

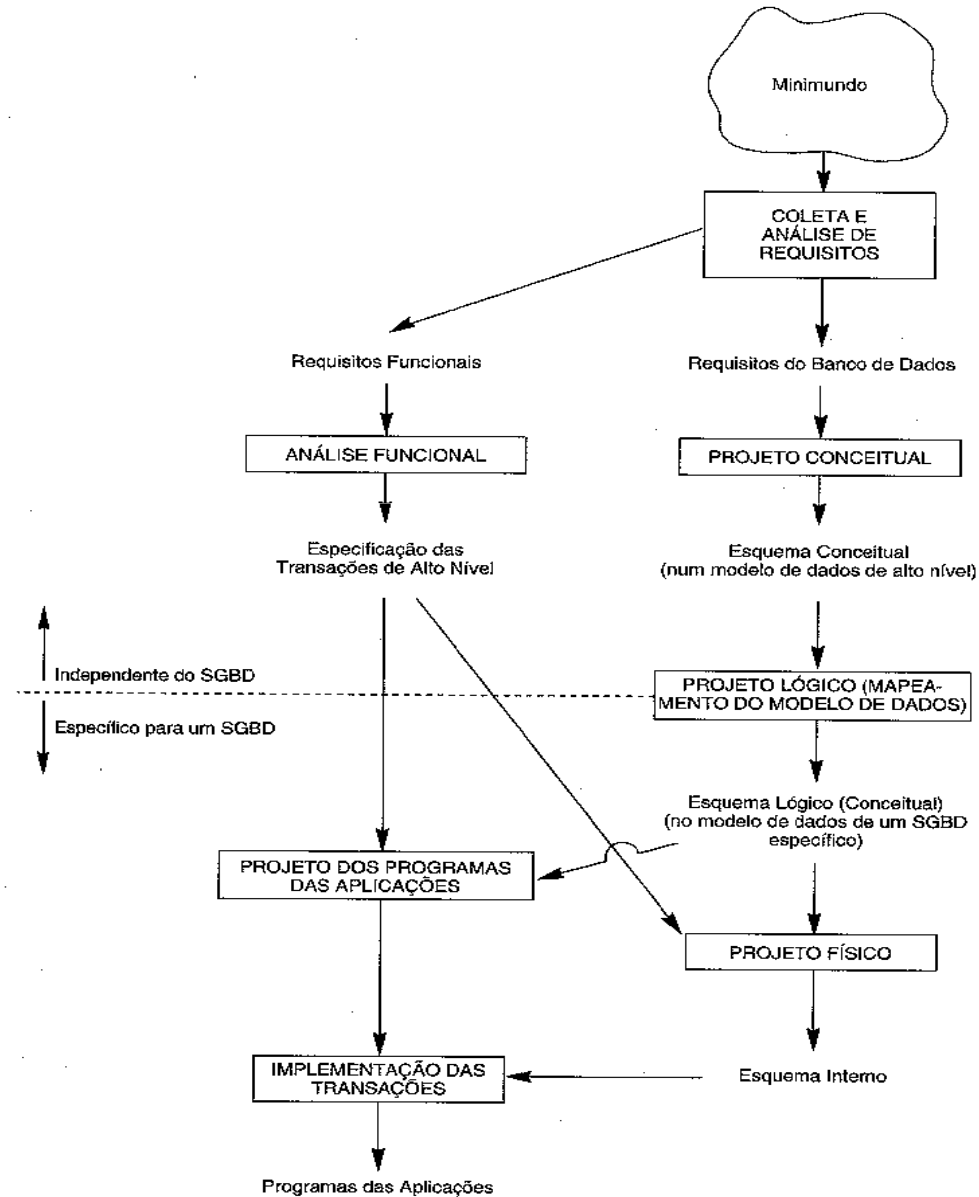
Modelagem de Dados

Banco de Dados I

Prof. Guilherme Tavares de Assis

Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM

Fases do Projeto de um Banco de Dados



Modelo ER

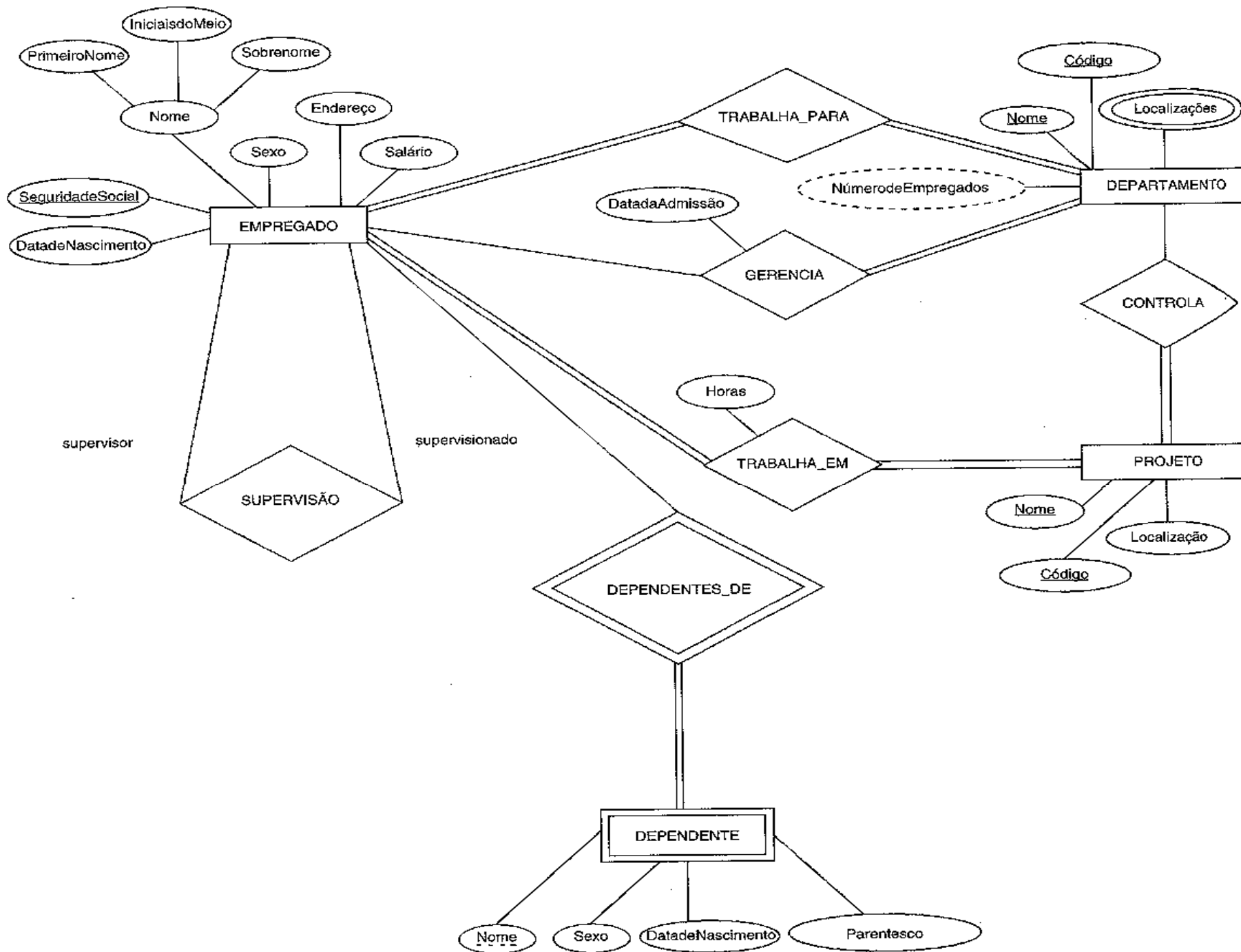
- O **Modelo de Entidades e Relacionamentos (MER)** é um modelo conceitual usado para projeto de aplicações de banco de dados.
- É um modelo baseado na percepção do mundo real como conjuntos de objetos básicos chamados entidades e nos relacionamentos entre esses objetos.
- É independente de aspectos de implementação.

Exemplo de Aplicação de Banco de Dados

- Deseja-se criar um BD "Empresa" que controla empregados, departamentos e projetos de uma empresa.
- Após a fase de coleta e análise dos requisitos, os projetistas declararam a seguinte descrição do "mini-mundo":
 - A empresa é organizada em departamentos. Cada departamento possui um nome único, um código único e um determinado empregado que gerencia o departamento. Acompanha-se a data inicial que o empregado começou a gerenciar o departamento. Um departamento pode possuir diversas localizações.
 - Um departamento controla um número de projetos onde cada um deles possui um nome único, um código único e uma localização única.

Exemplo de Aplicação de Banco de Dados

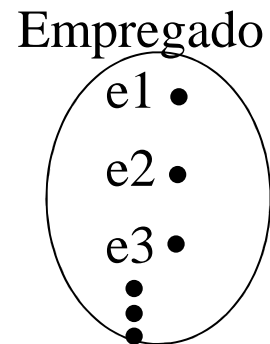
- Armazenam-se o nome, o número da seguridade social, o endereço, o salário, o sexo e a data de nascimento de cada empregado. Um empregado é alocado a um único departamento, mas pode trabalhar em diversos projetos, que não são necessariamente controlados pelo mesmo departamento. Acompanha-se o número de horas por semana que um empregado trabalha em cada projeto. Também acompanha-se o supervisor direto de cada empregado.
- Acompanham-se os dependentes de cada empregado para fins de seguridade social. São mantidos, para cada dependente, o nome, o sexo, a data de nascimento e o grau de parentesco com o empregado.



