

# Aula 7 – SBD

## ER para Relacional

Profa. Elaine Faria

UFU - 2018

# Introdução

- Modelo ER
  - Conveniente para representar um projeto de BD inicial de alto nível
- Modelo Relacional
  - Pode ser usado para suportar a implementação de aplicações
  - É necessário que exista um SGBD que se apóie no modelo relacional: um SGBDR
- Dado um DER é adotada uma estratégia padrão para gerar um esquema de BD relacional

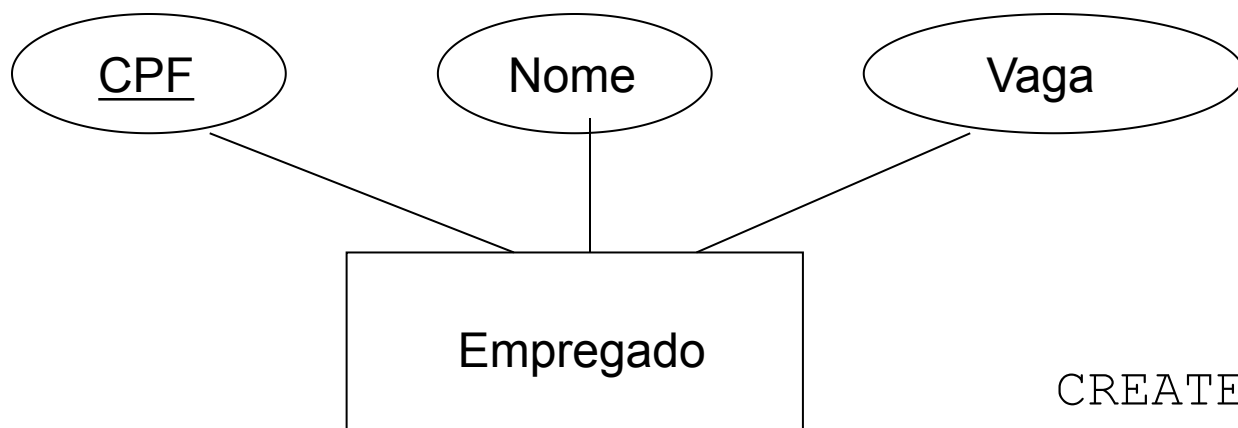
# Introdução

Como transformar um diagrama ER para um esquema de Banco de Dados Relacional?

# Tipo de Entidade para Tabelas

- Um tipo de entidade é mapeado em uma tabela (relação)
  - Cada atributo do tipo de entidade torna-se um atributo da tabela
  - O domínio de cada atributo e a chave primária são conhecidos

# Tipo de Entidade para Tabelas



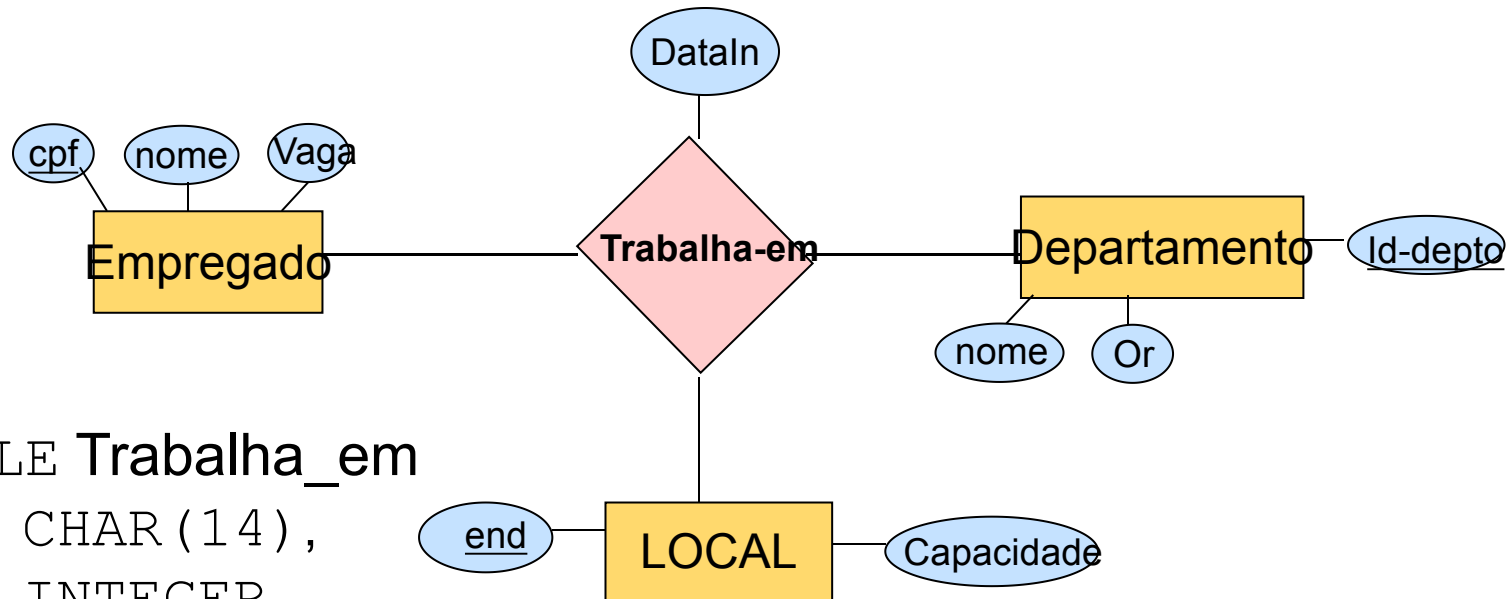
```
CREATE TABLE Empregado (  
    CPF CHAR(14),  
    Nome CHAR(30),  
    Vaga INTEGER,  
    PRIMARY KEY (CPF) )
```

