

# Extensões do Modelo Entidade-Relacionamento

Banco de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

# Modelo Entidade Relacionamento Estendido

## ◆ Características

- introduz semântica adicional ao modelo ER
- utilizado na modelagem de aplicações mais complexas, tais como CAD/CAM, BD gráficos, BD geográficos

## ◆ Conceitos

- subclasse, superclasse, hierarquia de herança
- generalização, especialização, e restrições
- agregação

# Subclasse/Superclasse

## ◆ Subclasse

- subagrupamento das entidades de um tipo-entidade

## ◆ Exemplo

- superclasse: tipo-entidade **empregado**
- subclasses: **secretário, engenheiro, técnico**

cada entidade que é membro de qualquer uma das subclasses também **é um** empregado



# Herança

- ◆ de atributos
  - atributos da superclasse são herdados pelas subclasses
- ◆ de relacionamentos
  - instâncias de relacionamento da superclasse são herdados pelas entidades das subclasses
- ◆ Observação
  - qualquer entidade membro de uma subclasse deve ser também membro da superclasse
  - qualquer entidade membro da superclasse pode ser opcionalmente incluída como membro de qualquer número de subclasses

# Generalização/Especialização

## ◆ Especialização

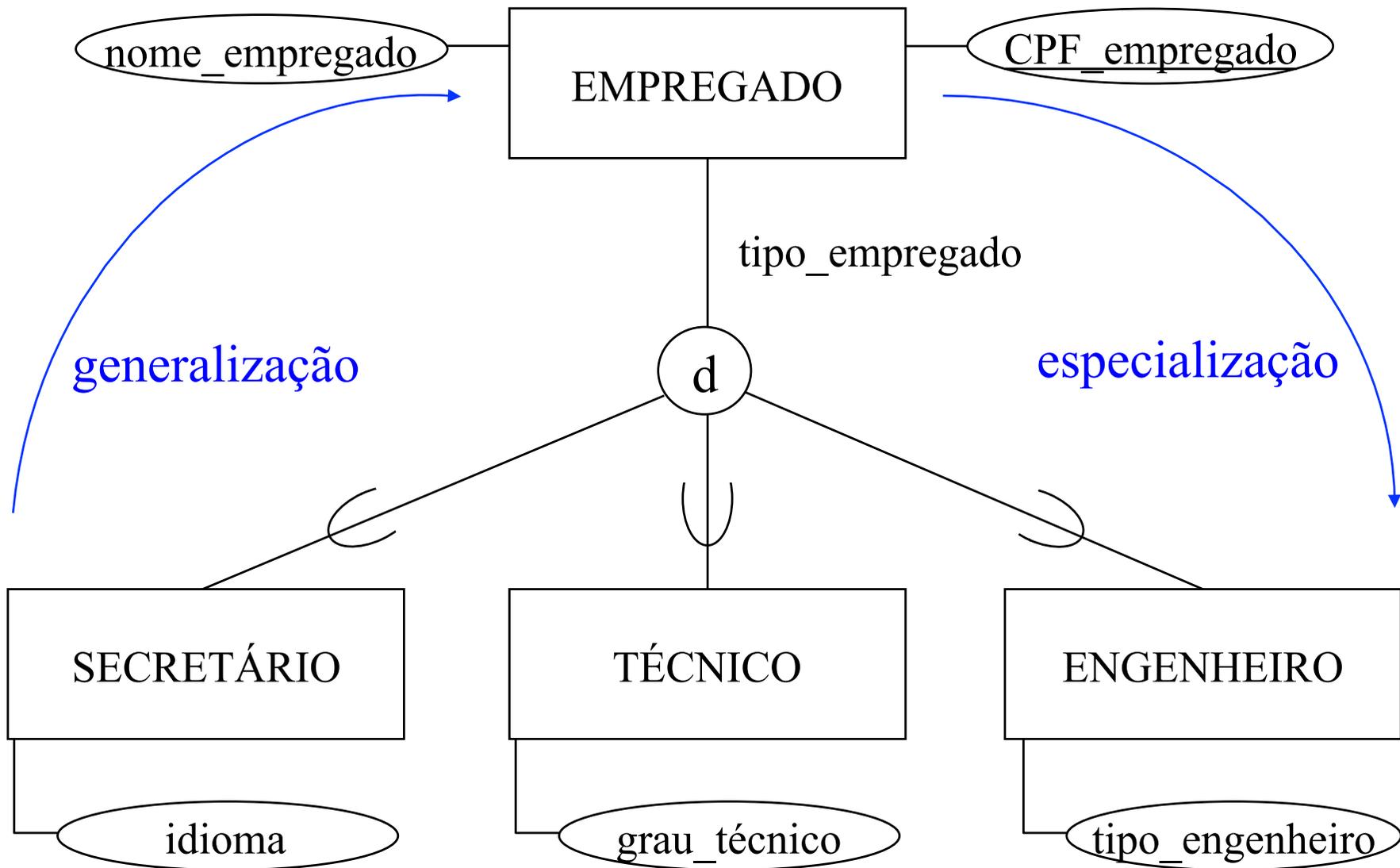
- resultado da separação de um tipo-entidade de nível mais alto (superclasse), formando vários tipos-entidade de nível mais baixo (subclasse)
- passos:
  - ◆ define-se um conjunto de subclasses de um tipo-entidade
  - ◆ associa-se atributos adicionais específicos às subclasses
  - ◆ estabelece-se tipos-relacionamento adicionais específicos às subclasses, caso necessário

