



Resumo II: Modelo Relacional e Mapeamento



Informações

- Alunos:
 - Lucas de Araújo: 18.2.4049
 - Bárbara Letícia Milagres: 18.2.4056
- Professor Doutor: Guilherme Tavares de Assis

Modelo Relacional

Conceitos Básicos

Característica das Relações

Esquema de um Banco de Dados Relacional

Exemplo de Instância de um Banco de Dados Relacional

Restrições do Modelo Relacional

Integridade Referencial com Opção de Exclusão

Operações de Atualização sobre Relações

Mapeamento ERE para Relacional

Entidade

Entidade Fraca

Atributo multivaleorado

Relacionamentos

Relacionamento binário 1:1

Relacionamento binário 1:N

Relacionamento binário N:M

Relacionamento n-ário (n >2)

Outras considerações sobre relacionamentos

Especialização e Generalização

Opção 1:

Opção 2:

Opção 3:

Opção 4:

Tipo União ou Categoria

Agregação

Modelo Relacional

Introduzido por Codd em 1970 através de uma pesquisa realizada na IBM, é um modelo formal fundamentado nos conceitos de uma relação matemática. Ou seja, a ideia principal utilizada é a **relação**

Portanto, podemos dizer que o modelo relacional representa o banco de dados como uma coleção de relações

Informalmente, cada relação é semelhante a uma tabela, sendo cada linha desta tabela um representação de uma coleção de valores de dados relacionados, que podem ser interpretados como fatos que descrevem uma entidade ou até um relacionamento.

O nome da tabela e os nomes das colunas são usados para facilitar na interpretação do significado dos valores em cada linha presente na tabela.

A relação corresponde, informalmente, a uma tabela

Cada coluna da relação corresponde a um atributo

Departamento

| NomeDept | NumDept | NumGerente | DataInicioGerencia |
|---------------|---------|------------|--------------------|
| Pesquisa | 5 | 333445555 | 22/05/98 |
| Administração | 4 | 987654321 | 01/01/95 |
| Diretoria | 1 | 888665555 | 19/06/01 |

Cada linha da relação corresponde a uma tupla

Podemos observar então, através da imagem a seguir, que uma relação nada mais é que uma coleção de **tuplas**.

Conceitos Básicos

Na terminologia do modelo relacional, temos:

- Uma linha: tupla
- Um cabeçalho de coluna: atributo
- Uma tabela: **relação**

O tipo de dado que descreve os valores que um atributo pode ter é chamado de **domínio** (similar ao conceito de domínio de funções). Um domínio D é um conjunto de valores atômicos, ou seja, **indivisíveis**. Exemplos:

- Nome: conjunto de nomes de pessoas. Cadeia de caracteres contendo apenas letras
- Idade de funcionários: os valores possíveis para idade de funcionários da empresa. Número inteiro entre 15 e 80
- CPFs: conjunto de números no formato ddd.ddd.ddd-dd, onde cada 'd' é um dígito de 0 a 9
- Salários: número real positivo com duas casas decimais

Temos também, os esquemas de relações

- **Esquema de relação:** é uma expressão da forma $R(A_1, A_2, \dots, A_n)$ onde:

- R: nome da relação
- A_i : nome de um atributo, cujo domínio em R é denotado por $\text{dom}(A_i)$
- n: grau da relação

Exemplos:

- Estudante (nome, matrícula, endereço, telefone, dataNasc)
- Disciplina (nome, código, cargaHorária, numCréditos)
- **Relação ou instância de uma relação:** uma relação r de um esquema R, denotado por $r(R)$, é um conjunto de tuplas: $r = \{t_1, t_2, \dots, t_n\}$
 - Cada tupla é uma lista ordenada de n valores:
 $t = <v_1, v_2, \dots, v_n>$, onde cada v_i , $1 \leq i \leq n$, é um elemento do domínio $\text{dom}(A_i)$ ou um valor especial nulo.

Formalmente, uma relação $r(R)$ é um subconjunto do produto cartesiano dos domínios que definem R:

►

O produto cartesiano especifica todas as possíveis combinações de valores dos domínios fundamentais

Característica das Relações

A ordem das tuplas é irrelevante em si. Porém, a ordem dos valores dentro de uma tupla é relevante, a menos que se estabeleça uma correspondência entre esses valores e os atributos definidos, como por exemplo:

- T = <BD, 032, 72, 4>
- T = <(nome, BD), (código, 032), (cargaHorária, 72), (numCreditos, 4)>
- T = <(código, 032), (numCreditos, 4), (cargaHorária, 72), (nome, BD)>

O valor de cada atributo na tupla é atômico (indivisíveis), logo atributos compostos e multivalorados não são permitidos. As tuplas de uma relação são únicas, ou seja, suas chaves primárias sempre terão um valor diferente de todas as outras tuplas.

Esquema de um Banco de Dados Relacional

Um esquema S de um banco de dados relacional define um conjunto de esquemas de relação:

$$R = \{R_1, R_2, \dots, R_n\}$$

E um conjunto de restrições de integridade:

$$I = \{I_1, I_2, \dots, I_m\}$$

Já uma instância BD de um esquema S é um conjunto de instâncias de relação:

$$BD = \{r_1, r_2, \dots, r_n\}$$

Tal que cada r_i é uma instância de R_i e as relações de r_i satisfazem as restrições de integridade em I . Portanto, podemos dizer que $S = (R, I)$

| Empregado | | | | | | | | | |
|-------------------|----------------|-------------|--------------------|----------------|----------|------|---------|---------------|---------|
| PrimeiroNome | InicialMeio | UltimoNome | NumEmpregado | DataNascimento | Endereco | Sexo | Salario | NumSupervisor | NumDept |
| Departamento | | | | | | | | | |
| NomeDept | NumDept | NumGerente | DataInicioGerencia | | | | | | |
| Localizacao_Depto | | | | | | | | | |
| NumDepart | Localizacao | | | | | | | | |
| Trabalha_em | | | | | | | | | |
| NumEmpregado | NumProj | Horas | | | | | | | |
| Projeto | | | | | | | | | |
| NomeProj | NumProj | Localizacao | NumDept | | | | | | |
| Dependente | | | | | | | | | |
| NumEmpregado | NomeDependente | Sexo | DataAniversario | Parentesco | | | | | |

