



Universidade de Brasília

Departamento de Ciência da Computação



Bancos de Dados

CIC0097



Prof. Pedro Garcia Freitas

<https://pedrogarcia.gitlab.io/>

pedro.garcia@unb.br

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciências da Computação



Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



Módulo 10

Modelo Relacional

– (MR) –

Parte 4:

Mapeamento MER → Relacional:

**Relacionamentos n-ários,
especialização/generalização**

CIC0097/2023.1

T1/T2



1. Objetivos

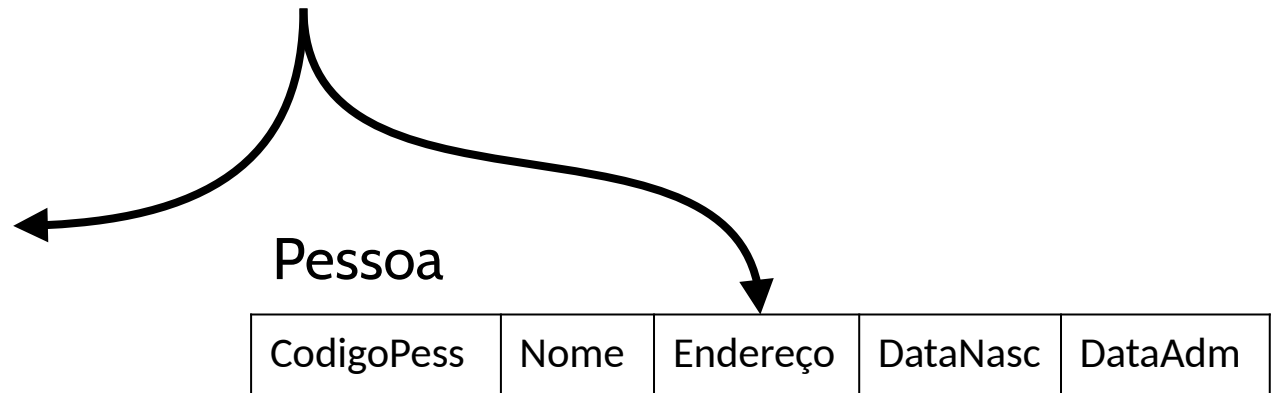
Esta aula apresenta o mapeamento $MER \rightarrow MR$, considerando as situações onde aparecem os relacionamentos n-ários, as especializações/generalizações e agregações.

2. Recap

2.1 Mapeamento de entidades fortes



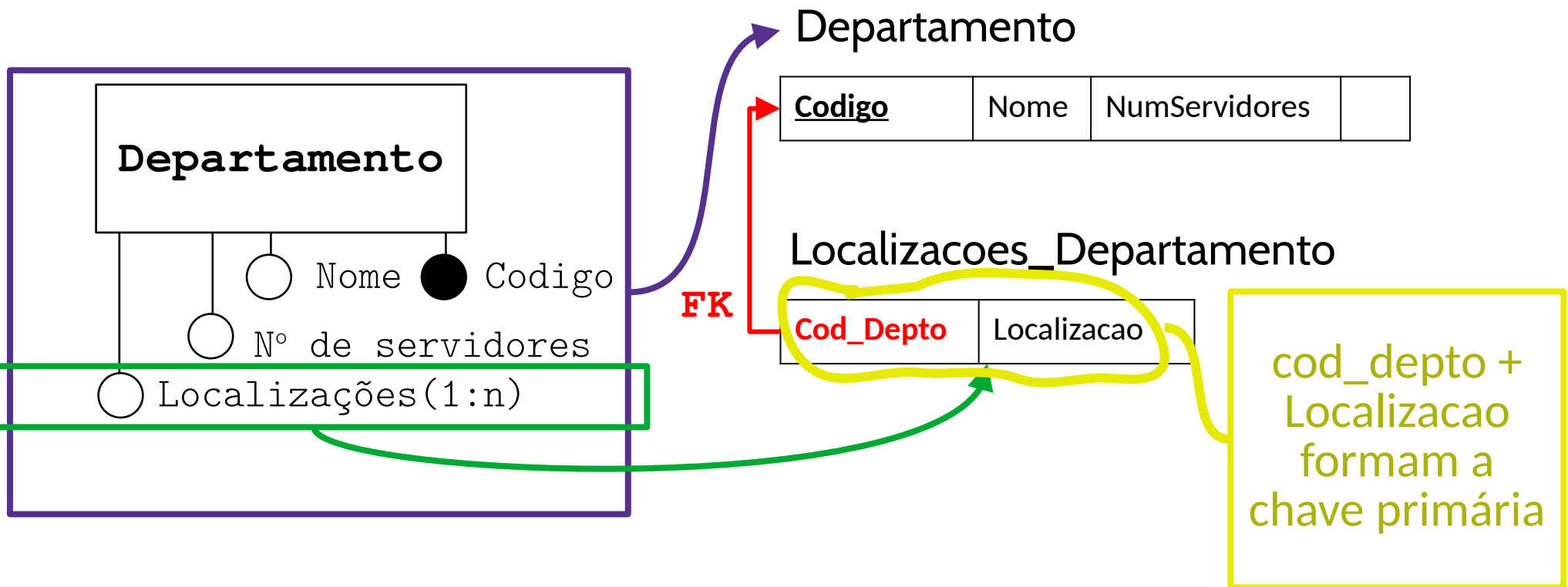
Pessoa (CodigoPess,
Nome,
Endereço,
DataNasc,
DataAdm)





2. Recap

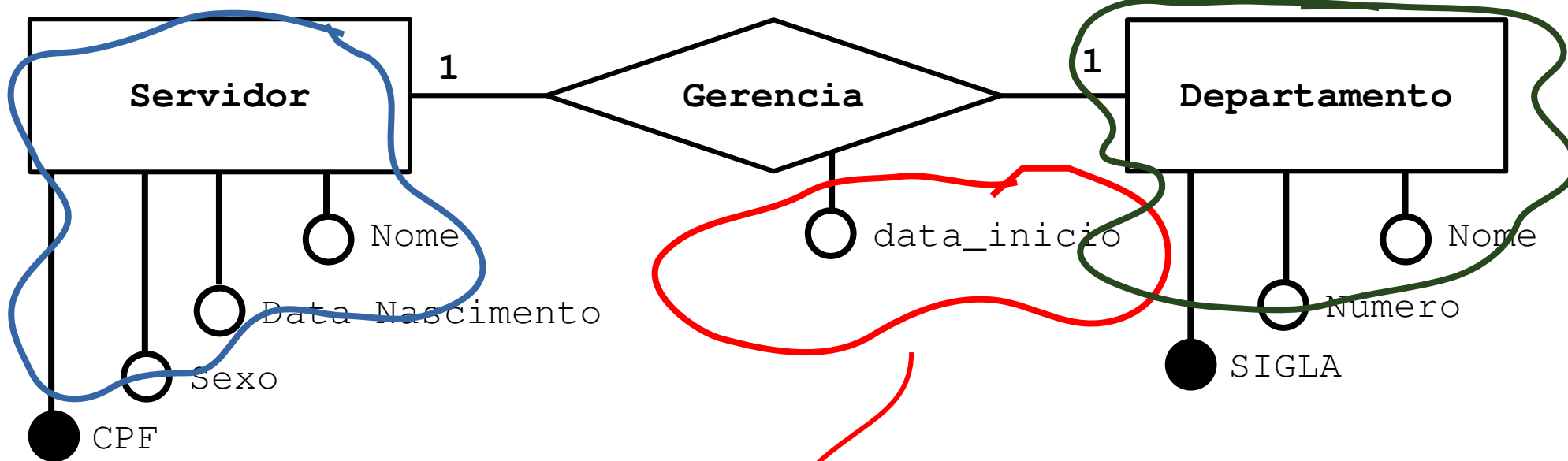
2.2 Mapeamento: atributos multivalorados





2. Recap

2.4 Relacionamentos binários 1:1 (FK)

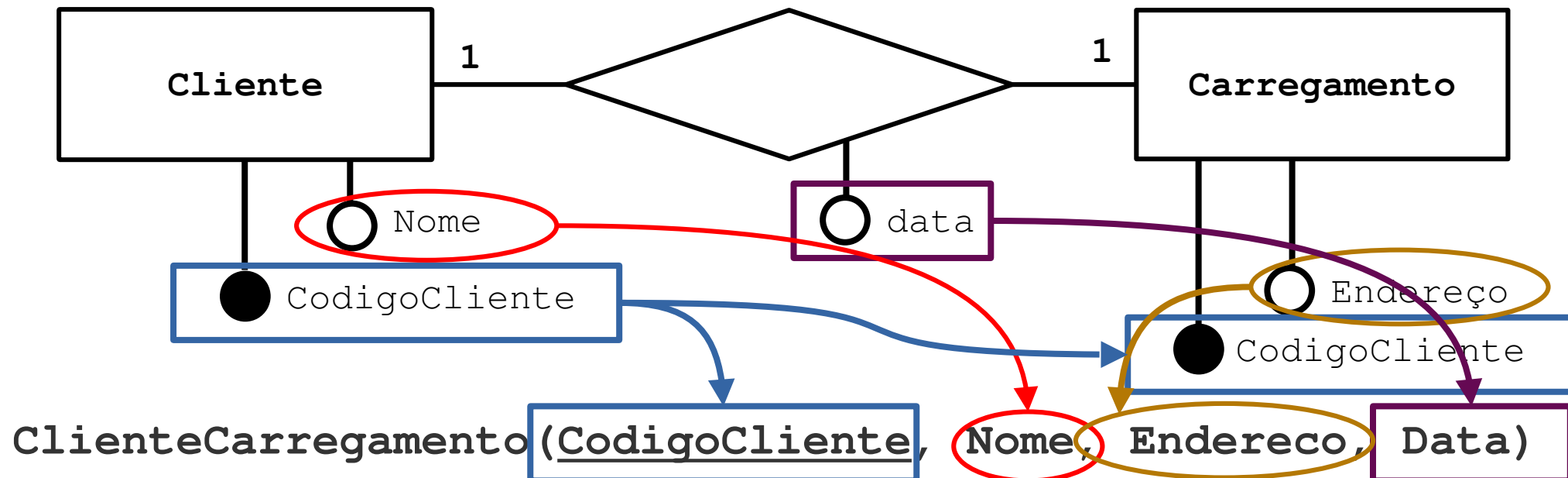


Servidor (CPF,
Nome,
DataNascimento,
Sexo)

Departamento (SIGLA,
Nome,
NumeroDepto,
data_inicio_gerente,
CPF_gerente)

