

INSTITUTO FEDERAL  
PIAUÍ  
Campus Parnaíba

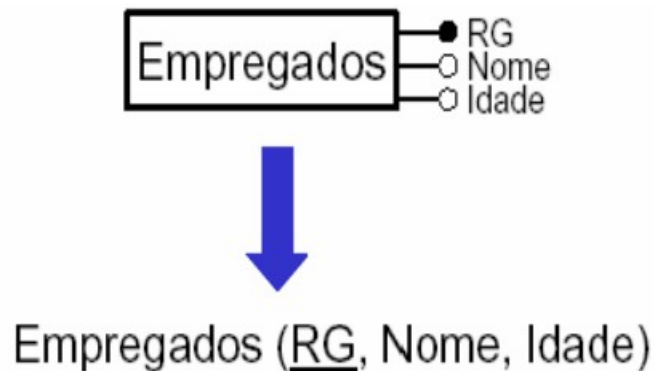
# Mapeamento ER - Relacional

Prof. Msc Denival A. dos Santos

# Mapeamento

## ■ Entidade

- Para cada entidade (não fraca) deve ser criada uma tabela.
- Cada atributo da entidade é definido como uma coluna.
- O atributo identificador se transforma em chave primária (PK).



# Mapeamento

## ▪ Atributo Multivalorados (2 soluções)

### 1) Primeira:

- Cria uma uma uma tabela
- Empregado(CPF, nome, idade)
- Telefone (CPF, Telefone)



### 2) Substitui por atributos fixos

- Deve ser realizada somente quando a quantidade for pequena.
- Empregado(CPF, nome, idade, foneCom, foneRes, celular)

# Mapeamento

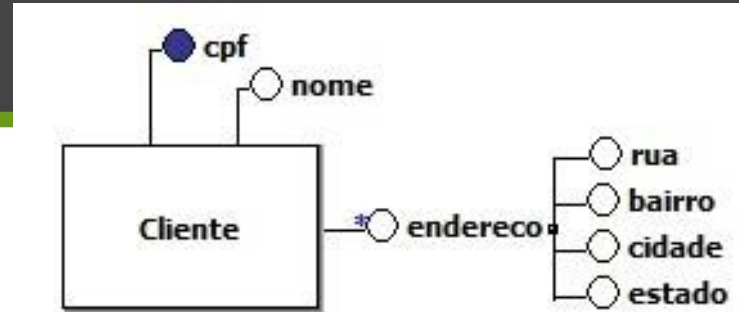
## ■ Atributo Composto (2 soluções)

### 1) Primeira:

- Combinar (esquece da composição)
- Cliente (CPF, nome, endereco)

### 2) Substitui por atributos fixos

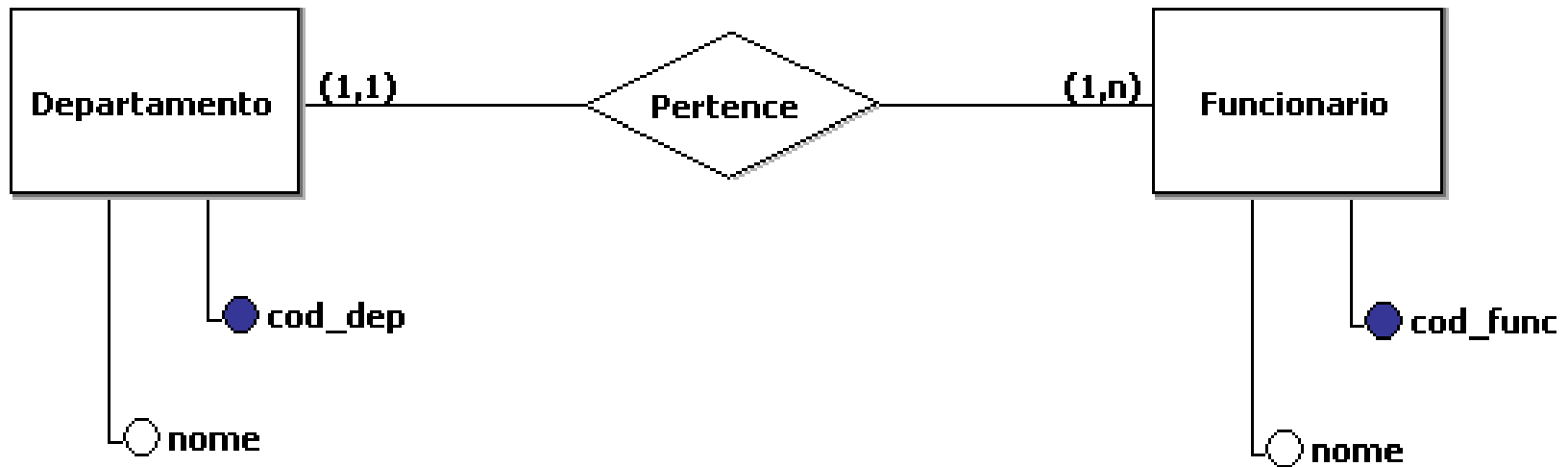
- Aplainar (Esquece as agregações lógicas)
- Cliente (CPF, nome, rua, bairro, cidade, estado)