CPF	Nome	Vaga
111.111.111-11	João	48
222.222.222.22	Maria	33
333.333.333-33	Pedro	22

# Tipo de Relacionamento N:M para Tabelas

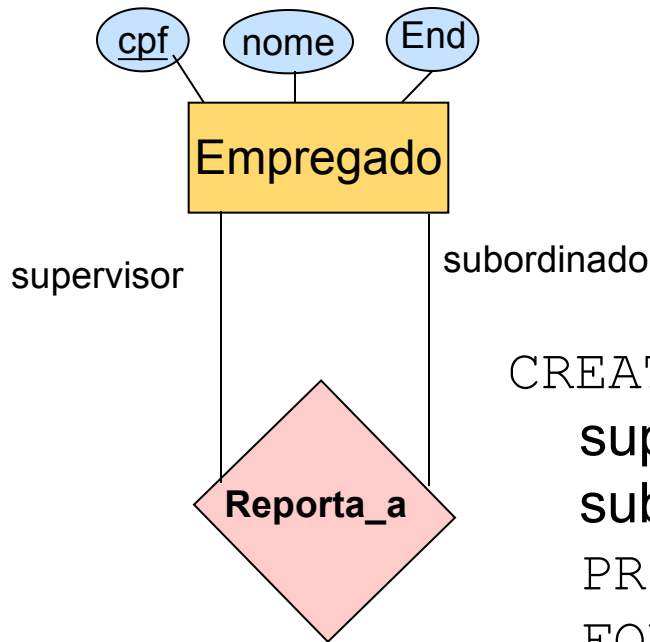
- Tipo de relacionamento → tabela
- Os atributos incluem
  - Atributos chave primária de cada tipo de entidade participantes → chave estrangeira
  - Atributos descritivos do conjunto de relacionamentos
- O conjunto de atributos não descritivos é uma chave candidata para a tabela

# Tipo de Relacionamento N:M para Tabelas



```
CREATE TABLE Trabalha_em
  cpf          CHAR(14),
  id_depto    INTEGER,
  end         CHAR(20),
  DataIn      DATE,
  PRIMARY KEY (cpf, id_depto, end),
  FOREIGN KEY (cpf) REFERENCES Empregado,
  FOREIGN KEY (id_depto) REFERENCES Departamento,
  FOREIGN KEY (end) REFERENCES Local )
```