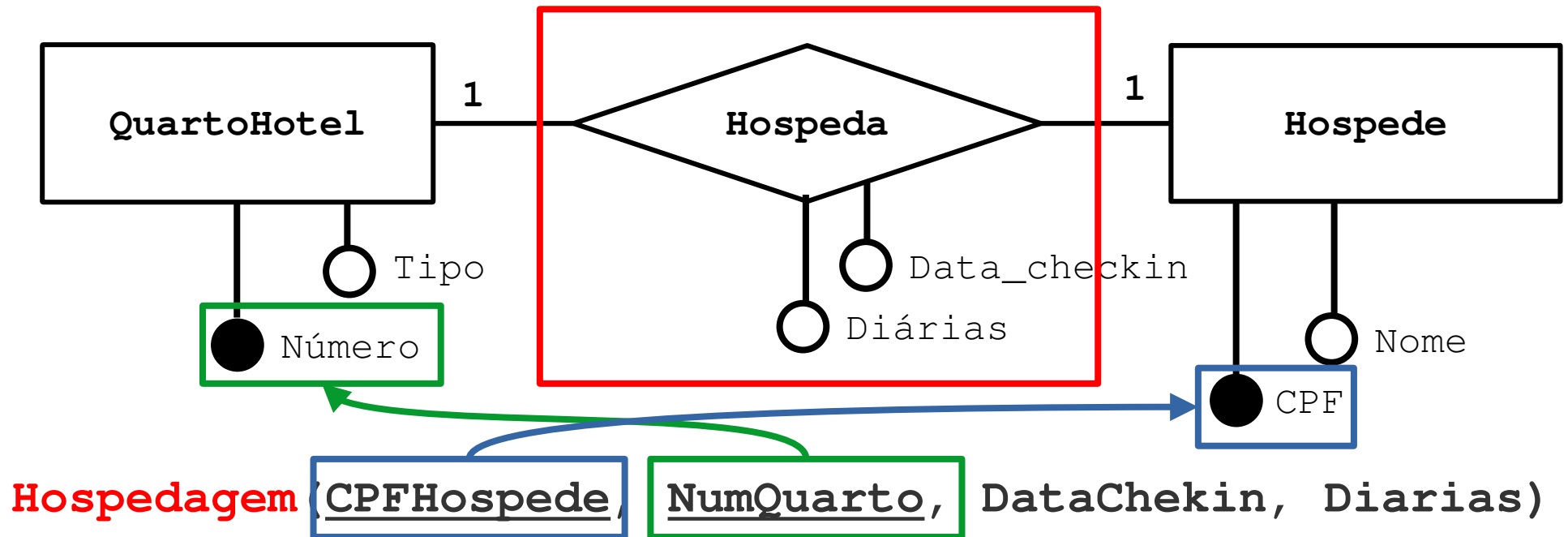
2. Recap

2.5 Relacionamentos binários 1:1 (relação unificada)



2. Recap

2.6 Relacionamentos binários 1:1 (referência cruzada)





2. Recap

2.7 Relacionamentos binários 1:n

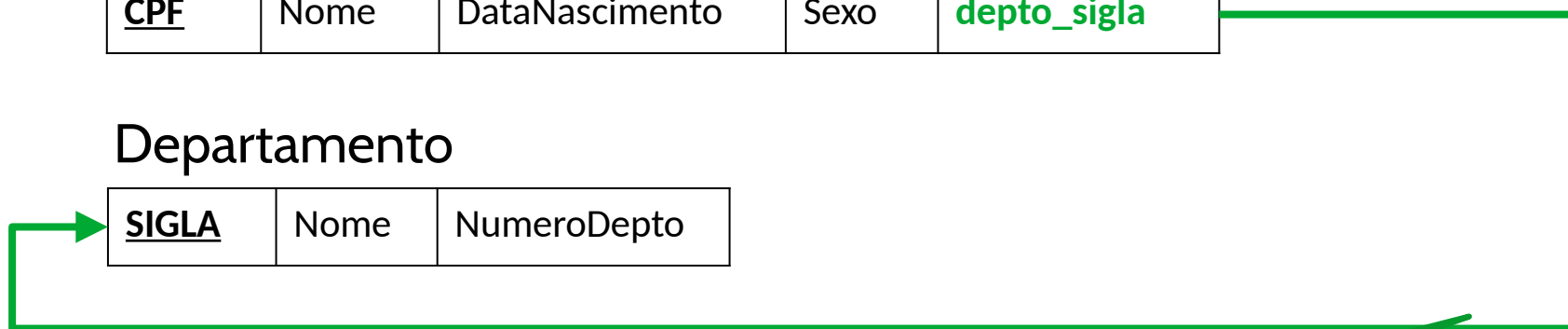


Servidor

<u>CPF</u>	Nome	DataNascimento	Sexo	depto_sigla
------------	------	----------------	------	-------------

Departamento

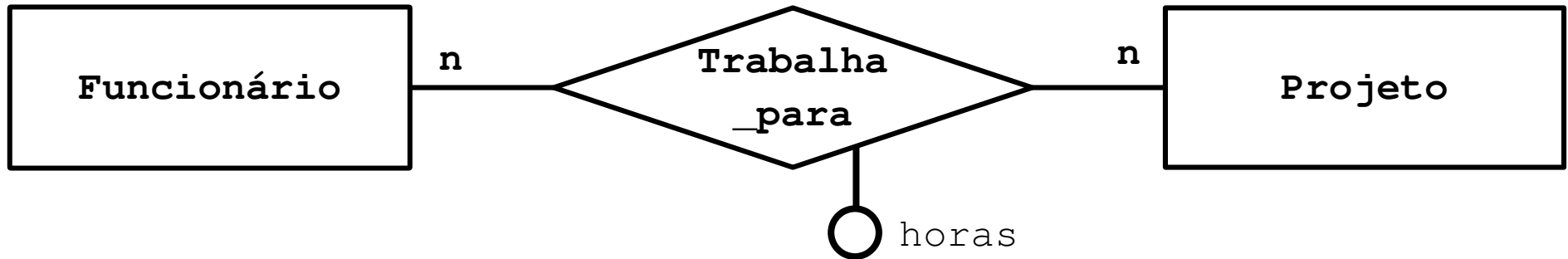
<u>SIGLA</u>	Nome	NumeroDepto
--------------	------	-------------



Parecido com o que acontece
com atributos multivalorados

2. Recap

2.8 Relacionamentos binários n:n



FUNCIONARIO (Pnome, NomeM, Unome, CPF, Data_nasc, Endereco, Salario, Sexo)

PROJETO (PNome, PNumero, PLocal, Depto)

TRABALHA_EM (CPF, PNumero, Horas)

Referência cruzada
obrigatória



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

→ Para cada não-binário/n-ário (i.e., $n > 2$), cria-se uma **nova relação S** para representar o relacionamento;



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

- Para cada cada não-binário/n-ário (i.e., $n > 2$), cria-se uma **nova relação S** para representar o relacionamento;
- As **chaves estrangeiras** de S são formadas **pelas chaves primárias** das relações que participam do relacionamento (igual na referência cruzada)



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

→ Os atributos simples ou componentes simples dos atributos compostos do relacionamento são incluídos em **S**.



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

→ A chave primária de S é, usualmente, a combinação de todas chaves estrangeiras em S .



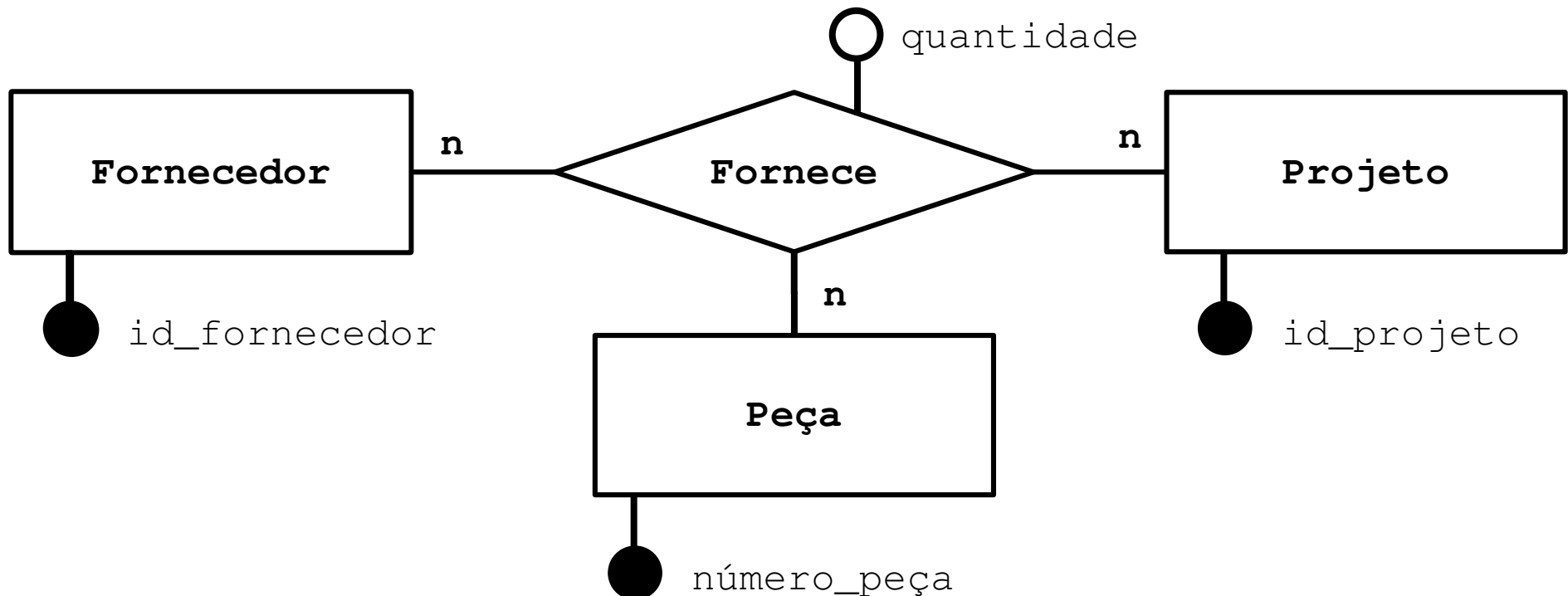
3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

- A chave primária de **S** é, usualmente, a **combinação de todas chaves estrangeiras em S.**
- Mas se a cardinalidade em qualquer participação do relacionamento é 1, então a chave primária de **S não deveria incluir a chave estrangeira que referencia a relação que representa esse tipo de entidade.**