Exemplo de Instância de um Banco de Dados Relacional

| Empregado | | | | | | | | | |
|--------------|-------------|-------------|--------------|----------------|-----------------------|------|---------|---------------|---------|
| PrimeiroNome | InicialMeio | UltimoNome | NumEmpregado | DataNascimento | Endereco | Sexo | Salario | NumSupervisor | NumDept |
| João | B | Silva | 123456789 | 09/01/65 | R. da Bahia, 2557 | M | 300.00 | 333445555 | 5 |
| Frank | T | Santos | 333445555 | 08/12/55 | Av. Afonso Pena, 3005 | M | 4000.00 | 888665555 | 5 |
| Alice | J | Pereira | 999887777 | 19/07/68 | Av. do Contorno, 2534 | F | 2500.00 | 987654321 | 4 |
| Luciene | S | Ferreira | 987654321 | 20/06/51 | R. Irajá, 175 | F | 430.00 | 888665555 | 4 |
| Pedro | K | Magalhães | 666884444 | 15/09/52 | Av. Silva Lobo, 2050 | M | 1200.00 | 333445555 | 5 |
| Daniela | A | Oliveira | 453453453 | 31/07/62 | R. Ataliba Lago, 250 | F | 2500.00 | 333445555 | 5 |
| Mateus | V | Mascarenhas | 987987987 | 29/03/79 | R. Contraí, 12 | M | 2500.00 | 987654321 | 4 |
| Fábio | E | Lemos | 888665555 | 10/11/47 | R. Chile, 425 | M | 5500.00 | null | 1 |

| Departamento | | | | | | | | | |
|---------------|---------|------------|--------------------|--|--|--|--|--|--|
| NomeDept | NumDept | NumGerente | DataInicioGerencia | | | | | | |
| Pesquisa | 5 | 333445555 | 22/05/98 | | | | | | |
| Administração | 4 | 987654321 | 01/01/95 | | | | | | |
| Diretoria | 1 | 888665555 | 19/06/01 | | | | | | |

| Localizacao_Depto | | | | | | | | | |
|-------------------|-------------|--|--|--|--|--|--|--|--|
| NumDepart | Localizacao | | | | | | | | |
| 1 | Savassi | | | | | | | | |
| 4 | Centro | | | | | | | | |
| 5 | Buritis | | | | | | | | |
| 5 | Pampulha | | | | | | | | |
| 5 | Contagem | | | | | | | | |

| Projeto | | | | | | | | | |
|-----------------|---------|-------------|---------|--|--|--|--|--|--|
| NomeProj | NumProj | Localizacao | NumDept | | | | | | |
| Produto X | 1 | Buritis | 5 | | | | | | |
| Produto Y | 2 | Pampulha | 5 | | | | | | |
| Produto Z | 3 | Contagem | 5 | | | | | | |
| Informatização | 10 | Centro | 4 | | | | | | |
| Reorganização | 20 | Savassi | 1 | | | | | | |
| NovosBeneficios | 30 | Centro | 4 | | | | | | |

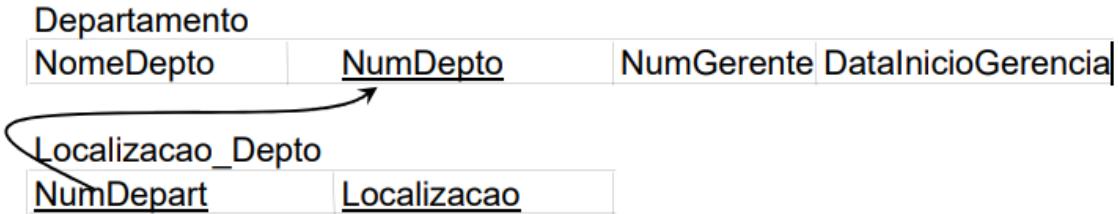
| Trabalha_em | | | | | | | | | |
|--------------|---------|-------|--|--|--|--|--|--|--|
| NumEmpregado | NumProj | Horas | | | | | | | |
| 123456789 | 1 | 32 | | | | | | | |
| 123456789 | 2 | 7 | | | | | | | |
| 666884444 | 3 | 40 | | | | | | | |
| 453453453 | 1 | 20 | | | | | | | |
| 453453453 | 2 | 20 | | | | | | | |
| 333445555 | 2 | 10 | | | | | | | |
| 333445555 | 3 | 10 | | | | | | | |
| 333445555 | 10 | 10 | | | | | | | |
| 333445555 | 20 | 10 | | | | | | | |
| 999887777 | 30 | 30 | | | | | | | |
| 999887777 | 10 | 10 | | | | | | | |
| 987987987 | 10 | 35 | | | | | | | |
| 987987987 | 30 | 5 | | | | | | | |
| 987654321 | 30 | 20 | | | | | | | |
| 987654321 | 20 | 15 | | | | | | | |
| 888665555 | 20 | null | | | | | | | |

| Dependente | | | | | | | | | |
|--------------|----------------|------|-----------------|------------|--|--|--|--|--|
| NumEmpregado | NomeDependente | Sexo | DataAniversario | Parentesco | | | | | |
| 333445555 | Aline | F | 03/04/76 | Filha | | | | | |
| 333445555 | Vitor | M | 25/10/73 | Filho | | | | | |
| 333445555 | Joana | F | 03/05/98 | Cônjuge | | | | | |
| 987654321 | Igor | M | 29/02/52 | Cônjuge | | | | | |
| 123456789 | Michel | M | 01/01/88 | Filho | | | | | |
| 123456789 | Aline | F | 31/12/98 | Filha | | | | | |
| 123456789 | Elizabeth | F | 05/05/57 | Cônjuge | | | | | |

Restrições do Modelo Relacional

Dentro do modelo relacional, os valores de atributos possuem restrições que são classificadas da seguinte forma:

- **Restrição de Domínio:** o valor de cada atributo deve ser um valor atômico presente no domínio do atributo
- **Restrição de Chave:** uma chave é um conjunto mínimo de valores dos atributos que identificaunicamente uma tupla.
 - Se um esquema de relação tem mais de uma chave, cada uma dessas ganha a alcunha de **chave candidata**. Logo após, uma das chaves candidatas é escolhida arbitrariamente para se tornar a **chave primária** e as outras não selecionadas tornam-se **chaves alternativas** (ou secundárias)
 - Tem se em mente que é melhor escolher como chave primária aquela com o menor número de atributos.
 - Cada esquema de relação R deve ter uma chave primária que é indicada no esquema por um sublinhado
- **Restrição de Integridade de Entidade:** a chave primária de um esquema de relação R não pode ter valor nulo
- **Restrição de Integridade Referencial:** é uma restrição especificada entre outras duas relações, sendo utilizada para manter a consistência entre as tuplas das duas relações. Utiliza-se o conceito de chave estrangeira para definir tais restrições
 - Um conjunto de atributos FK em um esquema de relação R_1 é uma chave estrangeira de R_1 que referencia uma relação R_2 caso satisfaça as duas regras seguintes:
 - 1) Os atributos de FK referenciam a chave primária PK de R_2 tendo os mesmos domínios dos atributos de PK.
 - 2) Um valor de FK na tupla t_1 da instância $r_1(R_1)$ ocorre como um valor PK para alguma tupla t_2 da instância $r_2(R_2)$ ou é nulo
 - São obtidas, geralmente, a partir dos relacionamentos entre as entidades representadas pelos esquemas de relação;
 - podem ser representadas graficamente usando-se setas que partem de uma chave estrangeira para a chave primária na relação referenciada, da seguinte maneira:



Integridade Referencial com Opção de Exclusão

Notação: $R_1[fk] \rightarrow^{op} R_2[pk]$

Onde **op** é a opção de exclusão, que pode ser:

- **bloqueio (b)**: se alguma tupla referencia a tupla a ser excluída, através de uma chave estrangeira, a exclusão não é efetuada;
- **propagação (p)**: todas as tuplas que referenciam a tupla a ser excluída, através de uma chave estrangeira, são excluídas também automaticamente;
- **substituição por nulos (n)**: todas as tuplas que referenciam a tupla a ser excluída, através de uma chave estrangeira, têm os valores dos atributos da chave estrangeira modificados para nulo (se for permitido nulo) e a exclusão é efetuada.

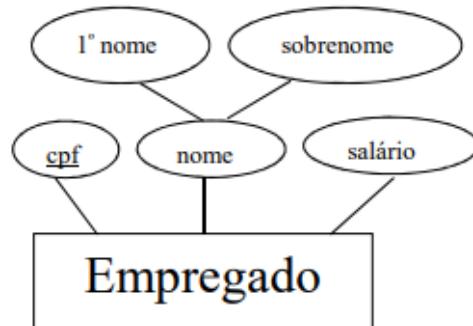
Operações de Atualização sobre Relações

- **Inserção (insert)**: Operações de Atualização sobre Relações.
 - Pode violar qualquer dos quatro tipos de restrições discutidas.
- **Exclusão (delete)**: Exclui tuplas de uma relação.
 - Pode violar somente restrição de integridade referencial.
- **Modificação (update ou modify)**: Muda os valores de alguns atributos em tuplas existentes.
 - Modificar um atributo que não é chave primária nem chave estrangeira pode violar somente a restrição de domínio;
 - Modificar a chave primária é similar a excluir uma tupla e inserir uma outra no seu lugar; assim, pode violar qualquer das quatro restrições discutidas;
 - Modificar um atributo de uma chave estrangeira pode violar a restrição de integridade referencial ou de domínio.

Mapeamento ERE para Relacional

Entidade

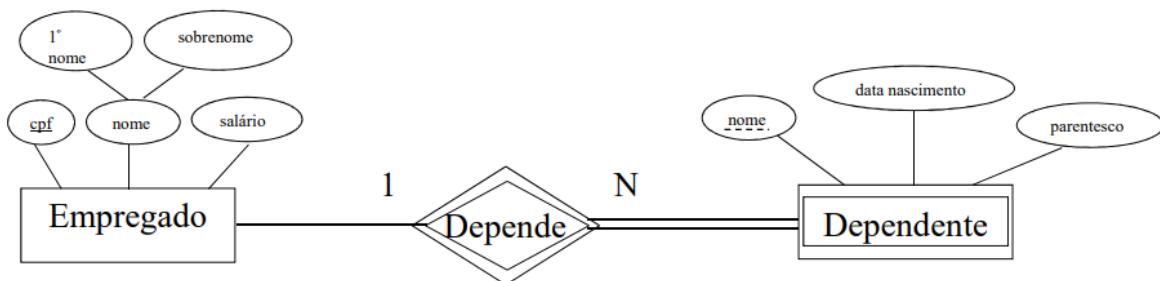
Para cada Entidade no esquema ERE é criada uma relação R e são inclusos todos os atributos simples. No caso dos atributos compostos, serão considerados os atributos simples ligados a ele. Além disso, a chave primária será selecionada a partir das chaves candidatas. No esquema à seguir temos um atributo composto, abaixo é apresentado como ele fica na relação R gerada:



Relação R: **Empregado**(cpf, primeironome, sobrenome, salário)

Entidade Fraca

No caso das entidades fraca (EF), é criada uma relação R com todos atributos simples e componentes simples de atributos compostos da EF. Além disso, também são incluídos como atributos de R todos os atributos componentes das chaves primárias de todas as entidades fortes da EF. Essas inclusões correspondem a chaves estrangeiras de R. A chave primária de R é a combinação dos atributos das chaves primárias das entidades fortes da EF mais a chave parcial da EF. No exemplo à seguir, temos uma entidade fraca e a representação dela em uma relação R:



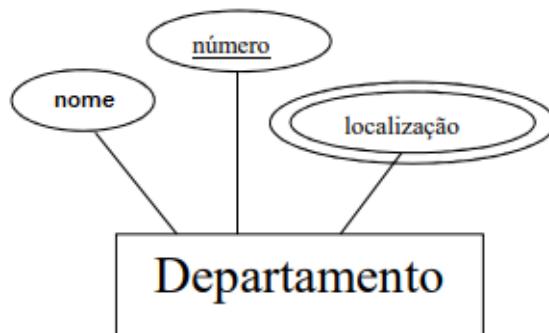
Relação: **Dependente**(cpfEmp, nome, datanascimento, parentesco)

Como **cpfEmp** se trata de uma chave estrangeira, então deve ser feita a **Restrição de Integridade Referencial**:

$$\text{Dependente}[\text{cpfEmp}] \rightarrow^p \text{Empregado}[\text{cpf}]$$

Atributo multivalorado

No caso dos atributos multivalorados, vamos chamá-los de A, deve ser criada uma relação que inclua o atributo A e a chave primária K, uma chave estrangeira em R, da relação que representa o tipo de entidade ou relacionamento que tem A como atributo. No caso de atributos compostos, vale a mesma regra anteriormente citada. A chave primária de R será a combinação de K e A. No exemplo abaixo, é criada uma relação para o atributo multivalorado **localização**:



Relação: Localização_Dpto(númeroDept, localização)

Restrição de Integridade Referencial: Localização_Dpto[númeroDept] →^p
Departamento[número]

Relacionamentos

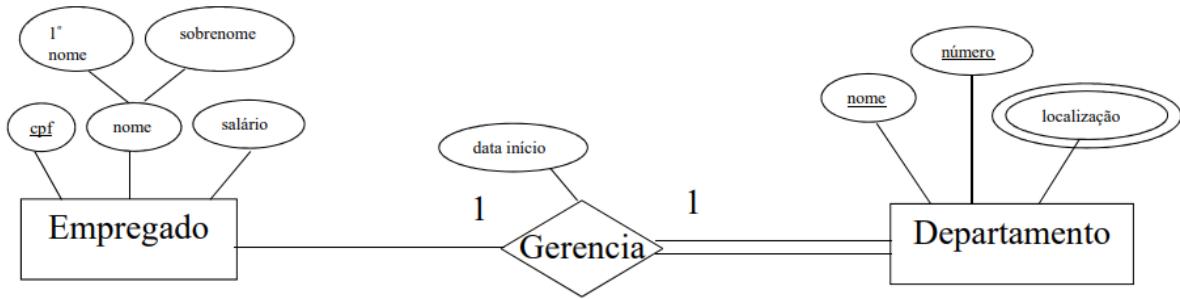
Para cada tipo de relacionamento binário no esquema ERE, o mapeamento será feito de forma diferente. À seguir são listados os tipos de relacionamento e a forma como são mapeados.

Relacionamento binário 1:1

No caso dos relacionamentos binários 1:1, inicialmente deve-se identificar as relações S e T que correspondem aos tipos de entidades participantes do relacionamento R. Em seguida, é feita a escolha de uma das relações, S ou T (vamos escolher S por exemplo), e incluir como chave estrangeira em S a chave primária de T e todos os atributos simples do relacionamento R, no caso de atributos compostos, cada atributo simples, como explicado anteriormente. Para a escolha da entidade, é melhor escolher uma que possua participação total em R.

OBS: Um mapeamento alternativo é juntar os dois tipos de entidades e o tipo de relacionamento em uma única relação. Isso é particularmente apropriado quando ambas as participações são totais e os tipos de entidades não participam de nenhum outro tipo de relacionamento.

No exemplo abaixo é possível visualizar como é tratado um relacionamento 1:1 quando apenas um dos lados possui participação total:



Como **Departamento** possui participação total no relacionamento, então criamos uma chave estrangeira *cpfGerente* que vem da chave primária de **Empregado**. Além disso, o atributo "dataInício" também passa a fazer parte da relação **Departamento**. Assim temos:

Relação: **Departamento**(número, nome, cpfGerente, dataInício)

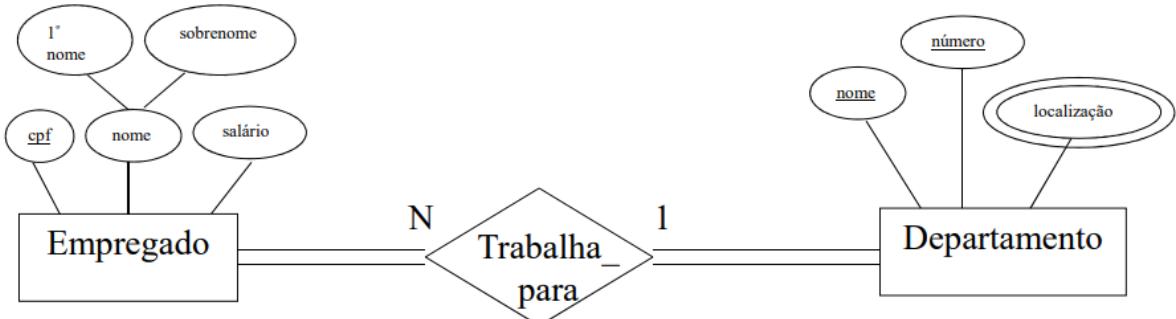
Obs: Vale lembrar que o atributo multivalorado "localização" foi tratado anteriormente.