Entidade

- **Entidade** é um objeto que existe no mundo real e é distinguível dos outros objetos. Exemplos:
 - O empregado João da Silva com o CPF 890.123.456
 - O estudante José Carlos Albuquerque com matrícula 2.034
- **Tipo de Entidade** define uma coleção de entidades que possuem os mesmos atributos. Descreve o esquema para um conjunto de entidades.
- **Conjunto de Entidades** é uma coleção das instâncias de entidades de um determinado tipo de entidade.

Exemplo: o conjunto de todos os empregados pode ser definido como um conjunto de entidades Empregado.



Atributo

- **Atributo** é uma propriedade que descreve uma característica particular de uma entidade.
 - Ex.: uma entidade *Empregado* tem os atributos *nome*, *CPF*, *endereço*, *salário*, *sexo* e *data de nascimento* com os valores "João da Silva", "890.123.456" , "Contorno, 1900, Centro", R\$ 1.500,00, "M" e 20/10/1970, respectivamente.
- Os atributos podem ser dos seguintes tipos:
 - **Simples (atômicos)** ou **Compostos**: os atributos simples são indivisíveis e os atributos compostos podem ser divididos em subpartes menores.
 - Ex.: o atributo *nome* da entidade *Empregado* pode ser dividido em primeiro nome e sobrenome. Assim, *nome* é um atributo composto. O atributo *salário* é simples.

Atributo

- **Mono-valorados** ou **Multi-valorados**: os atributos mono-valorados têm um único valor para uma entidade particular e os atributos multi-valorados podem ter um conjunto de valores.
 - Ex.: o atributo *nome* da entidade *Departamento* é um atributo mono-valorado e o atributo *localizações* é multi-valorado (um departamento pode estar localizado em mais de lugar).
- **Armazenados** ou **Derivados**: os atributos derivados podem ser obtidos a partir dos atributos armazenados (e portanto não precisam ser armazenados no banco de dados).
 - Ex.: o atributo *no_de_empregados* da entidade *Departamento* pode ser calculado através do relacionamento *trabalha_para* com a entidade *Empregado*. Um atributo *idade* pode ser calculado a partir de um atributo *data de nascimento*.

Atributo

- **Atributos complexos:** os atributos podem ser compostos e multi-valorados ao mesmo tempo.
 - Exemplo: um atributo *endereço* de uma entidade *Pessoa* pode ser composto por *rua*, *numero*, *complemento*, *cidade*, *estado* e *cep*. Uma mesma pessoa pode ter mais de um endereço (comercial e residencial).
- **Valores nulos:** uma entidade pode não ter valor aplicado a um atributo. Neste caso, é usado o valor especial nulo. É também usado quando não se conhece o valor de um atributo.
 - Exemplo: um atributo *apartamento* do endereço de uma entidade *empregado* será nulo se ele morar em uma casa.

Atributo

- Cada atributo de um tipo de entidade está associado com um **conjunto de valores** (ou **domínio de valores**), o qual especifica os valores que podem ser atribuídos àquele atributo para cada instância de entidade.
 - Exemplo:
 - O conjunto de valores para o atributo *nome* de *Empregado* pode ser definido como uma cadeia de caracteres.
 - Se o intervalo de salários permitidos para os empregados está entre 700,00 e 7.000,00 reais, então o conjunto de valores para o atributo *salário* de *Empregado* deve ser um número real entre 700 e 7.000.

Chave

- **Atributo Chave** (identificador): é um atributo cujos valores são distintos (únicos) para cada instância de entidade de um mesmo tipo de entidade.
 - A chave pode ser formada por mais de um atributo (chave composta). Neste caso, a combinação dos valores desses atributos é que é única para cada instância de entidade.
 - Exemplos:
 - O atributo *seguridade_social* do tipo de entidade *Empregado*.
 - Os atributos *seguridade_social_empregado* + *nome_dependente* do tipo de entidade *Dependente*.

Chave

- **Superchave** é qualquer conjunto de um ou mais atributos cujos valores são distintos para cada instância de entidade.
 - Exemplos:
 - O atributo *seguridade_social* do tipo de entidade *Empregado* é uma superchave.
 - Os atributos *seguridade_social* + *nome* também é uma superchave do tipo de entidade *Empregado*.
 - Os atributos *seguridade_social* + *nome* + *salario* também é uma superchave do tipo de entidade *Empregado*.

Chave

- **Chave Candidata** é uma superchave que não possui subconjuntos próprios de superchaves, ou seja, é uma superchave com um conjunto mínimo de atributos.
 - Exemplos:
 - No exemplo anterior, apenas o atributo *seguridade_social* é uma chave candidata para o tipo de entidade *Empregado*.
 - Para o tipo de entidade *Departamento*, tanto o atributo *código* quanto o atributo *nome* do departamento são chaves candidatas.

Chave

- **Chave Primária** é a chave candidata escolhida arbitrariamente pelo projetista do banco de dados como meio principal de identificação de entidades para o tipo de entidade em questão.
 - As demais chaves candidatas são chamadas de **chaves alternativas** ou **chaves secundárias**.
 - Exemplo:
 - Para o tipo de entidade *Departamento*, o atributo *código* poderia ser escolhido para ser a chave primária; neste caso, o atributo *nome* seria uma chave alternativa ou secundária.

Projeto Conceitual inicial do BD Empresa

- De acordo com a descrição do "mini-mundo" da empresa, identifica-se os seguintes tipos de entidade:
 - *Departamento* com os atributos *nome*, *código*, *localizações*, *gerente*, *data_início_gerência*. *Localizações* é o único atributo multivalorado. Pode-se especificar que tanto *nome* quanto *código* são atributos chaves (separados), porque cada um foi especificado para ser único.
 - *Projeto* com os atributos *nome*, *código*, *localização* e *departamento_controlador*. Tanto o *nome* quanto o *código* são atributos chaves (separados).

Projeto Conceitual inicial do BD Empresa

- *Empregado* com os atributos *nome*, *seguridade_social* (carteira profissional), *sexo*, *endereço*, *salário*, *data_nascimento*, *departamento*, *supervisor*. Tanto o *nome* quanto o *endereço* podem ser atributos compostos (não foi especificado). O fato de que um empregado poder trabalhar em vários projetos, um certo número de horas por semana para cada projeto, pode ser representado pelo atributo composto multi-valorado *trabalha_em* com os componentes (*projeto*, *horas*).
- *Dependente* com os atributos *empregado*, *nome_dependente*, *sexo*, *data_nascimento*, *parentesco* (com o empregado).

Relacionamento

- **Relacionamento** é uma associação entre entidades que representa um fato do mundo real.
- **Tipo de Relacionamento R** sobre **n** tipos de entidade **E₁**, **E₂**, ..., **E_n** define um conjunto de associações entre entidades destes tipos.
 - Exemplo:
 - O tipo de relacionamento *trabalha_para* entre os tipos de entidade *Empregado* e *Departamento* associa cada empregado com o departamento para o qual ele trabalha.

Relacionamento

- **Grau de um tipo de relacionamento** é o número de tipos de entidades participantes.
 - Exemplos:
 - Grau 2 (binário): o tipo de relacionamento *trabalha_para*.
 - Grau 3 (ternário): o tipo de relacionamento entre *fornecedor*, *peça* e *projeto*, onde fornecedores fornecem peças para os projetos.
- Tipos de relacionamento podem ter atributos, similares àqueles dos tipos de entidade.
 - Exemplo:
 - Atributo *horas* do tipo de relacionamento *trabalha_em* entre *Empregado* e *Projeto*, que indica o número de horas por semana que um empregado trabalha em um projeto.

Relacionamento

- **Tipo de relacionamento recursivo** ocorre quando o mesmo tipo de entidade participa mais de uma vez em um tipo de relacionamento desempenhando diferentes papéis.
 - Exemplo:
 - Tipo de relacionamento *supervisão* entre dois tipos de entidade *Empregado*, indicando que um determinado empregado supervisiona outros empregados. Neste caso, deve-se dar nome aos papéis de cada um dos lados do tipo de relacionamento (um empregado participa do relacionamento no papel de supervisor ou de supervisionado).

Restrição de Cardinalidade

- **Restrição de cardinalidade** especifica o número de instâncias de um relacionamento que uma instância de entidade pode participar.
- Para um tipo de relacionamento binário **R** entre os tipos de entidade **A** e **B**, tem-se as relações de cardinalidade:
 - **1:1** (um-para-um): uma entidade de **A** está associada a, no máximo, uma entidade de **B**, e uma entidade de **B** está associada a, no máximo, uma entidade de **A**.
 - **1:N** (um-para-muitos) - uma entidade de **A** está associada a várias (zero ou mais) entidades de **B**, mas uma entidade de **B** está associada a, no máximo, uma entidade de **A**.
 - **M:N** (muitos-para-muitos) - uma entidade de **A** está associada a várias entidades de **B** (zero ou mais), e uma entidade de **B** está associada a várias entidades de **A** (zero ou mais).