# Generalização/Especialização

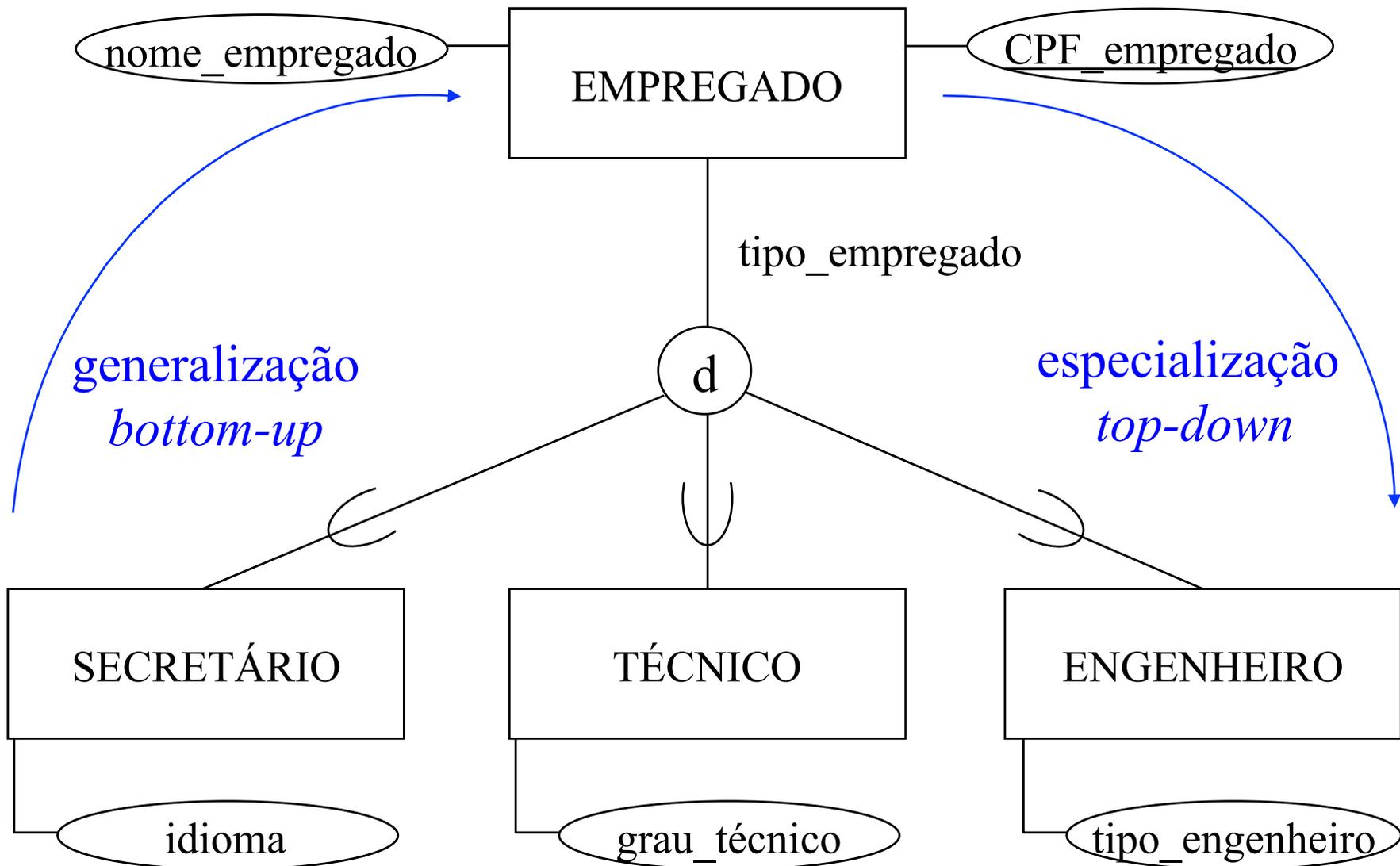
## ◆ Generalização

- resultado da união de dois ou mais tipos-entidade de nível mais baixo (subclasse), produzindo um tipo-entidade de nível mais alto (superclasse)
- é uma abstração de um conjunto de entidades
- passos:
  - ◆ suprime-se as diferenças entre os tipos-entidade
  - ◆ identifica-se os atributos em comum
  - ◆ generaliza-os em uma superclasse

