

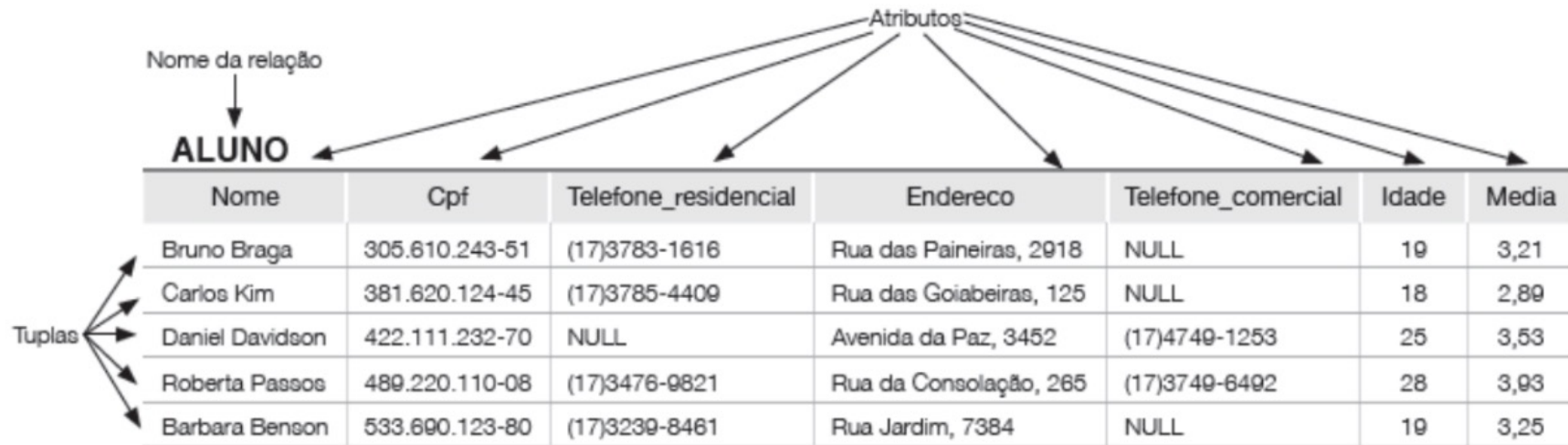
# Banco de Dados

# Modelo Relacional

# Modelo Relacional

- Proposto em 1970 pelo Dr. Edgar Frank Codd (IBM).
- Consiste em uma coleção de tabelas de nomes únicos.
- Estrutura de dados simples e uniforme:
  - Um banco de dados relacional é um conjunto de relações;
  - Cada relação é um conjunto de linhas (ou tuplas);
  - Cada tupla é uma lista de valores de atributos;
  - Cada valor de atributo é retirado de um domínio.

# Modelo Relacional



# Modelo Relacional

## Domínio

-O tipo de dado que descreve os tipos de valores que podem aparecer em cada coluna é representado por um **domínio** de valores possíveis.

# Modelo Relacional

## Características das Relações

- Uma relação é definida como um conjunto de tuplas;
- As tuplas em uma relação, assim como os elementos de um conjunto, não possuem ordem entre eles.
- A ordem dos atributos e seus valores não é tão importante, desde que a correspondência entre atributos e valores seja mantida.
- Cada valor em uma tupla é um valor atômico.

# Modelo Relacional

-Como não pode haver uma tupla repetida (duplicada) em uma instância da relação, isto significa que é possível identificar cada tupla separadamente uma da outra por meio da escolha de algum atributo.

## Chaves

- Chave primária (PK)
- Chave única (UK)
- Chave estrangeira (FK)

# Modelo Relacional

## Mapeamento do MER para Relacional

O Modelo Entidade-Relacionamento (MER) é responsável por realizar uma representação mais conceitual dos dados de uma aplicação. Esta representação é um pouco distante da forma como realmente os elementos (entidades e relacionamentos) serão implementados.

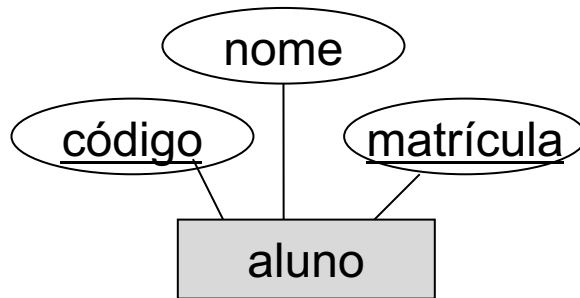
O modelo relacional fornece uma representação dos dados de forma mais próxima de como estes dados se encontrarão quando forem definidos os arquivos para o BD.



