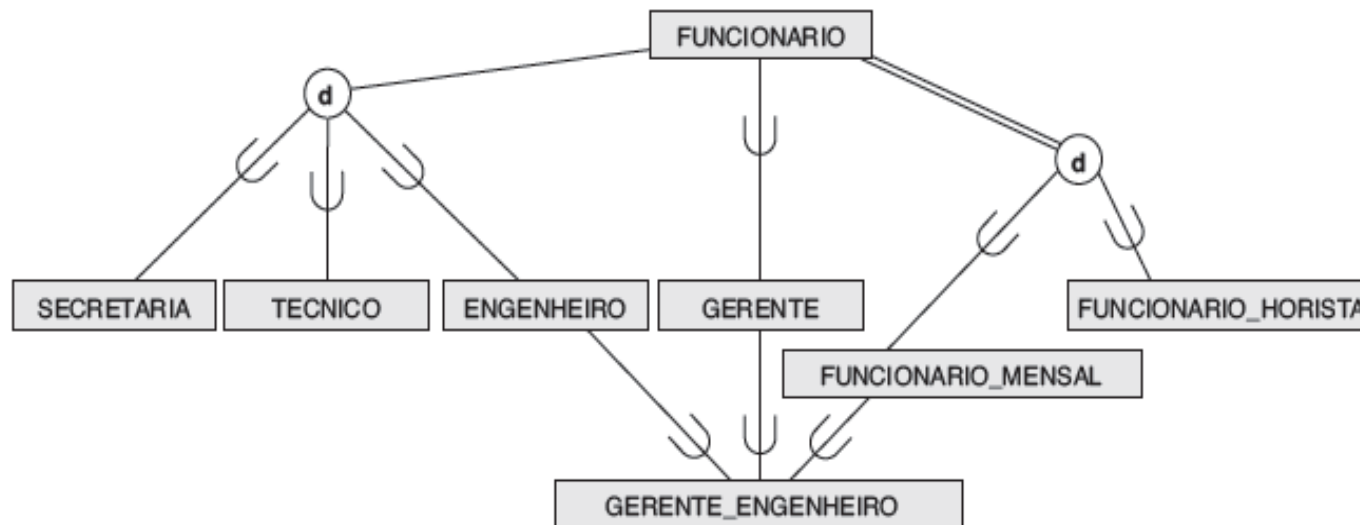
Restrições sobre Especialização/Generalização

■ Restrição de Completude:

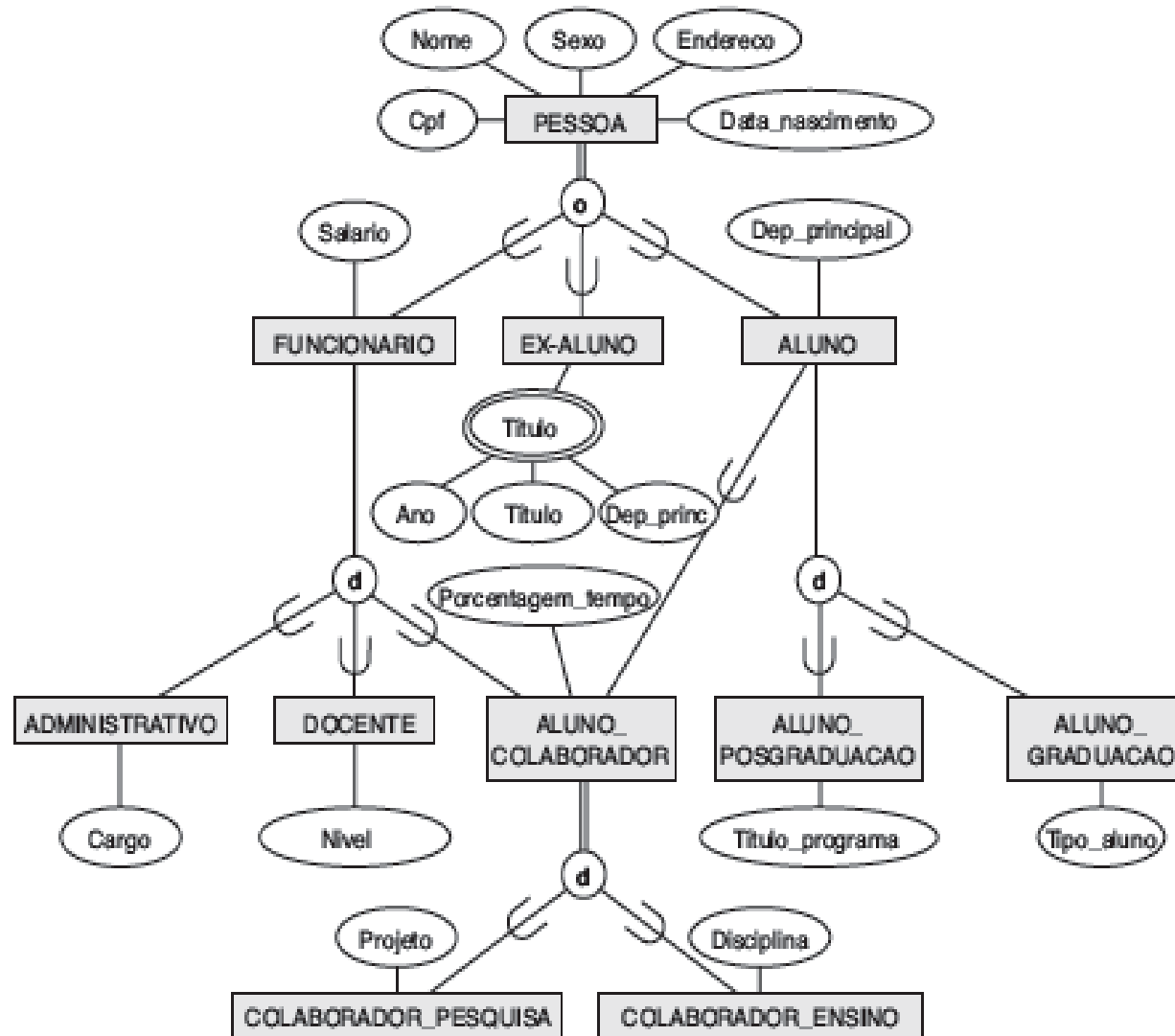
- ◆ **Total:** toda entidade na superclasse deve ser membro de pelo menos uma subclasse na especialização.
 - Representada pelas linhas duplas ligando a superclasse ao círculo.
- ◆ **Parcial:** uma entidade na superclasse não precisa ser membro de nenhuma subclasse na especialização.
 - Representada pela linha simples ligando a superclasse ao círculo.

