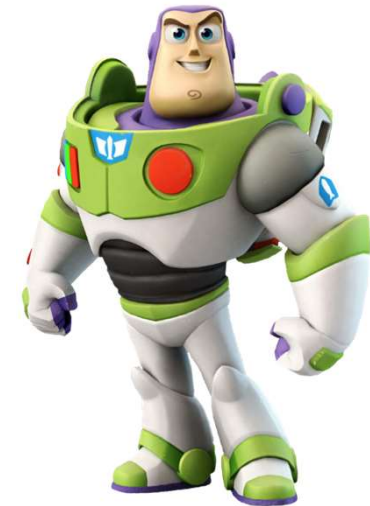


# Aula 4 – Projeto de Banco de Dados

## Modelo Lógico

- Convertendo ER em Tabelas Relacionais
- Generalizações
- Agregações
- Relacionamentos recursivos
- Relacionamentos N-ários
- Atividades



**Prof.:Antonio (Buzz)**

# Modelo Relacional

**Relação** = Tabela bi-dimensional, composta de linhas e colunas de dados.

**Relação recursiva** = relaciona o objeto a si mesmo.

**Atributo** = Coluna

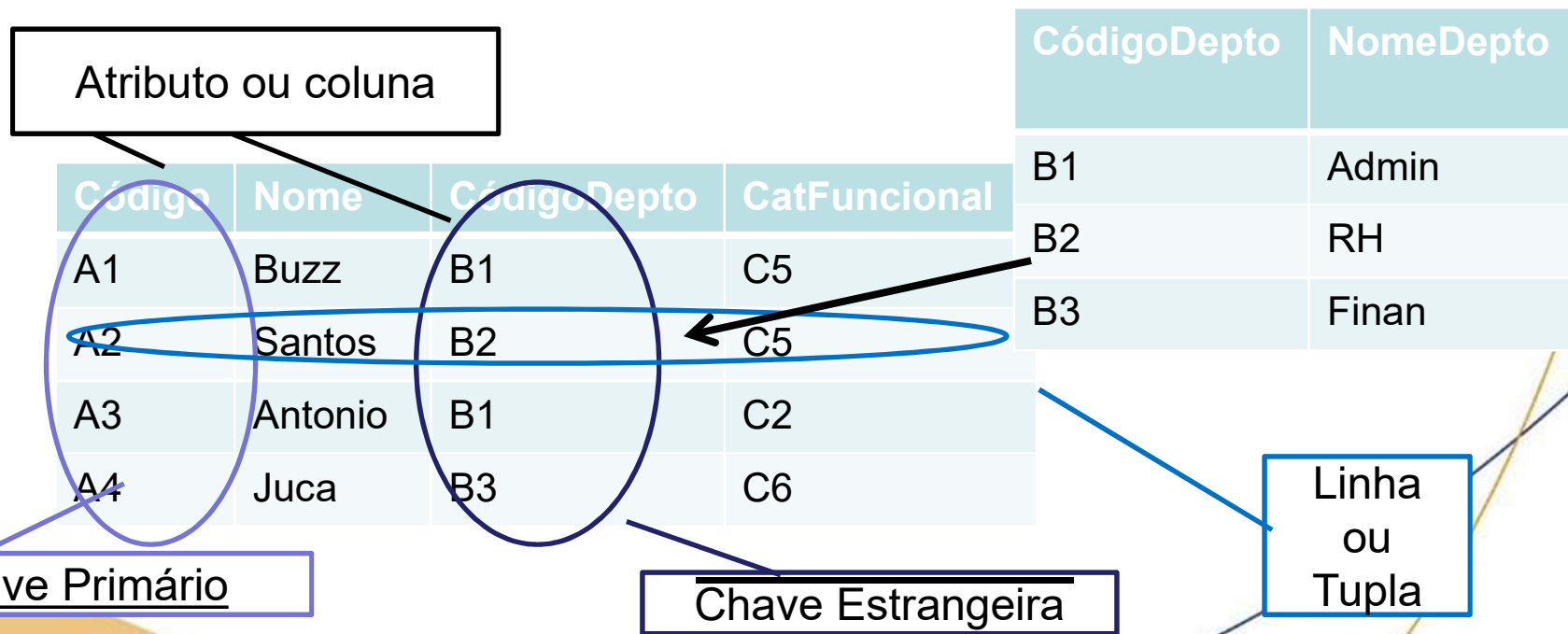
**Grau de uma relação** = Número de atributos

**Tupla** = Cada linha da relação

**Domínio de um atributo** = Conjunto ao qual pertence os valores de um atributo

Valor nulo de um atributo de uma tupla = Atributo inaplicável ou com valor desconhecido

**Esquema de Banco de Dados Relacional** = Nomes das relações seguidos pelos nomes dos atributos, com os atributos chaves sublinhados e com as chaves estrangeiras identificadas.



# Modelo Relacional

## **Atributo Identificador:**

O atributo ou conjunto de atributos que será utilizado para identificar instâncias de uma entidade. Faz parte do modelo conceitual.

## **Chave:**

Designa o conceito de item de busca, isto é, um atributo(simples) ou conjunto de atributos(composta) que será utilizado nas consultas à base de dados. É um conceito lógico da aplicação.

## **Índice:**

É um recurso físico que visa otimizar a recuperação de uma informação, via um método de acesso. Seu objetivo principal está relacionado com a performance de uma aplicação.

Uma chave pode ser utilizada como índice, mas um índice não é, necessariamente, uma chave.

# Modelo Relacional

## **Chave Candidata**

- Deve ser única, ou seja, nenhuma tupla de uma mesma relação pode ter o mesmo valor para o atributo escolhido como chave candidata
- Deve ser irreduzível, nenhum subconjunto da chave candidata pode ter sozinho a propriedade de ser único.
  - Pode ser :
    - . Simples : quando é composta por apenas um atributo
    - . Composta : quando possui mais de um atributo para formar a chave

## **Chave primária**

- É um caso especial da chave candidata. É a escolhida entre as candidatas para identificar unicamente uma tupla.