Como a relação possui chave estrangeira, então deve ser feita a **Restrição de Integridade Referencial**:

$$\text{Departamento}[\text{cpfGerente}] \rightarrow^b \text{Empregado}[\text{cpf}]$$

Relacionamento binário 1:N

No caso de um relacionamento binário R 1:N regular, deve-se identificar a relação S que representa o tipo de entidade participante do lado N de R. Novamente, a chave primária de T, o lado 1 de R, deve ser incluída em S como chave estrangeira. Isso ocorre porque cada instância do lado N está relacionada a, no máximo, uma instância do lado 1 de R. Assim como citado anteriormente, todos os atributos simples (ou componentes simples de atributos compostos) de R devem ser incluídos como atributos de S. A seguir é apresentado o mapeamento de um relacionamento 1:N:



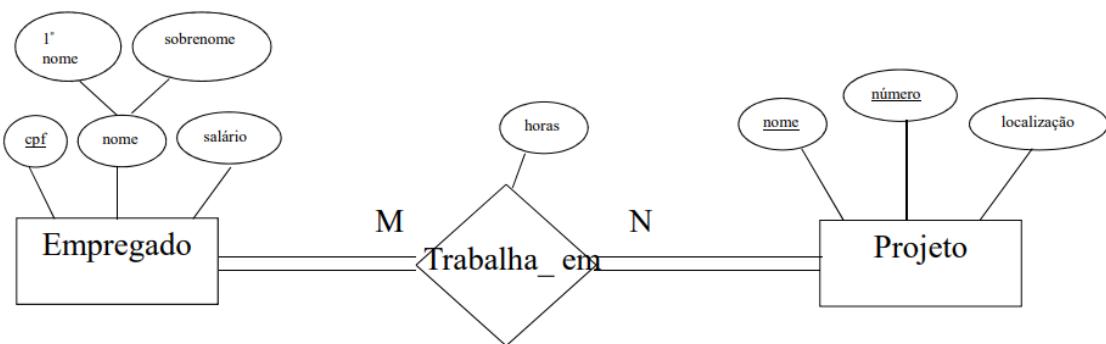
Empregado é o lado N da relação, logo, a chave estrangeira "númeroDept" advinda da chave primária de **Departamento** será incluída nela.

Relação: Empregado(cpf, primeiroNome, sobrenome, salário, númeroDept)

Restrição de Integridade Referencial: Empregado[númeroDept] \rightarrow^b Departamento[número]

Relacionamento binário N:M

No caso de um relacionamento binário R N:M, deve ser criada uma nova relação S para representar R. Nessa nova relação S, devem ser incluídas como chaves estrangeiras as chaves primárias das relações que representam os tipos de entidades participantes. Também devem ser incluídos todos os atributos simples (ou componentes simples de atributos compostos) do tipo de relacionamento R como atributos de S. A chave primária de S será a combinação de suas chaves estrangeiras. No exemplo abaixo, é apresentado um relacionamento N:M e em seguida seu mapeamento:



Como o relacionamento é N:M, criamos a relação "Trabalha_em" para representar o relacionamento:

Relação: Trabalha_em(horas, cpf, númeroProjeto)

Como especificado anteriormente, **cpf** e **númeroProjeto** são chaves estrangeiras e chaves primárias de Trabalha_em.

Como há chaves estrangeiras, então deve ser feita a **Restrição de Integridade Referencial**:

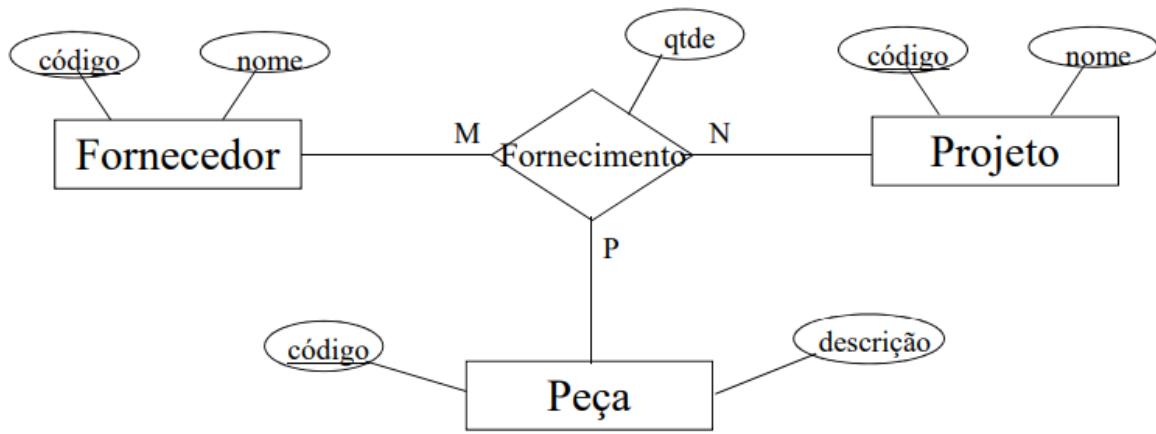
Trabalha_em[cpf] \rightarrow^p Empregado[cpf]

Trabalha_em[númeroProjeto] \rightarrow^p Projeto[número]

Relacionamento n-ário ($n > 2$)

No caso de um relacionamento n-ário R, com $n > 2$, deve ser criada uma nova relação S para representar R. Nesta nova relação, devem ser incluídos como atributos da chave estrangeira em S as chaves primárias das relações que representam os tipos de entidades participantes. Também devem ser incluídos todos os atributos simples (ou componentes simples de atributos compostos) do tipo de relacionamento R como atributos de S. A chave primária de S é normalmente a combinação de suas chaves estrangeiras. Entretanto, se as restrições de cardinalidade de qualquer um dos tipos de entidades E participante de R é 1, então a chave primária de S não deve incluir a

chave estrangeira que referencia a relação correspondente a E. Abaixo segue um exemplo de como se dá o mapeamento n-ário com n = 3:



Primeiro é gerada a relação "Fornecimento" que possui como atributos "qtde", do próprio tipo de relacionamento, seguido das chaves estrangeiras "codForn", "codProj" e "codPeça" que referenciam às chaves primárias de "Fornecedor", "Projeto" e "Peça", respectivamente.

Relação: Fornecimento(codForn, codProj, codPeça, qtde)

Como há chaves estrangeiras, então deve-se fazer a **Restrição de Integridade Referencial** para cada chave:

Fornecimento[codForn] →^p Fornecedor[código]

Fornecimento[codProj] →^p Projeto[código]

Fornecimento[codPeça] →^p Peça[código]

Outras considerações sobre relacionamentos

Os tipos de relacionamentos binários 1:1 e 1:N podem ser mapeados de forma similar ao tipo de relacionamento binário M:N. Uma alternativa útil quando existem poucas instâncias relacionadas a fim de evitar valores nulos nas chaves estrangeiras. Neste caso, a chave primária da relação que representa o tipo de relacionamento é a chave estrangeira de somente uma das relações que representam os tipos de entidades participantes.