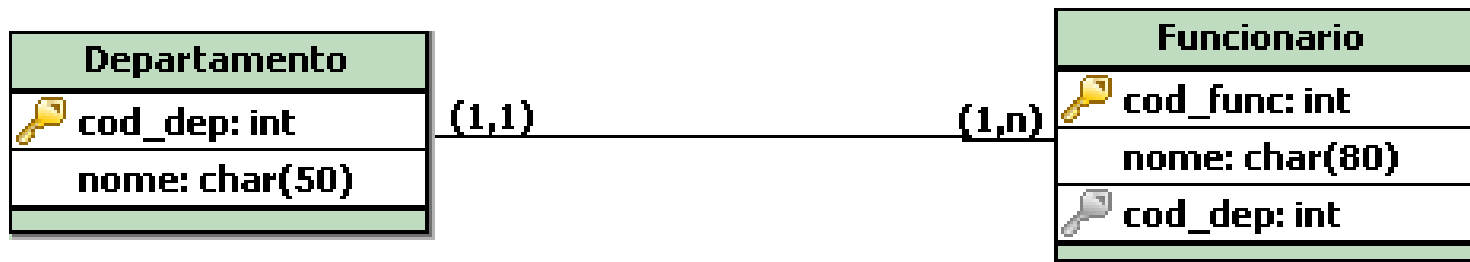
# Mapeamento dos relacionamento

- Relacionamento 1 para N
- Auto-relacionamento 1 para N
- Relacionamento 1 para 1
- Auto-relacionamento 1 para 1
- Relacionamento M para N
- Auto-relacionamento M para N
- Relacionamento ternário
- Generalização / Especialização

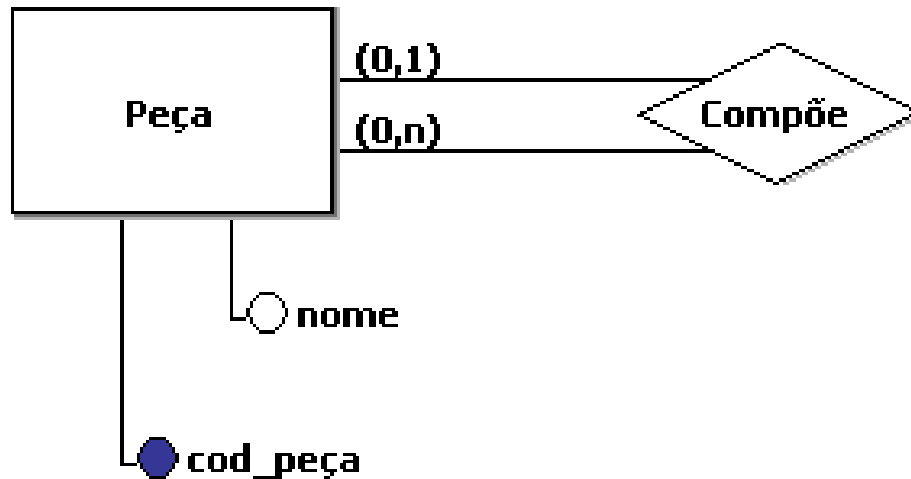
# Relacionamento 1 para N





- A entidade (funcionário) cuja cardinalidade é N incorpora o identificador (chave) da entidade (departamento) cuja cardinalidade é 1 (chave estrangeira).