# Tipo de Relacionamento N:M para Tabelas



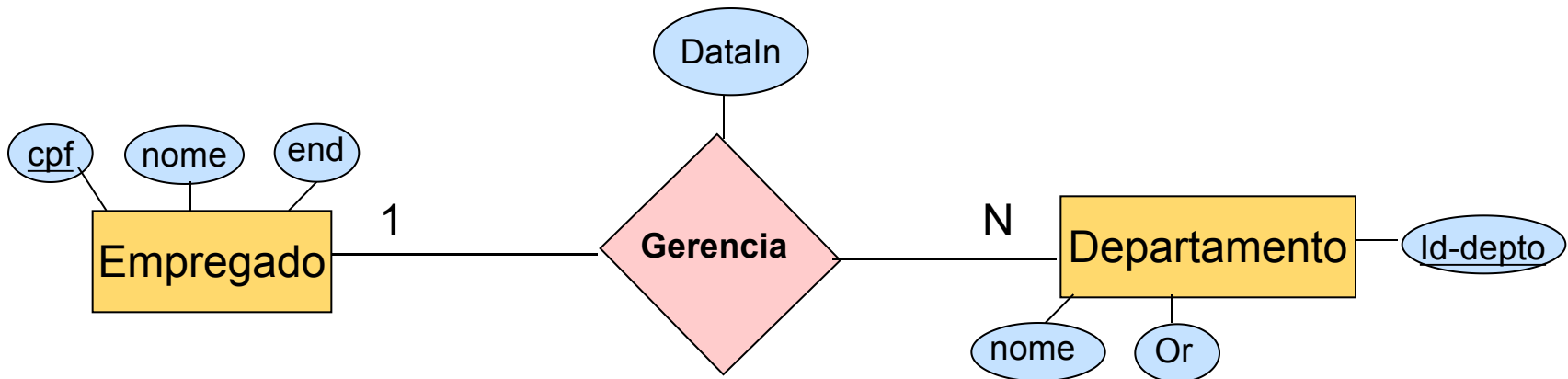
```
CREATE TABLE Report_a (  
  supervisor_cpf      CHAR(14),  
  subordinado_cpf    CHAR(14),  
  PRIMARY KEY (supervisor_cpf, subordinado_cpf),  
  FOREIGN KEY (supervisor_cpf) REFERENCES  
              Empregado (cpf),  
  FOREIGN KEY (subordinado_cpf) REFERENCES  
              Empregado (cpf) )
```



# Tipo de Relacionamento 1:N para Tabelas

- Primeira estratégia
  - Identificar a tabela S que representa o lado N do tipo de relacionamento
  - Incluir como chave estrangeira em S, a chave primária da tabela T que representa o lado 1 do tipo de relacionamento
- Inconveniente
  - Ex: espaço desperdiçado
    - Campos preenchidos com o valor *null*

# Tipo de Relacionamento 1:N para Tabelas

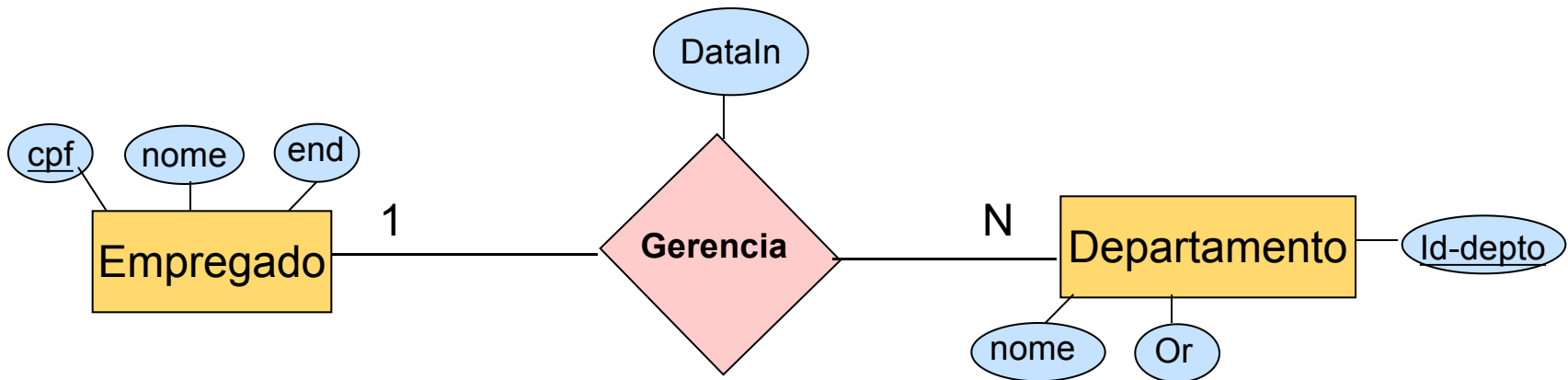


```
CREATE TABLE Departamento(  
  id_depto    INTEGER,  
  nome       CHAR(20),  
  Or         REAL,  
  cpf        CHAR(14),  
  DataIn     DATE,  
  PRIMARY KEY (id_depto),  
  FOREIGN KEY (cpf) REFERENCES Empregado )
```