# Modelo Relacional

## Regras para o mapeamento

1- Todas as entidades são mapeadas para uma relação contendo os mesmos atributos do DER.

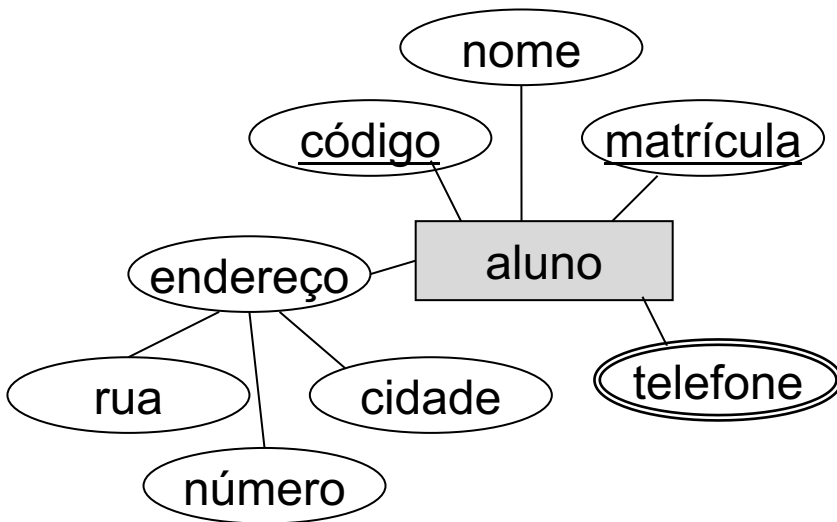


tbAluno
ALU_CODIGO (PK)
ALU_NOME
ALU_MATRICULA (UK)

# Modelo Relacional

## Regras para o mapeamento

2- Para os atributos multivalorados (A), é criada uma relação R que terá como atributos os mesmos de A mais a chave primária da entidade (ou relacionamento) no qual A é atributo. Se o atributo multivalorado é composto, será incluído seus componentes.



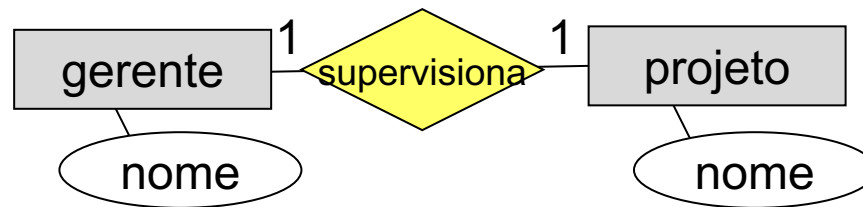
tbAluno
ALU_CODIGO (PK)
ALU_NOME
ALU_MATRICULA (UK)
ALU_RUA
ALU_NUMERO
ALU_CIDADE

tbAlunoTelefone
ALT_CODIGO (PK)
ALT_TELEFONE
<b>ALU_CODIGO (FK)</b>

# Modelo Relacional

## Regras para o mapeamento

3- Para relacionamento 1:1, dentre as relações que mapeiam as entidades participantes escolha uma delas (a que possuir participação total) e inclua como chave estrangeira a chave primária da outra.



**OU**

tbGerente
GER_CODIGO (PK)
GER_NOME
PRO_CODIGO (FK)