## **Chave estrangeira**

- É quando um atributo de uma relação é chave primária em outra.
- Constitui um conceito de vital importância no modelo relacional: é o elo de ligação lógica entre as tabelas (relacionamentos)
- Através das operações com as chaves estrangeiras que se garante a INTEGRIDADE REFERENCIAL do banco de dados:

# Modelo Relacional

## **Regras de Integridade**

- Regras que devem ser obedecidas em todos os estados válidos da base de dados (podem envolver uma ou mais linhas de uma ou mais tabelas)

## **Integridade da Entidade**

- O valor da chave não pode ser vazio
- A chave primária serve como representante na base de dados de uma entidade – se a chave primária for vazia, a linha não corresponde a nenhuma entidade

## **Integridade de Chave Primária**

- O chave primária tem que ser única

## **Integridade Referencial**

- As chaves estrangeiras tem que ser respeitadas, ou seja, se existe um determinado valor para o atributo na tabela onde ele é chave estrangeira este valor deve existir na tabela onde ele é chave primária.

## **Restrições de Integridades Semânticas**

- Todas as demais regras que devem ser obedecidas por todos os estados válidos da base de dados.

# Convertendo o Diagrama ER para Tabelas Relacionais

## **Para cardinalidade 1:1**

Incluir todos os atributos numa tabela simples. O nome da tabela relacional pode ser o nome de uma das entidades que participam do relacionamento, um nome composto formado pela combinação dos nomes das duas entidades ou um novo nome que represente o significado dos dados na tabela.

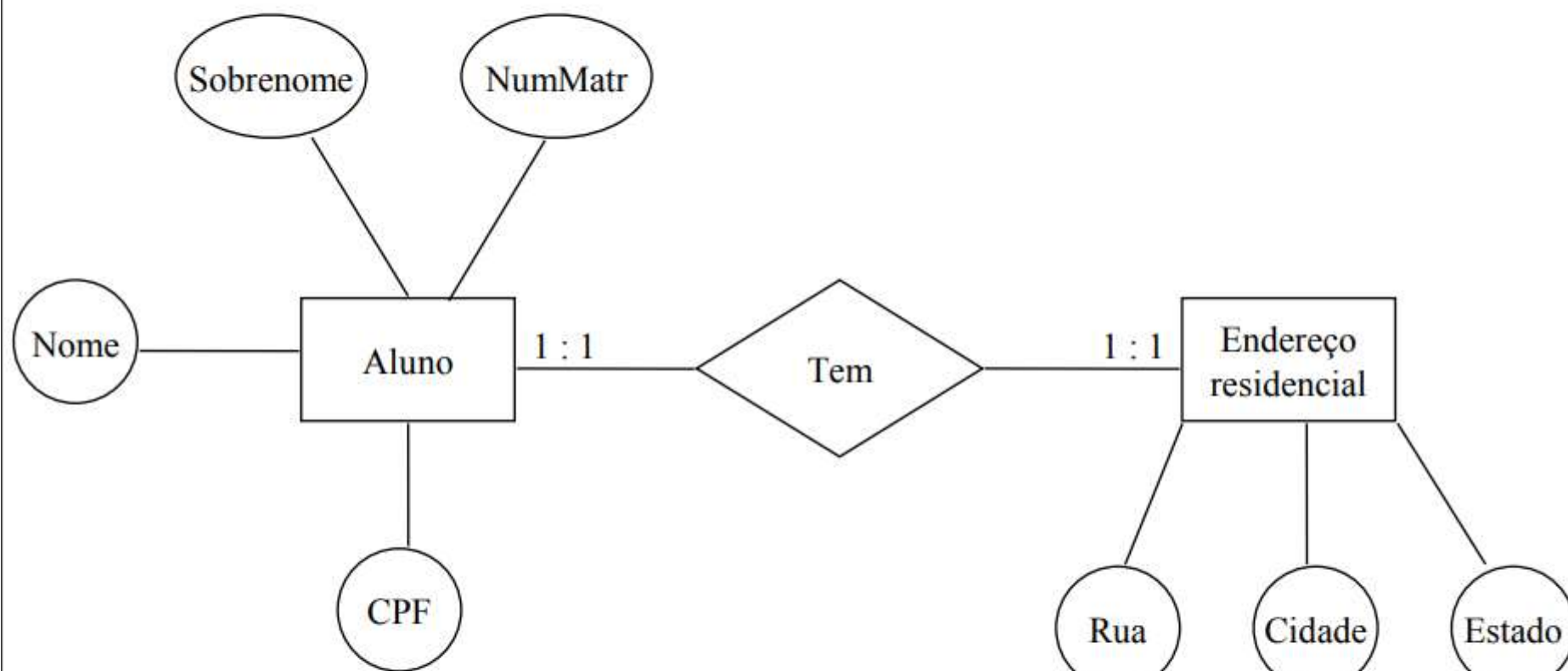
## **Para cardinalidade 1:N**

Incluir o “identificador” do lado “um” como um atributo no lado “muitos”. O identificador colocado do lado “muitos” é chamado de chave estrangeira.

## **Para cardinalidade N:M**

Criar uma nova tabela e colocar as chaves primárias de cada uma das entidades como atributos na nova tabela. A nova tabela é chamada de tabela associativa. O identificador da tabela é uma chave composta formada pelas chaves primárias das duas tabelas que participam do relacionamento. Cada identificador colocado na nova tabela é uma chave estrangeira.