# Auto-relacionamento 1 para N

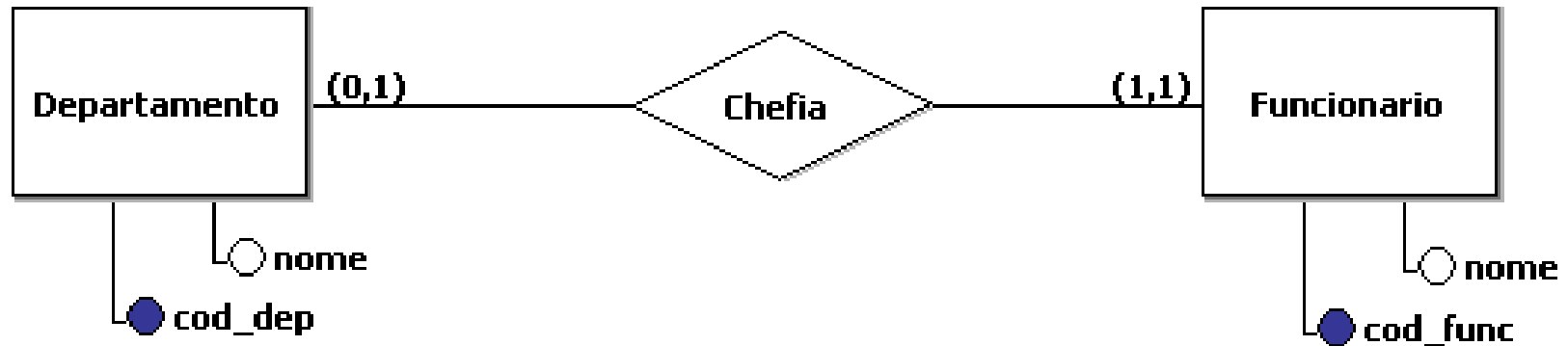


- Incluir a chave primária da entidade na própria entidade como chave estrangeira, gerando uma estrutura de acesso a partir desta chave estrangeira.

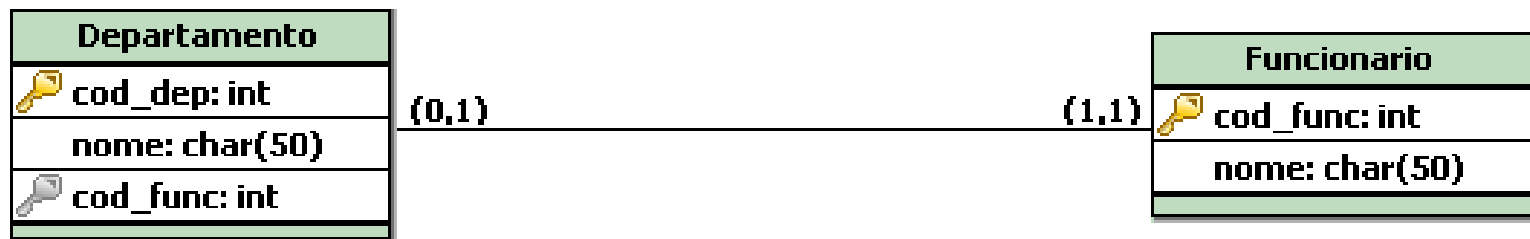
Peça	
	cod_peça: int
	nome: char(40)
	cod_peçache: int

Obs.: Compõe

# Relacionamento 1 para 1

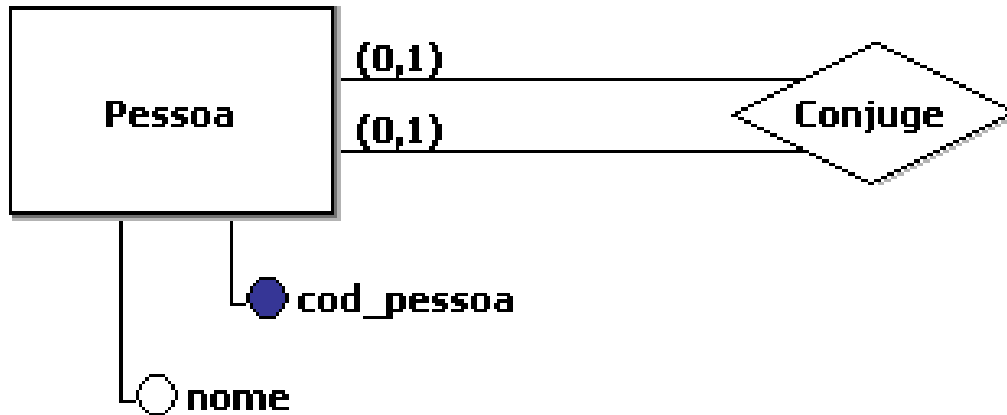


- As entidades (tabelas) envolvidas neste relacionamento carregarão o identificador da outra (uma ou outra ou ambas) conforme a conveniência do projeto (de acordo com o acesso a essas tabelas).