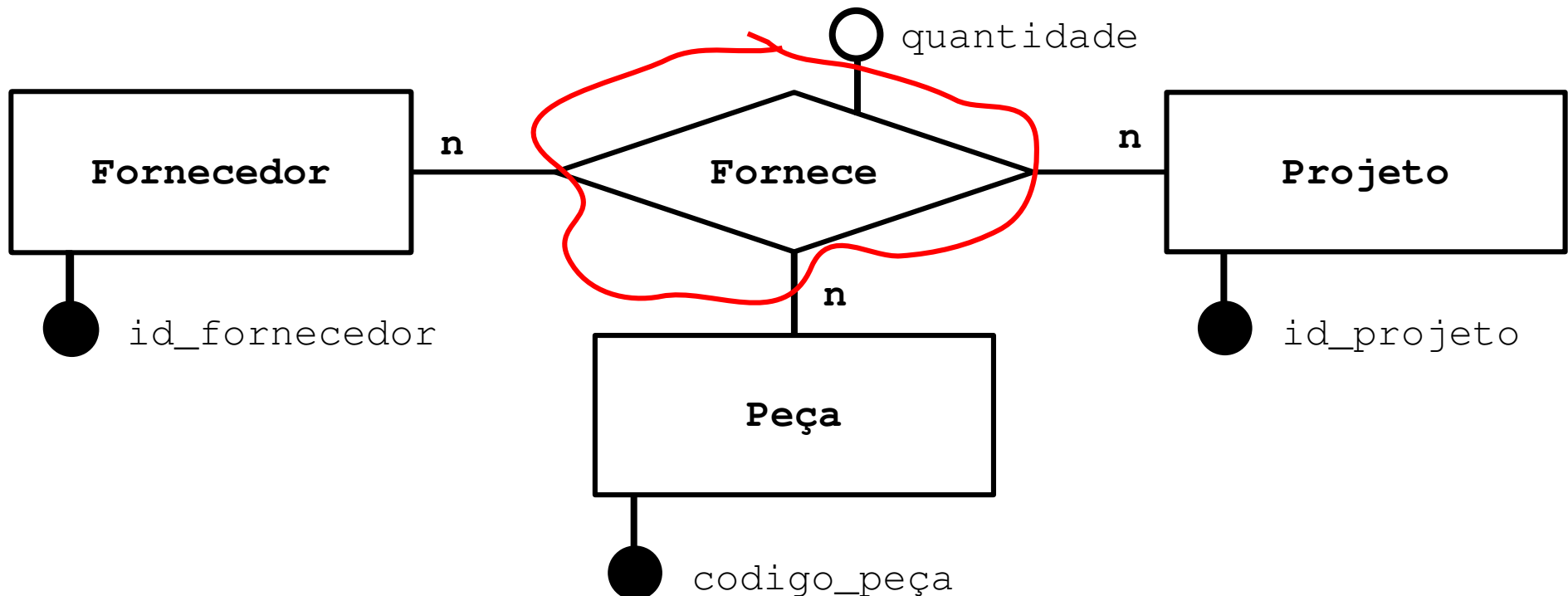
3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)



3. Mapeamento MER → Relacional:

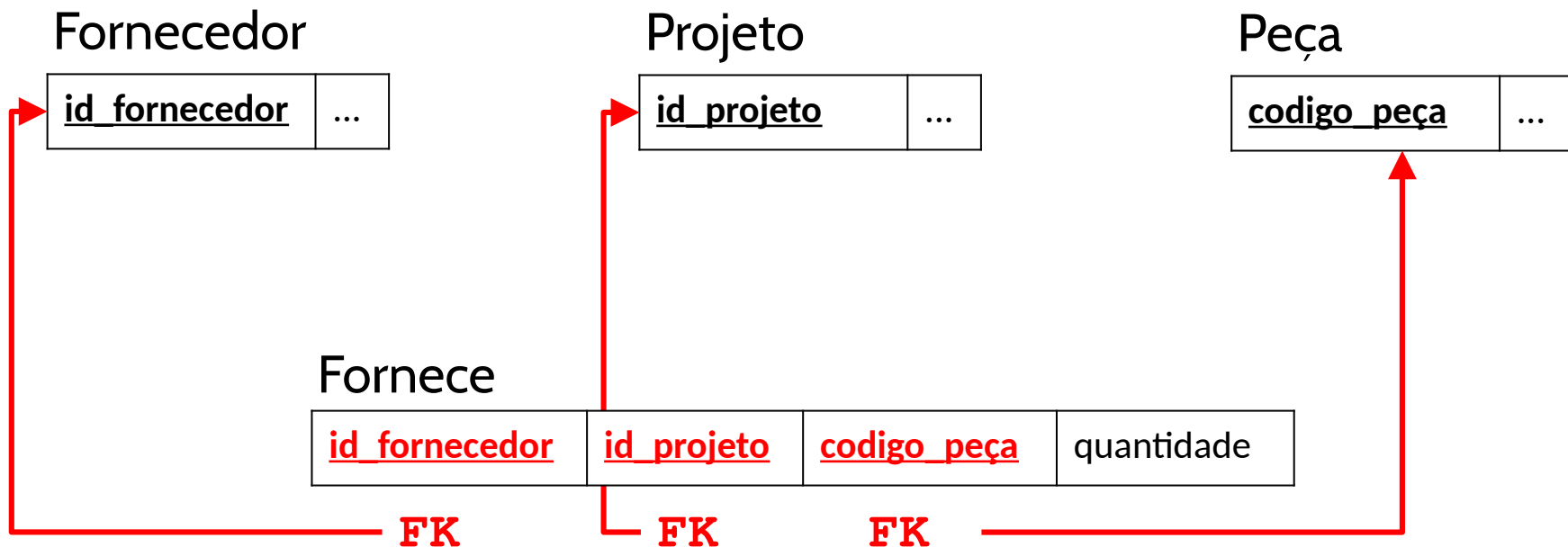
3.1 Relacionamentos não-binários (n-ários)





3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)





3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

Fornecedor

<u>id_fornecedor</u>	...
1001	...
1002	...
1003	...

Projeto

<u>id_projeto</u>	...
10	...
20	...
30	...

Peça

<u>número_peça</u>	...
111	...
333	...
222	...

Fornece

<u>id_fornecedor</u>	<u>id_projeto</u>	<u>codigo_peça</u>	quantidade
1001	10	111	20
1001	10	222	20
1002	10	111	20
1002	20	111	10
1003	20	111	30
1004	20	333	40



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

Fornecedor

<u>id_fornecedor</u>	...
1001	...
1002	...
1003	...

Projeto

<u>id_projeto</u>	...
10	...
20	...
30	...

Peça

<u>número_peça</u>	...
111	...
333	...
222	...

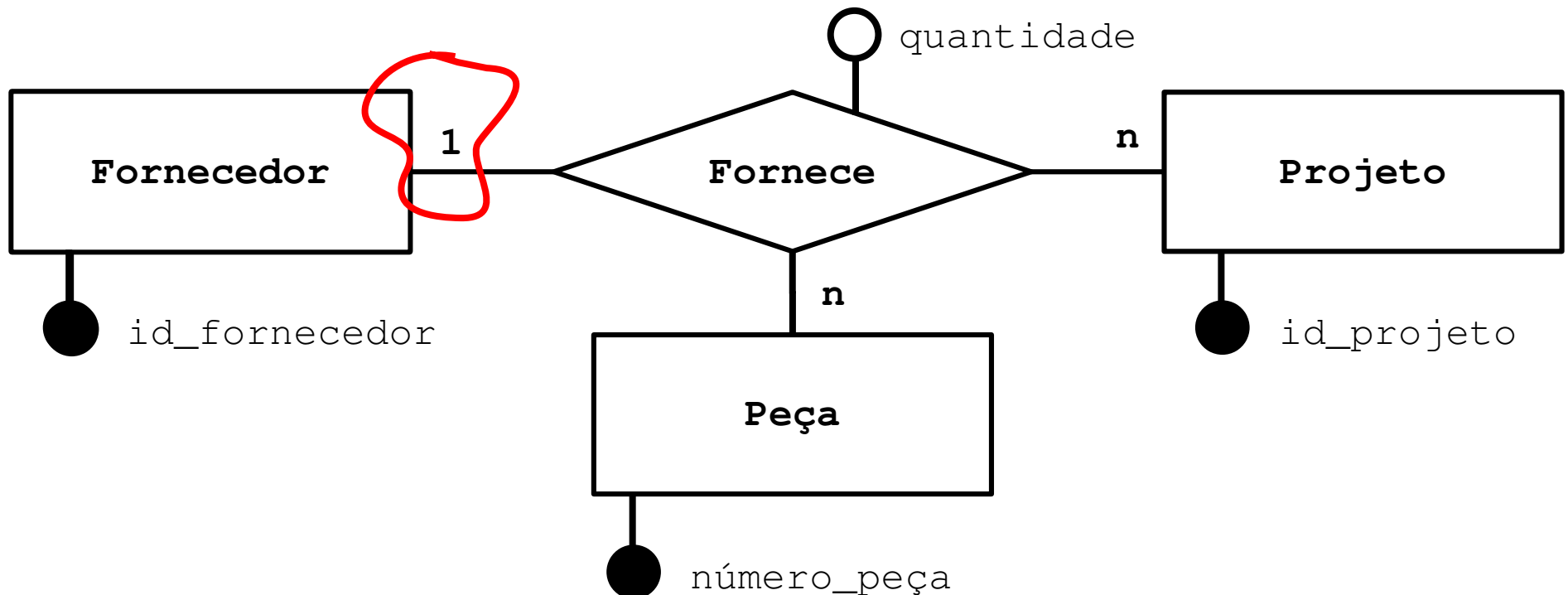
Fornece

Chave primaria

<u>id_fornecedor</u>	<u>id_projeto</u>	<u>codigo_peça</u>	quantidade
1001	10	111	20
1001	10	222	20
1002	10	111	20
1002	20	111	10
1003	20	111	30
1004	20	333	40

3. Mapeamento MER → Relacional:

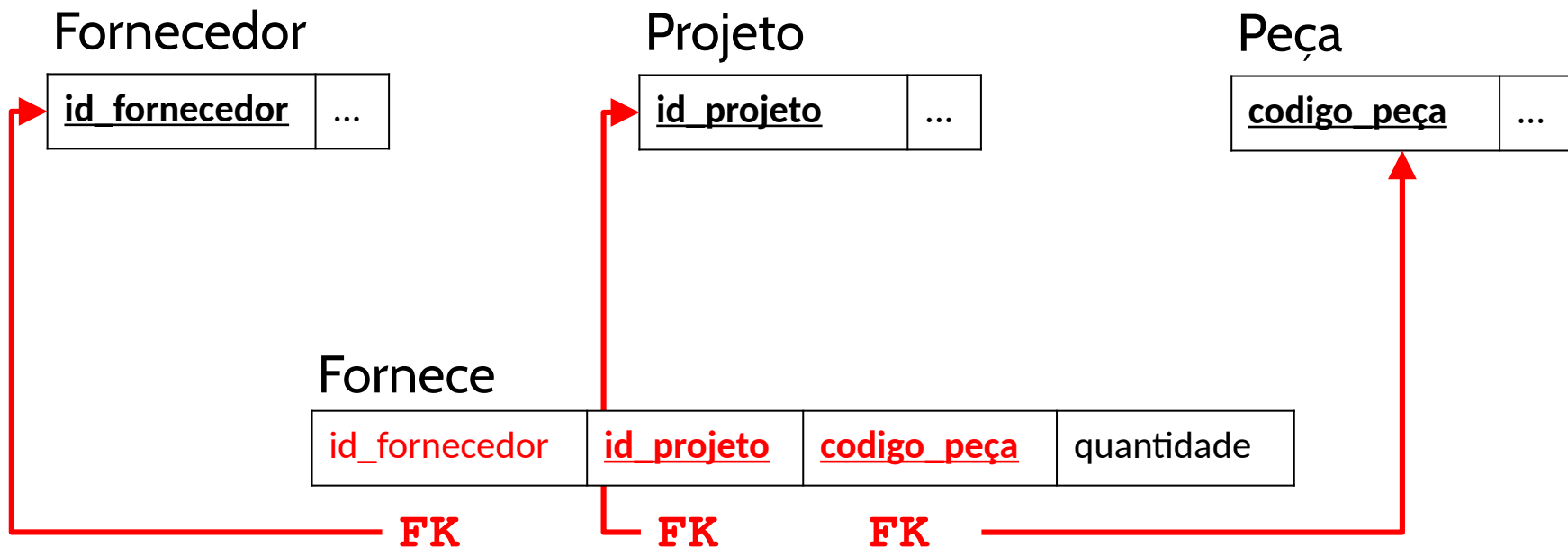
3.1 Relacionamentos não-binários (n-ários)