Herança Múltipla (ou “*Lattice*”)

- Ocorre quando uma subclasse descende de mais de uma superclasse.



Herança Múltipla (ou “*Lattice*”)



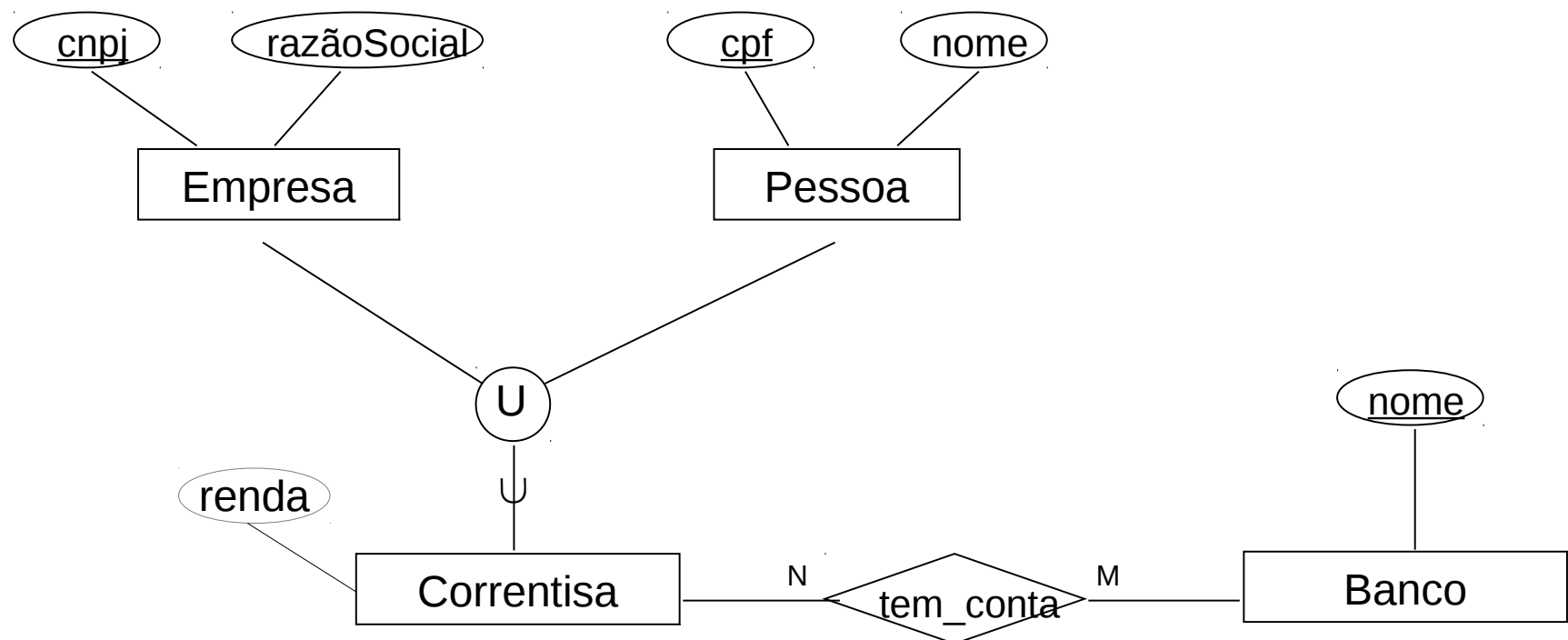
Tipo União (ou Categoria)

- Em algumas aplicações, é necessário modelar um único relacionamento superclasse/subclasse com mais de uma superclasse, onde as superclasses representam tipos entidades diferentes, e a subclasse representa uma coleção de objetos que é (um subconjunto de) a UNIÃO de tipos entidades distintos. Esta subclasse é chamada de **tipo união** (ou **categoria**).