tbProjeto
PRO_CODIGO (PK)
PRO_NOME

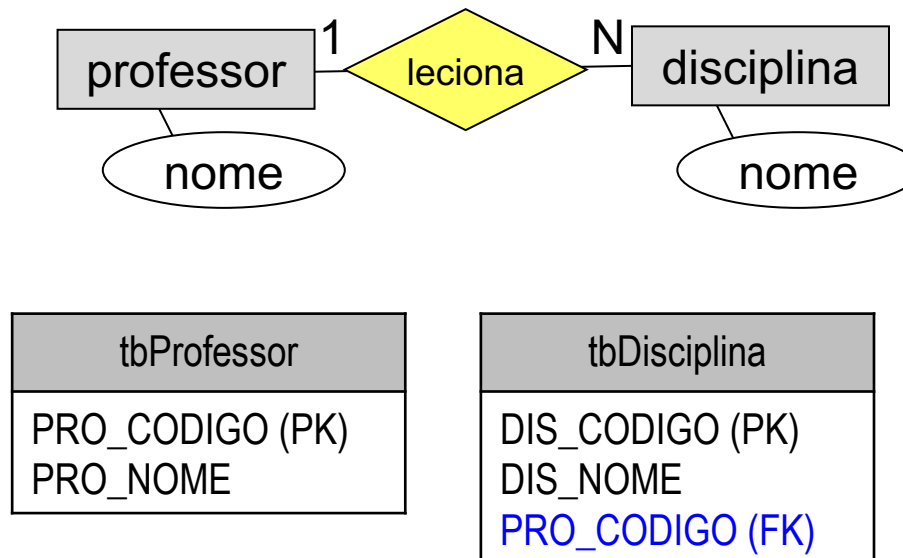
tbGerente
GER_CODIGO (PK)
GER_NOME

tbProjeto
PRO_CODIGO (PK)
PRO_NOME
GER_CODIGO (FK)

# Modelo Relacional

## Regras para o mapeamento

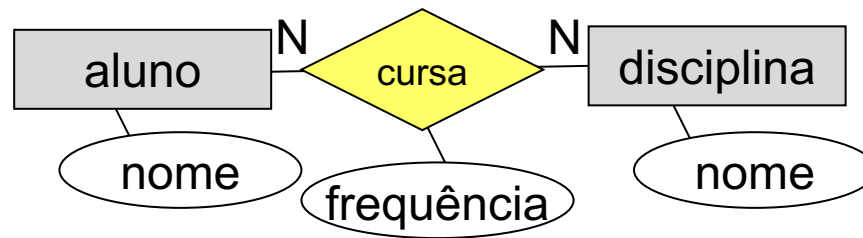
4- Para relacionamentos 1:N, escolha a relação que representa a entidade presente no lado N e acrescente como chave estrangeira a chave primária da entidade do lado 1.



# Modelo Relacional

## Regras para o mapeamento

5- Para relacionamentos N:N, é criada uma nova relação contendo como chaves estrangeiras as chaves primárias das entidades participantes, mais os atributos do relacionamento.



tbAluno
ALU_CODIGO (PK)
ALU_NOME

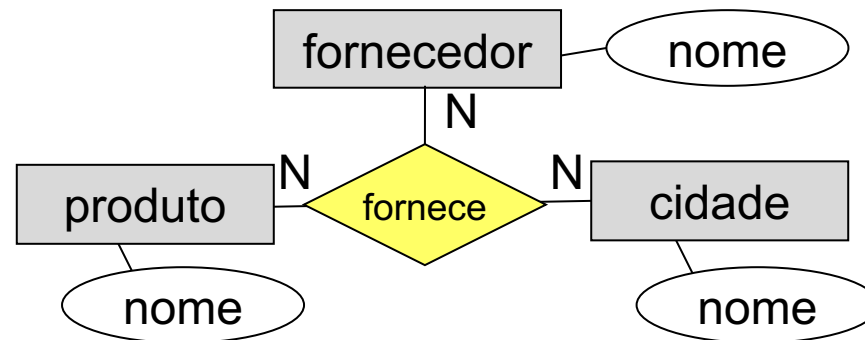
tbDisciplina
DIS_CODIGO (PK)
DIS_NOME

tbAlunoDisciplina
ALD_CODIGO (PK)
ALU_CODIGO (FK)
DIS_CODIGO (FK)
ALD_FREQUENCIA

# Modelo Relacional

## Regras para o mapeamento

6- Para relacionamentos triplos, o mapeamento ocorre de forma semelhante ao descrito pela regra 5.