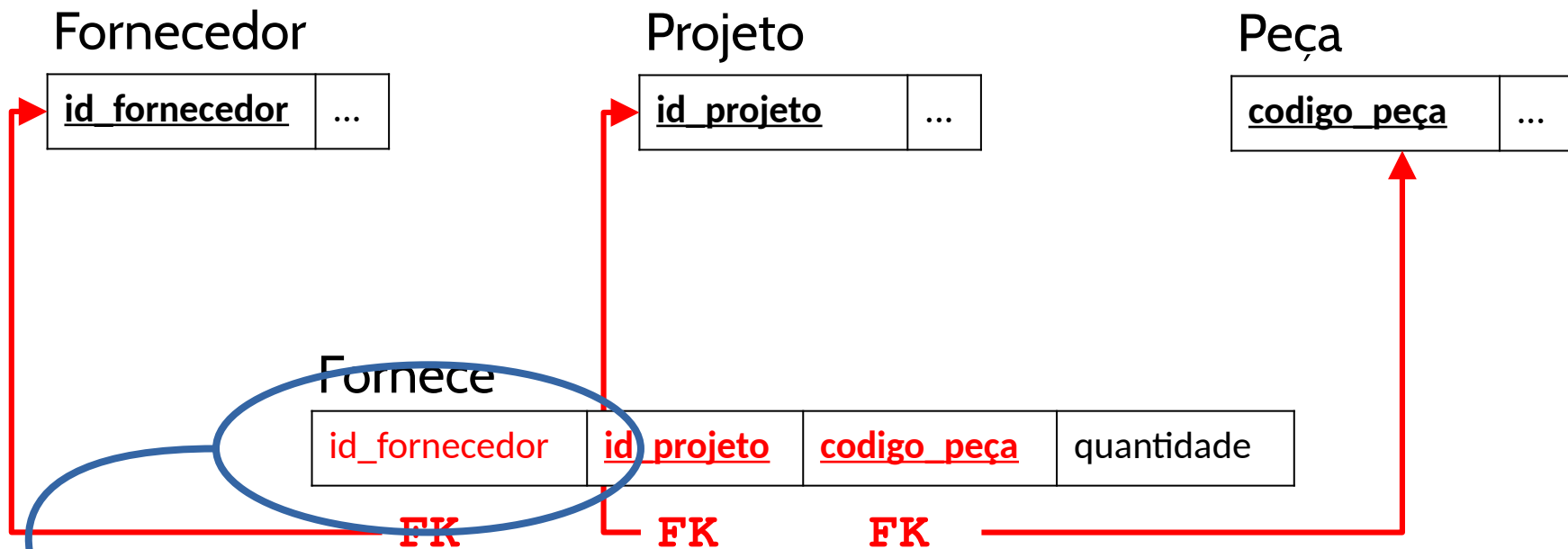
3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)



Note que a entidade de cardinalidade 1 entra mais na chave primária!



3. Mapeamento MER → Relacional:

3.1 Relacionamentos não-binários (n-ários)

Fornecedor

<u>id_fornecedor</u>	...
1001	...
1002	...
1003	...

Projeto

<u>id_projeto</u>	...
10	...
20	...
30	...

Peça

<u>número_peça</u>	...
111	...
333	...
222	...

Fornece

Chave primaria

<u>id_fornecedor</u>	<u>id_projeto</u>	<u>codigo_peça</u>	quantidade
1001	10	111	20
1001	10	222	20
1002	10	111	20
1002	20	111	10
1003	20	111	20
1004	20	333	40



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

→ Considerando superclasse C , cujos atributos são $\{k, a_1, a_2, \dots, a_n\}$ (sendo k a chave primária), podemos converter cada uma das especializações com m subclasses $\{S_1, S_2, \dots, S_m\}$ usando as 4 possíveis estratégias:



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

- 1)Múltiplas relações: superclasse + subclasses
- 2)Múltiplas relações: apenas subclasses
- 3)Única relação: atributo único especificando *tipo*
- 4)Única relação: vários atributos de *tipos*



3. Mapeamento MER \rightarrow Relacional:

3.2 Generalizações/Especializações

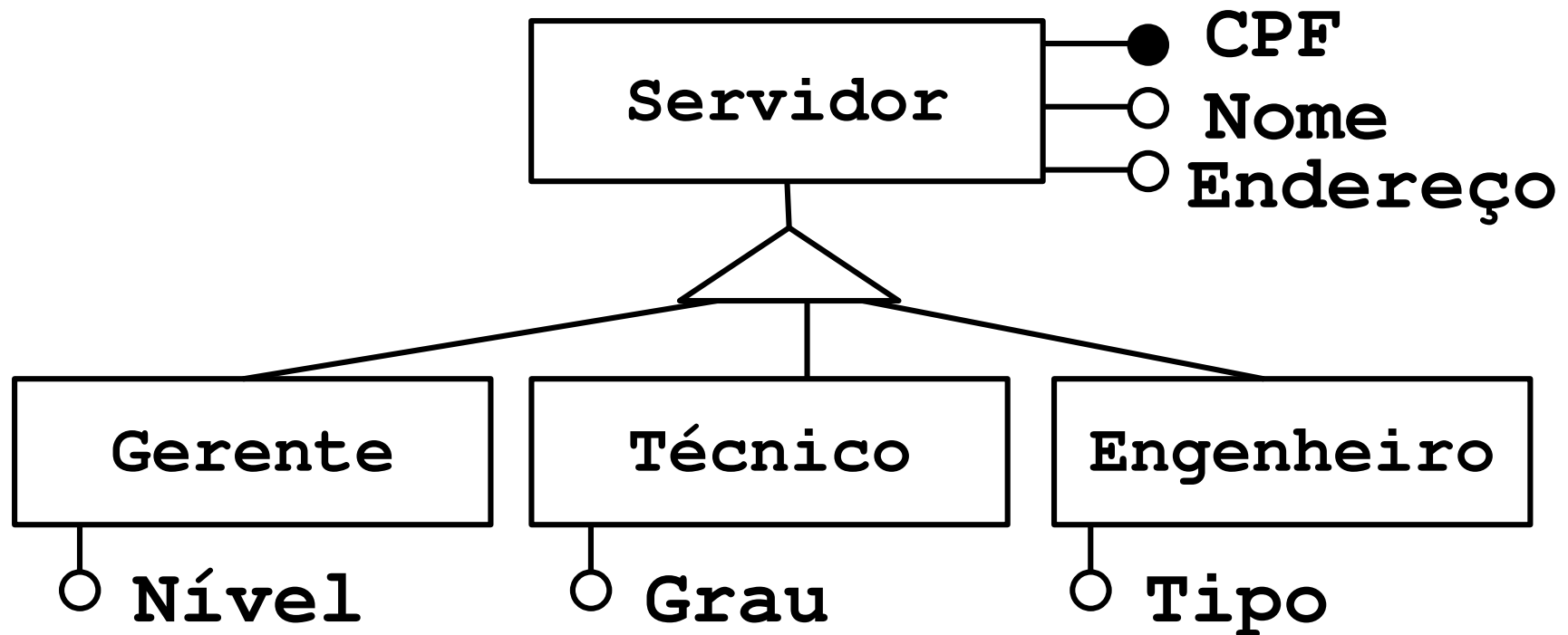
3.2.1 - Múltiplas relações: superclasse + subclasses

- Crie uma relação L para representar a superclasse C e inclua em L os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\}$ e a chave primária $\text{PK}(L) = k$;
- Crie uma relação L_i para cada subclasse S_i , onde $1 \leq i \leq m$ com os atributos $\text{attr}(L_i) = \{k\} \cup \{a_1, a_2, \dots, a_n\}$ e a chave primária $\text{PK}(L_i) = k$;

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.1 - Múltiplas relações: superclasse + subclasses

