

Banco de Dados

Prof. Dr. Ronaldo Celso Messias Correia

ronaldo.correia@unesp.br

Modelo Relacional

Modelo Relacional

- Foi introduzido por Codd (1970)
- Começou a ser realmente utilizado nas empresas a partir de 1987
- Tornou-se um padrão de fato para aplicações comerciais, devido a sua simplicidade e performance. Padrão este que ainda persiste até hoje
- É um modelo formal, baseado na teoria matemática das relações. Fortemente fundamentada na Álgebra Relacional e no Cálculo Relacional
- Um dos SGBD's precursores que implementaram este modelo foi o System R (IBM). Baseado em seus conceitos surgiram: DB2 (IBM), SQL-DS (IBM), Oracle, Informix, Ingres, Sybase entre outros

Modelo Relacional

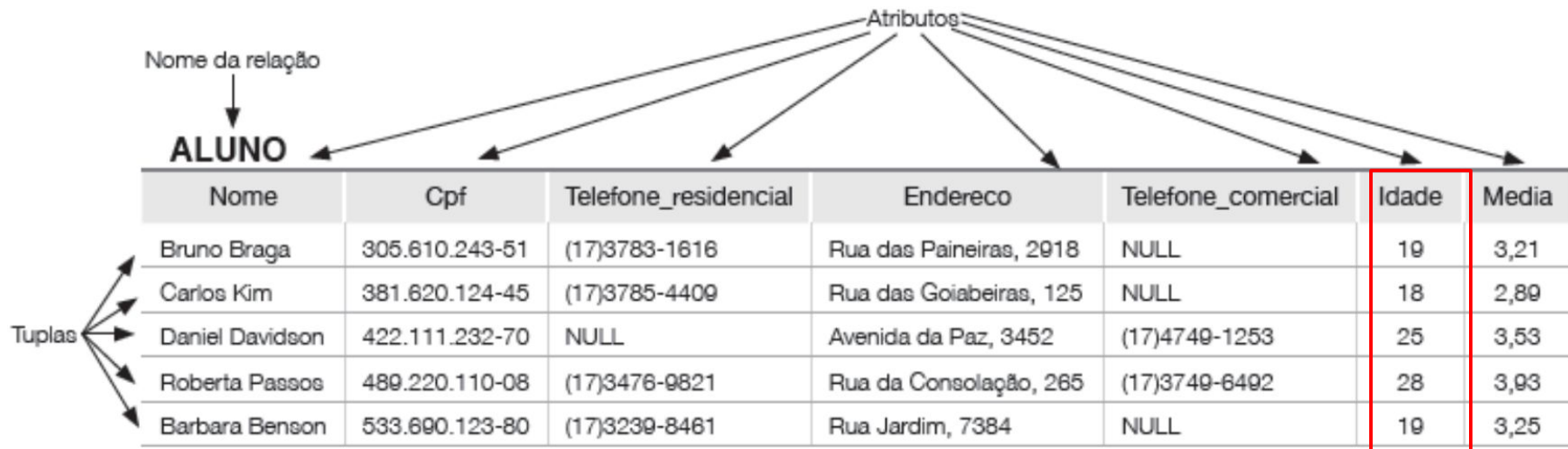
- O modelo relacional representa os dados num banco de dados como uma coleção de tabelas (relações)
- Cada tabela terá um nome, que será único, e um conjunto de atributos com seus respectivos nomes e domínios
- Todos os valores de uma coluna são do mesmo tipo de dados
- Terminologia:
 - Tabela é chamada de Relação
 - Linha é chamada de Tupla
 - Coluna é chamada de atributo

Nome	CPF	Rua	Cidade	Nr_Conta
José	015425446	Rua das Flores	São Paulo	5418-7
Maria	154879984	Rua Linda	Bauru	4876-9
José	015425446	Rua das Flores	São Paulo	8745-6

Nr_Conta	Saldo
5418-7	541,20
4876-9	145,00
8745-6	235,90

Modelo Relacional

➤ Exemplo de uma relação (tabela):



Atributo chave: ?

- CPF
- RA

Data_nascimento

Domínios

- Modelo relacional considera sempre que os valores são indivisíveis – atômicos
- Um Domínio é um conjunto de valores atômicos
- Exemplos de domínio
 - Nomes de Alunos
 - Inválido: M@ria ?
 - Códigos de Disciplinas
 - Idade: inteiro entre 15 e 70

- Especificação do Domínio
 - Nome
 - Definição lógica
 - Nomes de Alunos: conjunto de todos os nomes possíveis para pessoas
 - Códigos de Disciplinas: conjunto dos códigos das disciplinas oferecidas
 - Tipo de dado e/ou formato de dado
 - Nomes de Alunos – string de 60 caracteres
 - Códigos de Disciplinas – string com três letras seguidas de quatro dígitos:
MAT8850