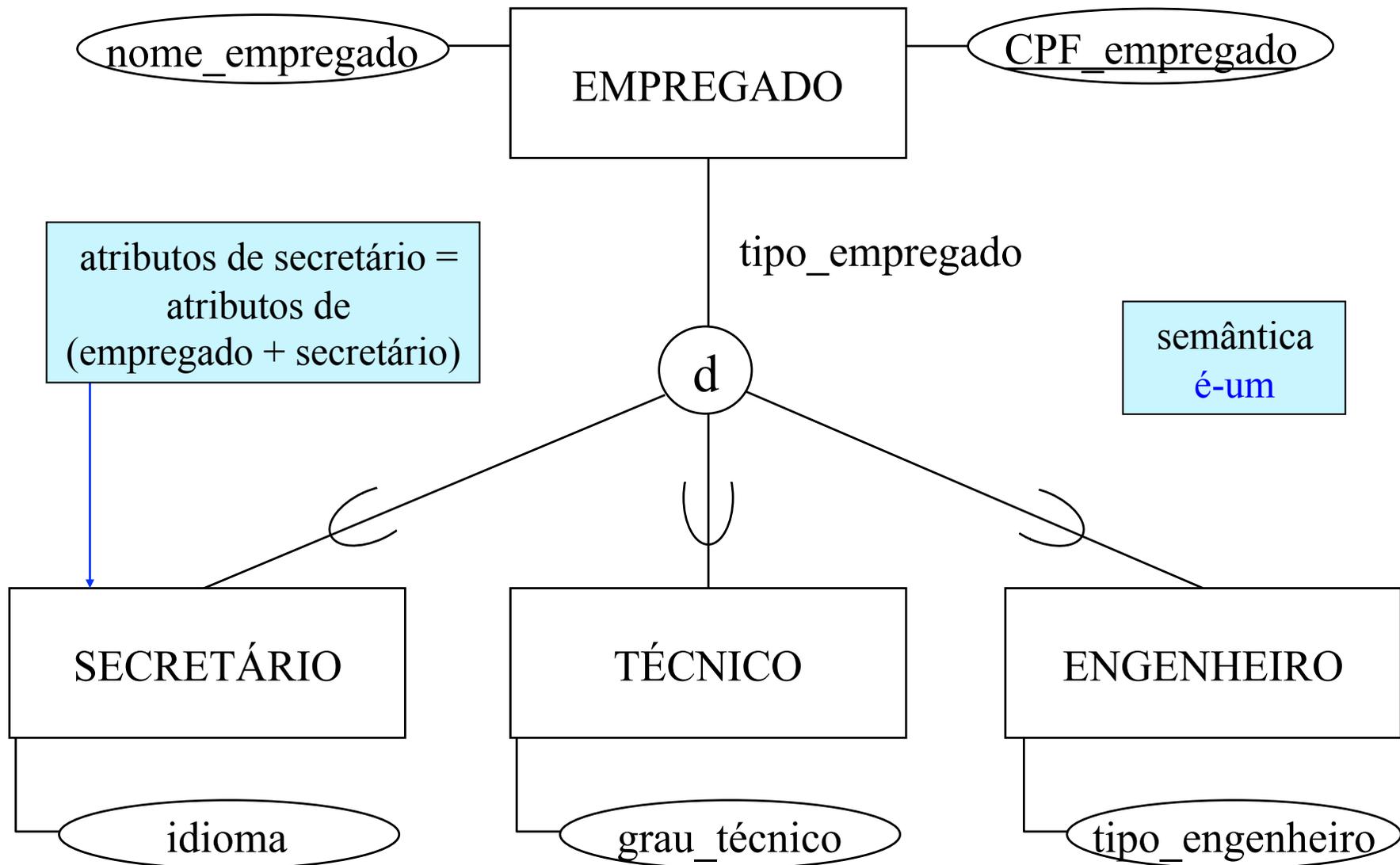
# Representação



# Representação



# Representação



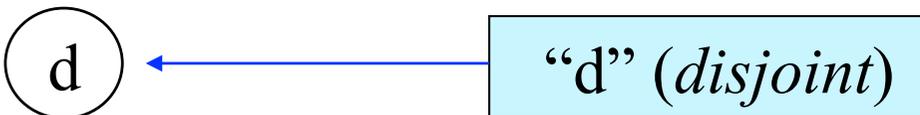
# Restrições

- ◆ Especialização definida pelo atributo
  - as subclasses que participam da hierarquia são determinadas por uma condição baseada em algum atributo da superclasse
  - exemplo: `tipo_empregado`
- ◆ Denominações
  - subclasses definidas pelo predicado
  - subclasses definidas pela condição

# Restrição de Disjunção

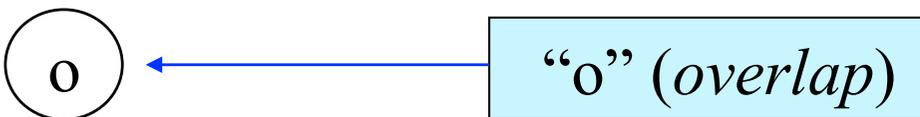
## ◆ Subclasses mutuamente exclusivas

- uma entidade de uma superclasse deve ser membro, quando muito, de apenas uma única subclasse

– representação:  “d” (*disjoint*)

## ◆ Subclasses que se sobrepõem

- uma entidade de uma superclasse pode ser membro de mais do que uma subclasse

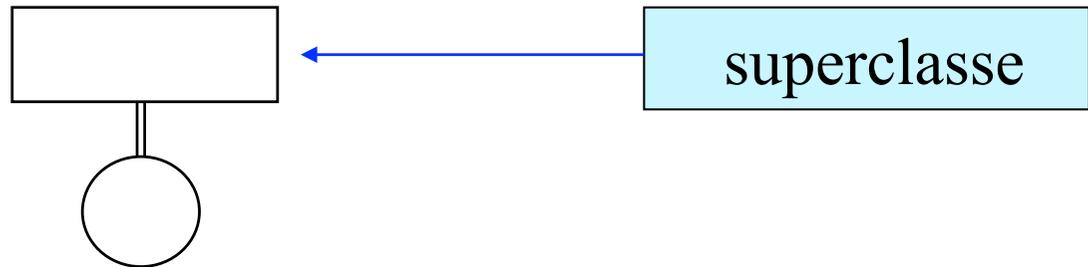
– representação:  “o” (*overlap*)

# Restrição de Completude

## ◆ Total

- cada entidade de uma superclasse deve ser membro de alguma subclasse na especialização

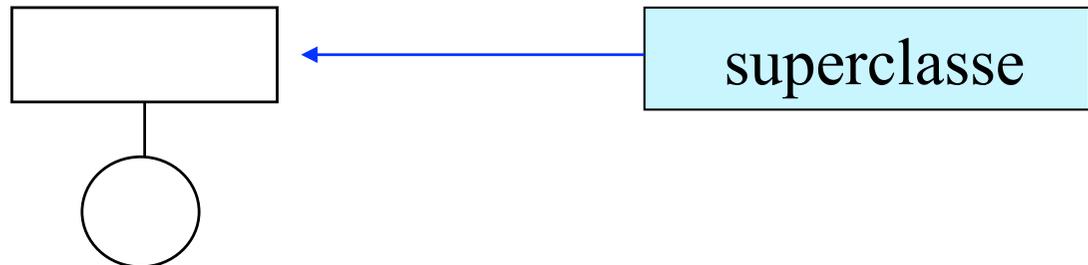
– representação:



## ◆ Parcial

- uma entidade de uma superclasse pode não pertencer a qualquer uma das subclasses

– representação:

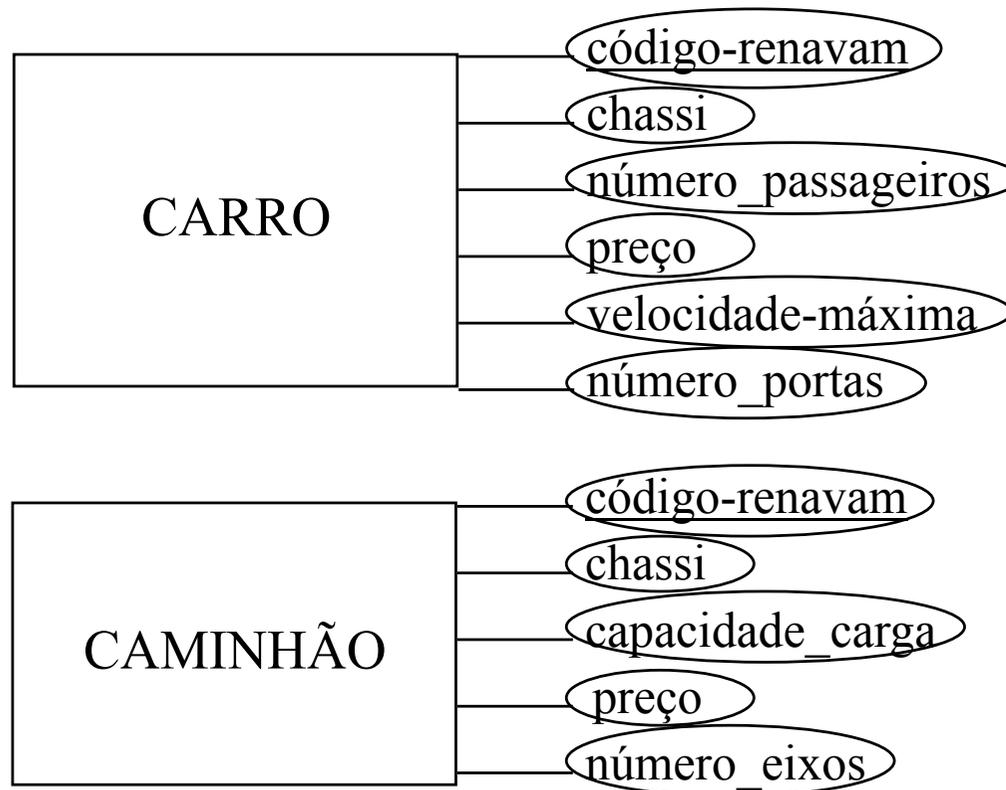


# Observações

- ◆ Restrições de disjunção e de completude são independentes
  - possibilidades de hierarquias
    - ◆ total disjunta
    - ◆ parcial disjunta
    - ◆ total com sobreposição
    - ◆ parcial com sobreposição

# Exercício

- ◆ Modele uma hierarquia de generalização/especialização para os tipos-entidade *carro* e *caminhão*. Defina as restrições de disjunção e de completude.



# Observações

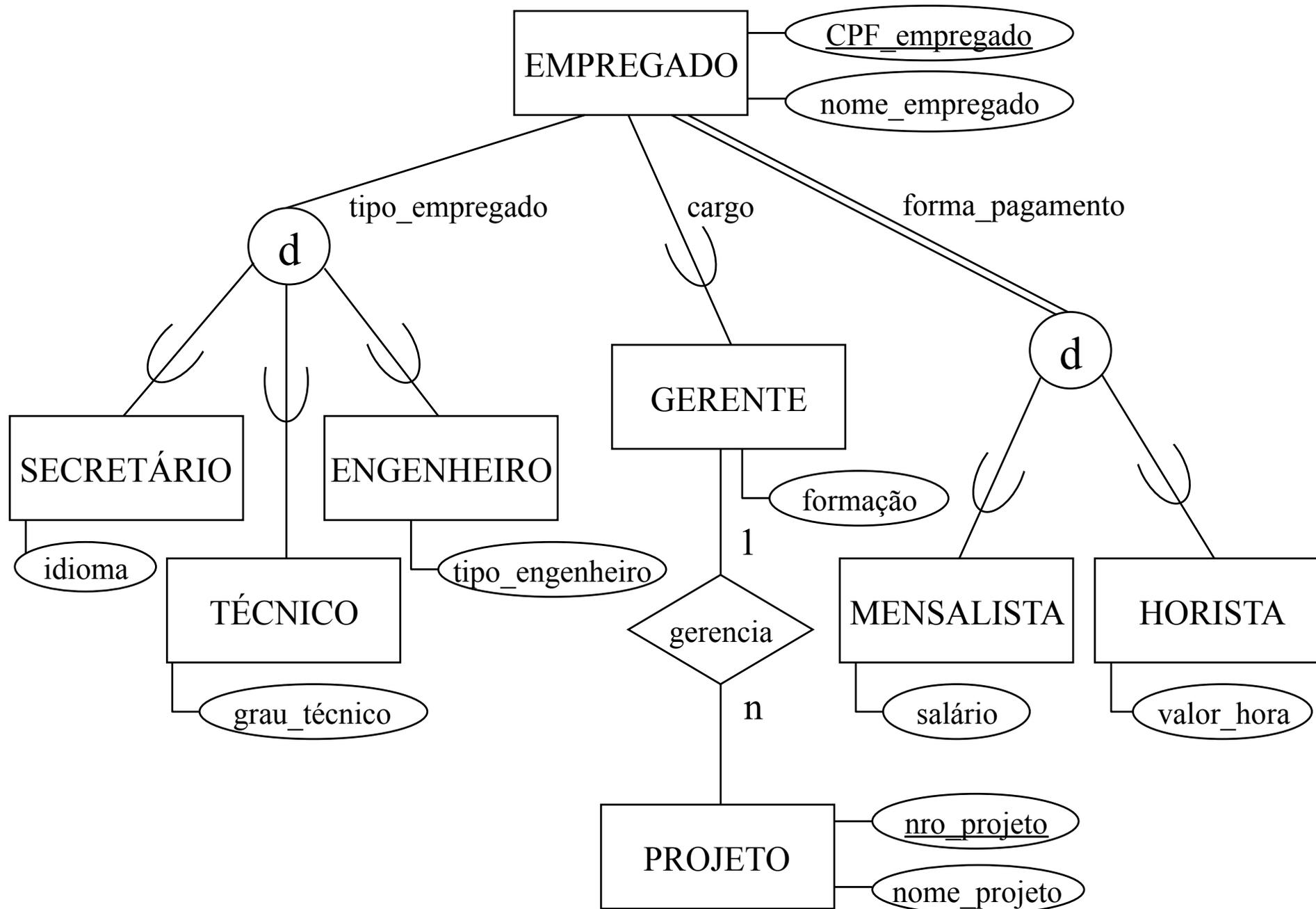
## ◆ Regras de inserção e remoção

- se uma entidade de uma superclasse for removida então ela deve ser automaticamente removida de todas as subclasses a que pertence
- se uma entidade for inserida em uma superclasse então ela deve ser necessariamente inserida em todas as subclasses definidas pelo atributo, quando este for satisfeito