# Convertendo o Diagrama ER para Tabelas Relacionais

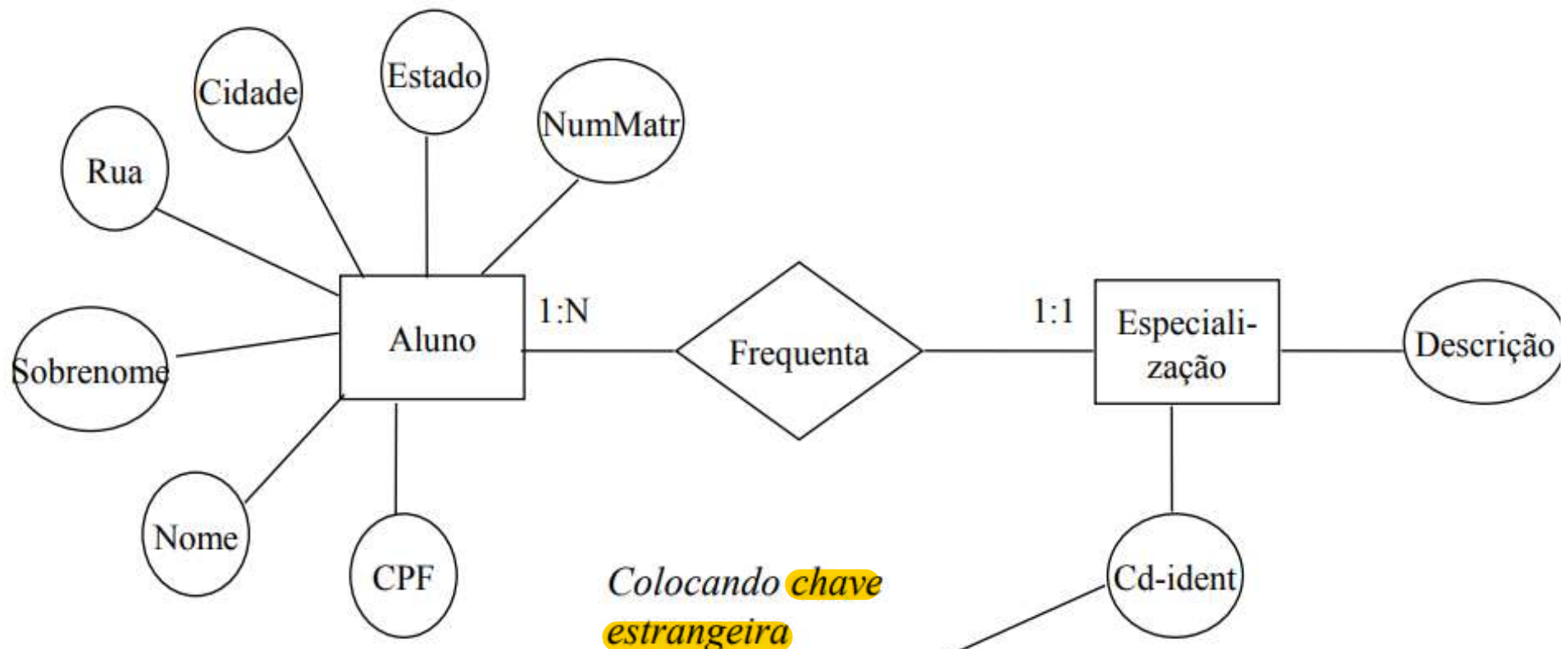


*Colocando Atributos*

**Aluno (NumMatr, CPF, Nome, Sobrenome, Rua, Cidade , Estado)**



# Convertendo o Diagrama ER para Tabelas Relacionais