Esquema de Relações

- Um esquema de relação R é denotado por $R(A_1, A_2, \dots, A_n)$, onde R representa um conjunto de atributos: $R = \{A_1, A_2, \dots, A_n\}$
- O **grau** de uma relação é o número de atributos que seu esquema contém
- $\text{Dom}(A_i)$ - Domínio do Atributo A_i
- Exemplo
 - Uma relação de Alunos que tenha os atributos Nome do aluno, RG, Data de Nascimento e E-mail tem o seguinte esquema:
 $\text{Aluno}(\text{Nome}, \text{RG}, \text{Data_nascimento}, \text{E-mail})$
 $\text{Alunos}(\text{Nome}, \text{RG}, \text{Data_nascimento}, \text{E-mail})$

Esquema de Relações

- Um esquema de relação R é denotado por $R(A_1, A_2, \dots, A_n)$, onde R representa um conjunto de atributos: $R = \{A_1, A_2, \dots, A_n\}$
- O **grau** de uma relação é o número de atributos que seu esquema contém
- $\text{Dom}(A_i)$ - Domínio do Atributo A_i
- Exemplo
 - Uma relação de Clientes que tenha os atributos Nome do Cliente, RG, Data de Nascimento e E-mail, tem o seguinte esquema:
 $\text{Cliente}(\text{Nome}, \text{RG}, \text{Data_nascimento}, \text{E-mail})$
 $\text{Clientes}(\text{Nome}, \text{RG}, \text{Data_nascimento}, \text{E-mail})$

Esquemas e Especificação dos domínios

- Especificação dos domínios:
 - Nomes do Cliente: conjunto de todos os nomes possíveis para pessoas – strings de 60 caracteres
 - RG: conjunto dos RGs válidos no Brasil – números de 9 dígitos
 - Data_nascimento: conjunto de datas de nascimento possíveis para clientes
- Esquema da relação Cliente:
 - Cliente={Nome, RG, Data_nascimento}
- Domínios dos atributos de Cliente
 - Dom(Nome) = Nomes dos Clientes
 - Dom(RG) = RG
 - Dom(Data_nascimento) = Data de Nascimento

Relações

- Relação r - Instância do Esquema de Relação $R(A_1, A_2, \dots, A_n)$
- Um instante (snapshot) de relação r , do esquema $R(A_1, A_2, \dots, A_n)$, denotado por $r(R)$, é o conjunto de n -tuplas $r = \{t_1, t_2, \dots, t_n\}$. Cada tupla t é uma lista ordenada de valores $t = \langle v_1, v_2, \dots, v_n \rangle$
- Uma instância $r(R)$ é um subconjunto do produto cartesiano dos domínios de R
- $r(R) \subset (\text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n))$
- Exemplo:
- Esquema da Relação Cliente:
 $\text{Cliente} = \{\text{Nome}, \text{RG}, \text{Data_Nascimento}, \text{E-mail}\}$
- Possível relação:
 $r(\text{Cliente}) = \{ \langle \text{Carlos}, 222345, 19/02/1978, \text{carlos@gmail.com} \rangle, \langle \text{Antonio}, 672561, 13/04/1985, \text{antonio@hotmail.com} \rangle, \langle \text{José}, 37321, 06/06/1976, \text{jose123@gmail.com} \rangle \}$

Aspectos Importantes das Relações

- A ordem das tuplas e dos atributos da relação não tem importância
 - Matematicamente não existe a idéia de ordem em conjuntos
 - OBS: na implementação de um SGBDR existe uma ordem física de armazenamento das tuplas, determinando uma ordem na recuperação das informações
- Todo atributo possui valor atômico
- Cada atributo numa relação tem um nome que é único dentro da relação
- Todas as tuplas devem ser únicas (conjunto)
- A fundamentação matemática está sempre presente

Restrições das Relações

- A ordem das tuplas e dos atributos da relação não tem importância
- **Restrição de domínio:** o valor de cada atributo A deve ser um valor atômico pertencente a $\text{Dom}(A)$
- **Restrição de unicidade:** deve ser possível identificar univocamente cada tupla da relação
 - para garantir esta propriedade: especifica-se uma Restrição de Unicidade
 - definição de superchaves e chaves
- **Restrição de vazio para atributo:** determina quando o valor especial null é ou não permitido para um atributo