# Tipo de Relacionamento 1:N para Tabelas

- Segunda estratégia
  - Criar uma tabela separada R cujos atributos são chaves primárias de S e T
    - Chaves estrangeiras
  - Definir a chave primária de R igual à chave primárias de S

# Tipo de Relacionamento 1:N para Tabelas



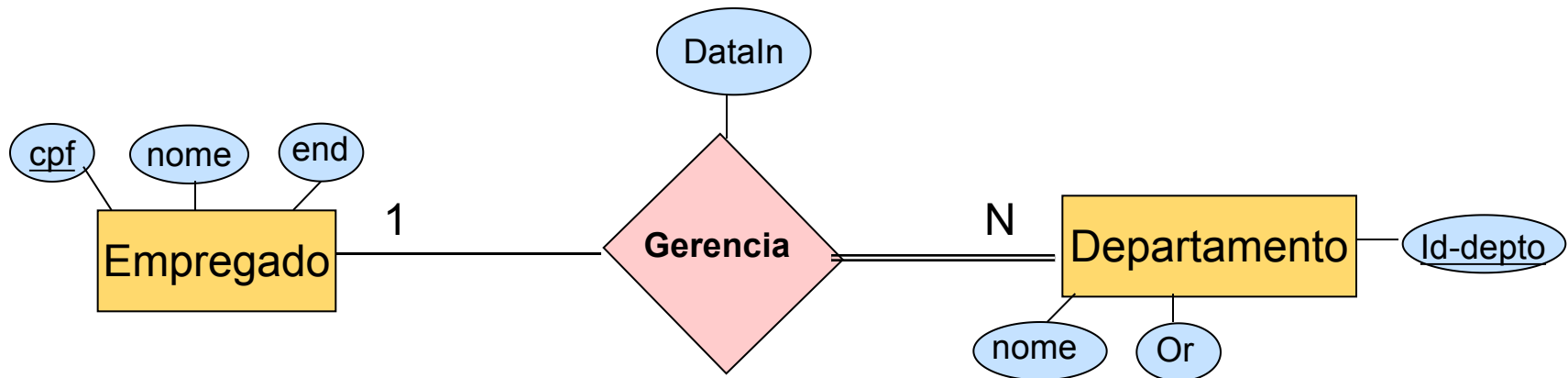
```
CREATE TABLE Gerencia(  
  cpf          CHAR (14) ,  
  id_depto    INTEGER,  
  DataIn      DATE,  
  PRIMARY KEY (id_depto),  
  FOREIGN KEY (cpf) REFERENCES Empregado,  
  FOREIGN KEY (id_depto) REFERENCES Departamento )
```

# Tipo de Relacionamento 1:N com Restrição de Participação Total

- Restrição “*Todo departamento deve ter um gerente*” → usar 1ª estratégia
  - não pode ser capturada usando-se a 2ª estratégia

Relacionamentos 1:N com restrição de participação total → 1ª estratégia

# Tipo de Relacionamento 1:N com Restrição de Participação Total



```
CREATE TABLE Departamento(  
  id_depto    INTEGER,  
  nome        CHAR(20),  
  Or          REAL,  
  cpf         CHAR(14)  not null,  
  DataIn     DATE  not null,  
  PRIMARY KEY (id_depto),  
  FOREIGN KEY (cpf) REFERENCES Empregado )
```

# Restrição de Participação Total

- Como garantir a participação Total de Departamentos em Trabalha\_em?