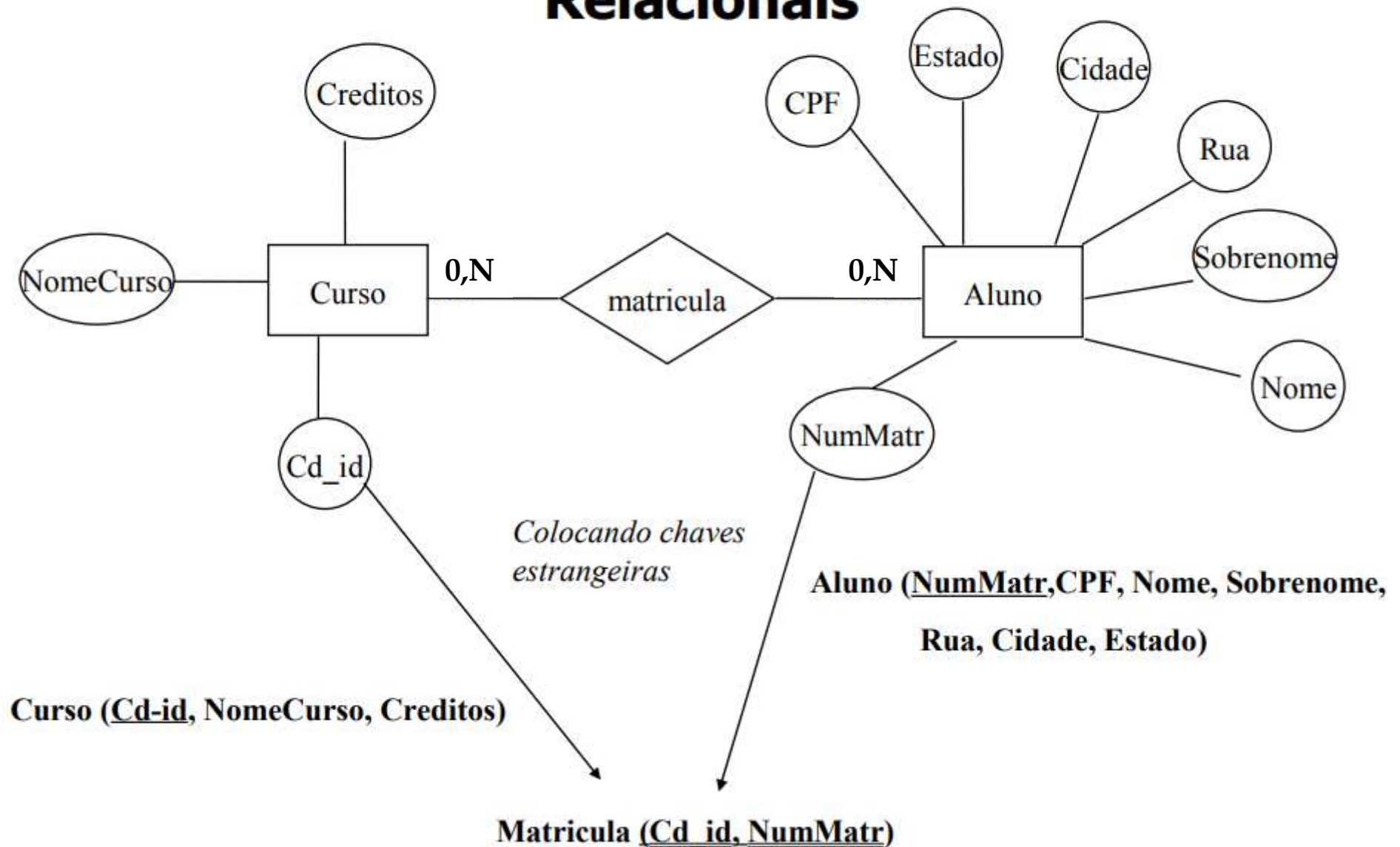


**Aluno** (NumMatr, CPF, Cd-ident, Nome, Sobrenome, Rua, Cidade, Estado)

**Especialização** (Cd-ident, Descrição)

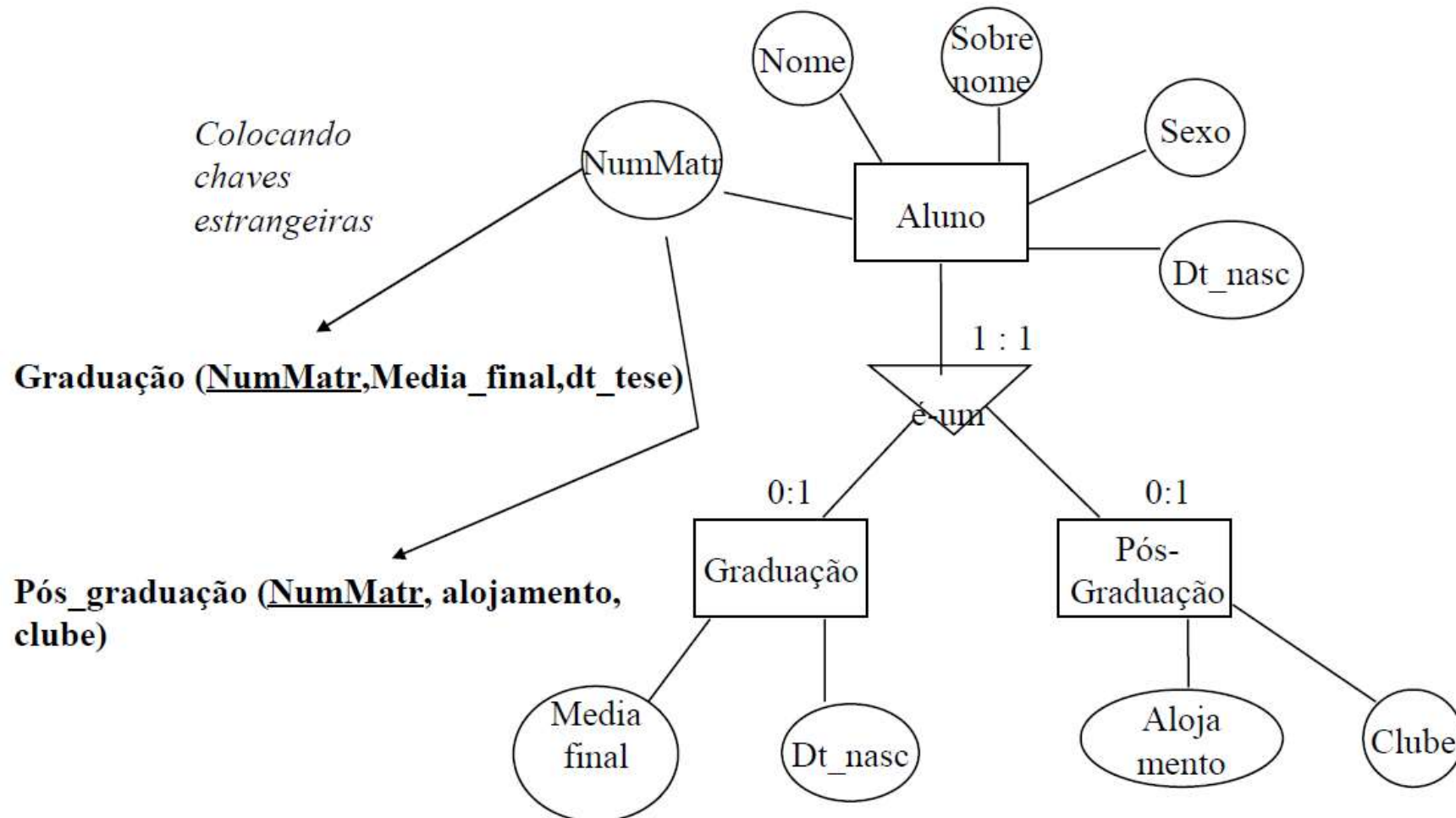