Conceitos de Chaves

- Chave designa o conceito de item de busca, ou seja, um dado que será empregado nas consultas à base de dados. É um conceito lógico da aplicação
- Superchave
 - Conjunto de um ou mais atributos que, tomados coletivamente nos permite identificar de maneira unívoca uma tupla em um conjunto de tuplas
 - Nome e CPF; RA e NOME; CPF e RA
 - Uma superchave pode ter atributos redundantes
- Chave
 - É uma superchave da qual não se pode retirar nenhum atributo e ainda preservar a propriedade de identificação unívoca (superchave mínima)
- Chave Candidata
 - É uma superchave para qual nenhum subconjunto possa ser uma superchave.
- Chave primária
 - É a chave candidata que é escolhida pelo projetista para identificar tuplas dentro de um conjunto de tuplas

Conceitos de Chaves e Índice

- Convenciona-se sublinhar os atributos que compõem a chave primária. Ex.:Empregado (Matrícula, Nome, Endereço, Função,Salário)
- Um mesmo atributo pode ter nomes diferentes nas diversas relações em que participa. Ex.: Empregado (Matrícula, Nome, Endereço, Função, Salário, Dep) e Departamento(CodDepart, Nome, Endereço)
- Atributos que representam diferentes conceitos podem ter o mesmo nome
- Índice é um recurso físico visando otimizar a recuperação de uma informação, via um método de acesso. Seu objetivo principal está relacionado com a performance de um sistema
- Uma chave pode ser utilizada como índice, mas um índice não é necessariamente uma chave
- A forma de criação do índice depende do ambiente relacional

Restrições de Integridade

- Regras a respeito dos valores que podem ser armazenados nas relações
 - objetivo: garantir consistência
 - quando definidas, devem ser sempre satisfeitas na base de dados
- **Unicidade da Chave:**
 - Toda tupla tem um conjunto de atributos que a identifica de maneira única na relação
- **Integridade de Entidade**
 - Nenhum valor de chave primária poderá ser NULO
 - Se a chave for composta por mais de um atributo, nenhum deles pode ser nulo
- **Integridade Referencial**
 - Uma relação pode ter um conjunto de atributos que contém valores com mesmo domínio de um conjunto de atributos que forma a chave primária de uma outra relação. Este conjunto é chamado chave estrangeira
 - Utilizada para manter consistência entre tuplas de duas relações
 - Não pode existir na chave estrangeira um valor que não exista na tabela na qual ela é chave primária
 - Define que se uma tupla t1 em uma relação R1 faz referência a uma relação R2, então t1 deve fazer referência a uma tupla existente em R2

Restrições de Integridade Referencial

- Um conjunto de atributos FK do esquema da relação R1 é uma Chave Estrangeira de R1, que faz referência à relação R2, se ele satisfizer as duas regras seguintes:
 - Os atributos de FK é compatível em domínio com a chave primária PK de R2
 - O valor dos atributos FK numa tupla t_i qualquer da relação R1: ou é igual ao valor do atributos PK de alguma tupla t_j da relação R2 ($t_i[FK] = t_j[PK]$), ou é nulo ($t_i[FK] = \text{null}$)

- **Exemplo :**

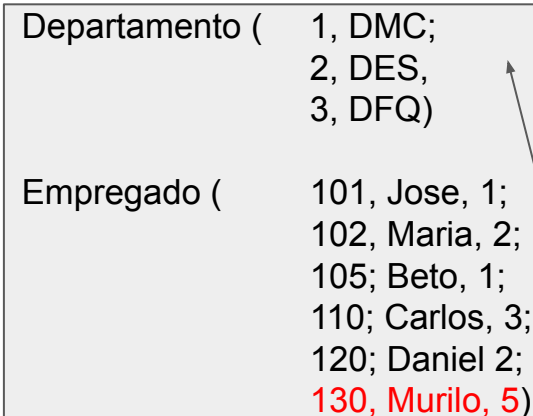
- Indicação da chave estrangeira esquema da relação:

Departamento = {CodigoD (PK), NomeD}

Empregado = {CodigoE (PK), NomeE, Departamento (FK)}

=====

Departamento (1, DMC;
	2, DES,
	3, DFQ)
Empregado (101, Jose, 1;
	102, Maria, 2;
	105; Beto, 1;
	110; Carlos, 3;
	120; Daniel 2;
	130, Murilo, 5)



- **Integridade Semântica:** Define aspectos comportamentais do BD. São especificadas e impostas dentro dos programas de aplicação.