Restrição de Participação

- **Restrição de participação** especifica se a existência de uma entidade depende de sua associação a outra entidade por meio de um relacionamento. A participação pode ser:
 - **Total** (dependência de existência): todas as instâncias do tipo de entidade **devem** participar de um relacionamento.
 - Exemplo: no tipo de relacionamento *gerencia* entre *Empregado* e *Departamento*, a participação é total do lado de *Departamento*, informando que todo departamento tem obrigatoriamente um gerente (indicado no diagrama pelas linhas duplas).
 - **Parcial**: as instâncias **podem** participar de um relacionamento.
 - Exemplo: no tipo de relacionamento *gerencia* entre *Empregado* e *Departamento*, a participação é parcial do lado de *Empregado*, informando que nem todo empregado é gerente de departamento (indicado no diagrama por uma linha simples).

Restrição de Cardinalidade Mínima e Máxima

- Uma restrição estrutural alternativa em relacionamentos no Diagrama ER é a **Cardinalidade Mínima e Máxima**.
 - Envolve associar um par de números inteiros (min , max) a cada participação de um tipo de entidade E num tipo de relacionamento, onde $0 \leq min \leq max$ e $max \geq 0$.
 - Os números min e max significam que para cada instância de entidade e em E , e deve participar de, no mínimo, min e, no máximo, max instâncias do relacionamento em questão.
 - Neste método, $min=0$ implica participação parcial e $min>0$ implica participação total.

Entidade Fraca

- **Tipo de Entidade Fraca** é um tipo de entidade que não possui atributos chaves (não tem identificação própria).
 - Está sempre associada a um **tipo de entidade forte** (ou **proprietária**) através de um **relacionamento identificador**.
 - Sua chave é formada pela combinação da chave primária do tipo de entidade forte + uma **chave parcial** própria.
 - Sempre tem uma restrição de participação total (dependência de existência) em relação ao relacionamento identificador.
 - É representada no diagrama por contornos duplos.
 - Algumas vezes também pode ser modelada como atributos complexos (compostos e multi-valorados).
- Exemplo: tipo de entidade *Dependente*.

Refinamento do Projeto do BD Empresa

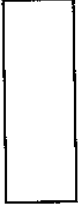
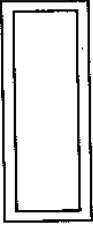
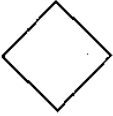
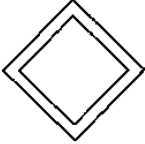



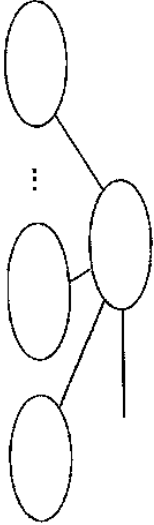

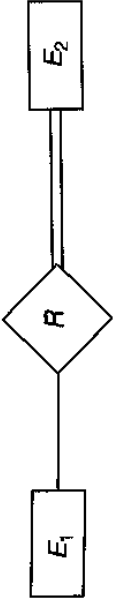
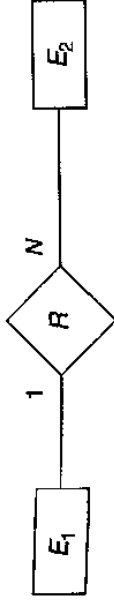
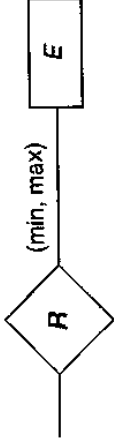
- Os atributos que representam relacionamentos devem ser transformados em tipos de relacionamento. Assim, têm-se os seguintes tipos de relacionamento no BD Empresa:
 - *Gerencia* é um tipo de relacionamento 1:1 entre *Empregado* e *Departamento*. A participação de *Empregado* é parcial. A participação de *Departamento* é total já que foi dito que um departamento sempre deve possuir um gerente. O atributo *data_inicio_gerência* é designado ao relacionamento.
 - *Trabalha_para* é um tipo de relacionamento N:1 entre *Empregado* e *Departamento*. Ambos têm participação total.

Refinamento do Projeto do BD Empresa

- *Controla* é um tipo de relacionamento 1:N entre *Departamento* e *Projeto*. A participação de *Projeto* é total, enquanto que a de *Departamento* foi determinada como sendo parcial (após consulta aos usuários).
- *Supervisão* é um tipo de relacionamento 1:N entre *Empregado* (supervisor) e *Empregado* (supervisionado). Ambas as participações são parciais, já que os usuários mencionaram que nem todo empregado possui um supervisor e vice-versa.

Refinamento do Projeto do BD Empresa

- *Trabalha_em* é um tipo de relacionamento M:N entre *Empregado* e *Projeto*, com o atributo *horas* indicando quantas horas um empregado trabalha em um projeto. Ambas as participações foram determinadas totais.
- *Dependentes_de* é um tipo de relacionamento N:1 entre *Dependente* e *Empregado*, sendo um relacionamento identificador para a entidade fraca *Dependente*. A participação de *Empregado* é parcial, enquanto que a de *Dependente* é total.

<u>Símbolo</u>	<u>Significado</u>
	ENTIDADE
	ENTIDADE FRACA
	RELACIONAMENTO
	RELACIONAMENTO IDENTIFICADOR
	ATRIBUTO
	ATRIBUTO CHAVE
	ATRIBUTO MULTIVALORADO
	ATRIBUTO COMPOSTO
	ATRIBUTO DERIVADO
	PARTICIPAÇÃO TOTAL DE E_2 EM R
	RAZÃO DA CARDINALIDADE 1:N PARA $E_1:E_2$ EM R
	RESTRIÇÃO ESTRUTURAL (min, max) NA PARTICIPAÇÃO DE E EM R

Herança

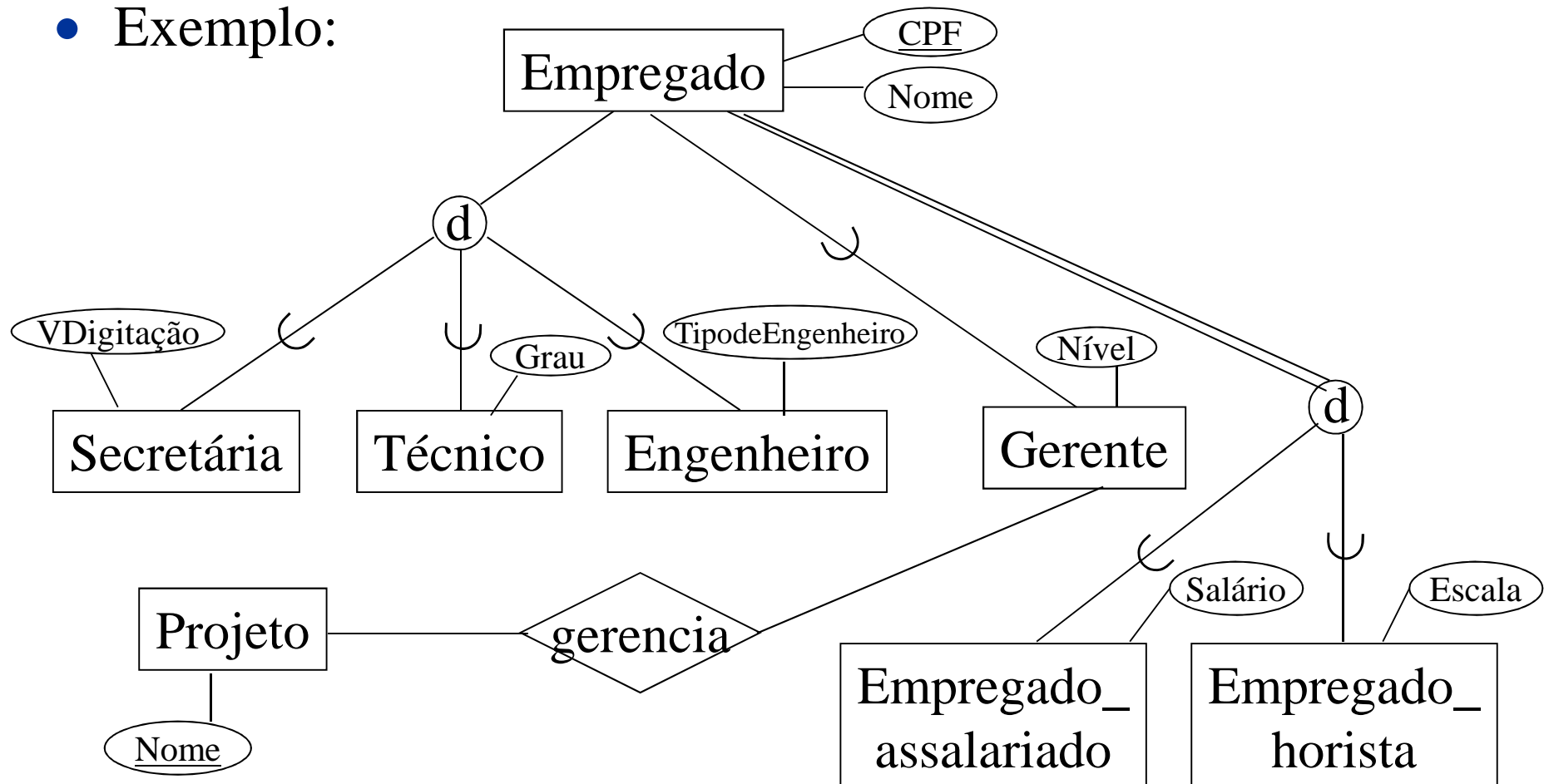
- Em muitos casos, um tipo de entidade tem vários subconjuntos de entidades que são significativos para a aplicação.
 - Exemplo: as entidades de um tipo de entidade *Empregado* podem ser agrupadas em *Secretária*, *Engenheiro*, *Gerente*, *Empregado_assalariado*, *Empregado_horista*, etc.
- Cada um dos subconjuntos é chamado de **subclasse** do tipo de entidade *Empregado*, e o tipo de entidade *Empregado* é chamado de **superclasse**.
 - Uma instância de entidade não pode existir no banco de dados somente como membro de uma subclasse; ela deve também ser membro da superclasse em questão.