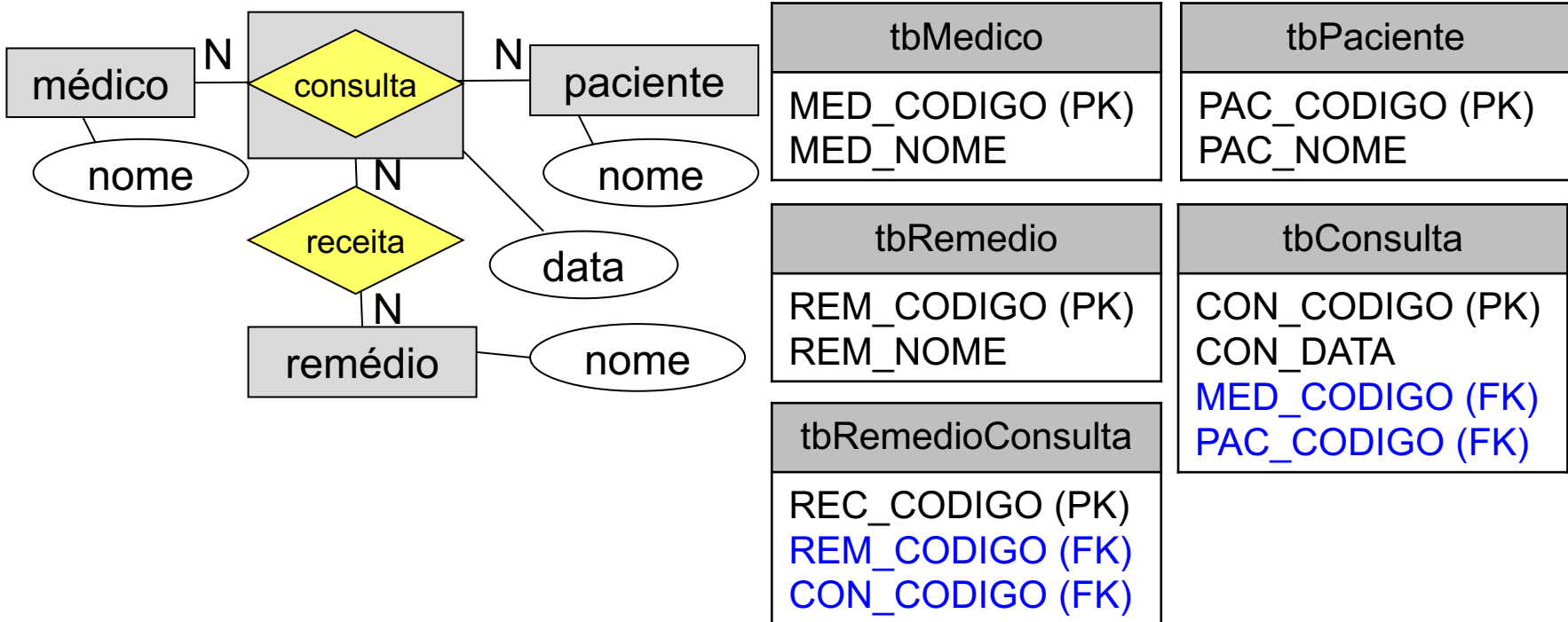


tbProduto	tbCidade	tbFornecedor	tbFornProdCidad
PRO_CODIGO (PK) PRO_NOME	CID_CODIGO (PK) CID_NOME	FOR_CODIGO (PK) FOR_NOME	FPC_CODIGO (PK) CID_CODIGO (FK) PRO_CODIGO (FK) FOR_CODIGO (FK)

# Modelo Relacional

## Regras para o mapeamento

7- Na agregação, crie uma relação para cada entidade, inclusive para a entidade associativa. Adicione as chaves estrangeiras conforme a cardinalidade dos relacionamentos.



# Modelo Relacional

## Regras para o mapeamento

8- Converta cada especialização com  $m$  subclasses  $\{S_1, S_2, \dots, S_m\}$  e superclasse  $SC$ , onde os atributos de  $SC$  são  $\{c, a_1, a_2, \dots, a_n\}$  onde  $c$  é a chave primária de  $SC$ , em tabelas utilizando uma das seguintes opções:

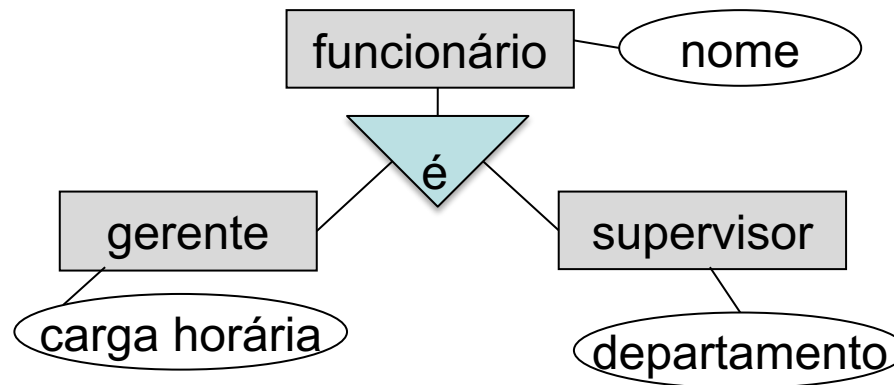
- a) Crie uma tabela  $T$  para  $SC$  com os atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\}$  e chave  $C(T) = c$ ; crie uma tabela  $T_i$  para cada subclasse  $S_i$ ,  $1 \leq i \leq m$ , com os atributos  $A(T_i) = \{c\} \cup A(S_i)$ , onde  $C(T) = c$ ;
- b) Crie uma tabela  $T_i$  para cada subclasse  $S_i$ ,  $1 \leq i \leq m$ , com os atributos  $A(T_i) = A(S_i) \cup \{c, a_1, a_2, \dots, a_n\}$  e  $C(T_i) = c$ ;
- c) Crie uma tabela  $T$  com os atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\} \cup A(S_1) \cup \dots \cup A(S_m) \cup \{t\}$ , onde  $t$  é um atributo tipo que indica a subclasse à qual cada tupla pertence, caso isto venha a ocorrer;
- d) Crie uma tabela  $T$  com atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\} \cup A(S_1) \cup \dots \cup A(S_m) \cup \{t_1, t_2, \dots, t_m\}$ ; cada  $t_i$ ,  $1 \leq i \leq m$ , é um atributo “booleano” indicando se a tupla pertence ou não à subclasse  $S_i$ ; embora funcional, esta opção pode gerar uma quantidade muito grande de valores nulos.



# Modelo Relacional

## Regras para o mapeamento

8- a) Crie uma tabela T para SC com os atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\}$  e chave  $C(T) = c$ ; crie uma tabela  $T_i$  para cada subclasse  $S_i$ ,  $1 \leq i \leq m$ , com os atributos  $A(T_i) = \{c\} \cup A(S_i)$ , onde  $C(T) = c$ ;



tbFuncionario
FUN_CODIGO (PK)
FUN_NOME

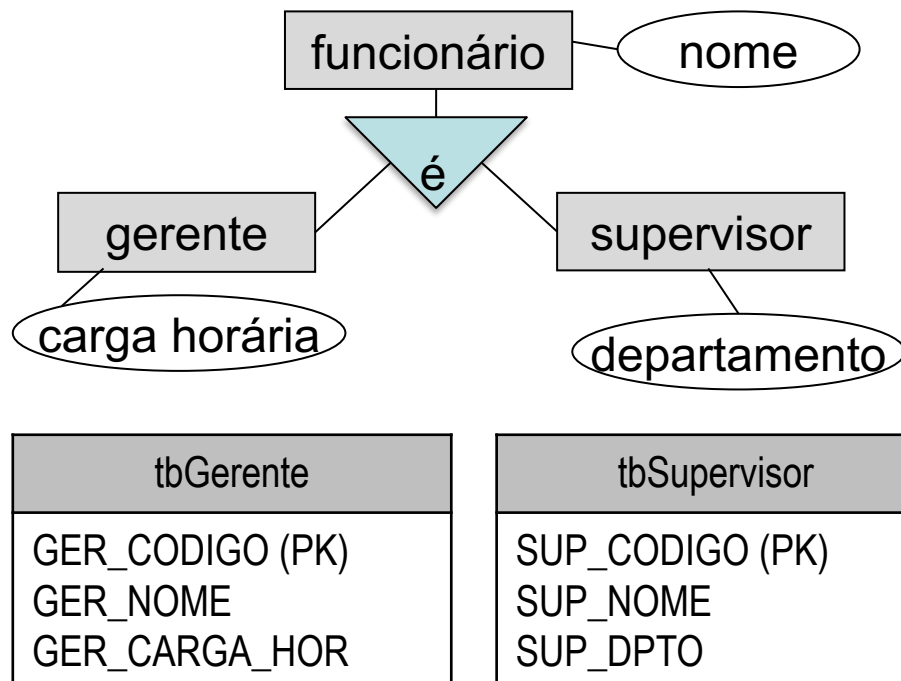
tbGerente
GER_CODIGO (PK)
GER_CARGA_HOR
FUN_CODIGO (FK)

tbSupervisor
SUP_CODIGO (PK)
SUP_DPTO
FUN_CODIGO (FK)