# Observações

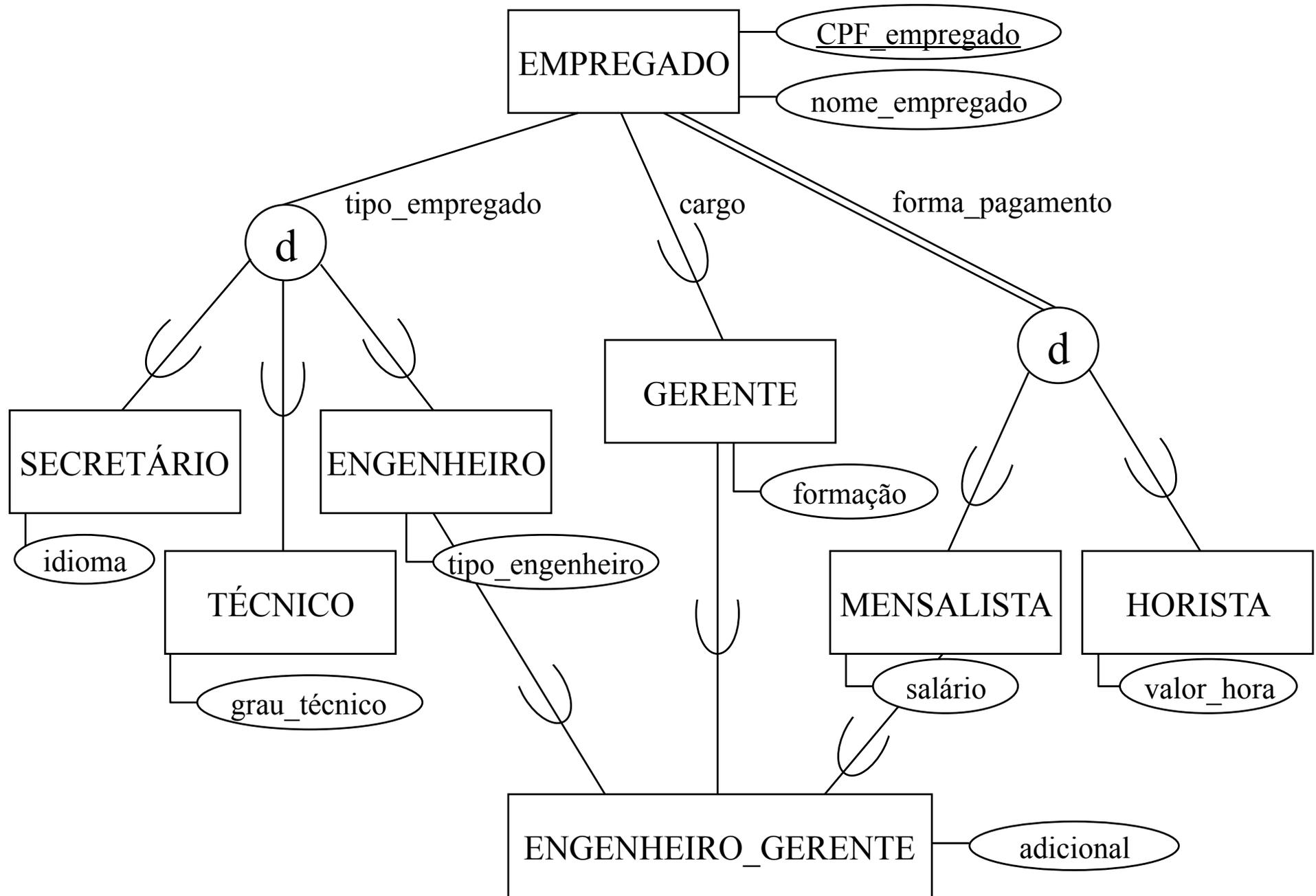
## ◆ Regras de inserção e remoção

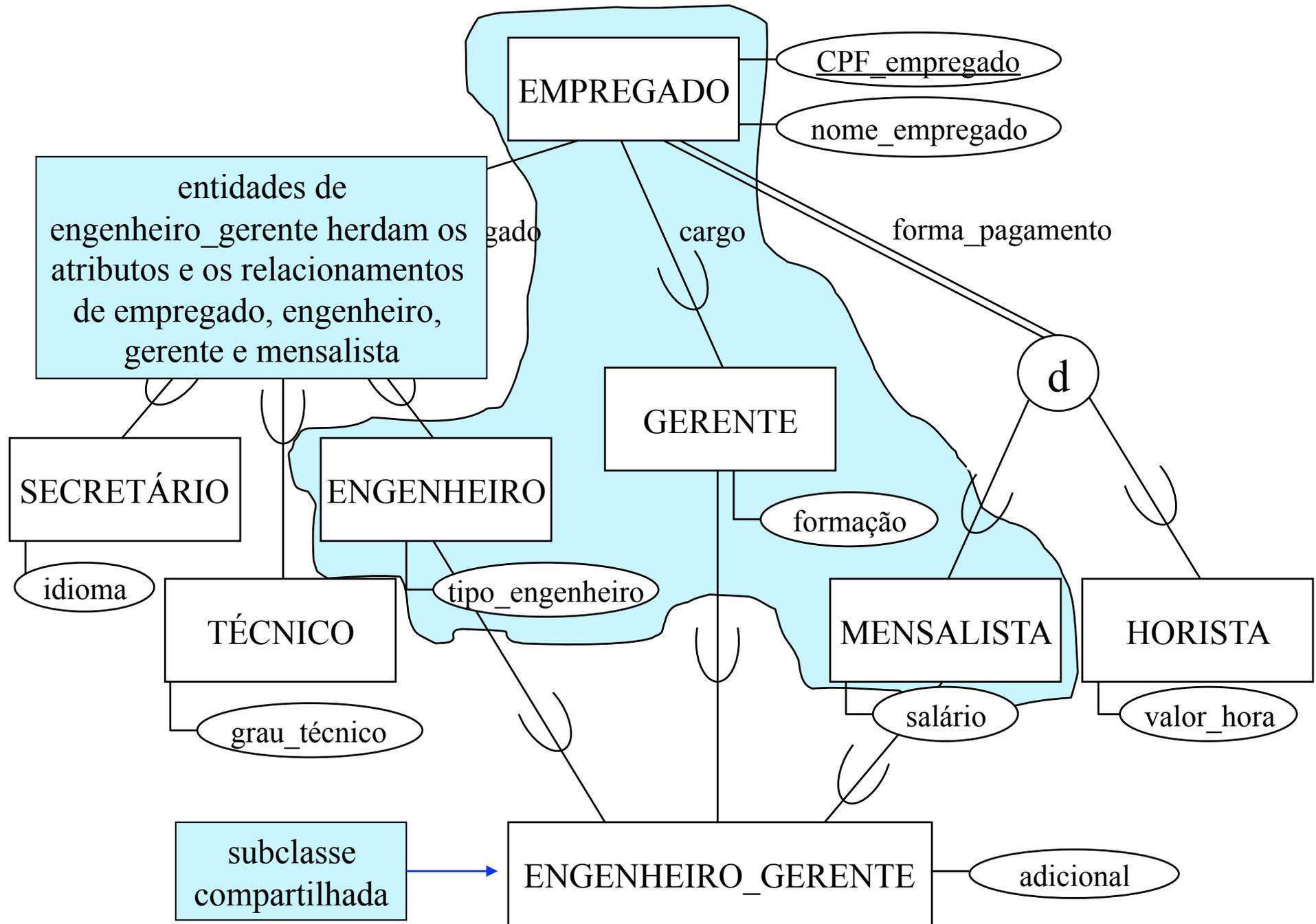
- se uma entidade for inserida em uma superclasse com especialização total então ela deve ser necessariamente inserida em pelo menos uma das subclasses da especialização
- ...