Herança

- Não é necessário que toda entidade em uma superclasse seja membro de alguma subclasse.
- Como uma entidade na subclasse representa a mesma entidade no mundo real da superclasse, ela possui valores de seus atributos específicos bem como valores de seus atributos como um membro da superclasse.
 - Logo, uma entidade que pertence a uma subclasse herda todos os atributos da superclasse.
- A entidade também herda todos os relacionamentos dos quais a superclasse participa. Também, pode ter seus próprios relacionamentos como subclasse.

Especialização

- **Especialização** é o processo de definir um conjunto de subclasses a partir de um tipo de entidade (superclasse).
- Exemplo:



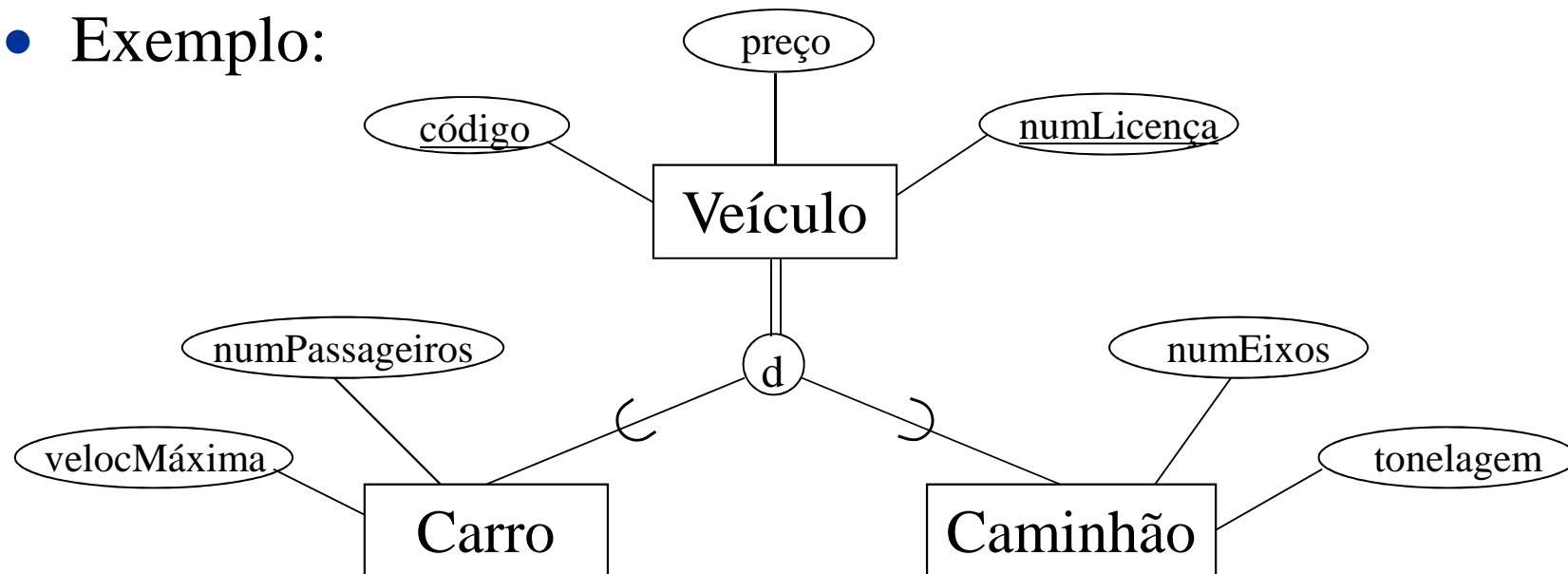
Especialização

- Algumas características do exemplo anterior são:
 - O conjunto de subclasses { *Secretária*, *Técnico* e *Engenheiro* } é uma especialização de *Empregado* que distingue as entidades pelo tipo de trabalho.
 - O conjunto de subclasses { *Empregado_assalariado* e *Empregado_horista* } é uma especialização de *Empregado* que distingue as entidades pela forma de pagamento.
 - Somente as entidades da subclasse *Engenheiro* possuem o atributo *TipodeEngenheiro*.
 - Os tipos de entidade *Secretária*, *Técnico*, *Engenheiro*, *Gerente*, *Empregado_assalariado* e *Empregado_horista* herdam os atributos da superclasse *Empregado*.
 - Somente as entidades da subclasse *Gerente* podem participar do tipo de relacionamento *gerencia*.

Generalização

- **Generalização** é o processo inverso da especialização.
 - As semelhanças entre dois ou mais tipos de entidades são identificadas e as diferenças são suprimidas, produzindo uma generalização em uma única superclasse da qual os tipos originais são subclasses especiais.

- Exemplo:



Restrições sobre Especialização/Generalização

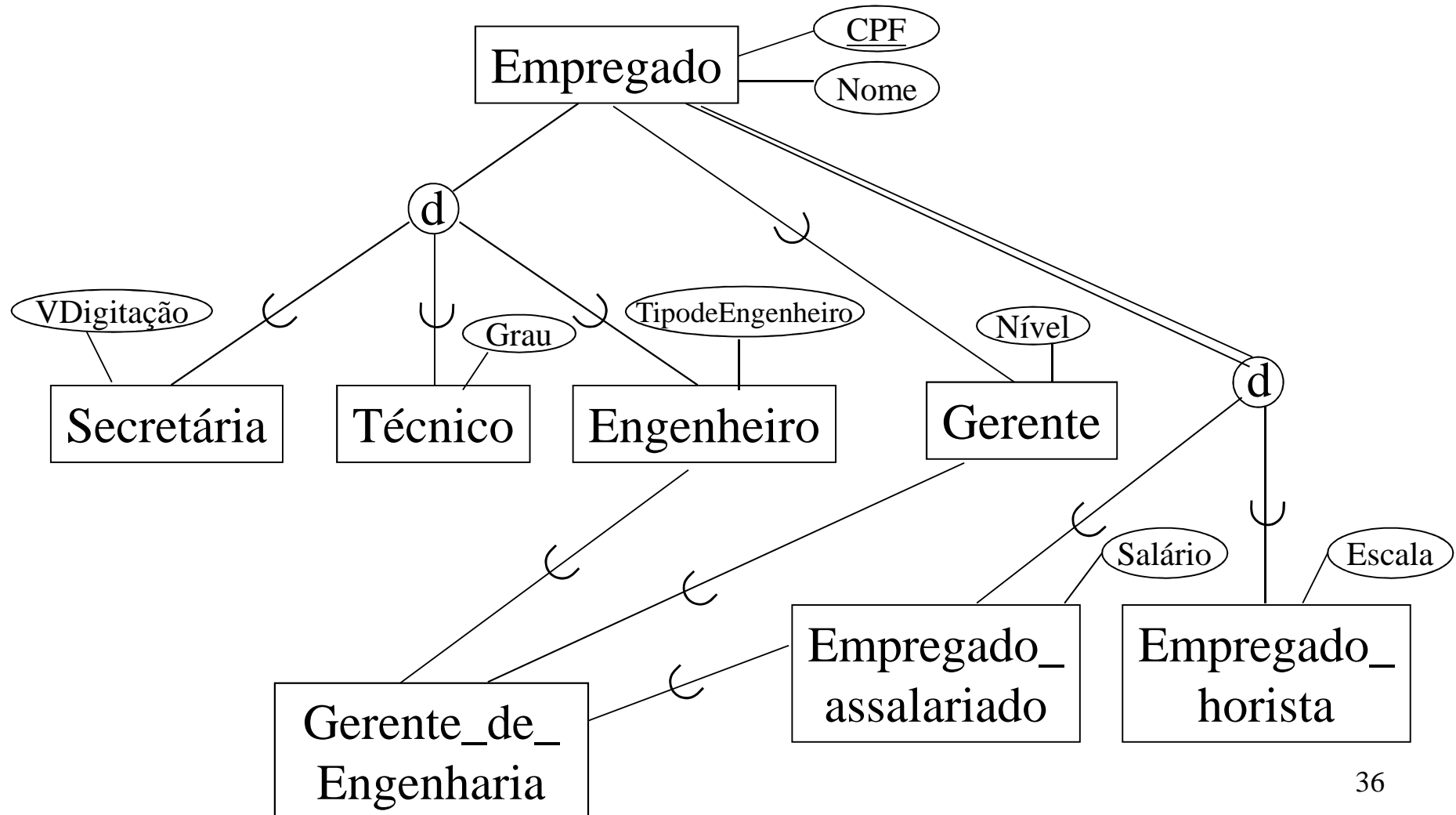
- Restrição de Disjunção:
 - **Disjunção:** uma entidade da superclasse pode ser membro de, no máximo, uma das subclasses da especialização ou generalização.
 - Representada pela letra **d** dentro do círculo da especialização ou generalização.
 - **Sobreposição:** uma entidade da superclasse pode ser membro de mais de uma subclasse da especialização ou generalização.
 - Representada pela letra **o** dentro do círculo da especialização ou generalização.