# Convertendo o Diagrama ER para Tabelas Relacionais

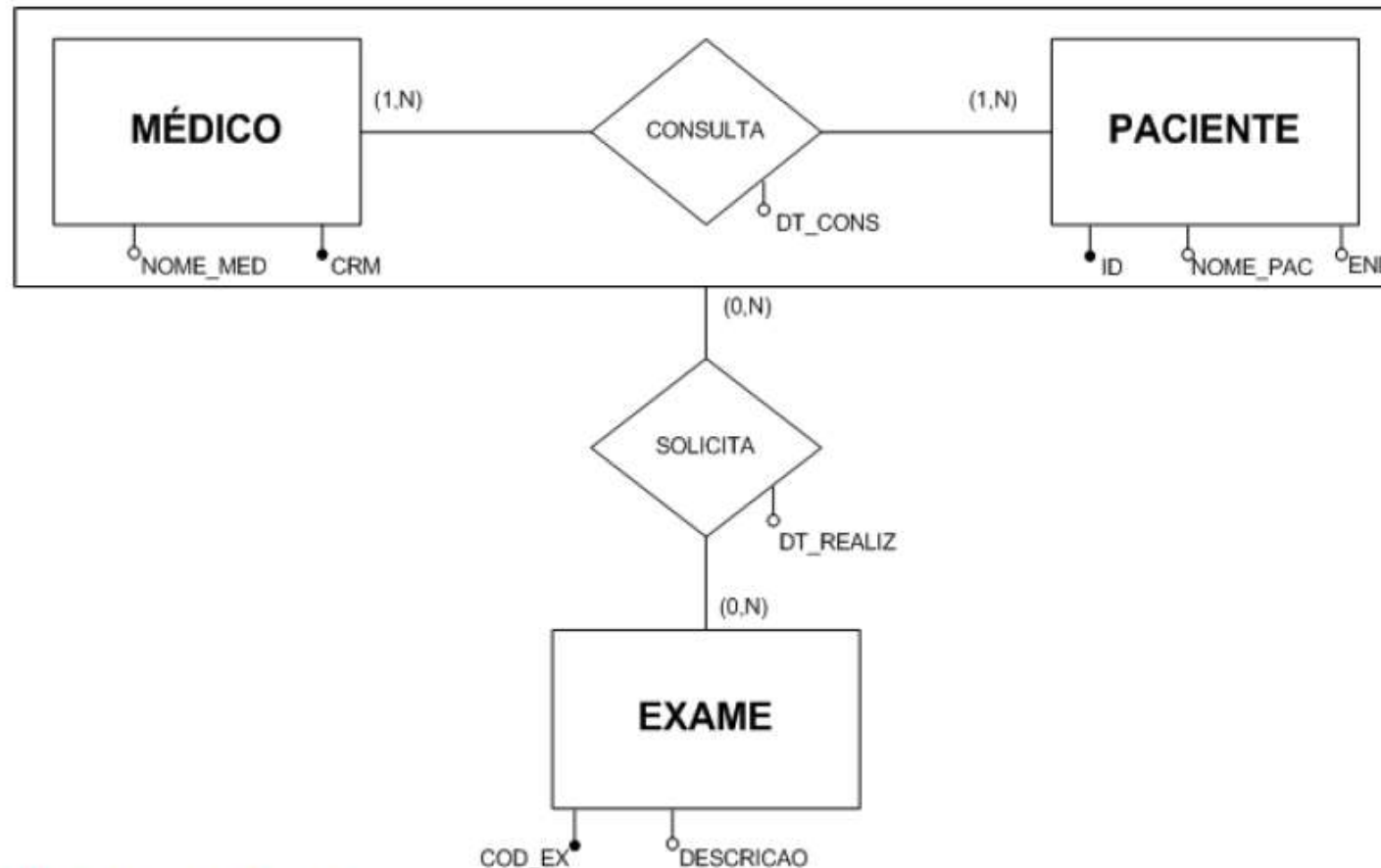


# Convertendo o Diagrama ER para Tabelas Relacionais - Generalizações

**Aluno** (NumMatr, Nome, Sobre nome, Sexo, Dt\_nasc)



# Convertendo o Diagrama ER para Tabelas