# Generalização/Especialização

- ◆ Uma subclasse pode possuir outras subclasses especificadas a partir dela
- ◆ Herança simples
  - cada subclasse participa como subclasse em **apenas um** relacionamento superclasse/subclasse
- ◆ Herança múltipla
  - cada subclasse pode participar como uma subclasse em **mais do que um** relacionamento superclasse/subclasse





# Herança Múltipla

## ◆ Regra

- se um mesmo atributo ou relacionamento for herdado mais do que uma vez por diferentes relacionamentos superclasse/subclasse então o atributo ou o relacionamento deve ser incluído apenas uma vez na subclasse

## ◆ Restrições

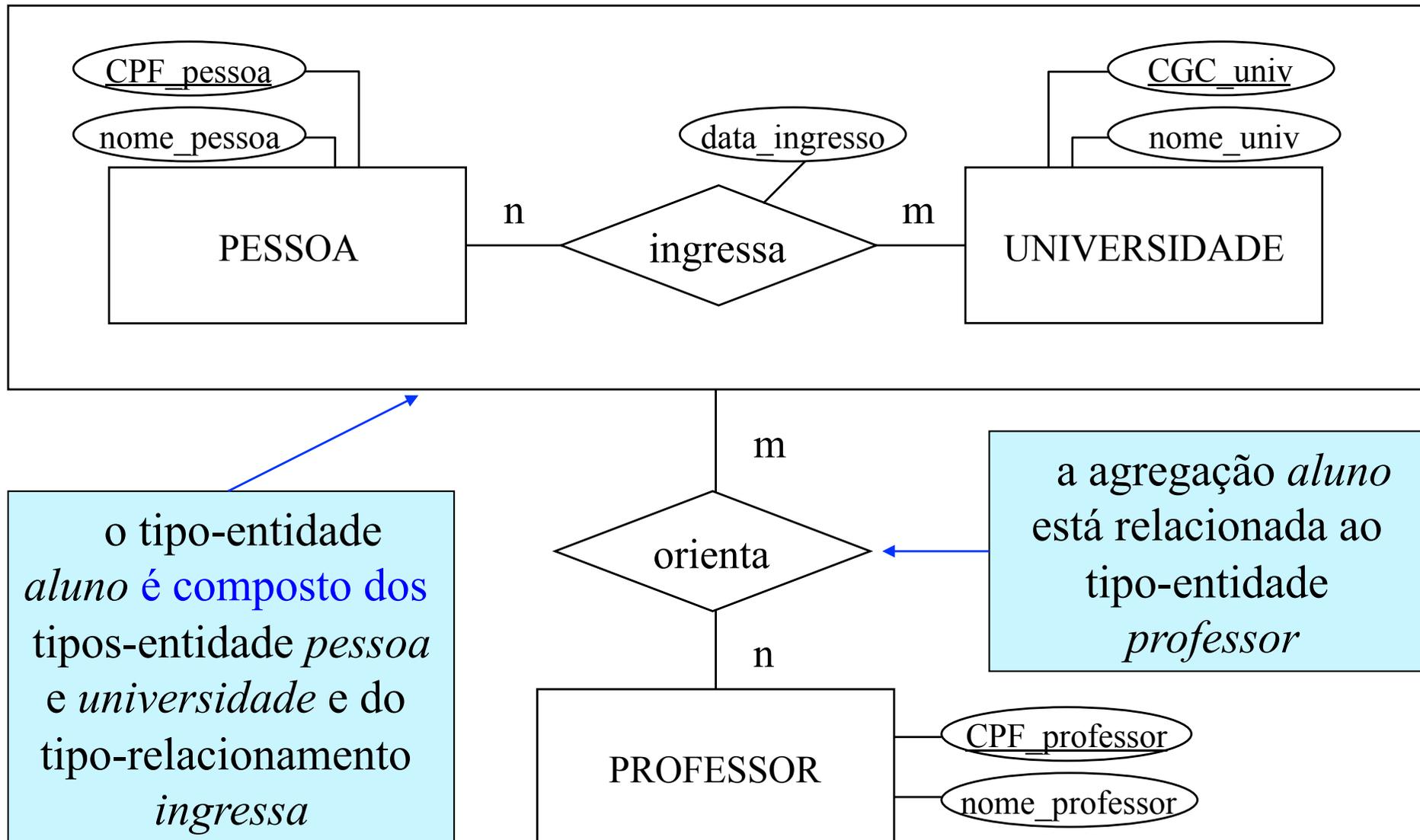
- alguns mecanismos de herança
  - ◆ não permitem herança múltipla
  - ◆ não permitem a especificação conjunta de herança múltipla e de diferentes predicados