Restrições sobre Especialização/Generalização

- Restrição de Completude:
 - **Total:** toda entidade da superclasse deve ser membro de, pelo menos, uma subclasse da especialização ou generalização.
 - Representada pelas linhas duplas ligando a superclasse ao círculo.
 - **Parcial:** nem toda entidade da superclasse precisa ser membro de alguma subclasse da especialização ou generalização.
 - Representada pela linha simples ligando a superclasse ao círculo.

Herança Múltipla

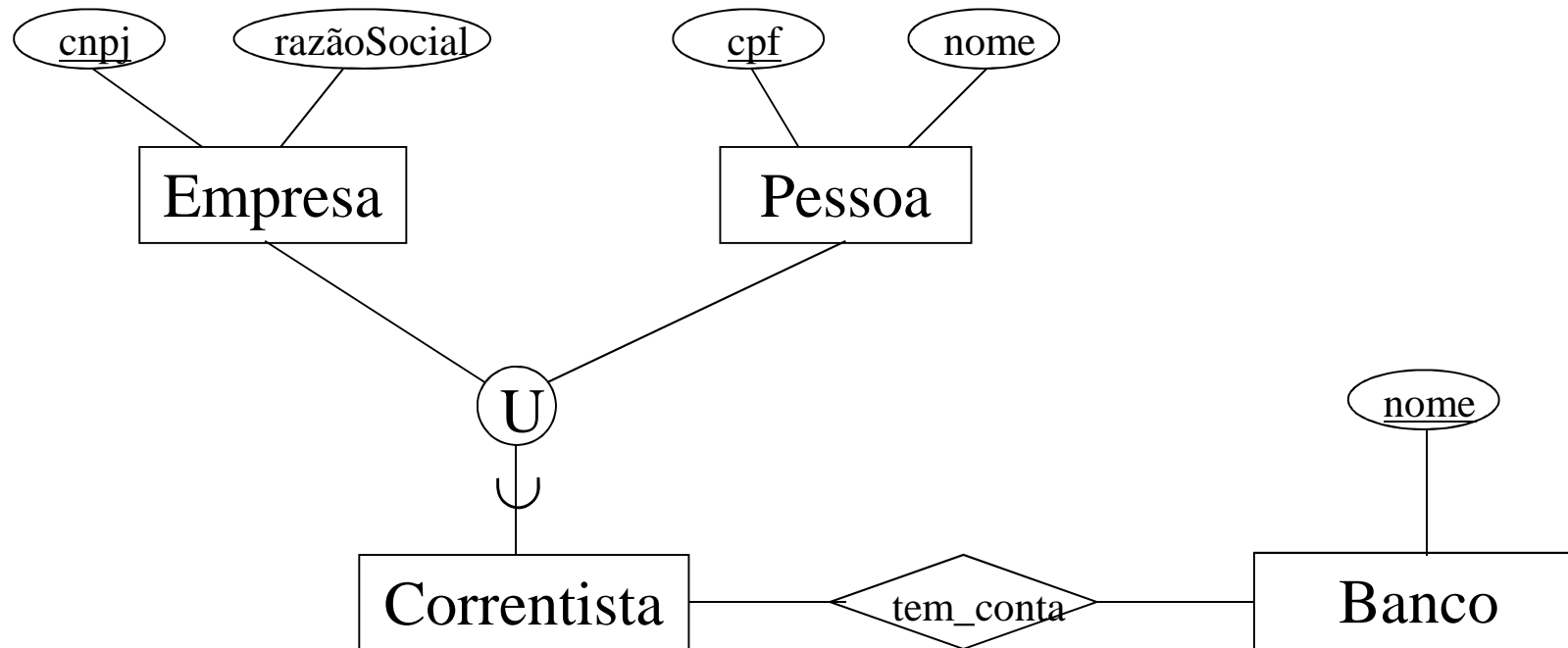
- Uma subclasse pode ser descendente de mais de uma superclasse. Neste caso, tem-se uma **herança múltipla**.



Tipo União ou Categoria

- Em algumas aplicações, é necessário modelar um único relacionamento superclasse/subclasse com mais de uma superclasse, onde as superclasses representam tipos de entidades diferentes e a subclasse representa uma coleção de objetos que é a UNIÃO de instâncias das superclasses (ou um subconjunto da união). Esta subclasse é chamada de **tipo união** ou **categoria**.
- Exemplo:
 - Em um sistema bancário, deseja-se representar um tipo de entidade *Correntista* como sendo o conjunto das entidades que possuem conta no banco. Um correntista pode ser uma pessoa física ou uma empresa. O diagrama seguinte representa este fato.

Tipo União ou Categoria



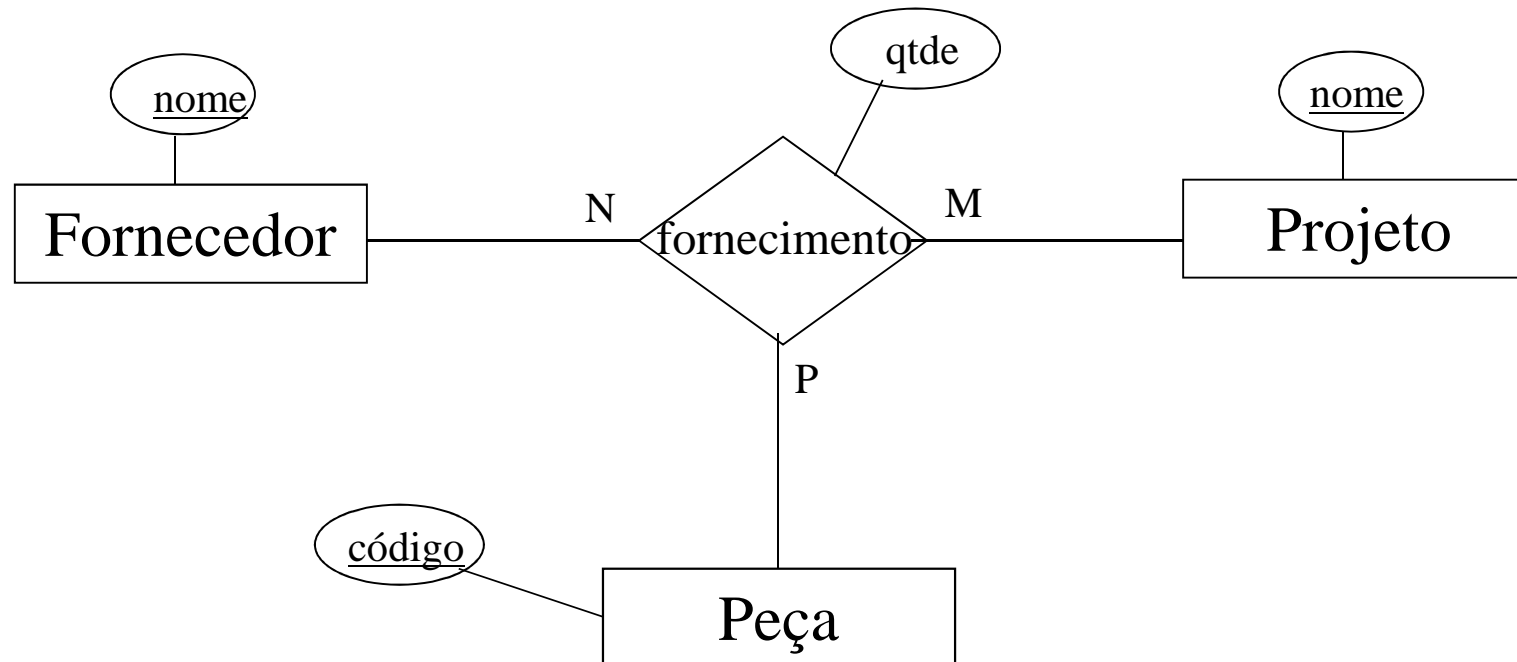
Uma categoria pode ser total ou parcial. No exemplo acima, a categoria **Correntista** é parcial (simbolizado pela linha simples), indicando que podem haver pessoas ou empresas que não são correntistas. Entretanto, todo correntista ou é uma pessoa ou é uma empresa.