Exemplo: O salário de um empregado não deveria exceder do supervisor do empregado

As 12 Regras de Codd

- Codd, ao definir o modelo relacional, estabeleceu um conjunto de 12 regras para a determinação de um banco de dados ser realmente relacional
- As regras são:
 - Toda informação num banco de dados relacional é apresentada a nível lógico por valores em tabelas
 - Todo dado em um banco de dados relacional tem a garantia de ser logicamente acessível, recorrendo-se a uma combinação do nome da tabela, um valor de chave e o nome da coluna
 - Tratamento sistemático de valores nulos (ausência de dado)
 - O dicionário de dados (catálogo) relacional ativo é baseado no modelo relacional
 - O SGBD relacional deve ter uma linguagem para definição, detalhamento e manipulação de dados
 - Tratamento das atualizações de visões dos dados
 - Tratamento de alto nível para inserção, atualização e eliminação de dados
 - Independência dos dados físicos (mudança na memória e no método de acesso)
 - Independência dos dados lógicos (mudanças de qualquer tipo nas tabelas básicas, ex: divisão de uma tabela por linha ou coluna)
 - Independência das restrições de integridade
 - Independência de distribuição
 - Não subversão das regras de integridade ou restrições quando se utiliza uma linguagem de baixo nível
- Raros são os bancos de dados que se enquadram em mais do que 10 destas regras

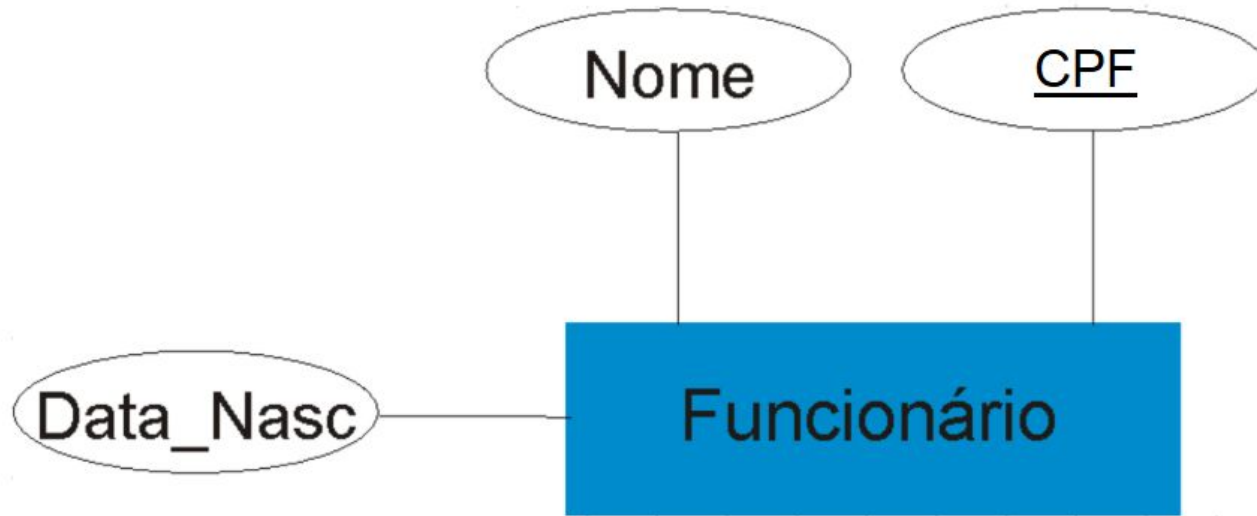
Mapeamento Modelo E-R para Modelo Relacional

- MER - modelo conceitual:
 - pode ser usado para especificar conceitualmente a estrutura dos dados de uma aplicação
- **Modelo Relacional - modelo lógico:**
 - pode ser usado para suportar a implementação de Aplicações
 - é necessário que exista um SGBD que se apoie no modelo relacional: Um SGBDR
- Mapeamento: permite que se traduzam esquemas concebidos com um modelo de conteúdo semântico mais alto para um esquema utilizando um modelo lógico, preservando as propriedades do modelo semântico (mais rico)

Mapeamento Modelo E-R para Modelo Relacional

- Para cada modelo conceitual E-R pode existir vários modelos Relacionais
- A maioria das ferramentas de modelagem conceitual automatizam o mapeamento. Porém é importante conhecer as etapas deste mapeamento
- A definição equivocada do modelo Relacional afeta a estrutura de todo o projeto
- Muitas vezes a maneira como implementar as tabelas no modelo relacional dependem de decisões de projeto e não de regras pré-estabelecidas