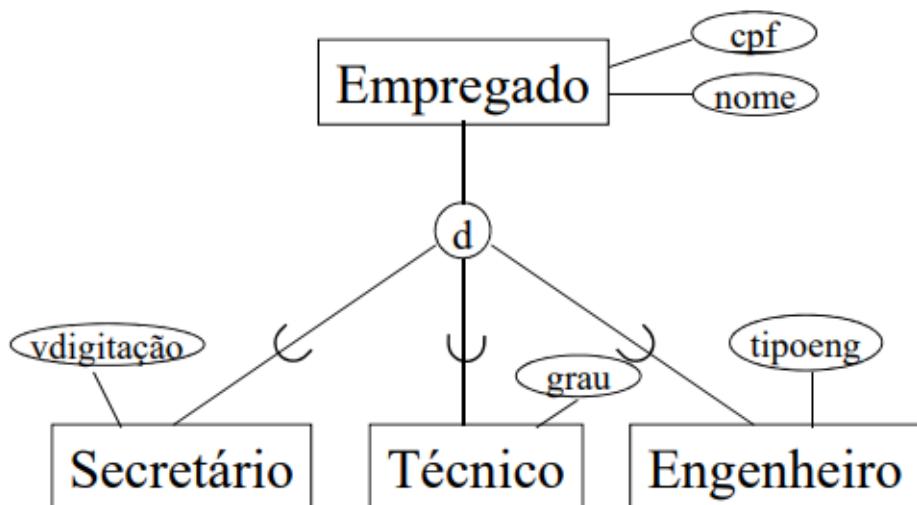
- Para um tipo de relacionamento 1:N, a chave primária vem da relação representando o lado N;
- Para um tipo de relacionamento 1:1, qualquer lado pode ser escolhido, mas é preferível escolher o lado com restrição de participação total (se houver), nesse caso, a outra chave estrangeira deve ser definida como chave secundária.

Especialização e Generalização

Para o mapeamento de uma Especialização ou Generalização, há quatro opções diferentes que são utilizadas dependendo do caso tratado. A seguir, uma descrição de cada uma das opções e exemplos de uso.

Opção 1:

Deve-se criar uma relação L para a superclasse C com todos os atributos de C . A chave primária de L é uma chave de C . Também deve ser criada uma relação L_i para cada subclasse S_i de C . Cada L_i inclui os atributos específicos de S_i mais a chave primária de L , a qual torna-se também a chave primária de L_i . É importante destacar que essa opção funciona para qualquer restrição i na especialização: disjunta/sobreposta, total/parcial. No exemplo a seguir, é apresentado o mapeamento de uma especialização utilizando a opção 1:



Primeiro cria-se a relação que representa a superclasse "Empregado":

Empregado(cpfo, nome)

Em seguida, cria-se uma relação L_i para cada subclasse S_i :

Secretário(cpfo, vdigitação)

Técnico(cpfo, grau)

Engenheiro(cpfo, tipoeng)

Como há chaves estrangeiras, então deve ser feita a **Restrição de Integridade Referencial**:

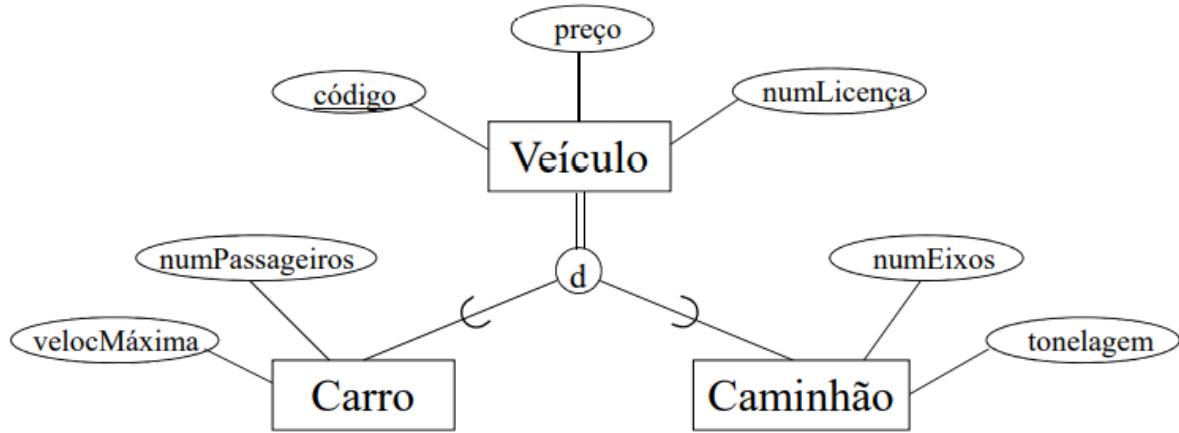
$\text{Secretário}[\text{cpfo}] \rightarrow^p \text{Empregado}[\text{cpfo}]$

$\text{Técnico}[\text{cpfo}] \rightarrow^p \text{Empregado}[\text{cpfo}]$

$\text{Engenheiro}[\text{cpfo}] \rightarrow^p \text{Empregado}[\text{cpfo}]$

Opção 2:

Deve-se criar uma relação L_i para cada subclasse S_i com os atributos da subclasse mais os atributos da superclasse C . Nesse caso não haverá uma relação apenas para a superclasse. A chave primária de L_i é uma chave da superclasse. Essa opção deve ser usada para restrições **total** e **disjunta**. No exemplo abaixo, há uma generalização com restrição total e disjunta:



São criadas relações para cada subclasse, tendo como chave primária a chave da superclasse, além de herdar todos atributos da superclasse:

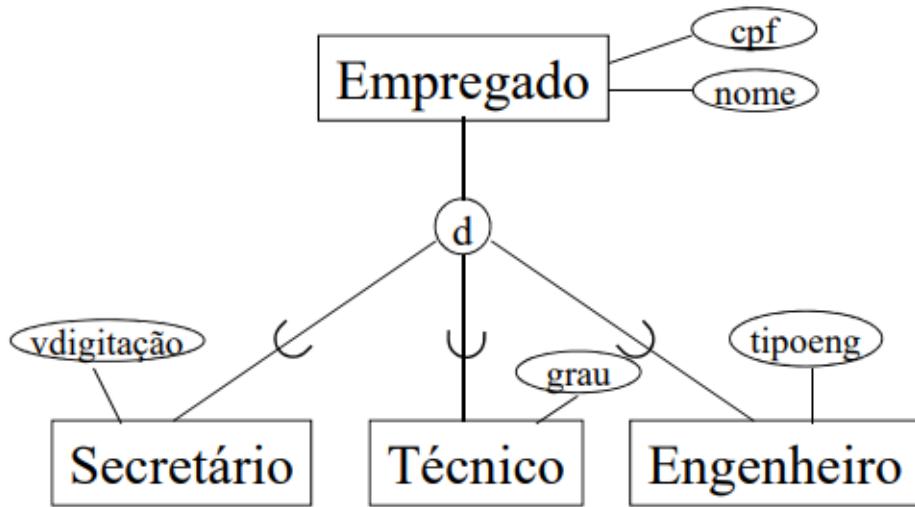
Carro(código, preço, numLicença, numPassageiros, velocMáxima)

Caminhão(código, preço, numLicença, numEixos, tonelagem)

Como não há chaves estrangeiras, então não é preciso fazer a Restrição de Integridade Referencial.

Opção 3:

Deve-se criar uma única relação L com todos os atributos da superclasse C e das subclasses S_i , e mais um atributo para indicar a qual subclasse cada tupla pertence. A chave primária de L é uma chave de C . Essa opção é para uma **especialização** cujas subclasses são **disjuntas**. Um possível problema é que essa opção pode gerar um grande número de valores nulos. O exemplo à seguir é o mesmo utilizado na opção 1:



A relação que representa um empregado, utilizando a opção 3, ficará da seguinte forma:

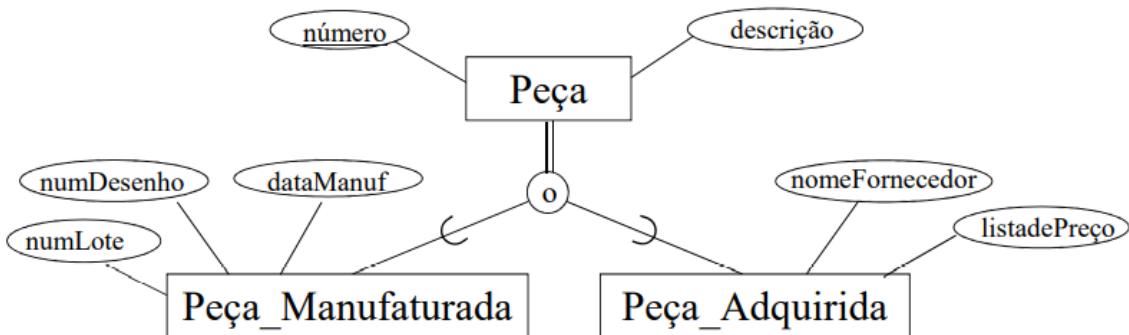
Empregado (cpf, nome, tipodetrabalho, vdigitação, grau, tipoeng)

Lembrando que, "tipodetrabalho" representa qual subclasse o empregado pertence.

Como não há chaves estrangeiras, então não é preciso fazer a Restrição de Integridade Referencial.

Opção 4:

Deve-se criar uma única relação L com todos os atributos da superclasse C e das subclasses S_i , e um atributo lógico (flag) t_i para cada subclasse para indicar se a tupla pertence à subclasse S_i . A chave primária de L é uma chave de C . Essa opção é indicada para especialização cujas subclasses são sobrepostas mas, também funciona para especialização disjunta. O exemplo a seguir apresenta o mapeamento de uma especialização sobreposta:



A relação que representa uma peça, utilizando a opção 4, ficará da seguinte forma:

Peça(número, descrição, **Mflag**, numDesenho, dataManuf, numLote, **Aflag**, nomeFornecedor, listaPreço)

MFlag: é uma flag para indicar que uma peça é manufaturada;

AFlag: é uma flag para indicar que uma peça é adquirida;

Vale lembrar que, como se trata de uma especialização sobreposta, a peça pode ser manufaturada e adquirida ao mesmo tempo.

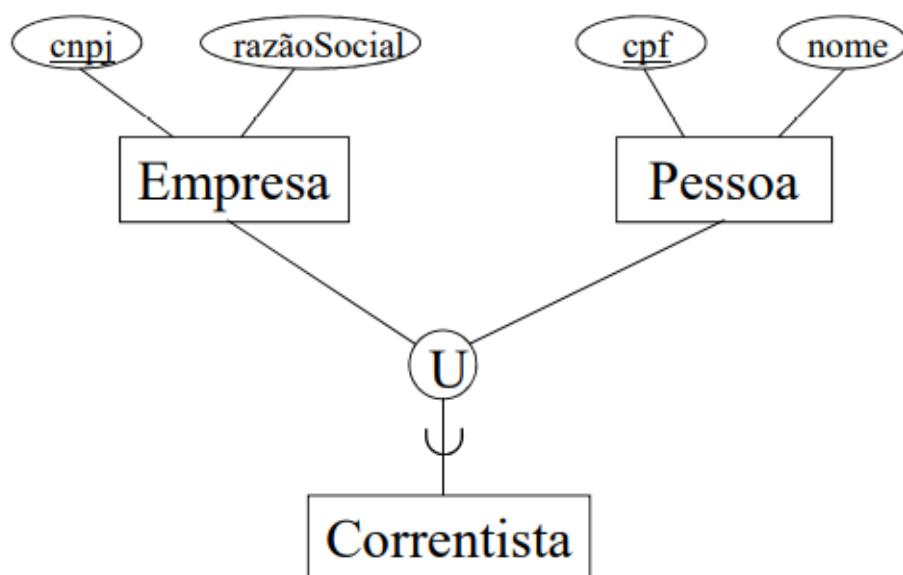
Como não há chaves estrangeiras, então não é preciso fazer a Restrição de Integridade Referencial.