Herança Múltipla versus Categoria

- Comparação Herança múltipla versus Categoria:
 - Na herança múltipla, uma entidade na subclasse deve existir também em todas as superclasses. Ex.: uma entidade em *Gerente_de_Engenharia* existe também em *Engenheiro*, em *Gerente* e em *Empregado_Assalariado*.
 - Uma categoria é um subconjunto da união de suas superclasses. Assim, uma entidade na subclasse (categoria) deve existir somente em uma das superclasses. Ex.: uma entidade em *Correntista* existe em *Empresa* ou em *Pessoa*.
 - Na herança múltipla, uma entidade na subclasse herda todos os atributos de todas as suas superclasses.
 - Na categoria, uma entidade na subclasse herda os atributos somente da superclasse da qual ela pertence.

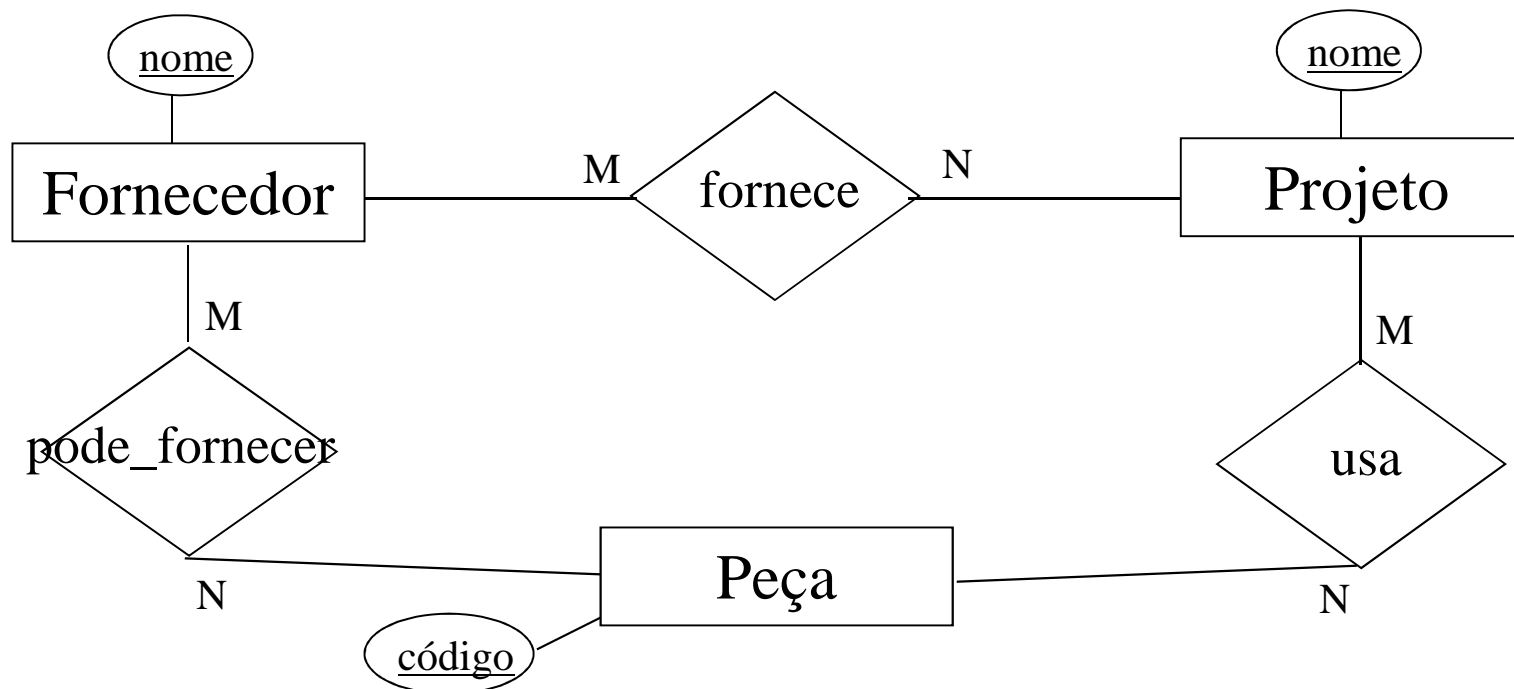
Relacionamento Ternário

- Um tipo de relacionamento ternário é um tipo de relacionamento de grau 3, ou seja, que possui 3 tipos de entidades participantes.
- Considere o exemplo abaixo, onde fornecedores fornecem peças a projetos.



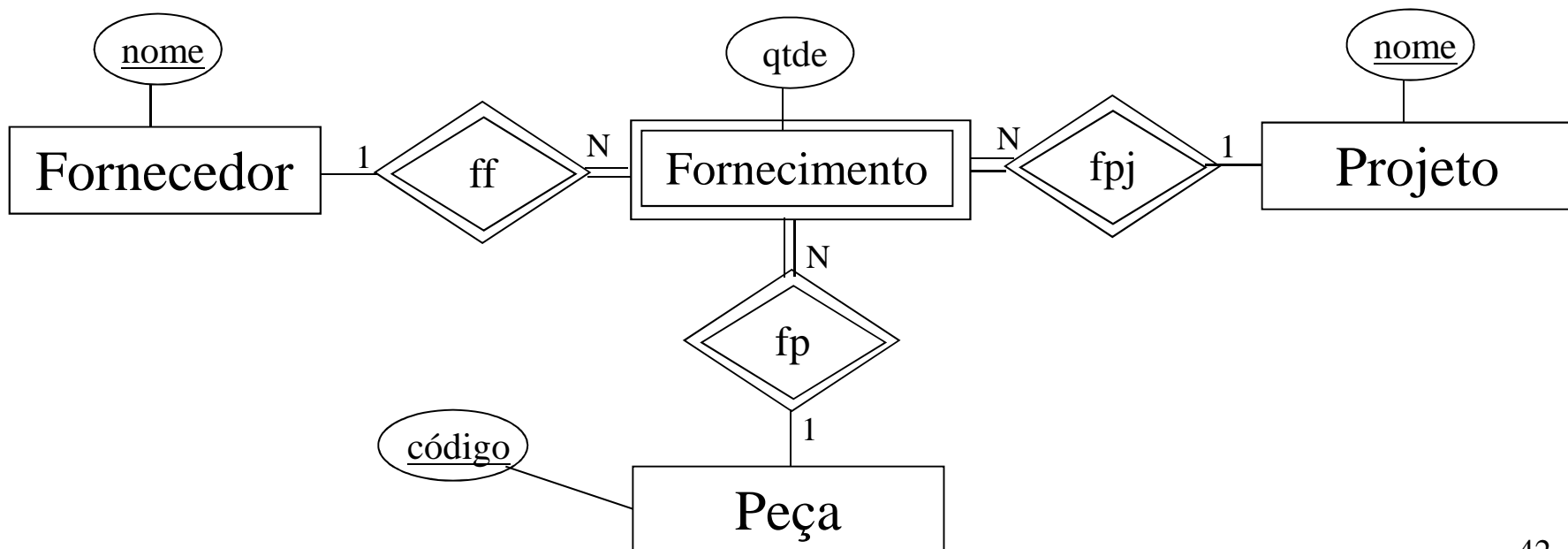
Relacionamento Ternário

- O exemplo abaixo representa os relacionamentos entre fornecedores, peças e projetos como três relacionamentos binários. Este diagrama **não** é equivalente ao anterior. Em geral, um tipo de relacionamento ternário representa mais informação do que três tipos de relacionamentos binários.



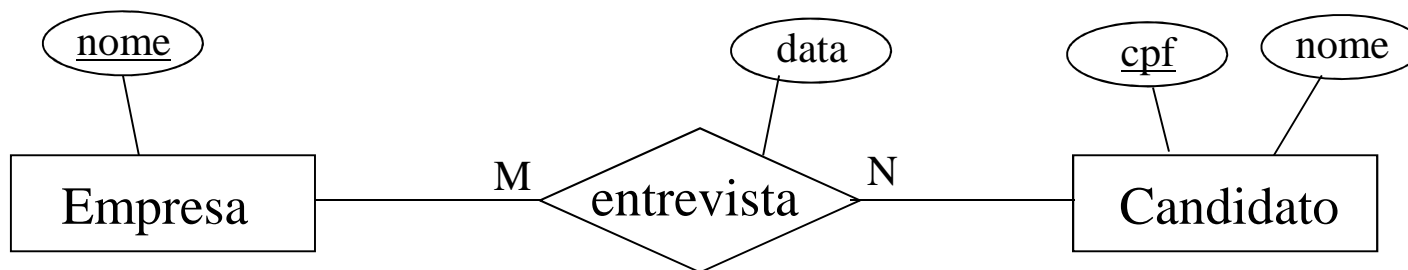
Relacionamento Ternário

- Uma representação equivalente ao tipo de relacionamento ternário, usando apenas tipos de relacionamentos binários, deve criar um tipo de entidade fraca, sem chaves parciais e com três relacionamentos identificadores. Uma instância do tipo de entidade fraca *Fornecimento* é identificada pela combinação de suas três entidades proprietárias.