Um Simples Exemplo de Mapeamento

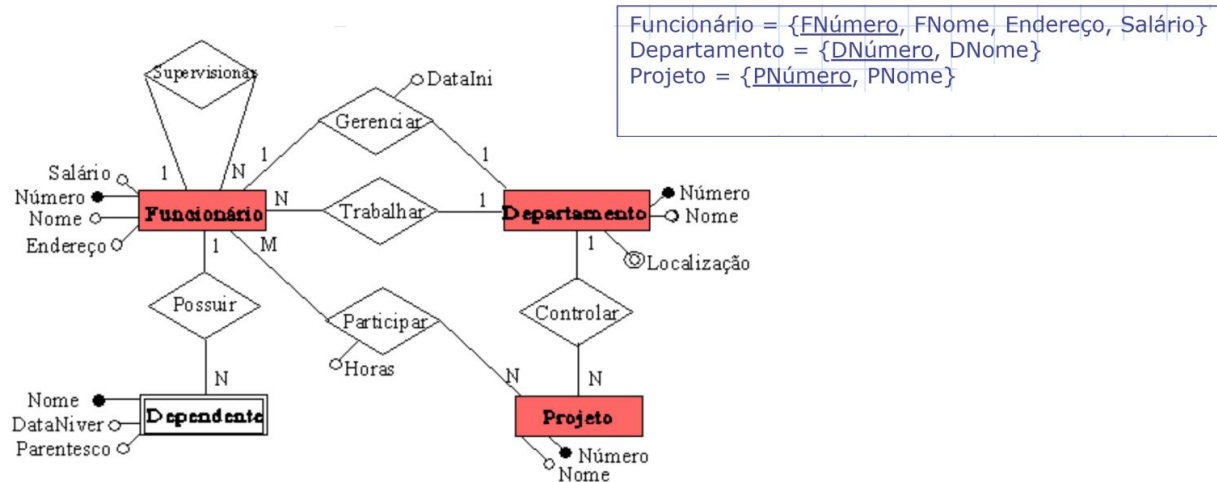


Funcionário(CPF (PK), Nome, Data_Nasc)

Etapas do Mapeamento

➤ Etapa 1:

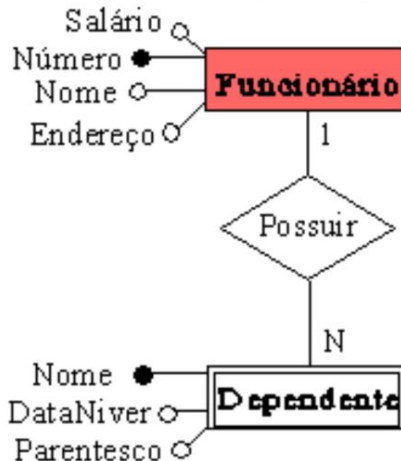
- Para cada entidade **E** no modelo ER é criada uma tabela (Relação) **T1** no Modelo Relacional que inclua todos os atributos simples de **E**
- Para cada atributo composto, são inseridos apenas os componentes simples de cada um
- Um dos atributos chaves de **E** deve ser escolhida como a chave primária de **T1**



Etapas do Mapeamento

➤ Etapa 2:

- Para cada entidade fraca **EF** com entidade proprietária **E** no modelo ER, é criada uma tabela **T1** no Modelo Relacional incluindo todos os atributos simples de **EF**
- Para cada atributo composto, são inseridos apenas os componentes simples de cada um
- A chave primária desta relação **T1** será composta pela chave parcial da entidade fraca **EF** mais a chave primária da entidade proprietária **E**



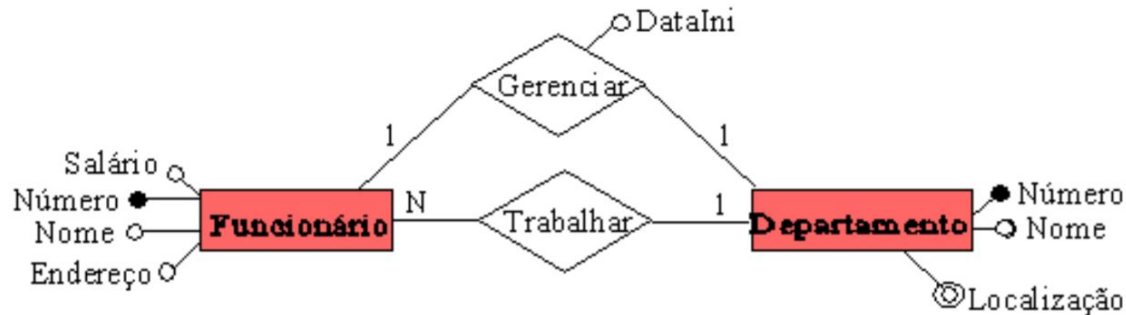
Dependente = {DependNome, **FNúmero**, DataNiver, Parentesco}

dependente ((dependnome, fnúmero(fk)))(pk)

Etapas do Mapeamento

➤ Etapa 3:

- Para cada relacionamento regular com cardinalidade 1:1 entre entidades **E1** e **E2** que geraram as tabelas **T1** e **T2** respectivamente, devemos escolher a chave primária de uma das relações (**T1**, **T2**) e inseri-la como chave estrangeira na outra relação;
- Se um dos lados do relacionamento tiver participação total e outro parcial, então é interessante que a chave do lado com participação **parcial** seja inserido como chave estrangeira no lado que tem participação **total**;