# Auto-relacionamento 1 para 1

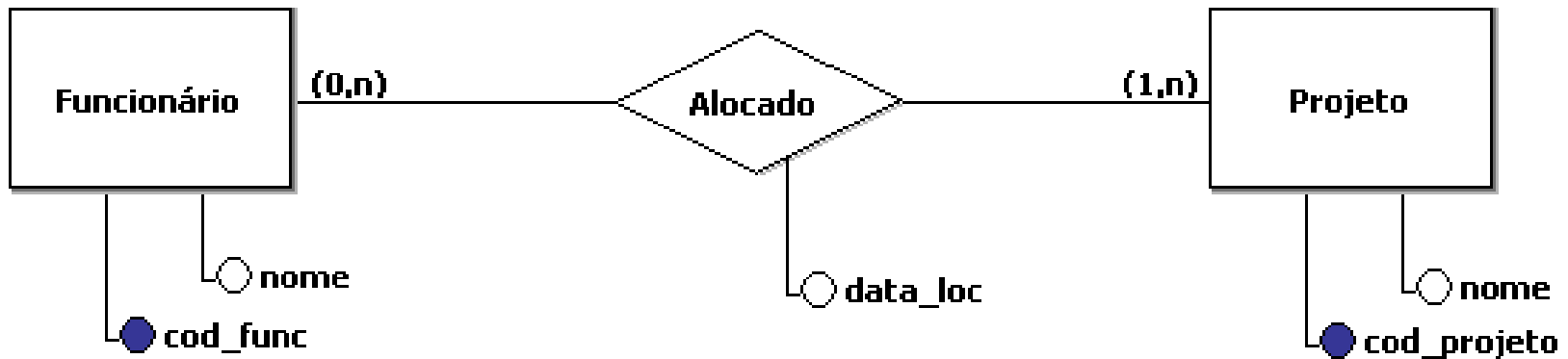


- Incluir a chave primária da entidade na própria entidade (chave estrangeira) e gerar uma estrutura de acesso para ela.

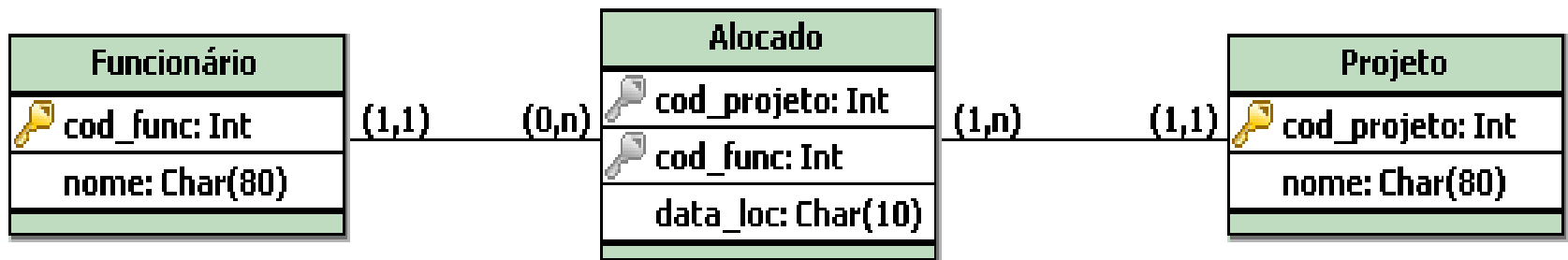
Pessoa	
	cod_pessoa: int
	nome: char(50)
	cod_pessoache: int

Obs.: C njuge

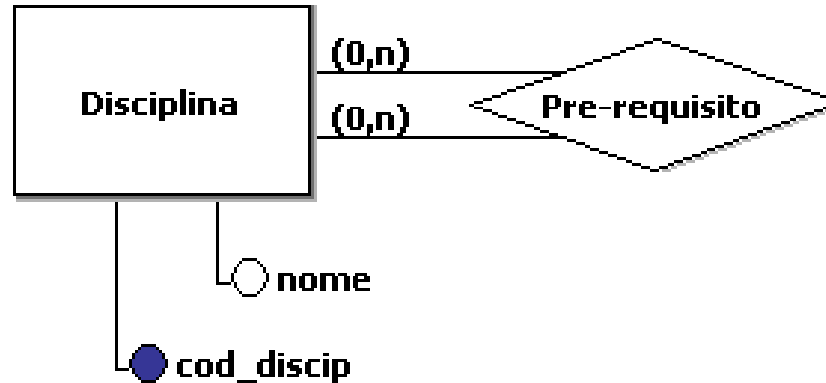
# Relacionamento M para N:



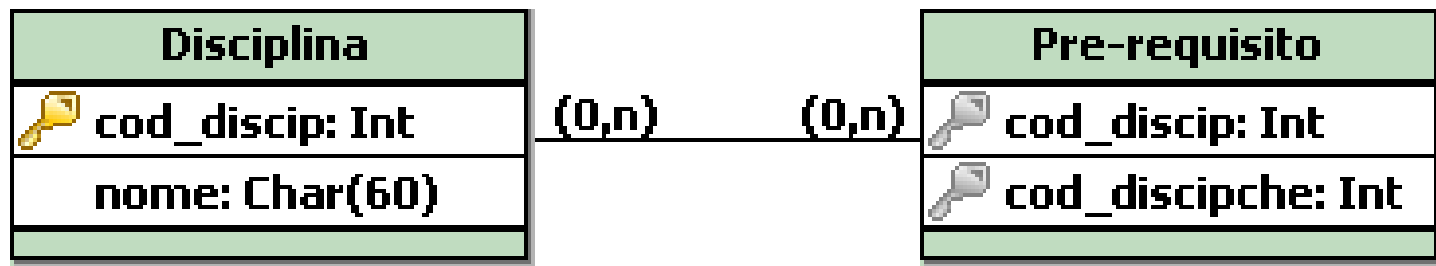
- O relacionamento torna-se uma tabela com atributos descritivos (caso haja) e com os identificadores das tabelas (entidades) que ele relaciona.