Relacionais - Agregações



**Medico**(CRM, Nome\_Med)

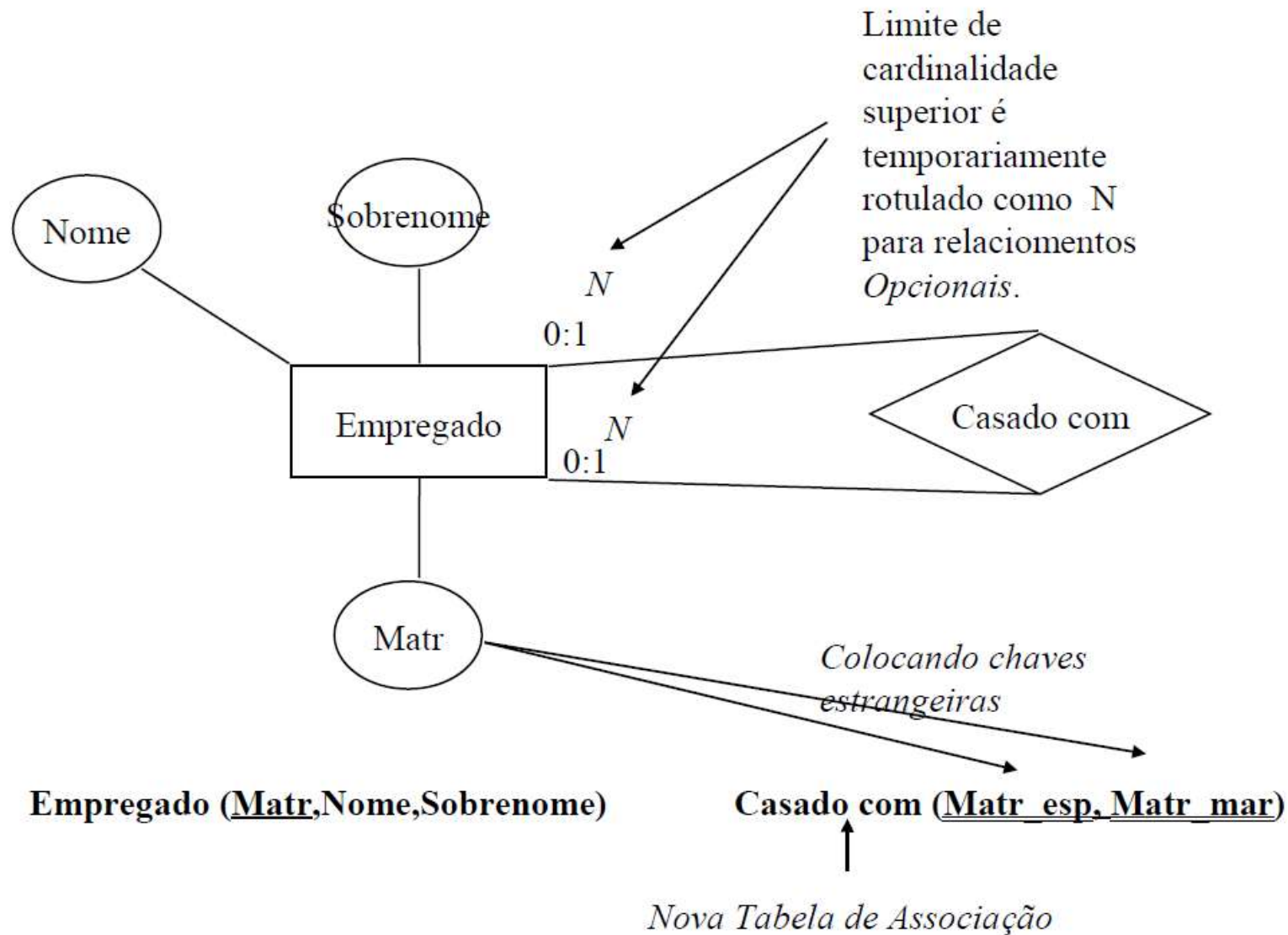
**Paciente** (ID, Nome\_Pac, End)

**Consulta**(CRM, ID, Data\_Cons)

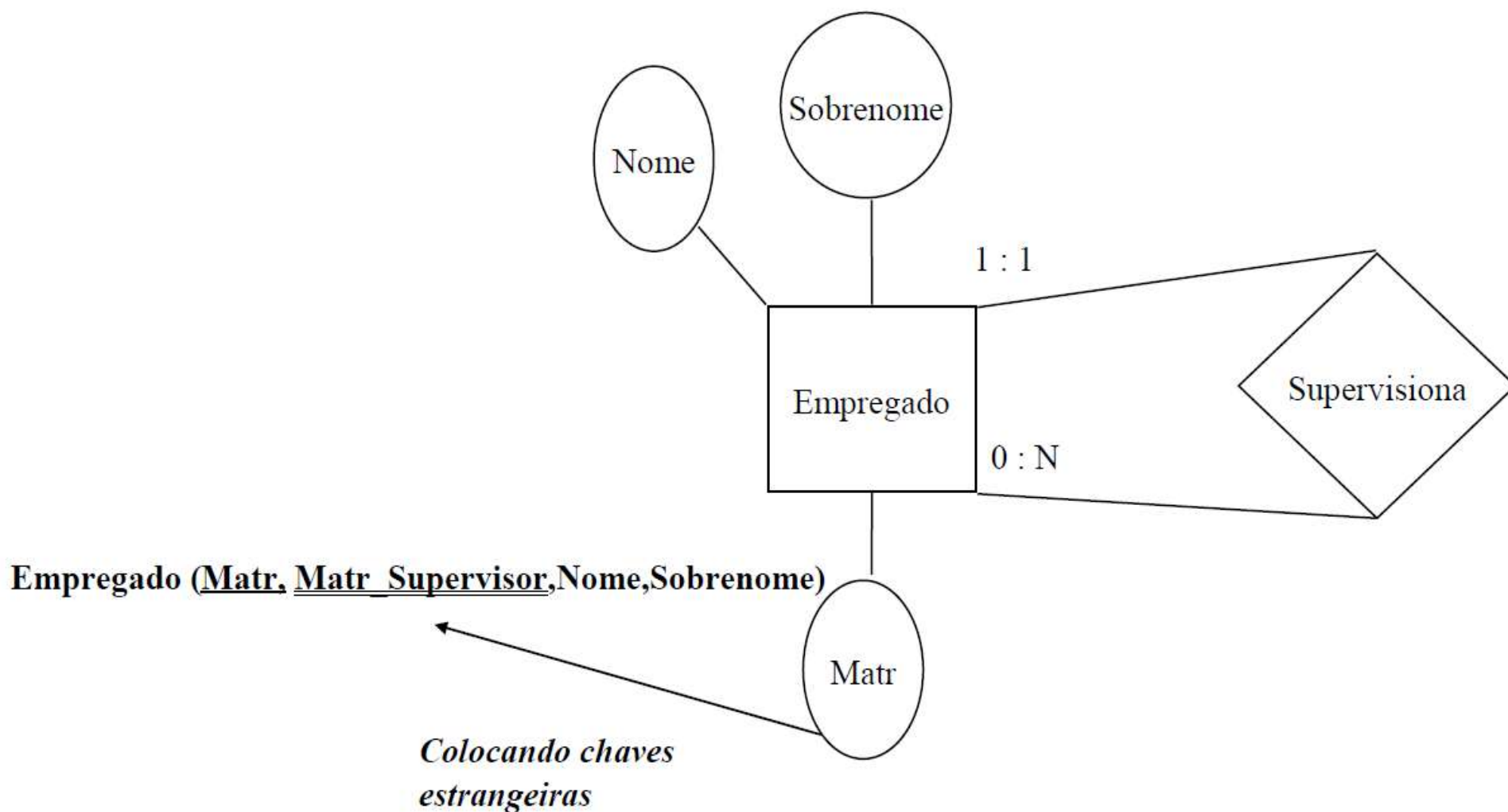
**Exame** (Cod\_Ex, Descricao)