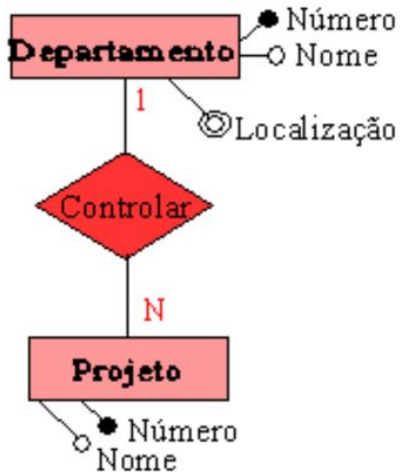


Funcionário = {FNúmero, FNome, Endereço, Salário}
Departamento = {DNúmero, Dnome, **FNúmero**, DataIni}

Etapas do Mapeamento

➤ Etapa 4:

- Para cada relacionamento regular com cardinalidade 1:N entre entidades **E1** e **E2** respectivamente e que geraram as tabelas **T1** e **T2** respectivamente, deve-se inserir a chave primária de **T1** como chave estrangeira em **T2**

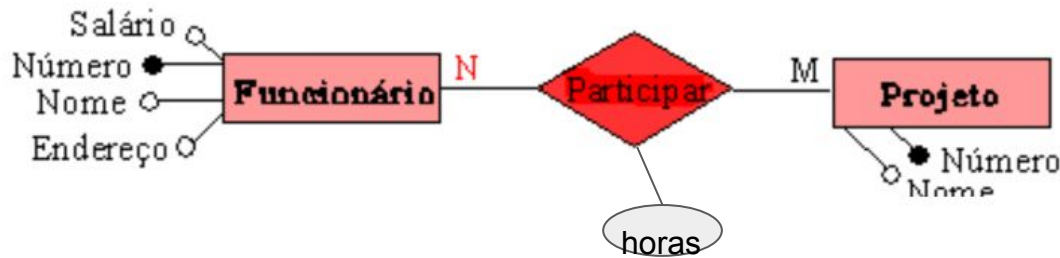


```
Departamento = {DNúmero, Dnome, FNúmero, DataIni}  
Projeto = {PNúmero, Pnome, DNro}
```

Etapas do Mapeamento

➤ Etapa 5:

- Para cada relacionamento regular com cardinalidade N:N entre entidades **E1** e **E2**, cria-se uma nova tabela **T1**, contendo todos os atributos do relacionamento mais o atributo chave de **E1** e o atributo chave de **E2**
- A chave primária de **T1** será composta pelos atributos chave de **E1** e **E2**;



Participar = {FNum, PNum, horas}

Participar ((Fnum (FK), PNum (FK)) (PK), horas)
PK (FNum, Pnum);
FK (FNum) ;FK (PNum)

Funcionário (
1, Jose, Rua A;
2, maria, Rua B,
5, Antonio, Rua D)

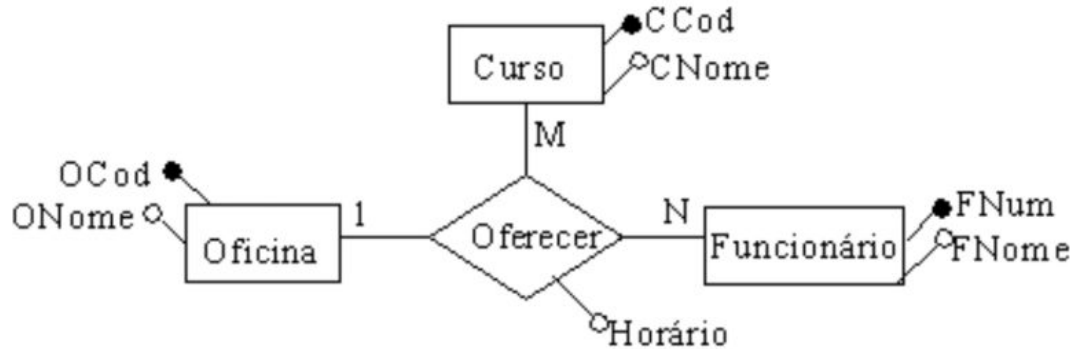
Projeto (
101, Cidades Inteligentes;
102, Arduino;
103, Mineração de Dados)

Participar (
1, 101, 10;
1, 103, 30;
5, 102, 20;
5, 101, 20)

Etapas do Mapeamento

➤ Etapa 6:

- Para cada relacionamento n-ário, $n > 2$, cria-se uma tabela **T1**, contendo todos os atributos do relacionamento
- A chave primária de **T1** será composta pelos atributos chaves das entidades participantes do relacionamento



Oferecer = {OCod, CCod, FNum, Horário}

Etapas do Mapeamento

➤ Etapa 7:

- Existem duas maneiras de tratar atributos multivalorados no mapeamento:
 - Sabendo uma estimativa do número de ocorrências do atributo. Assim, pode-se adicionar à relação quantos atributos forem necessários
 - Caso do número de ocorrências do atributo seja indefinido, cria-se uma nova relação.