# Modelo Relacional

## Regras para o mapeamento

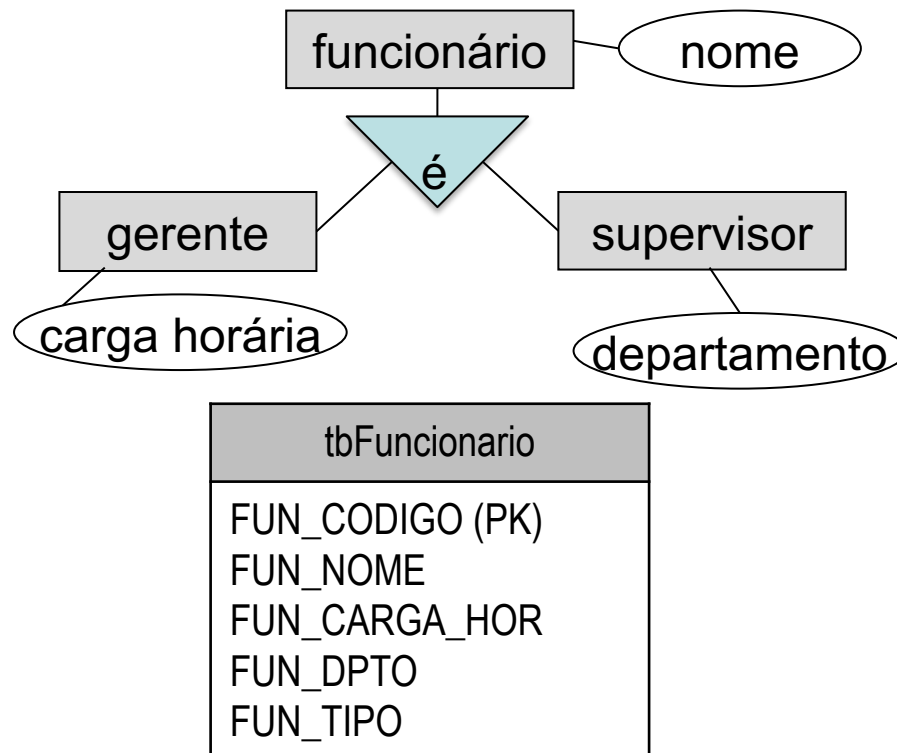
8- b) Crie uma tabela  $T_i$  para cada subclasse  $S_i$ ,  $1 \leq i \leq m$ , com os atributos  $A(T_i) = A(S_i) \cup \{c, a_1, a_2, \dots, a_n\}$  e  $C(T_i) = c$ ;



# Modelo Relacional

## Regras para o mapeamento

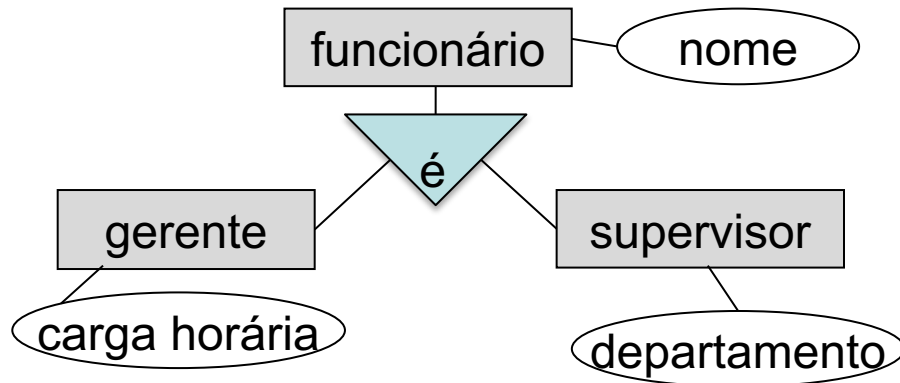
8- c) Crie uma tabela T com os atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\} \cup A(S_1) \cup \dots \cup A(S_m) \cup \{t\}$ , onde t é um atributo tipo que indica a subclasse à qual cada tupla pertence, caso isto venha a ocorrer;