**Solicita** (CRM, ID, Cod\_Ex, Dt\_Realiz)

# Convertendo o Diagrama ER para Tabelas Relacionais – Relacionamentos Recursivos

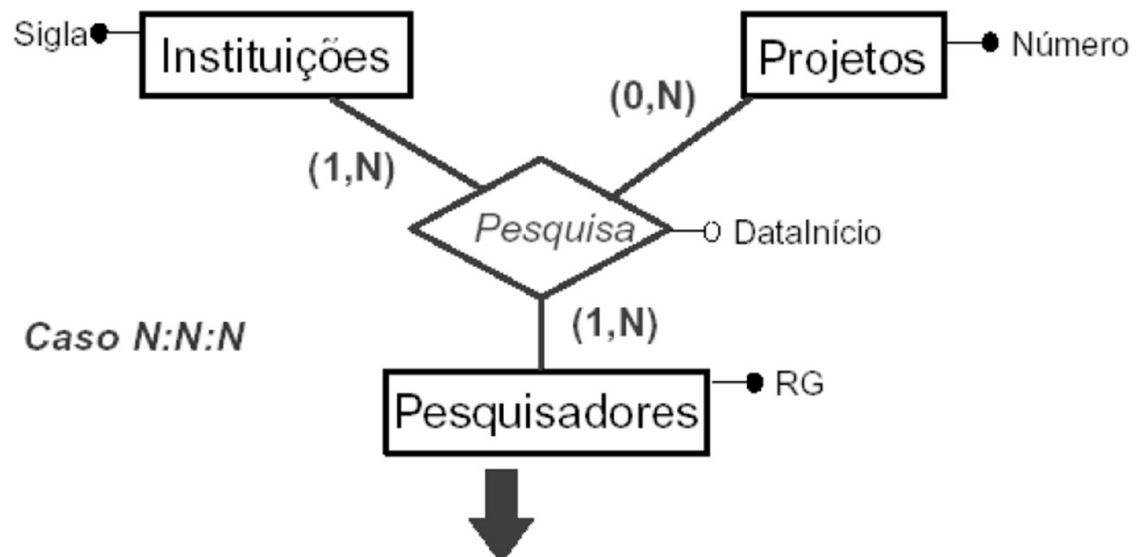


# Convertendo o Diagrama ER para Tabelas Relacionais – Relacionamentos Recursivos





## Convertendo o Diagrama ER para Tabelas Relacionais – Relacionamentos n-ários



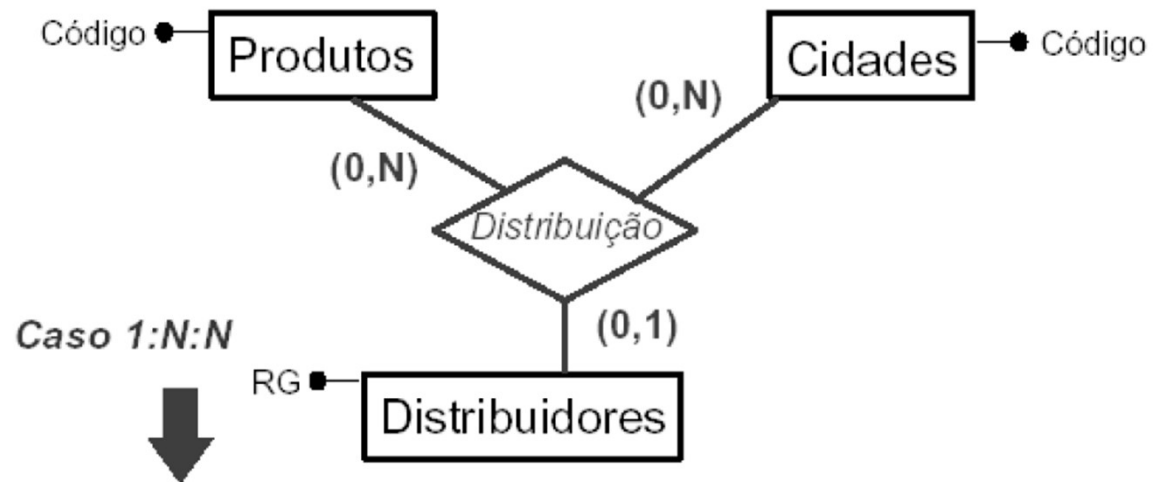
Instituições (Sigla, ...)

Projetos (Numero, ...)

Pesquisadores (RG, ...)

Pesquisa (Sigla, Numero, RG, DataInicio)

## Convertendo o Diagrama ER para Tabelas Relacionais – Relacionamentos n-ários



Produtos (Código, ...)