Agregação

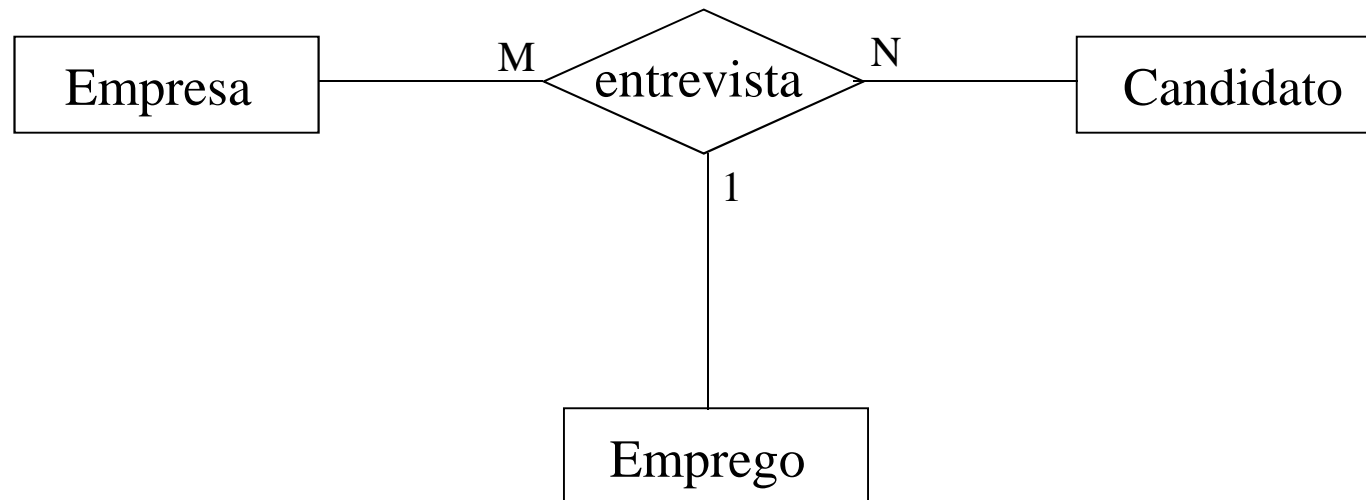
- **Agregação** é uma abstração usada para construir objetos compostos a partir de seus componentes.
 - Informalmente, uma agregação corresponde a um tipo de relacionamento que, devido aos requisitos de modelagem, deve ser transformado em um tipo de entidade.
- Exemplo: o diagrama abaixo representa um banco de dados de entrevistas para emprego de uma agência de RH.



- Suponha que algumas entrevistas resultam em oferta de emprego e outras não. Como representar este fato?

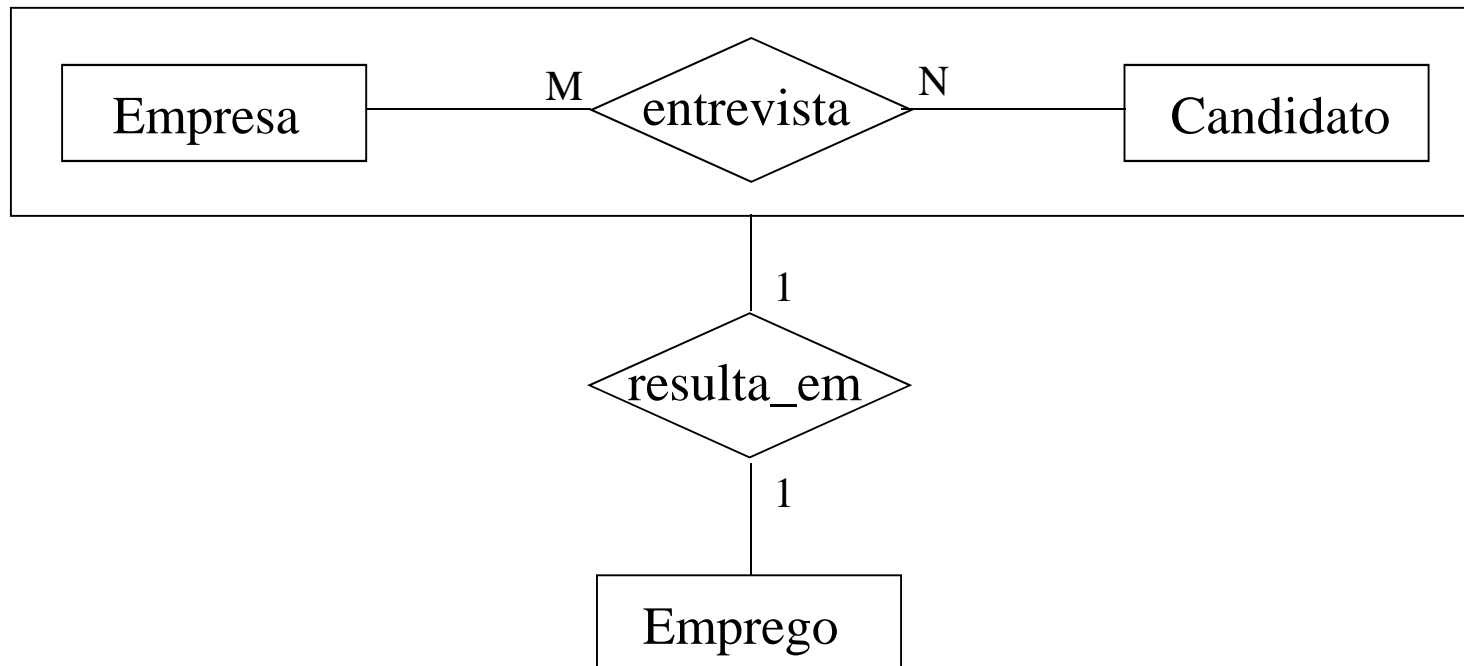
Agregação

- Usar um tipo de relacionamento ternário **não é correto**, pois requer que cada instância do relacionamento *entrevista* tenha uma oferta de emprego garantida.



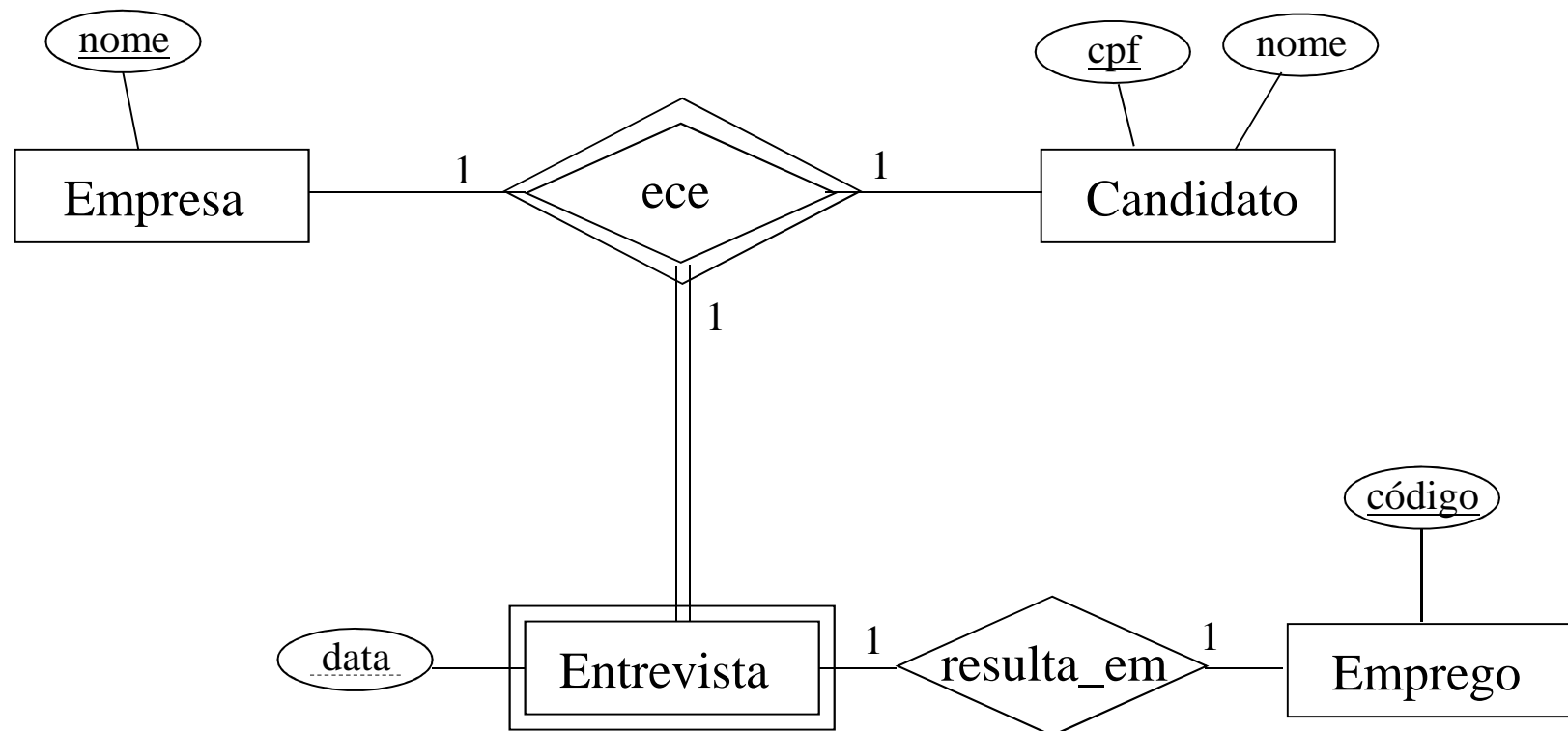
Agregação

- Uma maneira de representar essa situação é criar uma entidade agregada de nível mais alto composta por *Empresa*, *Candidato* e *entrevista* e, então, relacioná-la a *Emprego*, como mostrado abaixo:



Agregação

- Outra maneira de representar a mesma situação é criando um tipo de entidade fraca *Entrevista*, como mostrado abaixo:



Modelo Relacional

- Introduzido por Codd em 1970 (pesquisa da IBM).
- Modelo formal fundamentado nos conceitos de uma relação matemática (teoria de conjuntos).
 - O modelo relacional representa o banco de dados como uma coleção de relações.
- Informalmente, cada relação é semelhante a uma tabela.
 - Cada linha da tabela representa uma coleção de valores de dados relacionados, que podem ser interpretados como fatos descrevendo uma entidade ou um relacionamento.
 - O nome da tabela e os nomes das colunas são usados para ajudar na interpretação do significado dos valores em cada linha da tabela.