# Modelo Relacional

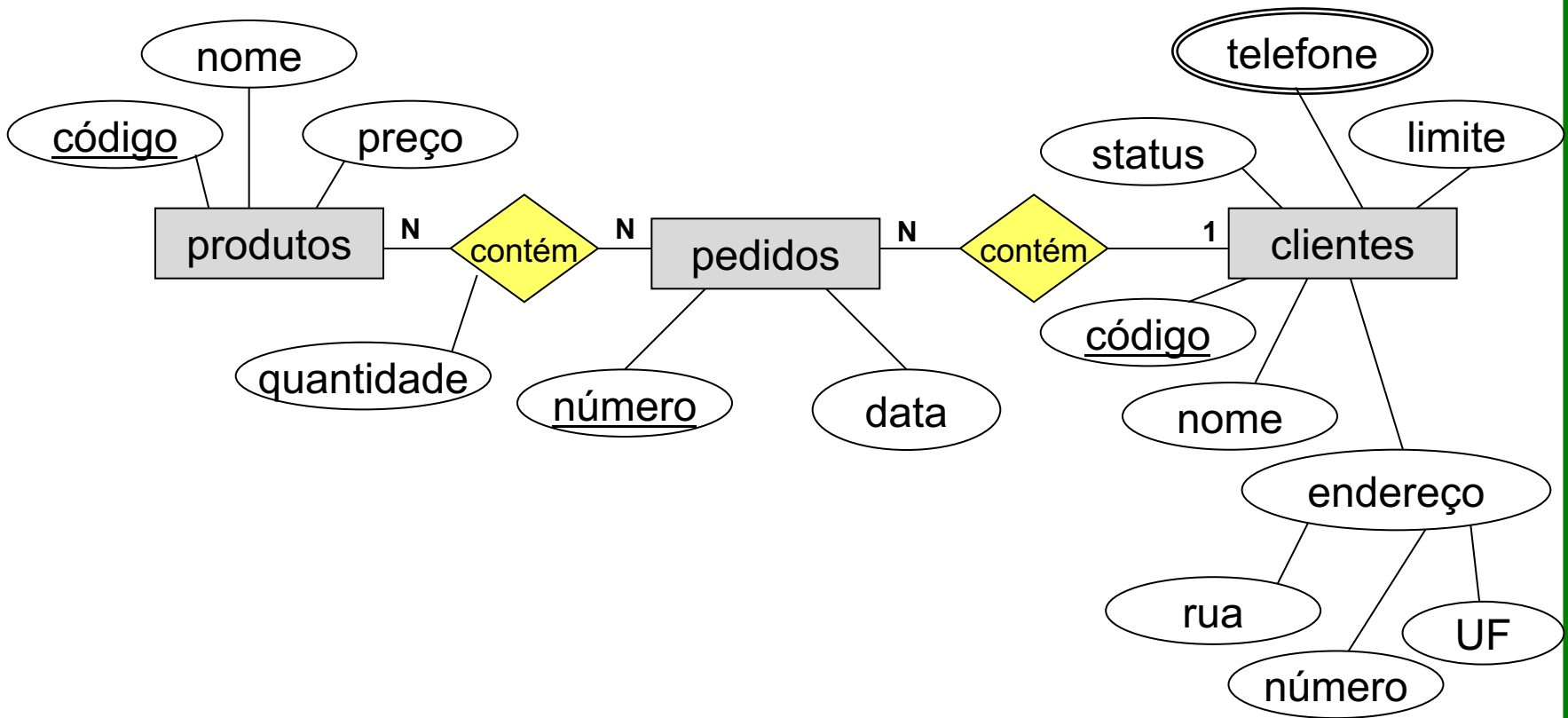
## Regras para o mapeamento

8- d) Crie uma tabela T com atributos  $A(T) = \{c, a_1, a_2, \dots, a_n\} \cup A(S_1) \cup \dots \cup A(S_m) \cup \{t_1, t_2, \dots, t_m\}$ ; cada  $t_i$ ,  $1 \leq i \leq m$ , é um atributo “booleano” indicando se a tupla pertence ou não à subclasse  $S_i$ ; embora funcional, esta opção pode gerar uma quantidade muito grande de valores nulos.