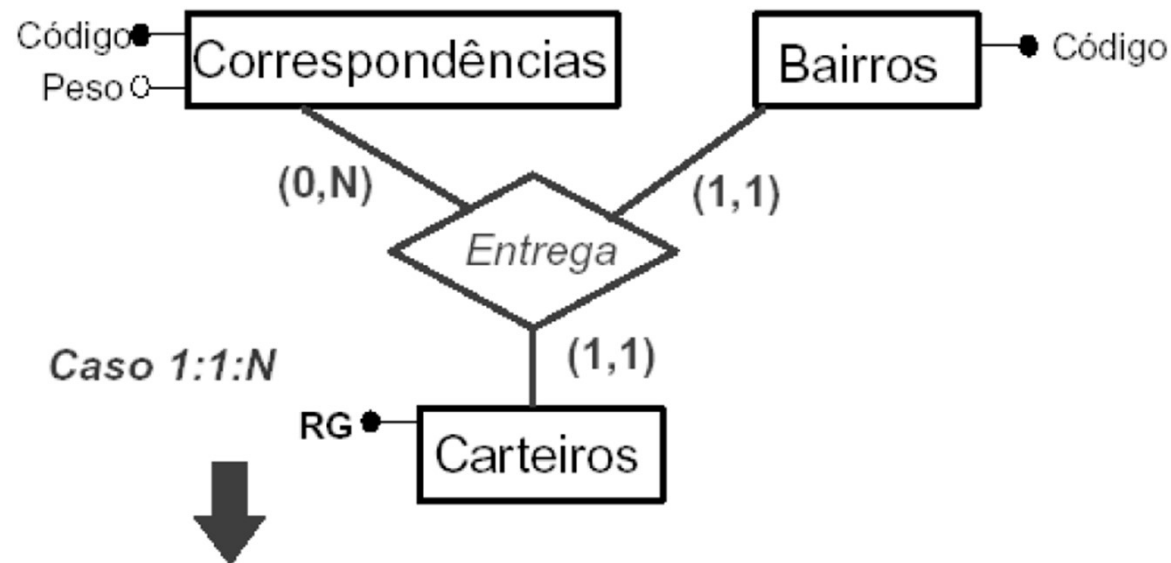
Cidades (Código, ...)

Distribuidores (RG, ...)

Distribuição (CodProduto, CodCidade, RG)



## Convertendo o Diagrama ER para Tabelas Relacionais – Relacionamentos n-ários



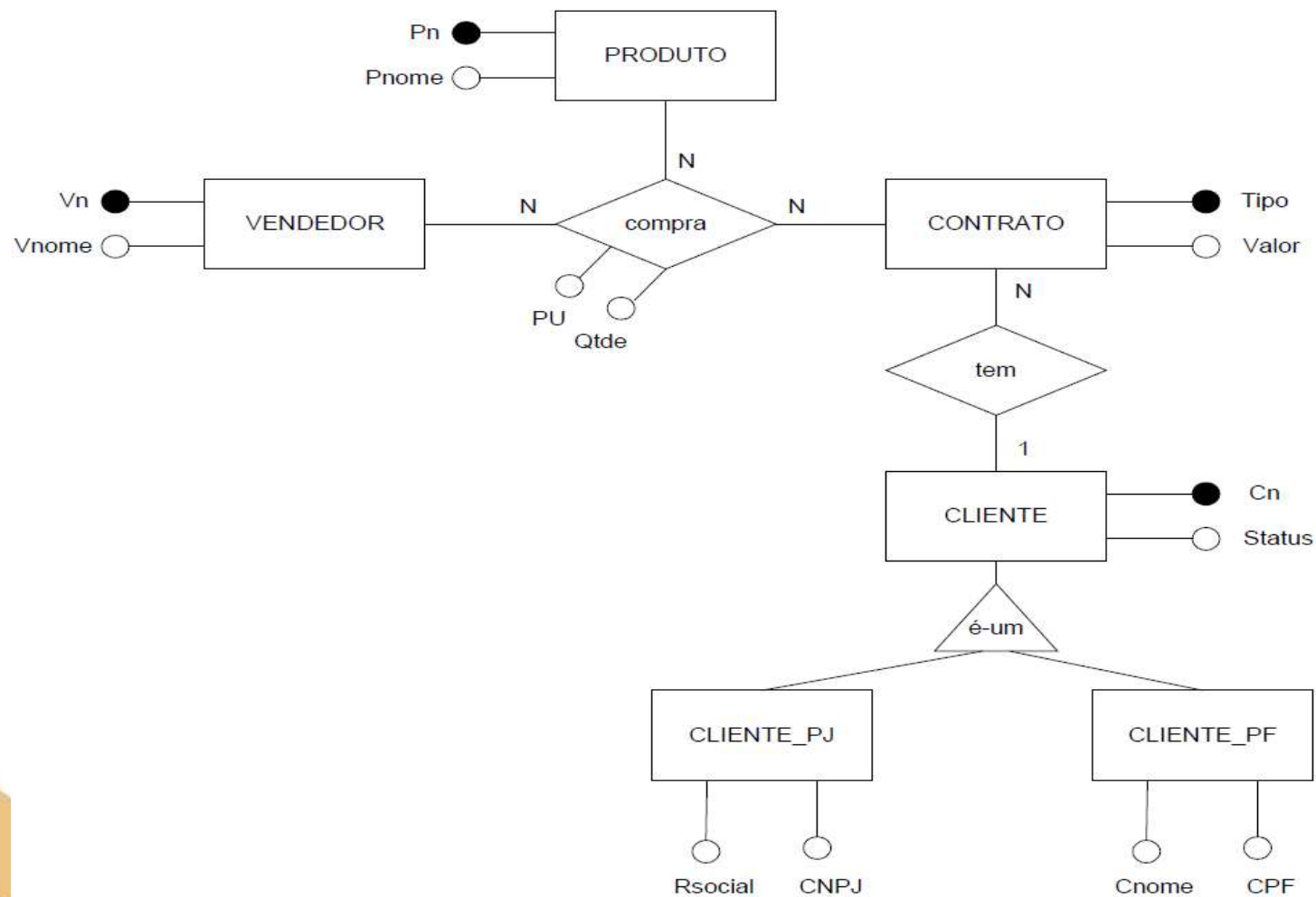
Bairros (Código, ...)

Carteiros (RG, ...)

Correspondências (CodCarta, *Peso*, CodBairro, RG, ...)

# Atividade 1

Converter o modelo conceitual abaixo em tabelas do modelo lógico relacional.



## **Atividade 2**

**Construa as tabelas relacionais do modelo lógico das atividades 1 de 2 das aulas anteriores.**