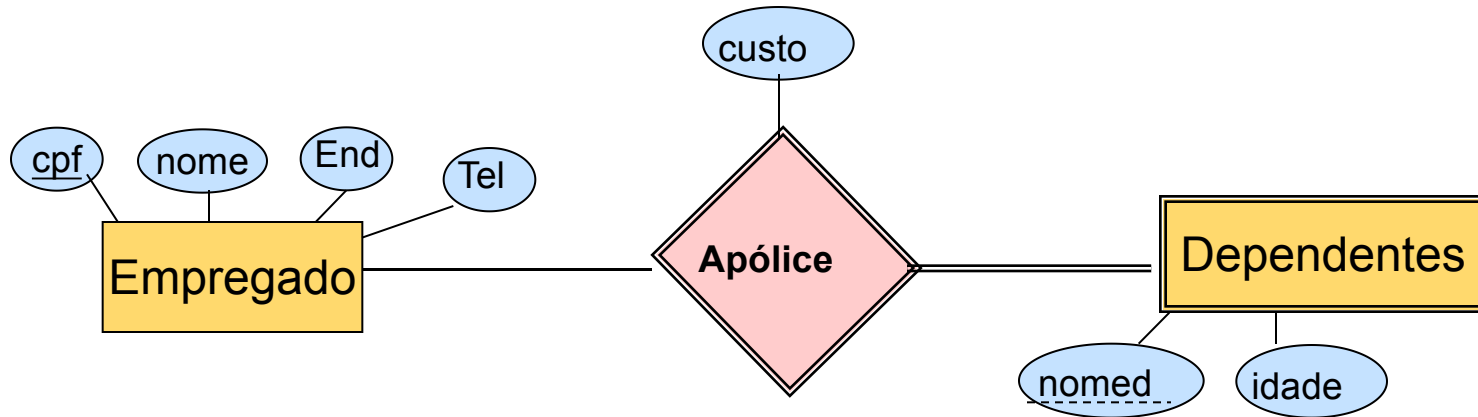
Uso de Asserções em SQL (programas específicos)

# Tipo de Entidade Fraca

- Um tipo de entidade fraca sempre
  - Participa de um relacionamento binário, 1:N e tem participação total
- Passos
  - Entidade fraca F com tipo de entidade proprietária E → tabela R
    - Todos os atributos simples de F
    - Chave estrangeira de R: atributos de chave primária de E
    - Chave primária de R: chave primária de E e chave parcial de F



# Tipo de Entidade Fraca



```
CREATE TABLE ApoliceDependente (  
  nomed    CHAR(20),  
  idade    INTEGER,  
  custo    REAL,  
  cpf      CHAR(14) NOT NULL,  
  PRIMARY KEY (nomed, cpf),  
  FOREIGN KEY (cpf) REFERENCES Empregado  
    ON DELETE CASCADE)
```

# Hierarquia de classe

Superclasse  $C$   $\{ch, a_1, \dots, a_n\}$  e  
 $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$

- Múltiplas Tabelas
  - Criar uma tabela  $L$  para  $C$
  - Criar uma tabela  $L_i$  para cada subclasse  $S_i$  com  $Atr(L_i) = \{ch\} \cup \{\text{atributos de } S_i\}$

Ideal para qualquer especialização: total ou parcial, disjunta ou sobreposta

# Hierarquia de classe

Superclasse  $C \{ch, a_1, \dots, a_n\}$  e  
m subclasses  $\{S_1, S_2, \dots, S_m\}$

- Múltiplas Tabelas – somente subclasse
  - Criar uma tabela  $L_i$  para cada subclasse  $S_i$  com  $Atr(L_i) = \{\text{atributos de } S_i\} \cup \{ch, a_1, \dots, a_n\}$

Só funciona para especialização cujas subclasses são totais

Recomendada se houver restrição de disjunção

# Hierarquia de classe

Superclasse  $C \{ch, a_1, \dots, a_n\}$  e  
m subclasses  $\{S_1, S_2, \dots, S_m\}$

- Tabela única com um atributo tipo
  - Criar uma única tabela L com  $\text{Attr}(L) = \{ch, a_1, \dots, a_n\} \cup \{\text{atributos de } S_i\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$
  - T é o atributo Tipo  $\rightarrow$  indica a subclasse