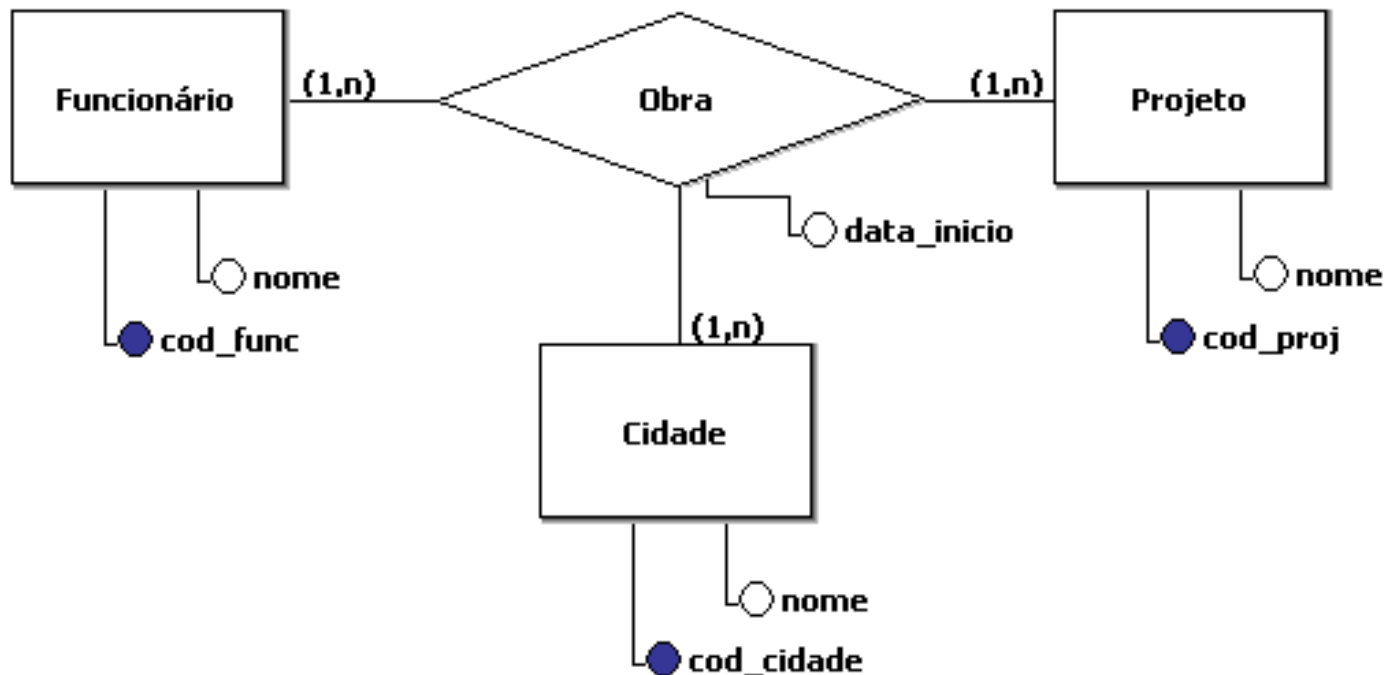
# Auto-relacionamento M para N



- No auto-relacionamento M para N também acontecerá de ser criada uma nova tabela composta pelos identificadores.

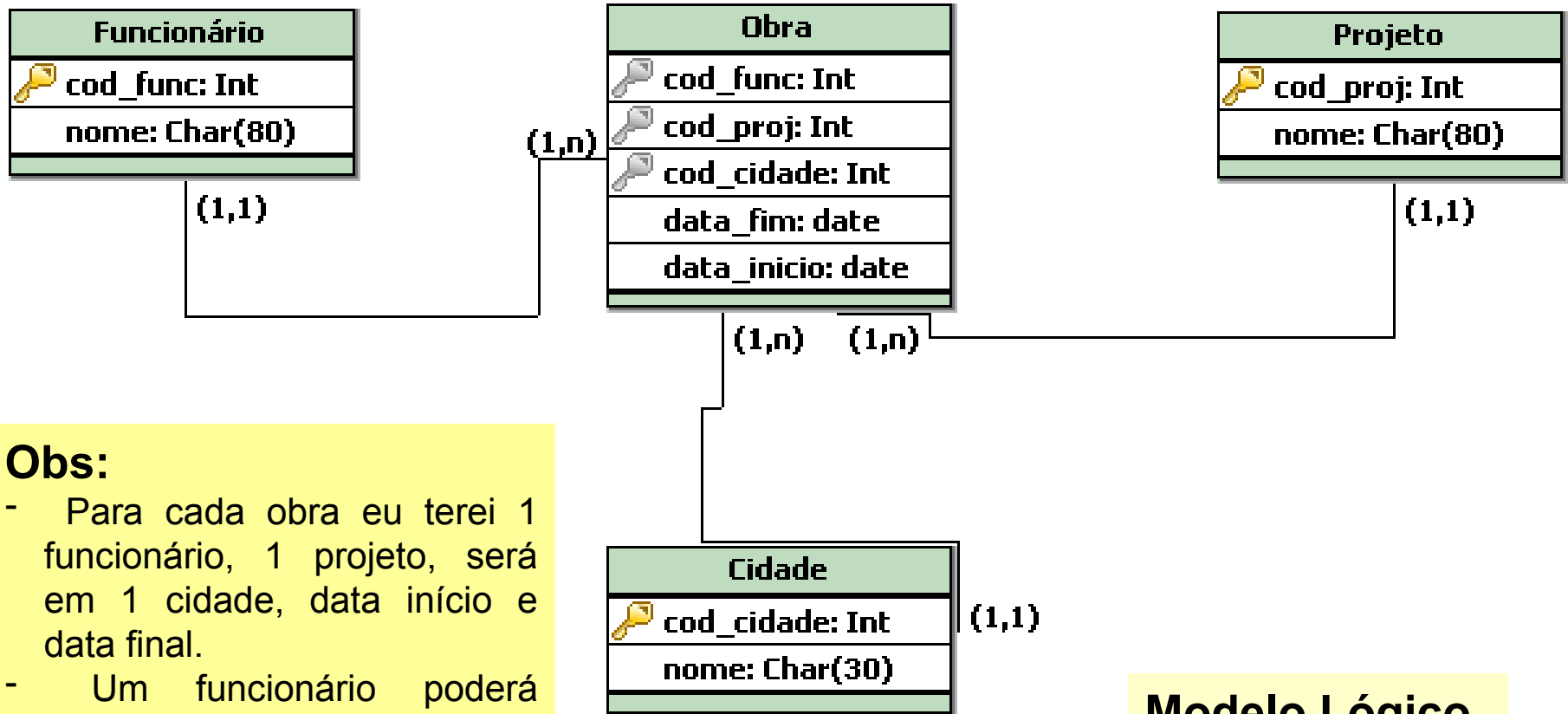


# Relacionamento ternário



- O relacionamento é mapeado em uma tabela e são geradas tantas estruturas de acesso quanto for o grau do relacionamento. A chave primária de cada uma das entidades associadas gera uma estrutura de acesso. A chave desta nova tabela é a concatenação das chaves estrangeiras.