LocalDep = {DNúmero, Localização}

LocalDep(

1, Bloco 1,
1, Bloco 2,
1, Bloco 3;
2; Bloco 4)

Departamento = {DNúmero, Dnome, FNúmero, DataIni,
local1, local2, local3}

Departamento (

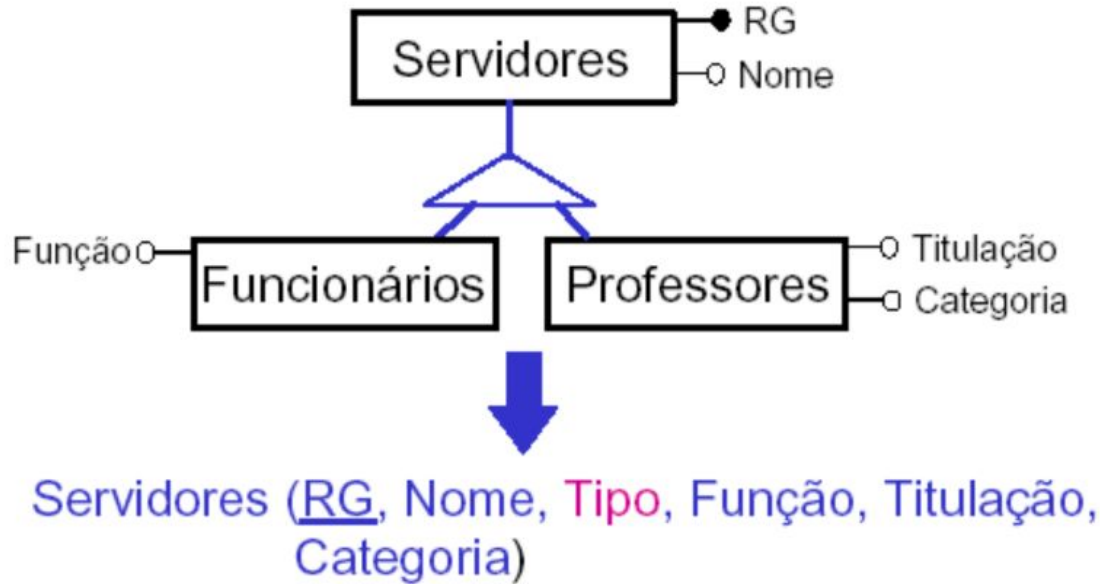
1, DMC, Bloco 1, Bloco 2, Bloco 3;
2; DFQ; Bloco 4, nulo, nulo)

Etapas do Mapeamento

- Etapa 8: Mapeamento de Especializações
 - Três alternativas para mapeamento
 - Tabela única para entidade genérica e suas especializações
 - Tabelas para a entidade genérica e as entidades especializadas
 - Tabelas apenas para as entidades especializadas