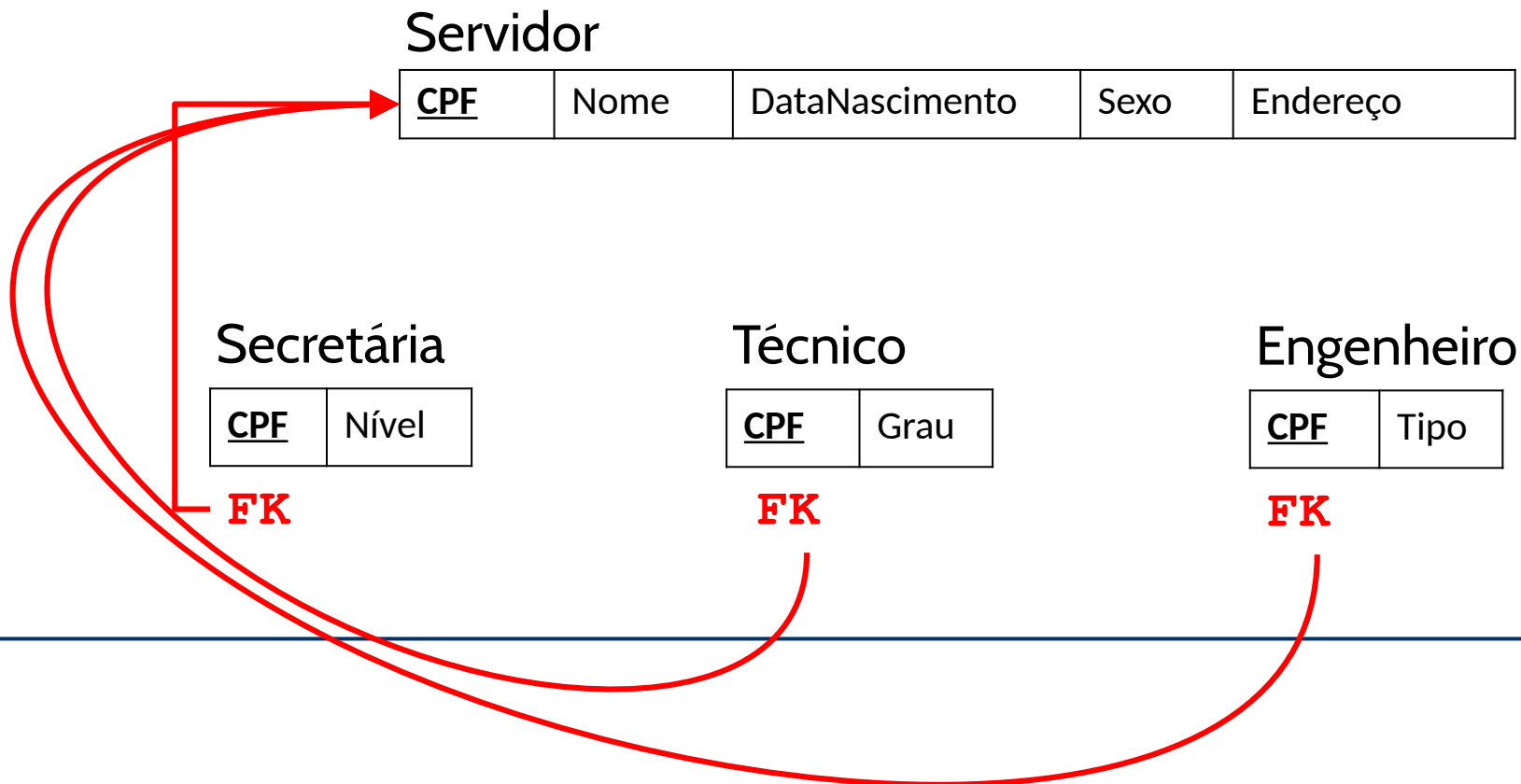




3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

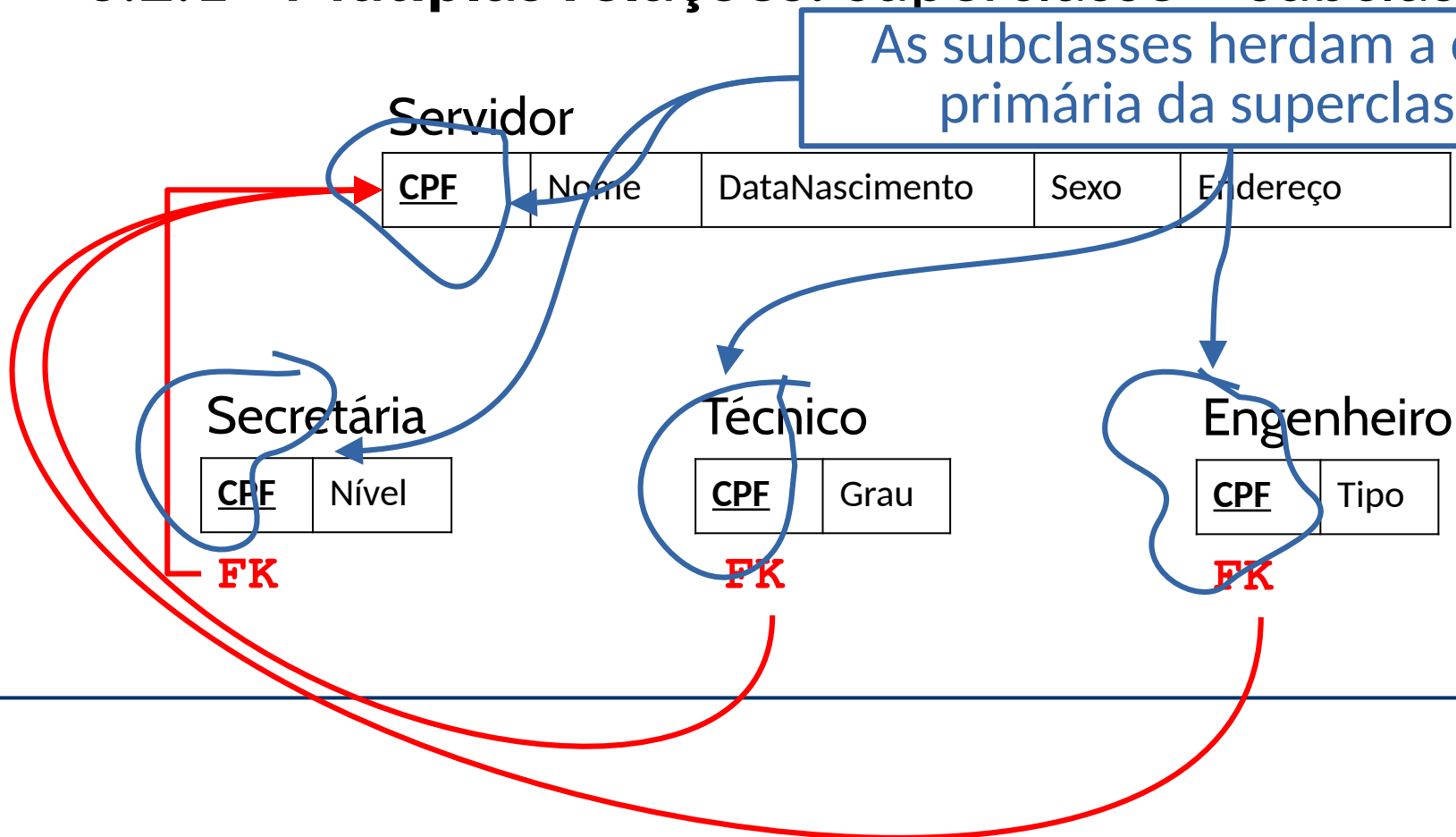
3.2.1 - Múltiplas relações: superclasse + subclasses



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.1 - Múltiplas relações: superclasse + subclasses





3. Mapeamento MER \rightarrow Relacional:

3.2 Generalizações/Especializações

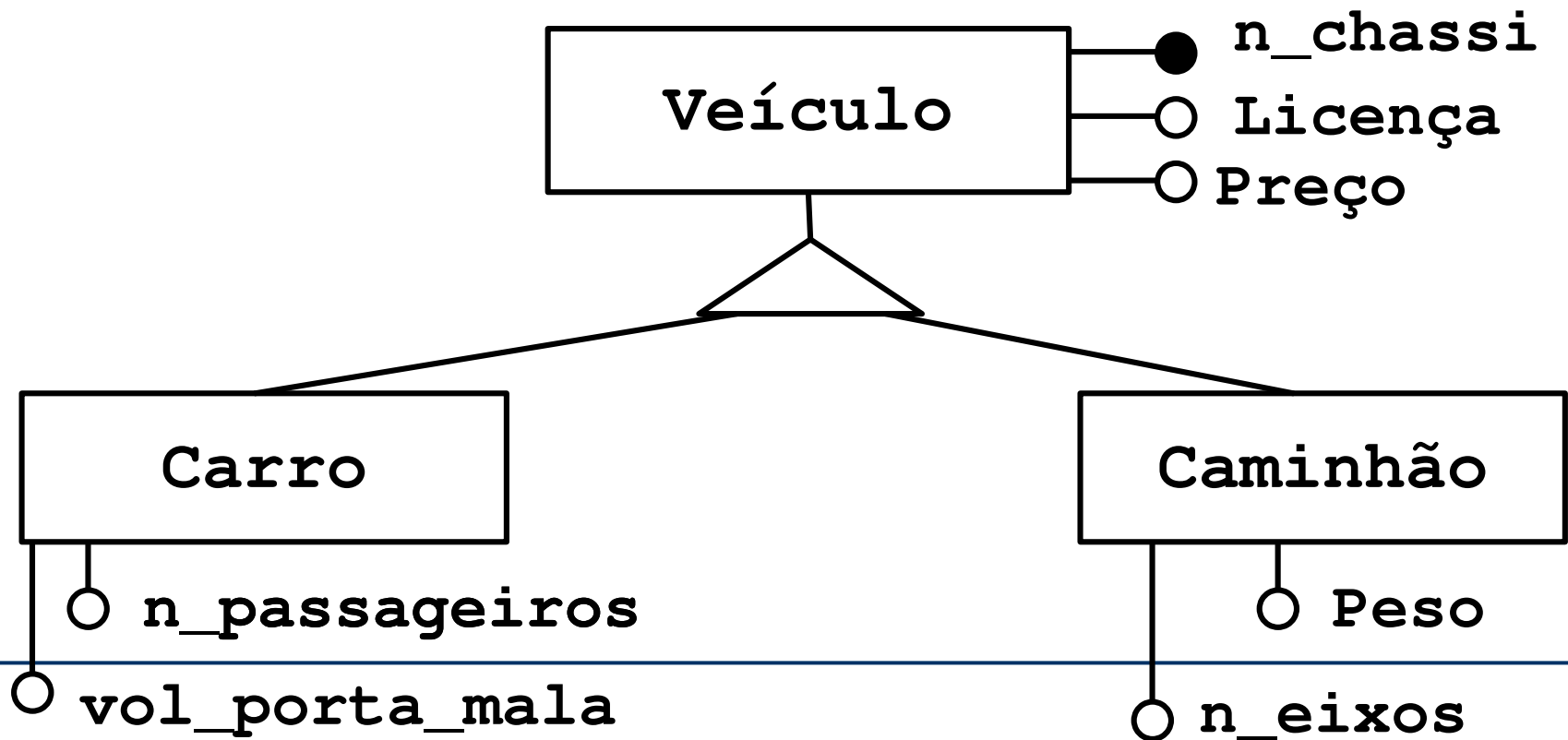
3.2.2 - Múltiplas relações: apenas subclasses

→ Crie uma relação L_i para cada subclasse S_i onde $1 \leq i \leq m$ com os atributos $\text{attr}(L_i) = \{\text{atributos de } S_i\} \cup \{k, a_1, a_2, \dots, a_n\}$ e a chave primária $PK(L_i) = k$;

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.2 - Múltiplas relações: apenas subclasses



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.2 - Múltiplas relações: apenas subclasses

Carro

<u>n_chassi</u>	Licença	preço	n_passageiros	vol_porta_mala
-----------------	---------	-------	---------------	----------------

Caminhão

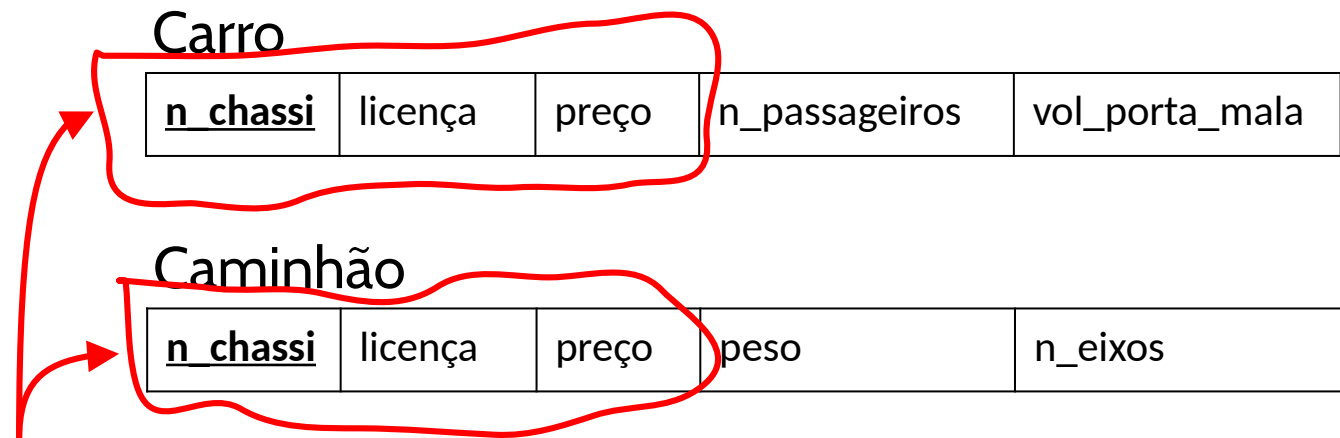
<u>n_chassi</u>	Licença	preço	peso	n_eixos
-----------------	---------	-------	------	---------