Tipo União ou Categoria

Deve ser criada uma relação para representar a categoria e nela incluir todos os seus atributos.

- Para uma categoria cujas superclasses têm chaves diferentes, deve ser adicionado um novo atributo chave, chamado “chave substituta”, para ser a chave primária da relação. Após criado, deve-se adicionar este atributo como chave estrangeira em todas as relações correspondentes às superclasses da categoria, para especificar a correspondência de valores entre a chave substituta e as chaves de cada superclasse;
 - Para uma categoria cujas superclasses têm as mesmas chaves, a chave substituta não é necessária. Deve-se adicionar à relação representante da categoria o atributo chave de uma superclasse para ser a sua chave primária. Cada chave primária das relações correspondentes às superclasses são também chaves estrangeiras referenciando a relação da categoria.

O exemplo abaixo apresenta uma categoria com chaves diferentes nas superclasses, logo, deverá ser criada a chave substituta:



São criadas as relações "Empresa", "Pessoa" e "Correntista", lembrando que, como as chaves das superclasses são diferentes, então será criada uma chave substituta em "Correntista" que será chave estrangeira nas superclasses. No caso, a chave criada será o número do correntista, que é um atributo único de cada correntista.

Relações:

Empresa(cnpj, razaoSocial, numCorrentista)

Pessoa(cpf, nome, numCorrentista)

Correntista(numCorrentista)

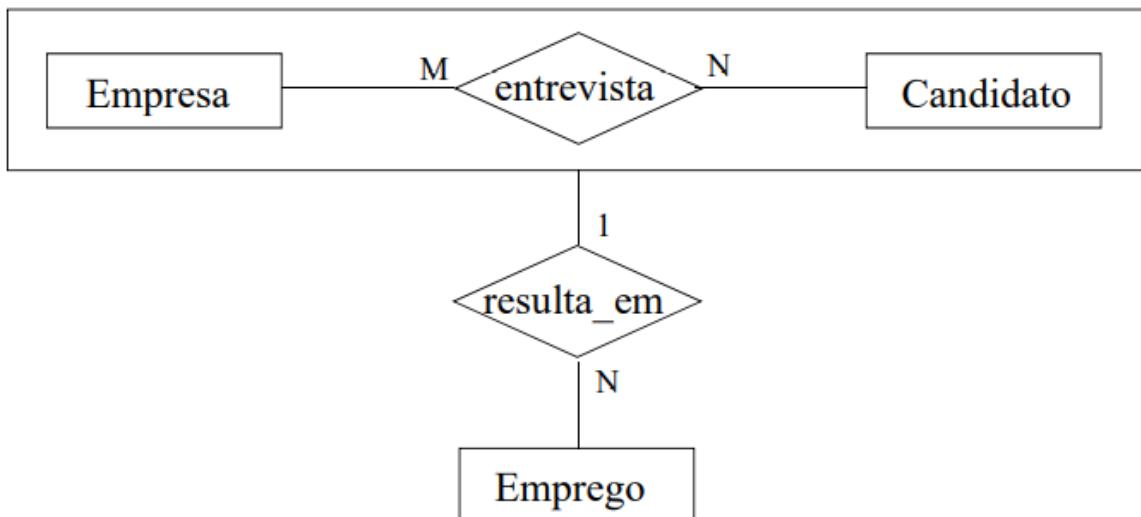
Como há chaves estrangeiras, então deve ser feita a **Restrição de Integridade Referencial**:

$\text{Empresa}[\text{numCorrentista}] \rightarrow^n \text{Correntista}[\text{numCorrentista}]$

$\text{Pessoa}[\text{numCorrentista}] \rightarrow^n \text{Correntista}[\text{numCorrentista}]$

Agregação

Não existe uma regra apropriada para agregação. Uma agregação é mapeada da mesma forma que os relacionamentos binários. No exemplo abaixo é possível observar como se dá o mapeamento de uma agregação:



Primeiro devem ser mapeados todos os tipos de entidade presentes:

Empresa(código, ...)

Candidato(numInscrição, ...)

Emprego(código, ...)

Feito isso, mapeia-se o relacionamento "entrevista", como se trata de um relacionamento binário N:M, então deve ser criada uma nova relação:

Entrevista(codEmpresa, numCand, data)

Como tem chaves estrangeiras em Entrevista, então deve ser feita a **Restrição de Integridade Referencial**:

Entrevista[codEmpresa] \rightarrow^p Empresa[código]

Entrevista[numCand] \rightarrow^p Candidato[numInscrição]

Como se trata de uma agregação, a relação "entrevista" foi convertida em uma entidade agregada, quando a relação "entrevista" é mapeada, a agregação já está sendo mapeada.

No caso da agregação, a partir do relacionamento envolvido, independente de ser 1:1, 1:N e N:M, deverá ser criada uma relação e, a escolha de sua chave primária se dará da seguinte maneira:

- N:M - combinação das chaves dos dois lados do relacionamento;
- 1:N - chave do lado N;
- 1:1 - analisar qual das duas chaves é melhor.

Após os passos anteriores, deve-se mapear o relacionamento que liga um tipo de entidade ao tipo de entidade agregada. Como a entidade agregada já foi mapeada, basta mapeá-lo como um relacionamento binário. No caso do exemplo, como "resulta_em" tem cardinalidade 1:N, basta colocar no lado N a chave primária do lado 1 como chave estrangeira. A chave primária do lado 1 é a chave primária do tipo de entidade agregada, que é a combinação de "codEmpresa" e "numCand". Logo, a relação "Emprego" ficará da seguinte forma:

Emprego(código, ..., codEmpresa, numCand)

Como há chaves estrangeiras, deve-se criar a **Restrição de Integridade Referencial**:

Emprego[codEmpresa, numCand] \rightarrow^b Entrevista[codEmpresa, numCand]

Temos apenas uma restrição de integridade referencial por se tratar de uma **única** chave estrangeira composta por dois atributos.