# Relacionamento ternário:

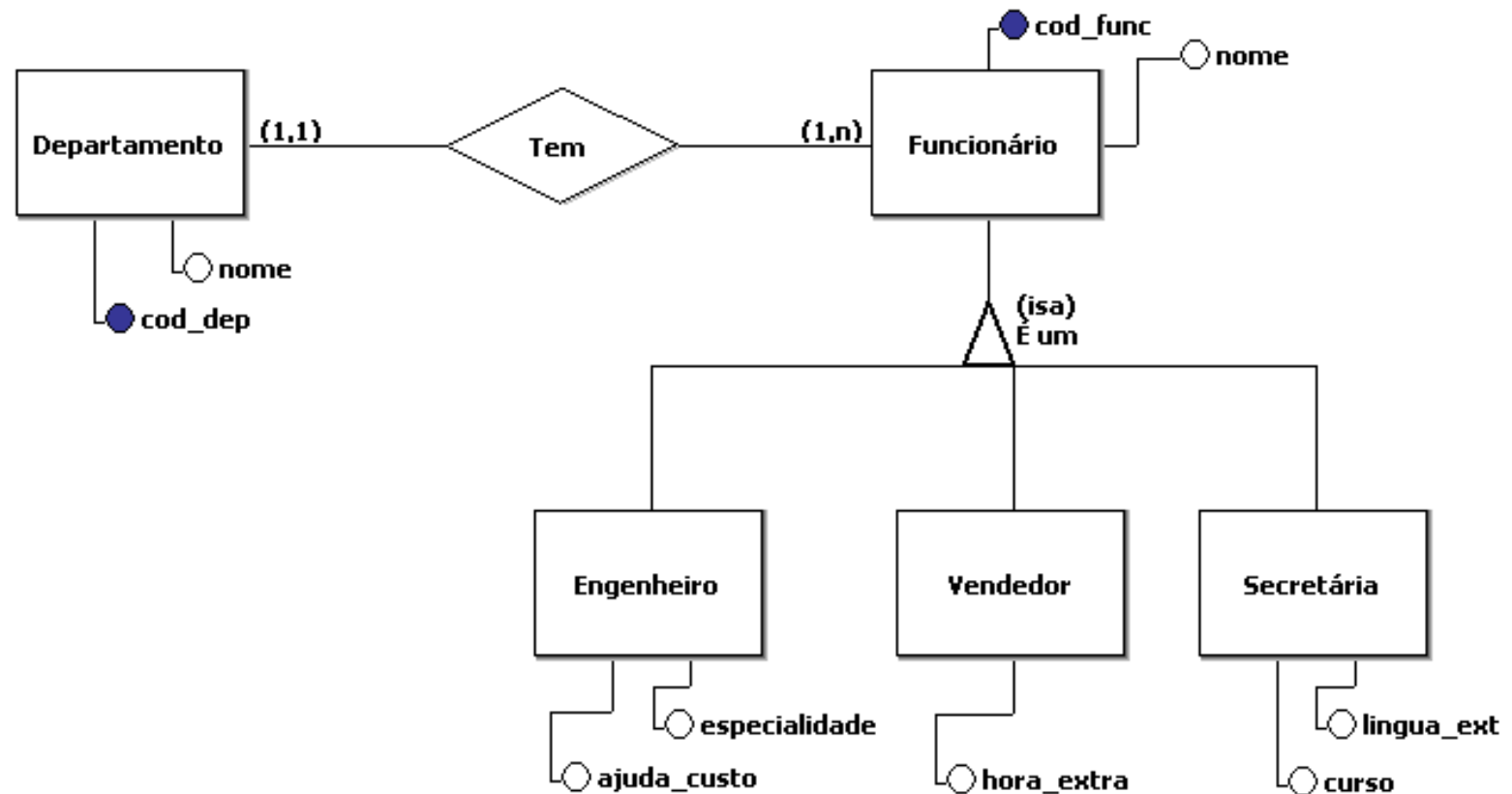


## Obs:

- Para cada obra eu terei 1 funcionário, 1 projeto, será em 1 cidade, data início e data final.
- Um funcionário poderá participar de 1 ou várias obras.
- Um projeto poderá ser executado 1 ou várias vezes em cidades ou períodos diferentes.