# Agregação

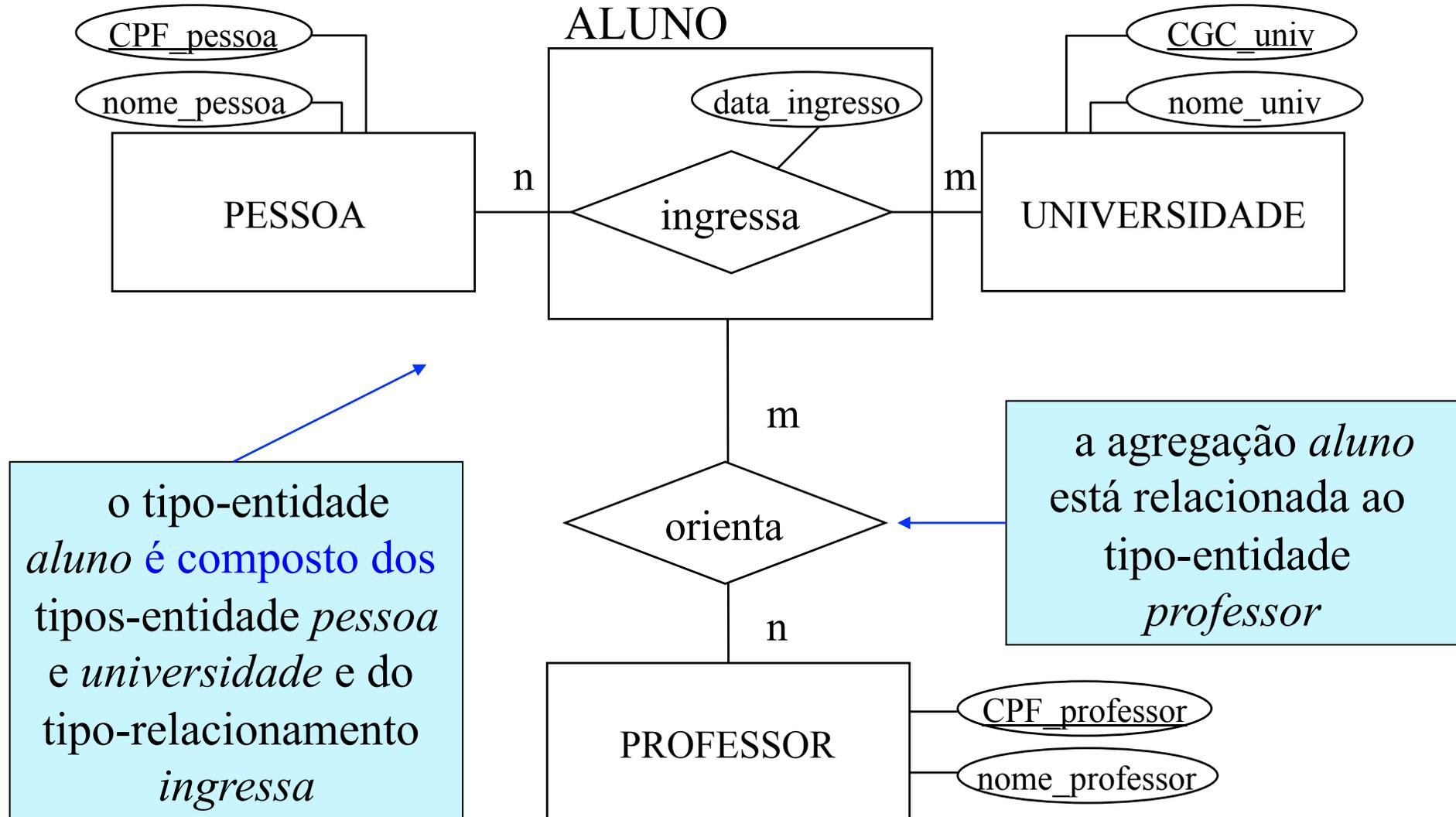
- ◆ Tipos-entidades agregados são representados como tipos-entidades comuns
- ◆ Pode englobar
  - dois tipos-entidades e um tipo-relacionamento
- ◆ Dados vistos em um nível mais baixo
  - atributos dos tipos-relacionamentos
  - chaves primárias dos tipos-entidades