Apenas as subclasses
da entidade
“Veículo” são
transformadas em
relações/tabelas

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.2 - Múltiplas relações: apenas subclasses

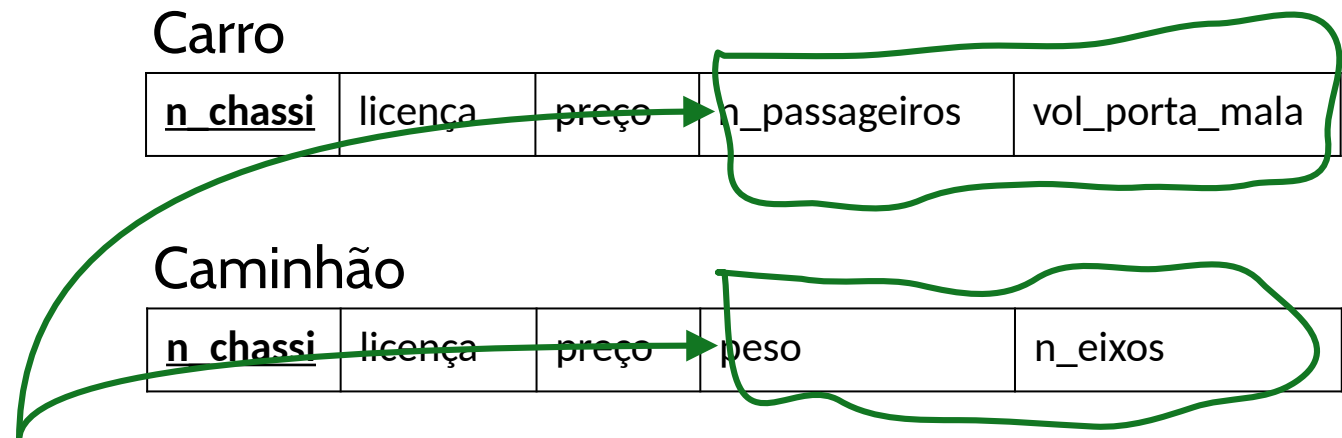


Note que os atributos (incluindo a chave) da superclasse são herdados e se repetem em ambas subclasses

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.2 - Múltiplas relações: apenas subclasses



Os atributos específicos não se sobrepõem



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

→ Crie uma relação L com os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S1\} \cup \{\text{atributos de } S2\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$ e a chave primária $PK(L) = k$;

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

→ Crie uma relação L com os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S1\} \cup \{\text{atributos de } S2\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$ e a chave primária $PK(L) = k$;

O atributo criado t é o atributo tipo (discriminador) e seu valor indica a subclasse à qual cada tupla pertence, caso ela pertença a alguma subclasse.



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

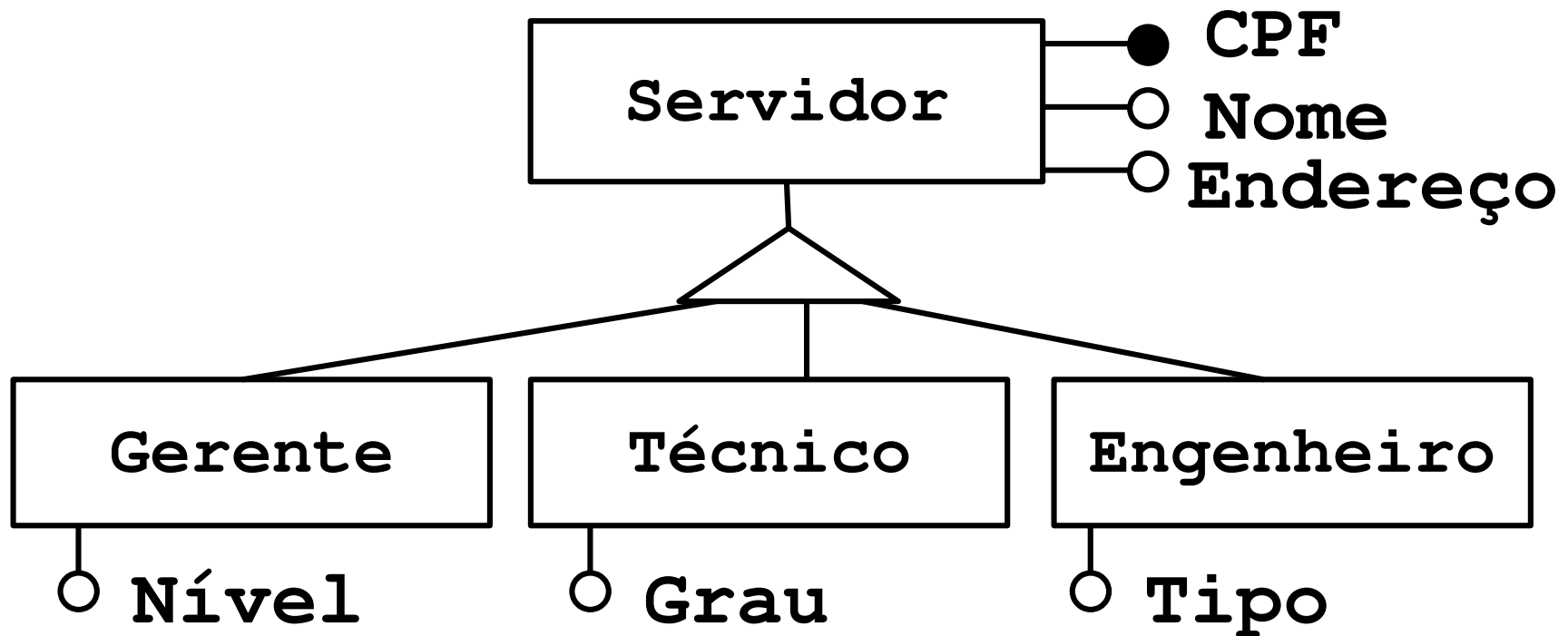
3.2.3 - Única relação: atributo único especificando tipo

- Crie uma relação L com os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S1\} \cup \{\text{atributos de } S2\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t\}$ e a chave primária $\text{PK}(L) = k$;
- Essa abordagem possui uma desvantagens: dependendo do relacionamento, ela pode requerer a criação de muitos valores **null**.

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo





3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

Servidor

<u>CPF</u>	Nome	DataNascimento	Sexo	Endereço	tipo	nível	grau	tipo_servidor
------------	------	----------------	------	----------	------	-------	------	---------------

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

Servidor

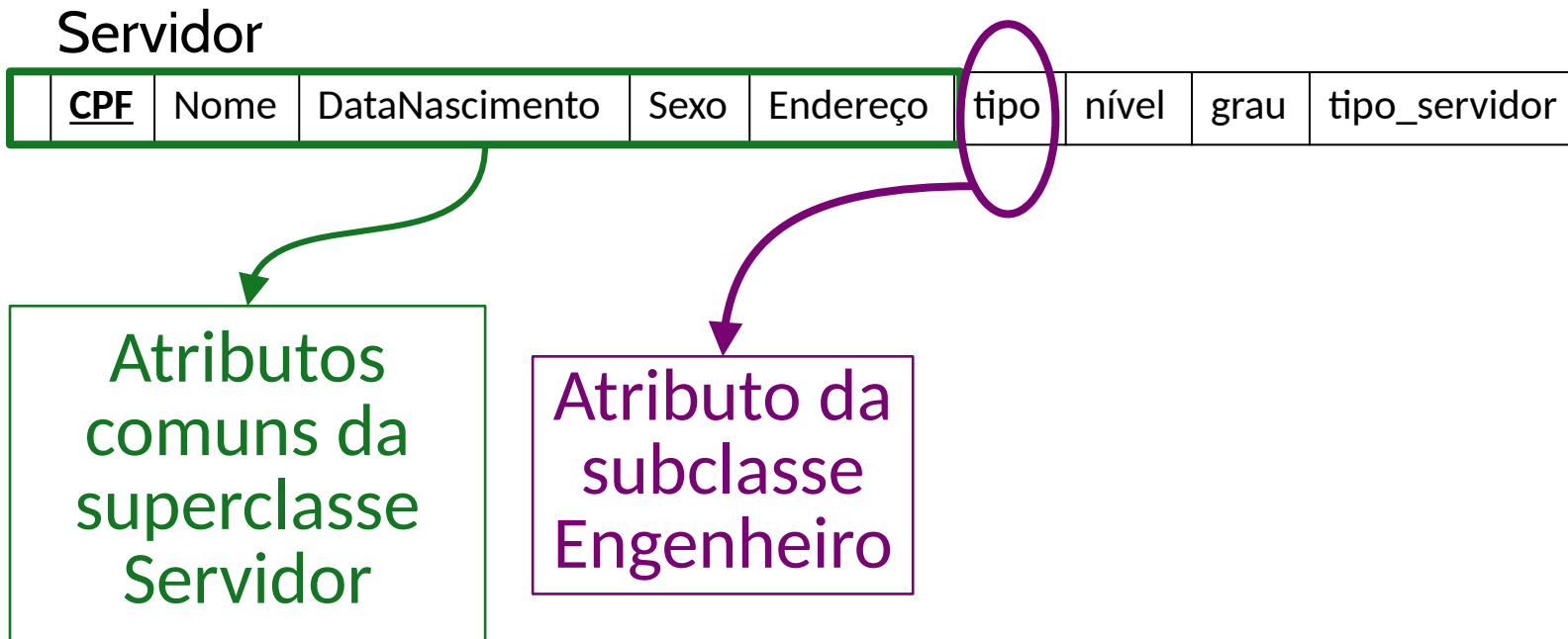
<u>CPE</u>	Nome	DataNascimento	Sexo	Endereço	tipo	nível	grau	tipo_servidor
------------	------	----------------	------	----------	------	-------	------	---------------