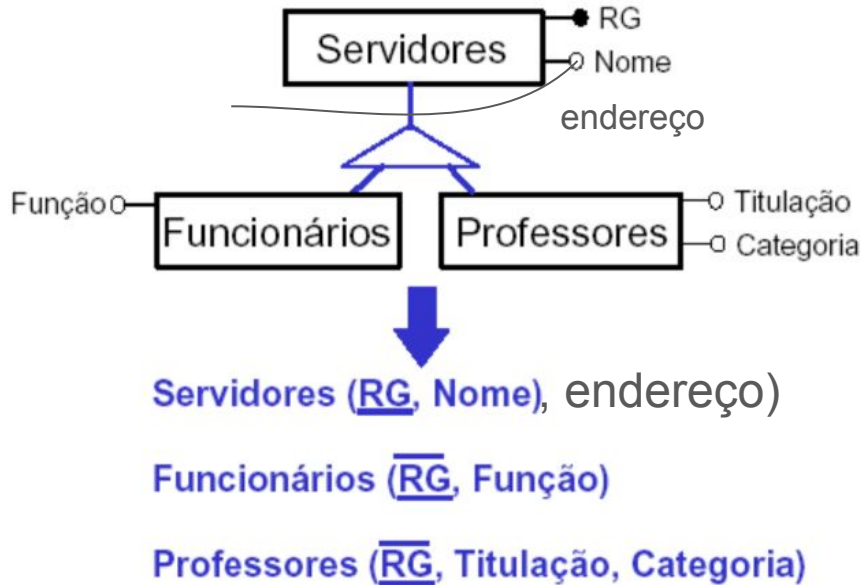
Etapas do Mapeamento

- Alternativa 1:
 - Tabela única para entidade genérica e suas especializações



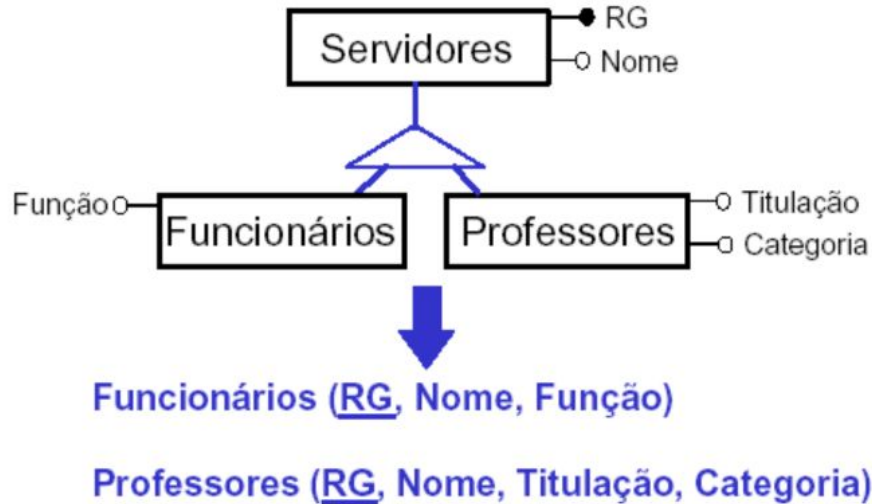
Etapas do Mapeamento

- Alternativa 2:
 - Tabelas para a entidade genérica e as entidades especializadas

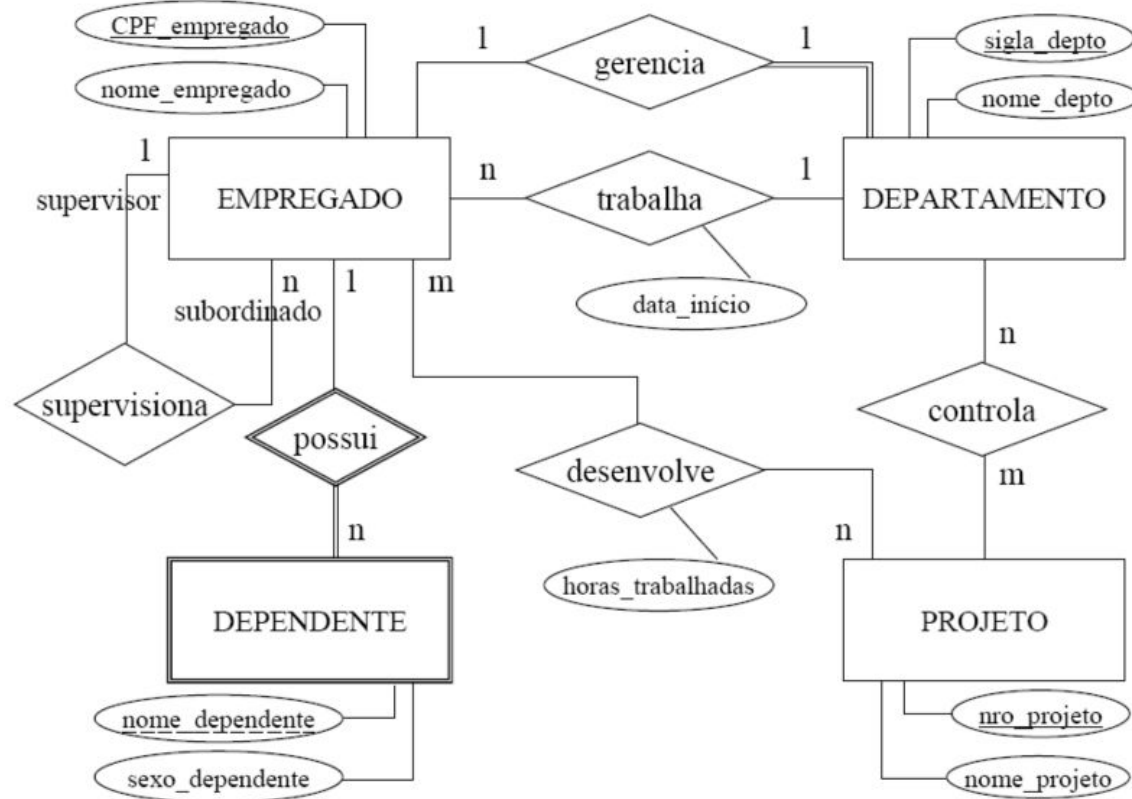


Etapas do Mapeamento

- Alternativa 3:
 - Tabelas apenas para as entidades especializadas
 - Não deve ser aplicado para especializações parciais



Exercício 1



Exercício 2