# Representação

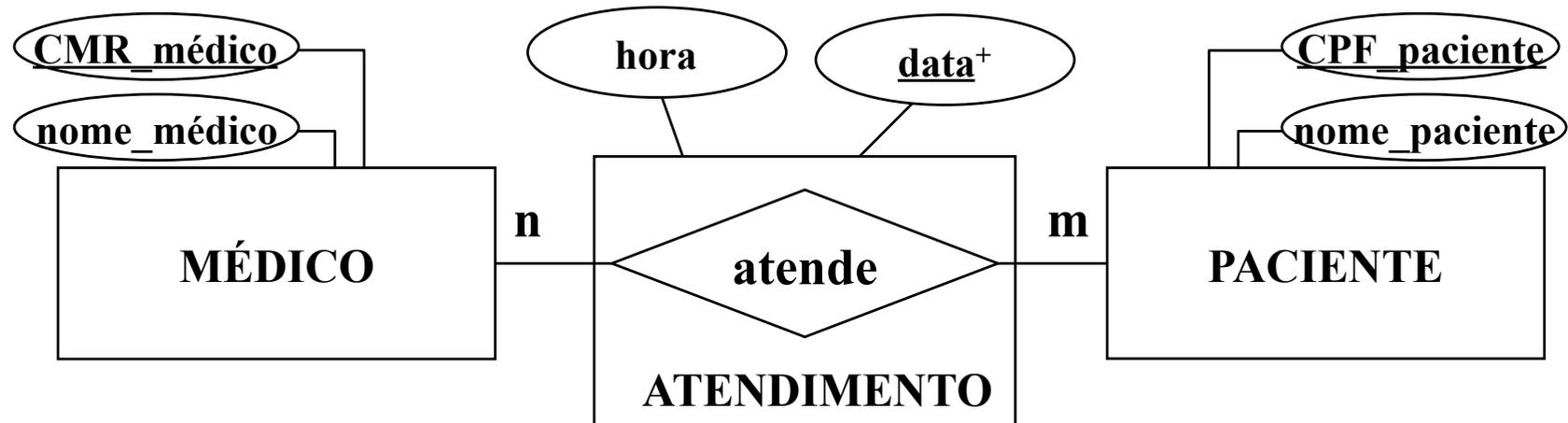
ALUNO



# Representação Simplificada

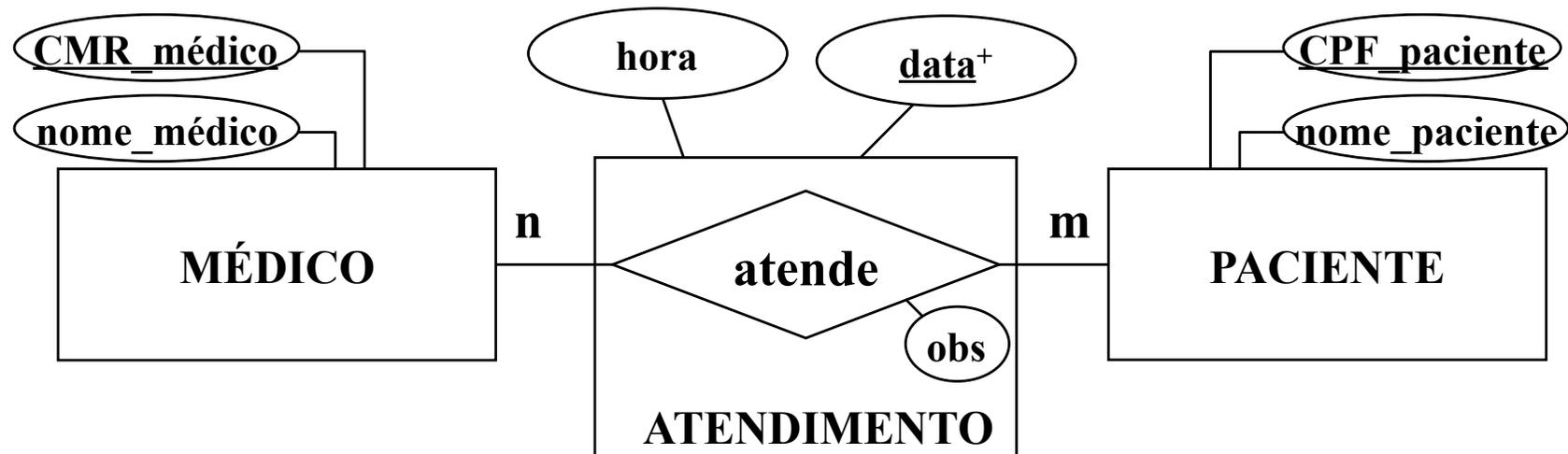


# Agregação



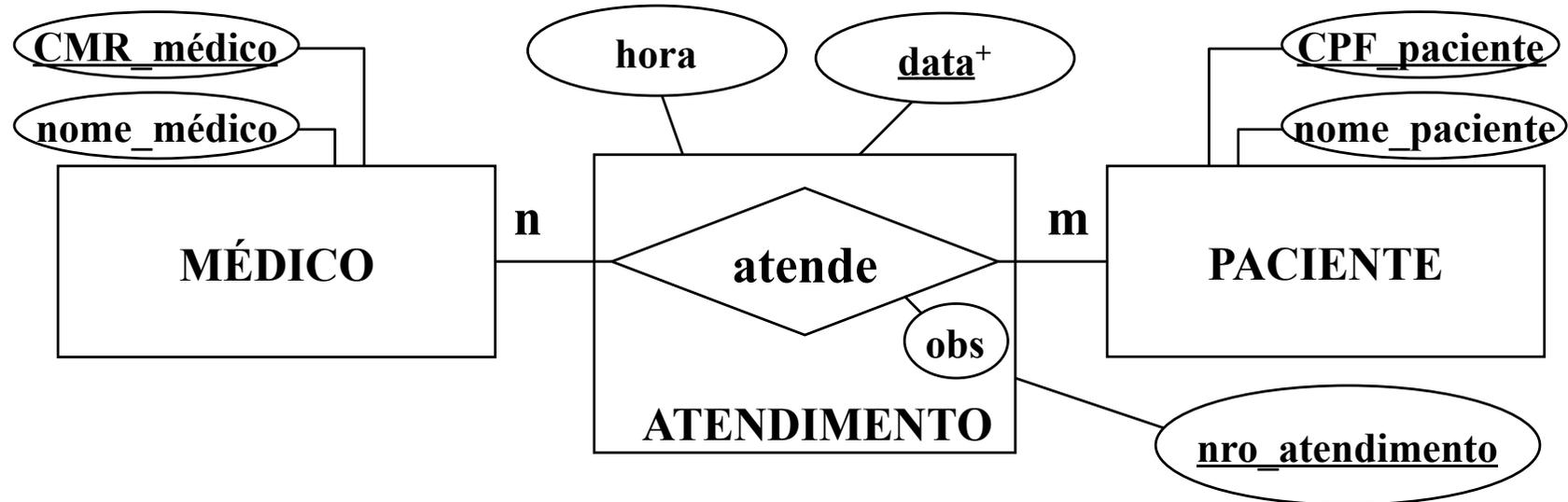
**Enfocando a possibilidade do mesmo médico atender o mesmo paciente em diferentes datas**

# Agregação



**Adicionando atributos também ao tipo-relacionamento**

# Agregação



**Adicionando uma chave primária à agregação**

# Projeto Lógico de BD

- ◆ Classificar tipos-entidades e atributos
  - tipos-entidade possuem informações descritivas, atributos não
  - atributos devem ser mantidos de forma atômica
  - atributos devem ser relacionados às entidades que eles descrevem
- ◆ Identificar chaves primárias

# Projeto Lógico de BD

- ◆ Identificar tipos-relacionamentos e seus atributos
  - determinar o grau dos tipos-relacionamentos
    - ◆ definir tipos-relacionamento ternários cuidadosamente
  - identificar as restrições que se aplicam sobre cada tipo-relacionamento
    - ◆ cardinalidade
    - ◆ participação
- ◆ Identificar tipo-entidade forte e tipo-entidade fraca

# Projeto Lógico de BD

- ◆ Modelar hierarquias de generalização
  - identificar atributos e relacionamentos comuns
  - determinar as restrições de disjunção e de completude
- ◆ Modelar agregações

*“Podemos notar que um projetista de BD necessita de um bom conhecimento do minimundo que está sendo modelado para que possa tomar essas decisões!”*

# Exercícios

- ◆ Fazer os exercícios da lista de exercícios “Modelo Entidade Relacionamento”
  - 3. Companhia
  - 4. Agência de Turismo
  - 5. Controle Acadêmico II
  - 6. Variações