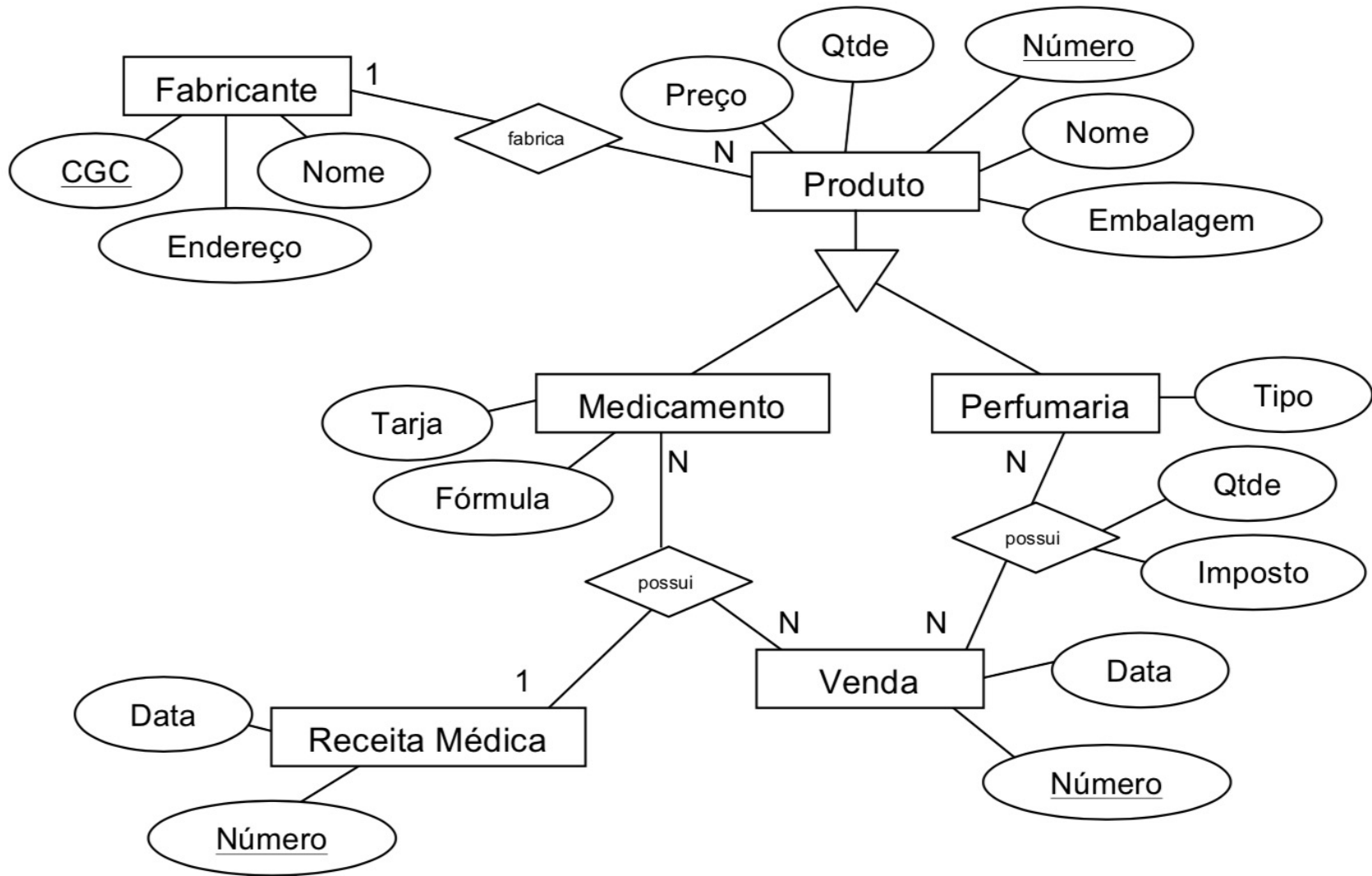


tbFuncionario
FUN_CODIGO (PK)
FUN_NOME
FUN_CARGA_HOR
FUN_DPTO
FUN_GERENTE
FUN_SUPERVISOR

# Modelo Relacional



# Modelo Relacional



# Modelo Relacional