Atributos
comuns da
superclasse
Servidor

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

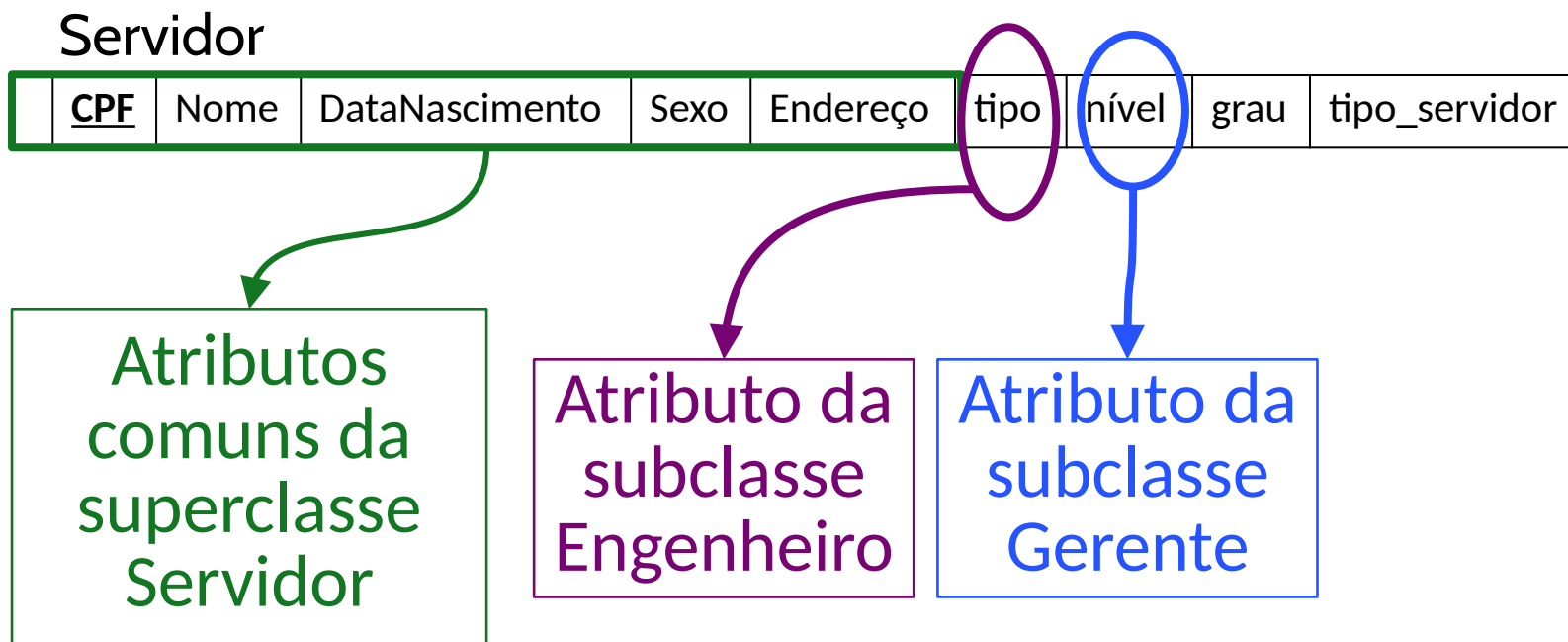
3.2.3 - Única relação: atributo único especificando tipo



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

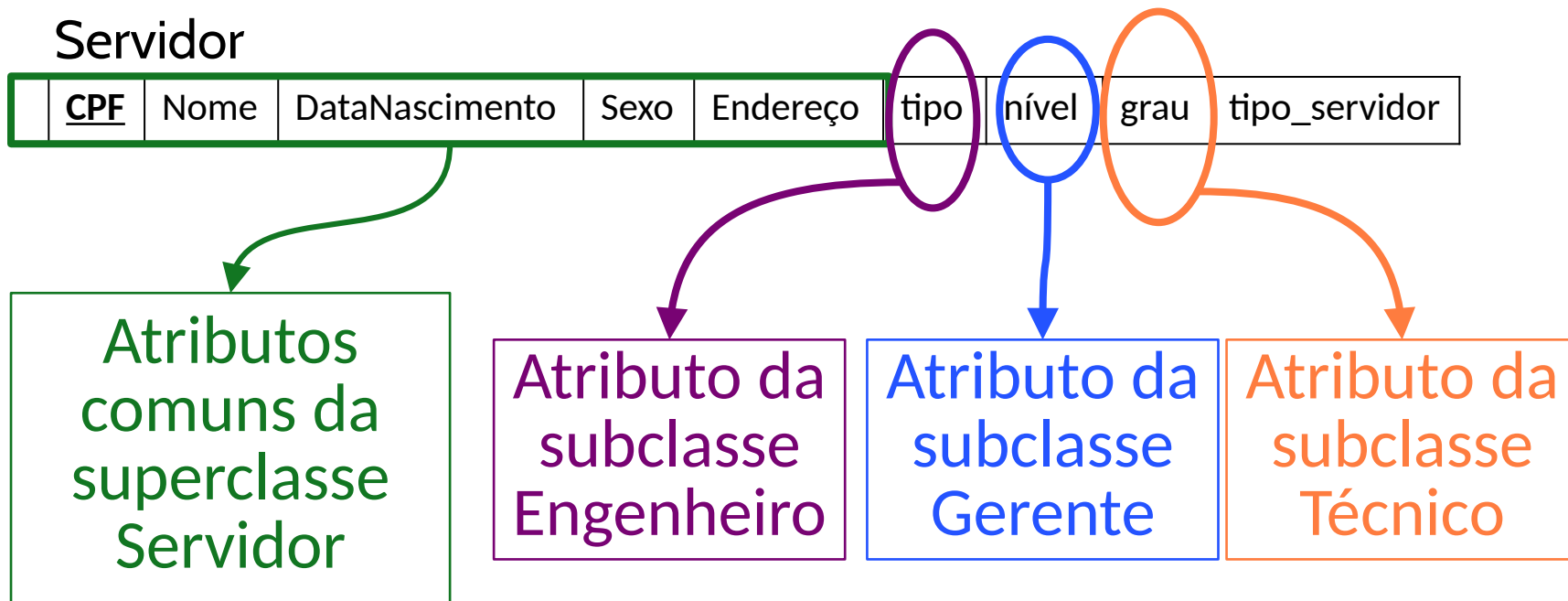
3.2.3 - Única relação: atributo único especificando tipo



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

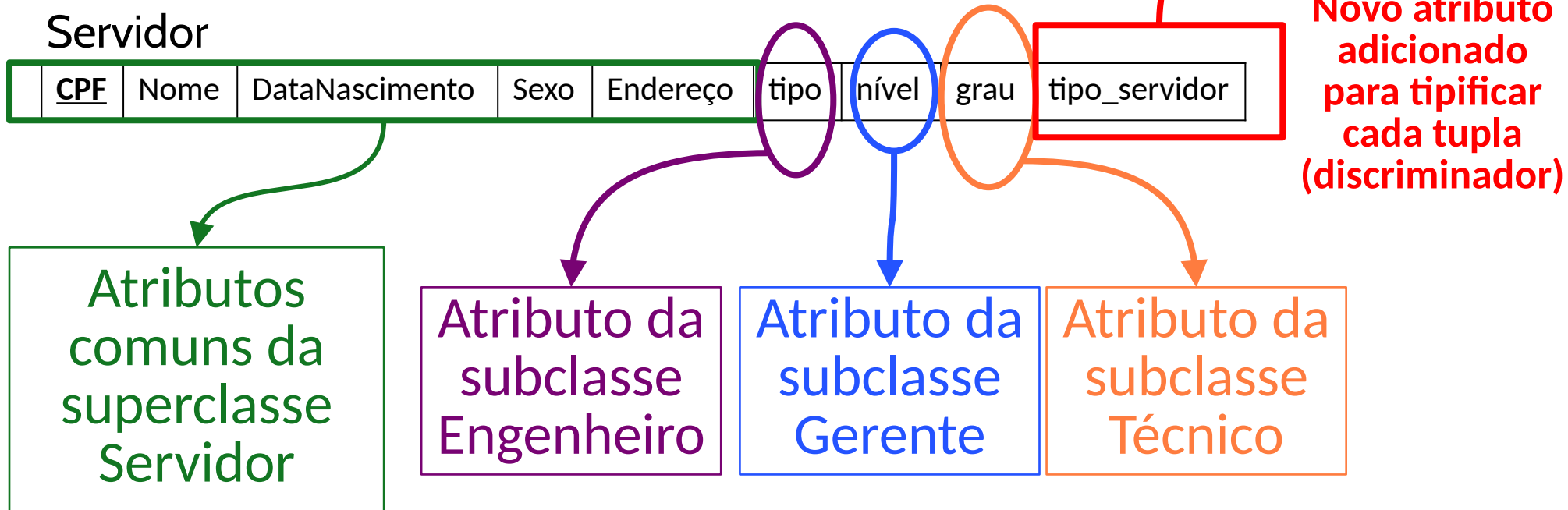
3.2.3 - Única relação: atributo único especificando tipo



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo





3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

Servidor

<u>CPE</u>	Nome	DataNascimento	Sexo	Endereço	tipo	nível	grau	tipo_servidor
01	Alice	06121984	F	DF	Civil	Null	Null	Engenheiro
10	Alice	01111942	F	SP	Null	C2	Null	Gerente
22	Paulo	03131985	M	RJ	Comp	Null	Null	Engenheiro
12	Beto	02021984	M	RJ	Null	Null	Medio	Técnico



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo

Servidor

<u>CPF</u>	Nome	DataNascimento	Sexo	Endereço	tipo	nível	grau	tipo_servidor
01	Alice	06121984	F	DF	Civil	Null	Null	Engenheiro
10	Alice	01111942	F	SP	Null	C2	Null	Gerente
22	Paulo	03131985	M	RJ	Comp	Null	Null	Engenheiro
12	Beto	02021984	M	RJ	Null	Null	Medio	Técnico

Note que essa abordagem pode gerar muitos valores Null na tabela.

3. Mapeamento MER \rightarrow Relacional:

3.2 Generalizações/Especializações

3.2.4 - Única relação: vários atributos de tipos

- Crie uma relação L com os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S1\} \cup \{\text{atributos de } S2\} \cup \dots \cup \{\text{atributos de } S_m\} \cup \{t_1, t_2, \dots, t_m\}$ e a chave primária $PK(L) = k$;
- Cada t_i , onde $1 \leq i \leq m$, é um atributo do tipo booleano que indica se uma tupla pertence a uma subclasse S_j .