Tipo União (ou Categoria)

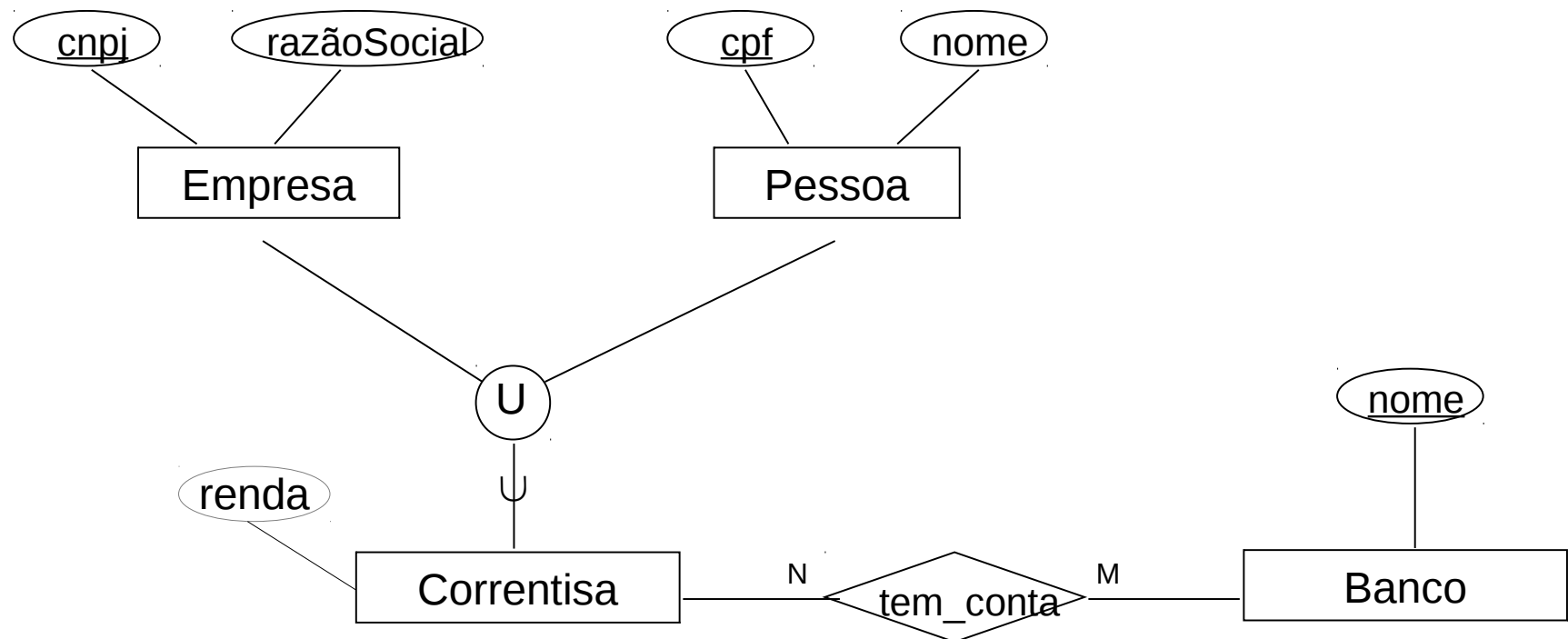
■ Exemplo:

Em um sistema bancário, deseja-se representar um tipo entidade Correntista como sendo o conjunto das entidades que possuem conta no banco. Um correntista pode ser uma pessoa física ou uma empresa.



Tipo União (ou Categoria)

- Uma categoria pode ser total ou parcial. No exemplo, a categoria Correntista é parcial (simbolizado pela linha simples ligando o Correntista), indicando que podem haver pessoas ou empresas que não são correntistas. Entretanto, todo correntista ou é uma pessoa ou é uma empresa.



Herança Múltipla versus Categoria

- Na herança múltipla, uma entidade na subclasse deve existir também em todas as superclasses.
 - ◆ Ex: uma entidade em “Eng_Gerente” existe também em “Engenheiro”, em “Gerente” e em “Assalariado”.
- Uma categoria é um subconjunto da união de suas superclasses. Assim, uma entidade na subclasse (categoria) deve existir somente em uma das superclasses.
 - ◆ Ex: uma entidade em “Correntista” existe em “Empresa” ou em “Pessoa”.
- Na herança múltipla, uma entidade na subclasse herda todos os atributos de todas as suas superclasses.
- Na categoria, uma entidade na subclasse herda os atributos somente da superclasse da qual ela pertence.

Bibliografia Básica

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de Bancos de Dados. Pearson Education, 6ª edição, 2011. ISBN-978-85-7936-085-5