Só funciona para especialização cujas subclasses são disjuntas

Problema: gera muitos valores null

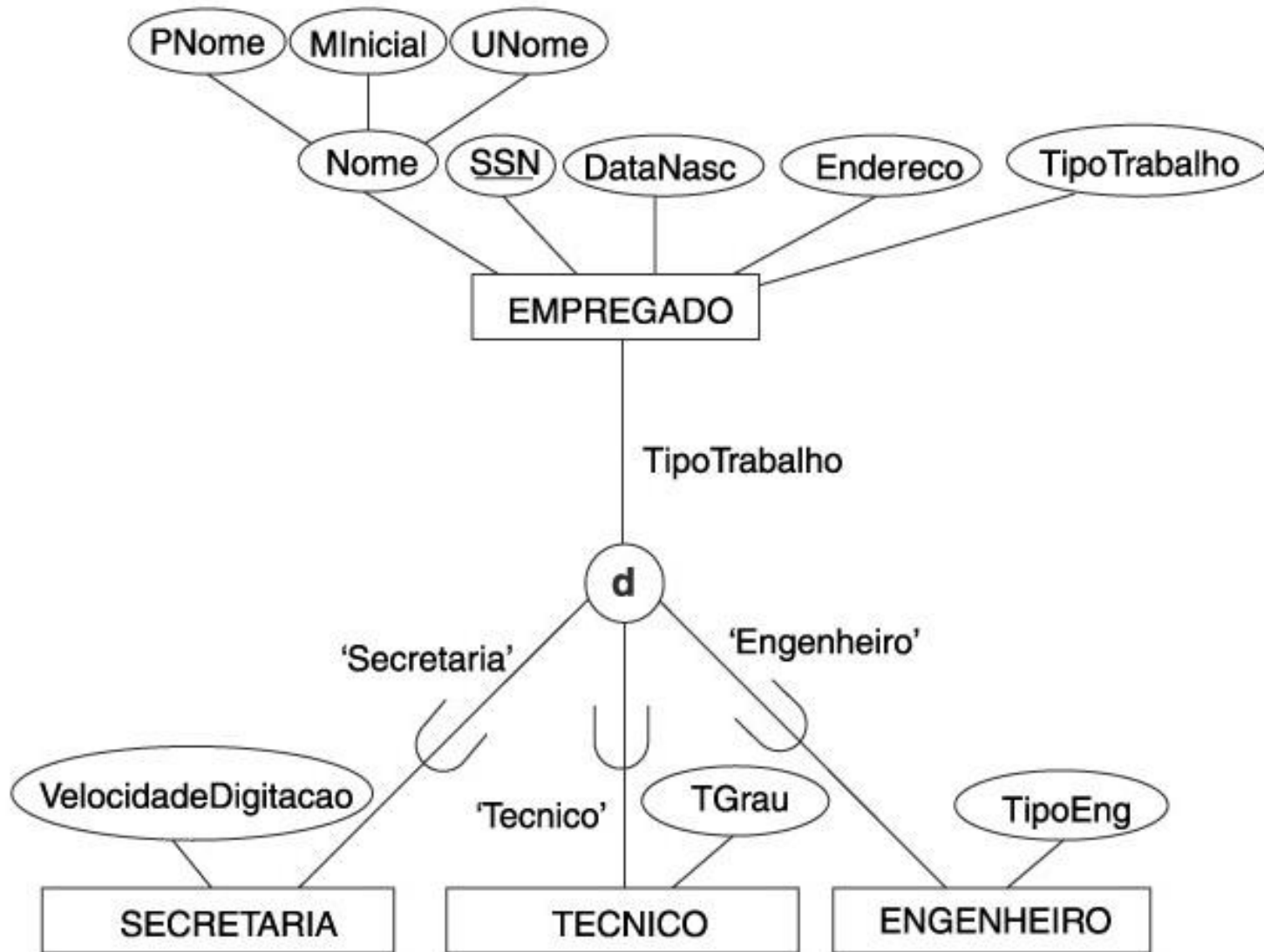
# Hierarquia de classe

Superclasse  $C \{ch, a_1, \dots, a_n\}$  e  
m subclasses  $\{S_1, S_2, \dots, S_m\}$

- Relação isolada com atributos de múltiplos tipos
  - Crie uma única tabela L com  $Atr(L) = \{ch, a_1, \dots, a_n\} \cup \{\text{atributos de } S_i\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t_1, t_2, \dots, t_m\}$
  - $T_i$  atributo do tipo booleano  $\rightarrow$  pertence a  $S_i$  ou não