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

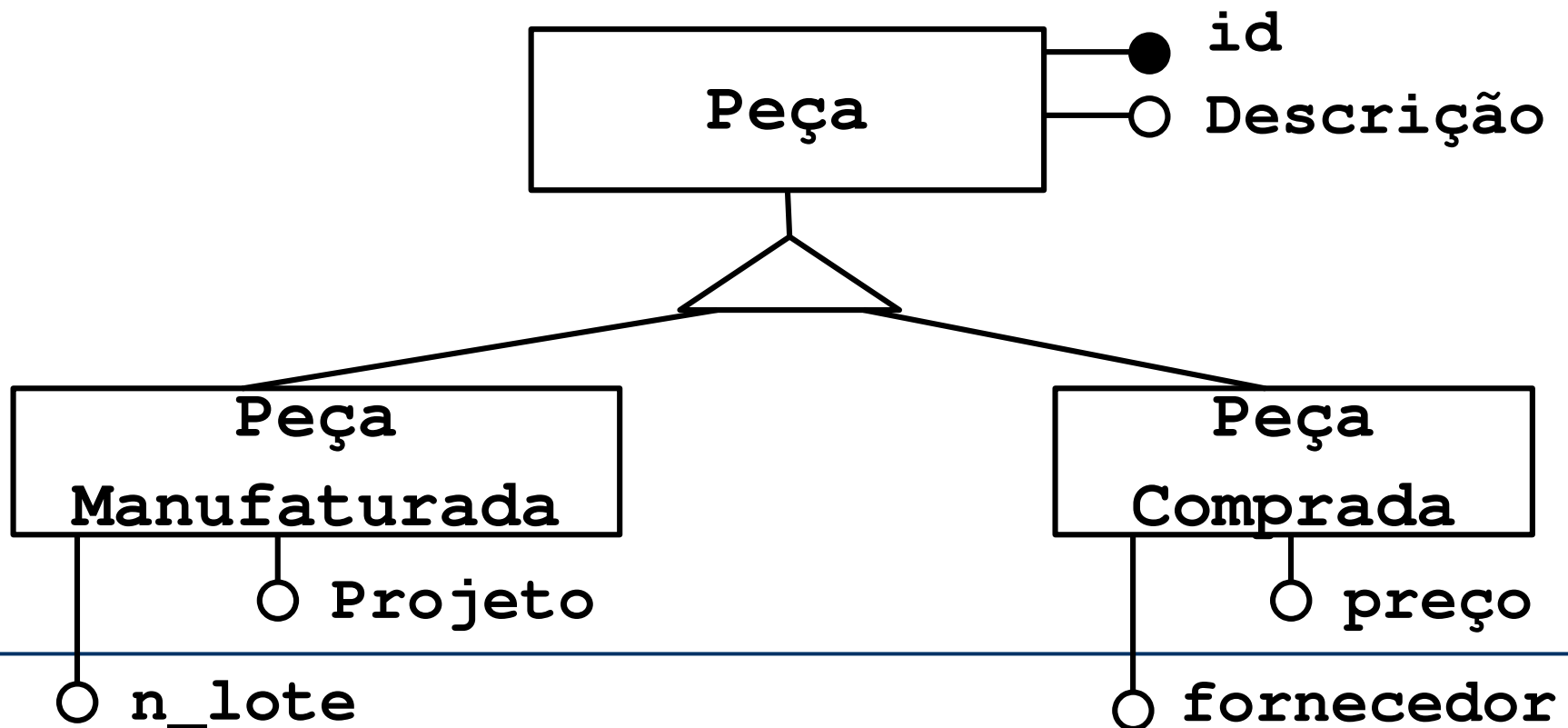
3.2.4 - Única relação: vários atributos de tipos

- Crie uma relação L com os atributos $\text{attr}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos } S_m\} \cup \{t_1, t_2, \dots, t_m\}$ e a chave primária $\text{PK}(L) = k$;
Várias colunas adicionadas
- Cada t_i , onde $1 \leq i \leq m$, é um atributo do tipo booleano que indica se uma tupla pertence a uma subclasse S_j .

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.3 - Única relação: atributo único especificando tipo





3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.4 - Única relação: vários atributos de tipos

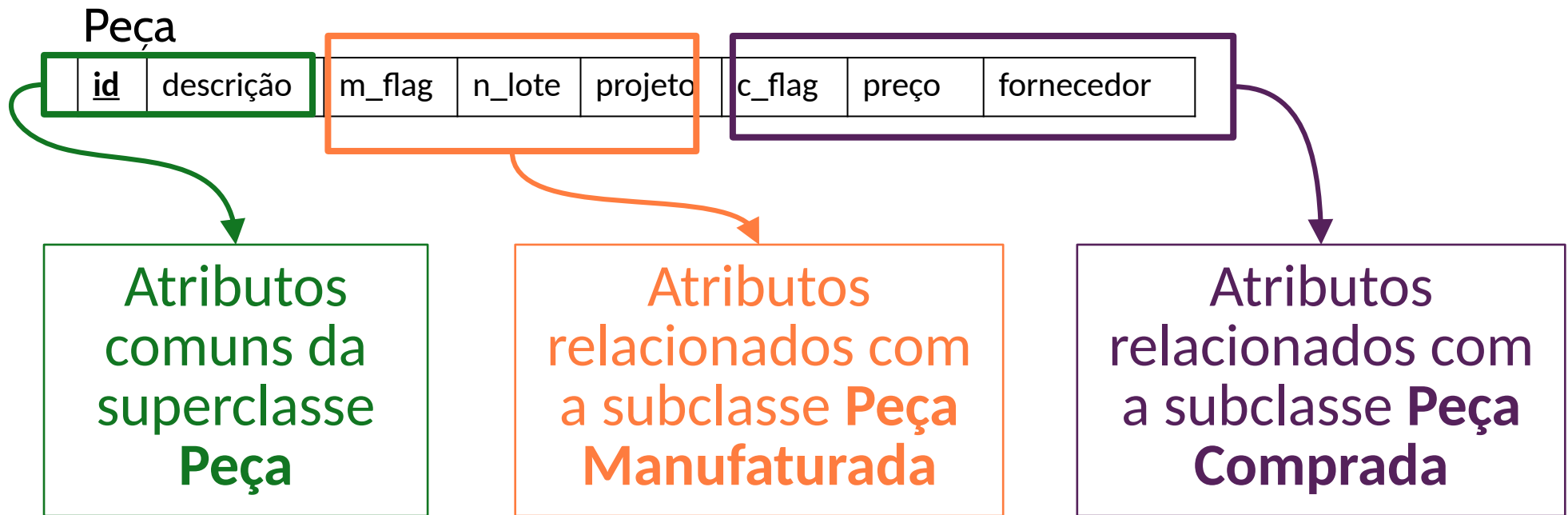
Peça

<u>id</u>	descrição	m_flag	n_lote	projeto	c_flag	preço	fornecedor
-----------	-----------	--------	--------	---------	--------	-------	------------

3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.4 - Única relação: vários atributos de tipos



3. Mapeamento MER → Relacional:

3.2 Generalizações/Especializações

3.2.4 - Única relação: vários atributos de tipos



4. Exercícios

Exercício 4.1: Considere o esquema textual de dados do modelo relacional definido parcialmente abaixo, onde **Dependente** é uma relação derivada de uma entidade fraca que se relaciona com Empregado.

Empregado (**CodigoEmpregado**, Nome, NoPIS-PASEP)

Dependente (CodigoEmpregado, NoDependente, Nome)

Nesse caso, na tabela **Empregado**, tanto **CodigoEmpregado** quanto **NoPIS-PASEP** podem ser chave primária. Qual você escolheria como chave primária? Porque?



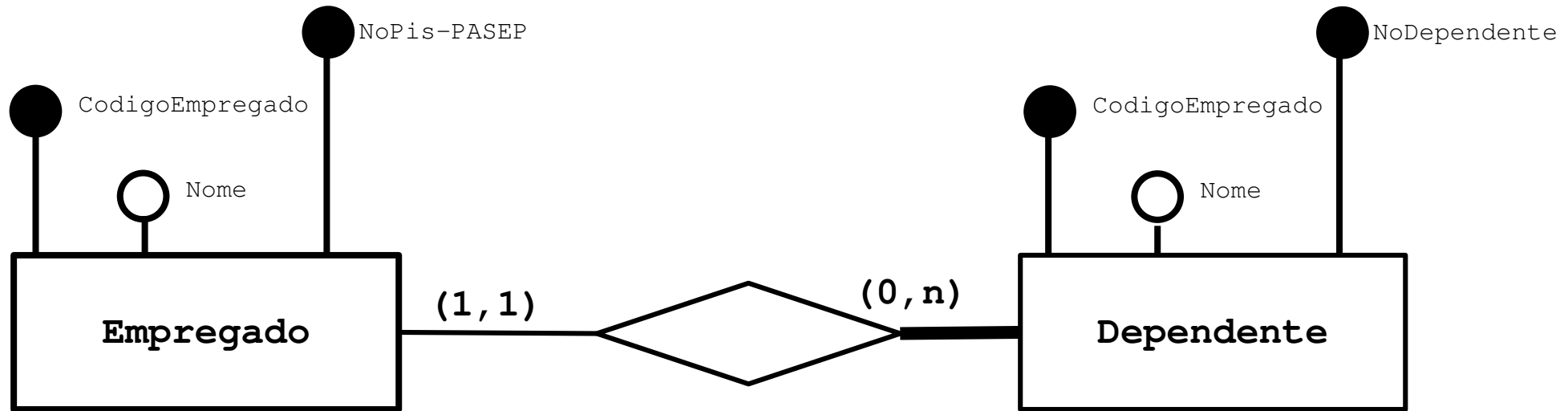
4. Exercícios

Exercício 4.1:

Escolheria o `CodEmpregado` como chave primária de **Empregado** pois este atributo está sendo usado na relação **Dependente** como chave estrangeira, sendo, desta maneira, estabelecida uma relação entre as duas relações.



4. Exercícios



Empregado

<u>CodEmpregado</u>	Nome	NoPis-PASEP
---------------------	------	-------------

Dependente

<u>CodigoEmpregado</u>	<u>NoDependente</u>	Nome
------------------------	---------------------	------



4. Exercícios

Exercício 4.2: Abaixo aparece um esquema textual parcial para um banco de dados relacional. Identifique neste esquema as chaves primárias e chaves estrangeiras:

Aluno (CodigoAluno, Nome, CodigoCurso)

Curso (CodigoCurso, Nome)

Disciplina (CodigoDisciplina, Nome, Creditos, CodigoDepartamento)

Curriculo (CodigoCurso, CodigoDisciplina, Obrigatória-Optional)

Conceito (CodigoAluno, CodigoDisciplina, Ano-Semestre, Conceito)

Departamento (CodigoDepartamento, Nome)



4. Exercícios

Exercício 4.2:

R: Chave primária / Chave estrangeira

Aluno (CodigoAluno, Nome, CodigoCurso)

Curso (CodigoCurso, Nome)

Disciplina (CodigoDisciplina, Nome, Creditos, CodigoDepartamento)

Curriculo (CodigoCurso, CodigoDisciplina, Obrigatória-Optional)

Conceito (CodigoAluno, CodigoDisciplina, Ano-Semestre, Conceito)

Departamento (CodigoDepartamento, Nome)

4. Exercícios

Exercício 4.3: Considere as seguintes alternativas de implementação de um banco de dados relacional:

- Alternativa 1
 - `Aluno (`
 `CodAl`,
 `Nome, CodCurso, Endereco)`
- Alternativa 2
 - `Aluno (CodAl, Nome, CodCurso)`
 `EnderecoAluno (CodAl, Endereco)`
 `CodAl referencia Aluno`

Em ambos os casos está sendo representado um conjunto de alunos e informações (código, nome, código de curso, endereço) a ele referentes. À luz dos princípios que baseiam as regras de tradução de diagramas ER, qual você escolheria? Por que?



4. Exercícios

Exercício 4.3:

- Alternativa 1

- `Aluno (`
 `CodAl`,
 `Nome, CodCurso, Endereco)`

- Alternativa 2

- `Aluno (CodAl, Nome, CodCurso)`
 `EnderecoAluno (CodAl, Endereco)`
 `CodAl referencia Aluno`

4. Exercícios

Exercício 4.3:

- Alternativa 1
 - `Aluno (`
 `CodAl`,
 `Nome, CodCurso, Endereco)`

- A alternativa representa o mesmo número de atributos usando apenas uma tabela.
- A alternativa não requer junções (join). Na outra alternativa, cada vez que os atributos do aluno forem necessários, será necessário fazer uma junção das 2 tabelas.
- A alternativa requer um menor número de chaves primárias.
- A alternativa não requer uso de chaves estrangeiras.
- Menor *overhead*.





Dúvidas?