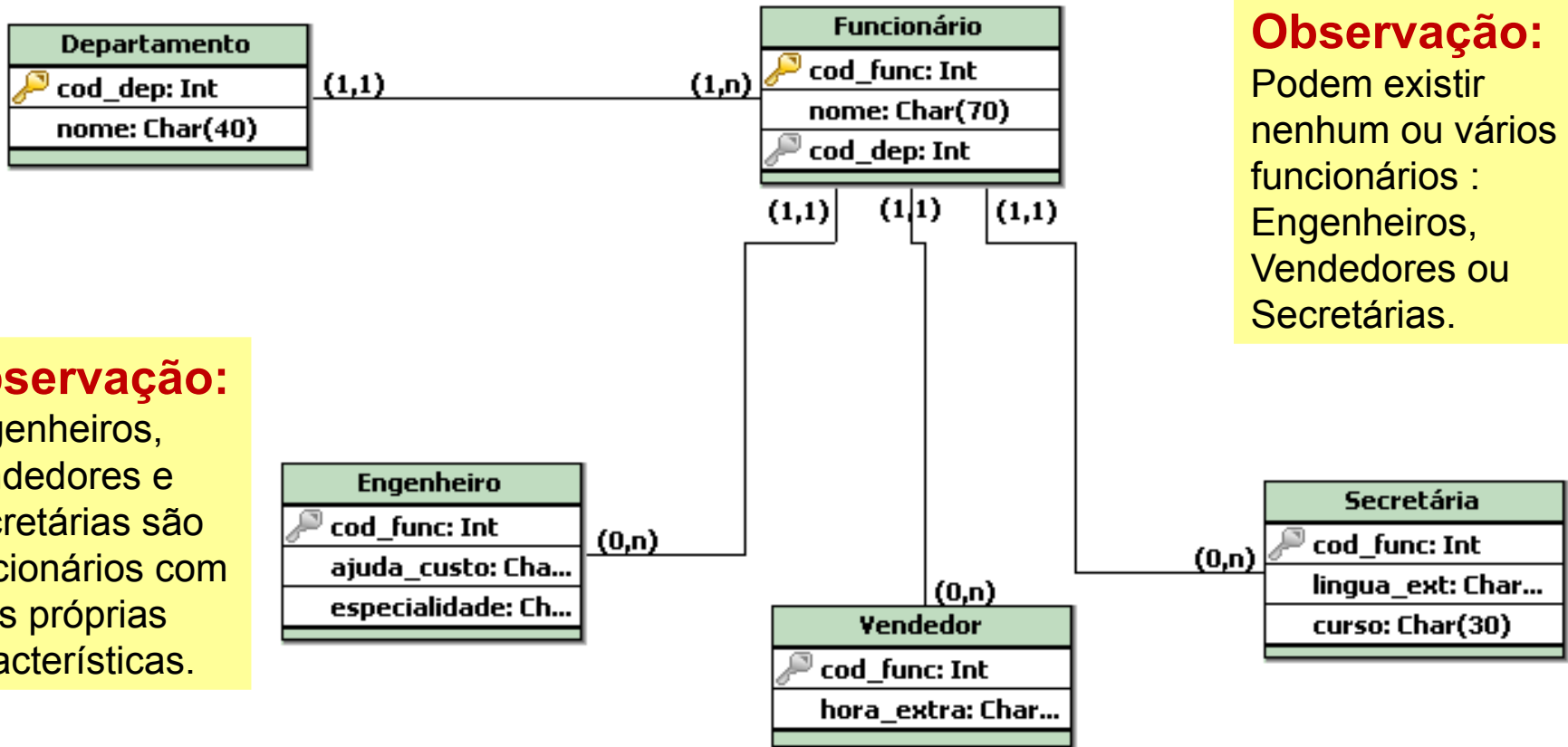
**Modelo Lógico**

# Generalizações



- Os subconjuntos tornam-se tabelas carregando o identificador do conjunto ao qual pertencem.

# Generalizações



## Observação:

Engenheiros, Vendedores e Secretárias são funcionários com suas próprias características.

## Observação:

Podem existir nenhum ou vários funcionários : Engenheiros, Vendedores ou Secretárias.

- O conjunto “funcionário” vira uma tabela (regra padrão) e os subconjuntos em outras tabelas, levando consigo a chave primária do “funcionário”