Usado para especialização cujas classes são sobrepostas

# Hierarquia de classe



# Hierarquia de classe

Empregado

CPF → chave P

PNome

Minicial

Unome

DataNasc

Endereco

TipoTrabalho

Secretaria

CPF → chave primária/estrangeira

VelocidadeDigitacao

Tecnico

CPF → chave primária/estrangeira

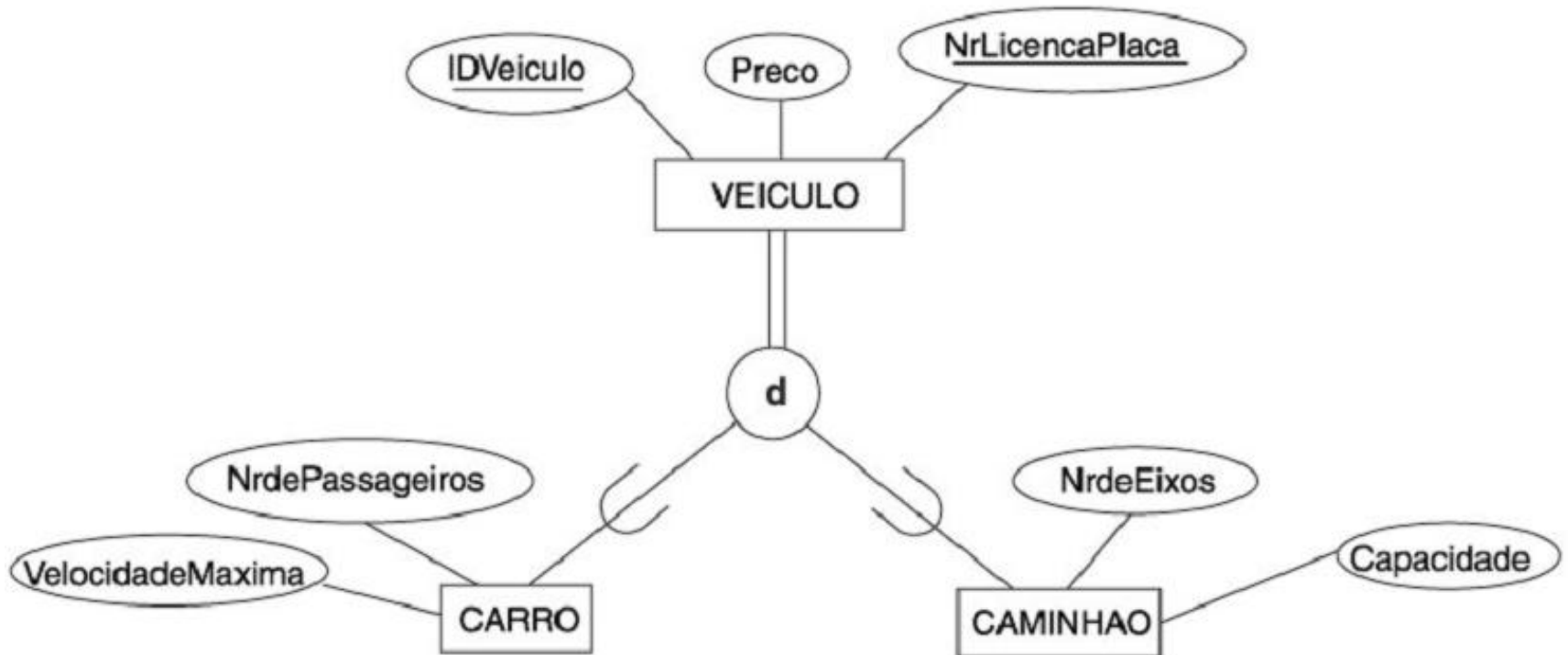
TGrau

Engenheiro

CPF → chave primária/estrangeira

TipoEng

# Hierarquia de classe





# Hierarquia de classe

## Carro

id\_veiculo → Chave P

Placa

Preco

Veloc\_Max

NroPass

## Caminhao

id\_veiculo → Chave P

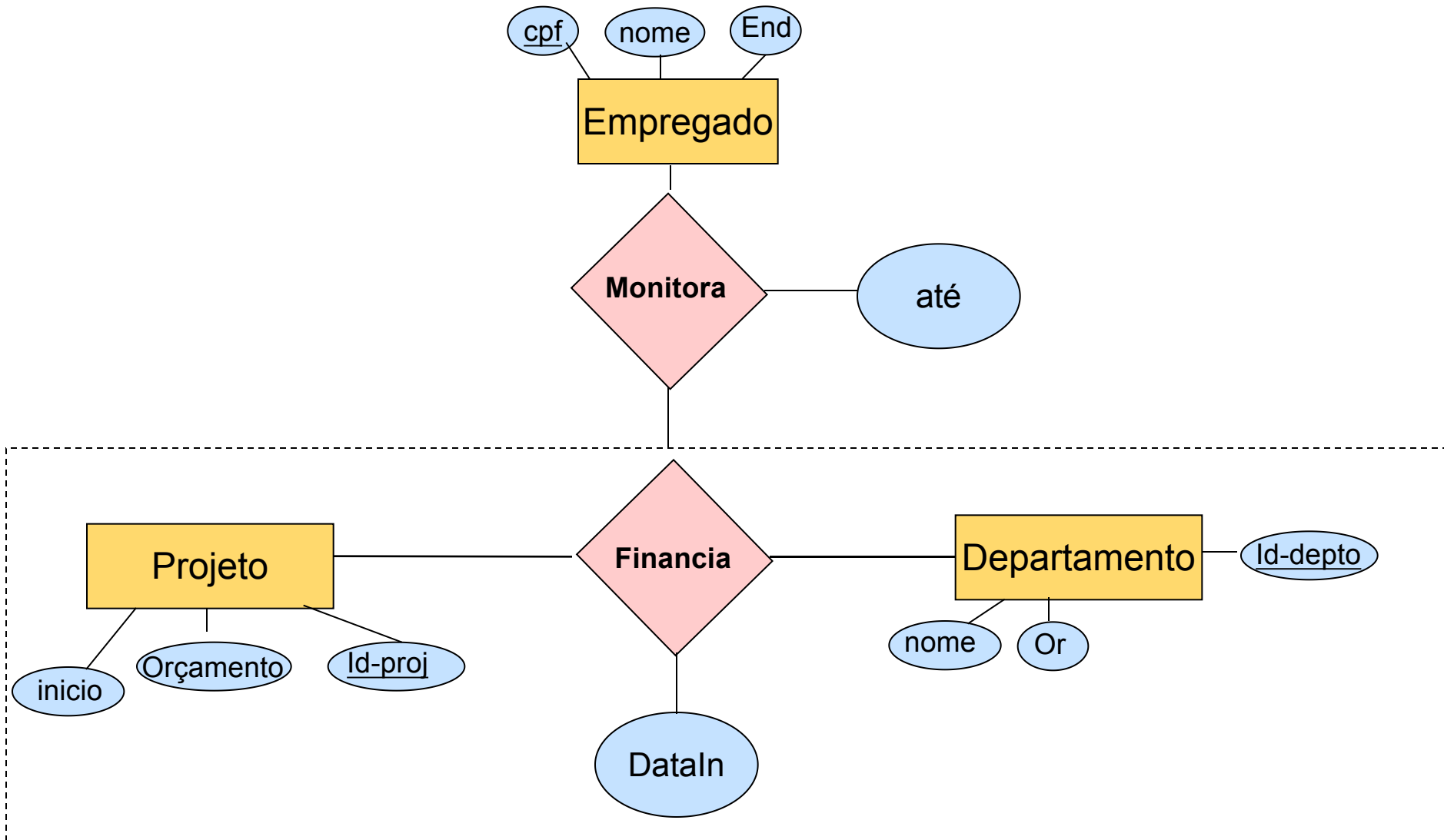
Placa

Preco

Nro\_Eixos

Capacidade\_Peso

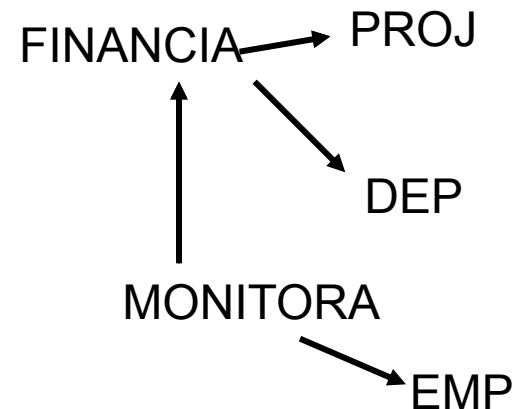
# Diagramas ER com Agregação



# Diagramas ER com Agregação

- Mapear os conjuntos de entidades Projeto, Empregado e Departamento para Tabelas
- Mapear o conjunto de relacionamentos Financia
  - id\_depto, id\_proj e DataIn
  - Pode-se criar uma nova chave para a tabela. Ex: ID\_Financia
- Para monitora, cria-se uma tabela contendo

- **Os atributos chave de Empregado (cpf)**
- **Os atributos chave de financia (id\_depto e id\_proj)**
- **Os atributos descritivos de Monitora (até)**



# Referências

- R. Elmasri e S. B. Navathe, Sistema de Banco de Dados, 6ª edição, Pearson, 2011.
- R. Ramakrishnan e J. Gehrke, *Database Management Systems*, 3a Edição, McGraw-Hill, 2003.
- Slides da Profa. Sandra de Amo – FACOM - UFU