Conceitos Básicos

- Na terminologia do modelo relacional, uma linha é chamada de uma **tupla**, um cabeçalho de coluna é chamado de um **atributo**, e uma tabela é chamada de uma **relação**.
- O tipo de dado que descreve os valores que um atributo pode ter é chamado de **domínio**. Um domínio D é um conjunto de valores atômicos (indivisíveis). Exemplos:
 - Nomes: o conjunto de nomes de pessoas. Cadeia de caracteres contendo apenas letras;
 - Idades_de_Funcionários: os valores possíveis para idades de funcionários de empresa. Número inteiro entre 15 e 80;
 - CPFs: ddd.ddd.ddd-dd (onde d é um dígito de 0 a 9);
 - Salários: número real positivo com duas casas decimais.

Conceitos Básicos

- **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)

Conceitos Básicos

- **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:
 $t = \langle v_1, v_2, \dots, v_n \rangle$, 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 :
$$r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$$
 - O produto cartesiano especifica todas as possíveis combinações de valores dos domínios fundamentais.

Características das Relações

- A ordem das tuplas é irrelevante.
- A ordem dos valores dentro de uma tupla é relevante, a menos que se estabeleça uma correspondência entre esses valores e os atributos definidos. Exemplos:

$t = \langle \text{BD}, 032, 72, 4 \rangle$

$t = \langle (\text{nome}, \text{BD}), (\text{código}, 032), (\text{cargaHorária}, 72), (\text{numCreditos}, 4) \rangle$

$t = \langle (\text{código}, 032), (\text{nome}, \text{BD}), (\text{numCreditos}, 4), (\text{cargaHorária}, 72) \rangle$

- O valor de cada atributo em uma tupla é atômico.
 - Atributos compostos e multi-valorados não são permitidos.
- As tuplas de uma relação são únicas.

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 $I = \{I_1, I_2, \dots, I_m\}$.
- Uma instância BD de 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 r_i satisfazem as restrições de integridade em I . Portanto, pode-se dizer que $S = (R, I)$.

Empregado

PrimeiroNome	InicialMeio	UltimoNome	<u>NumEmpregado</u>	DataNascimento	Endereco	Sexo	Salario	NumSupervisor	NumDeppto
--------------	-------------	------------	---------------------	----------------	----------	------	---------	---------------	-----------

Departamento

NomeDeppto	<u>NumDeppto</u>	NumGerente	DataInicioGerencia
------------	------------------	------------	--------------------

Localizacao_Depto

<u>NumDepart</u>	<u>Localizacao</u>
------------------	--------------------

Trabalha_em

<u>NumEmpregado</u>	<u>NumProj</u>	Horas
---------------------	----------------	-------

Projeto

NomeProj	<u>NumProj</u>	Localizacao	NumDeppto
----------	----------------	-------------	-----------

Dependente

<u>NumEmpregado</u>	<u>NomeDependente</u>	Sexo	DataAniversario	Parentesco
---------------------	-----------------------	------	-----------------	------------

Exemplo de Instância de um Banco de Dados

Empregado

PrimeiroNome	InicialMeio	UltimoNome	NumEmpregado	DataNascimento	Endereco	Sexo	Salario	NumSupervisor	NumDepto
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. Iraí, 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. Contria, 12	M	2500.00	987654321	4
Fábio	E	Lemos	888665555	10/11/47	R. Chile, 425	M	5500.00	null	1

Departamento

NomeDepto	NumDepto	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	NumDepto
Produto X	1	Buritis	5
Produto Y	2	Pampulha	5
Produto Z	3	Contagem	5
Informatização	10	Centro	4
Reorganização	20	Savassi	1
NovosBenefícios	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

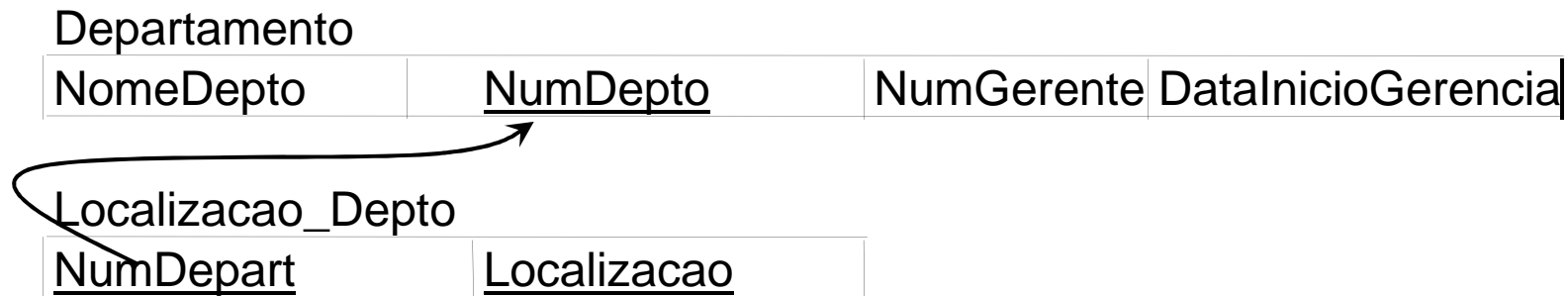
- **Restrição de Domínio:** o valor de cada atributo deve ser um valor atômico no domínio do atributo.
- **Restrição de Chave:** uma **chave** é um conjunto mínimo de valores dos atributos que identifica unicamente uma tupla.
 - Se um esquema de relação tem mais de uma chave, cada uma é chamada **chave candidata**. Uma das chaves candidatas é arbitrariamente escolhida para ser a **chave primária**, e as outras são chamadas **chaves alternativas** ou **chaves secundárias**.
 - 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.

Restrições do Modelo Relacional

- **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 duas relações, sendo usada para manter a consistência entre as tuplas das duas relações. Usa-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 se satisfaz as duas regras seguintes:
 - os atributos de FK referenciam a chave primária PK de R_2 , tendo os mesmos domínios dos atributos de PK.
 - um valor de FK na tupla t_1 da instância $r_1(R_1)$ ocorre como um valor de PK para alguma tupla t_2 da instância $r_2(R_2)$ ou é nulo.

Restrições do Modelo Relacional

- As restrições de integridade referencial são obtidas, geralmente, a partir dos relacionamentos entre as entidades representadas pelos esquemas de relação.
- As restrições de integridade referencial podem ser representadas graficamente usando-se setas que partem de uma chave estrangeira para a chave primária na relação referenciada.



Integridade Referencial com Opção de Exclusão

- Notação: $R_1[fk] \xrightarrow{op} R_2[pk]$

onde "op" é a opção de exclusão, dentre as seguintes:

- **bloqueio (restrict)**: se alguma tupla referencia a tupla a ser excluída, através de uma chave estrangeira, a exclusão não é efetuada.
- **propagação (cascade)**: todas as tuplas que referenciam a tupla a ser excluída, através de uma chave estrangeira, são excluídas também automaticamente.
- **substituição por nulos (set null)**: 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*):** insere novas tuplas em uma relação.
 - 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**

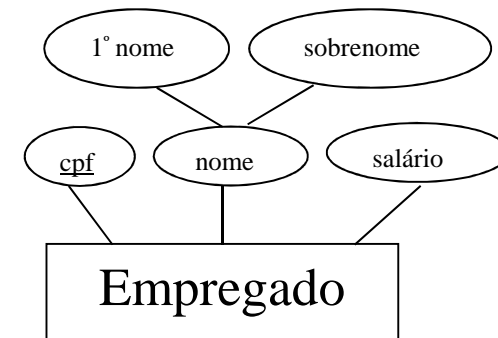
- Para cada tipo de entidade E no esquema ERE, crie uma relação R que inclua todos os atributos simples de E.
- Inclua também os atributos simples componentes de um atributo composto de E na relação R.
- Escolha uma das chaves candidatas de E para ser a chave primária de R.

Exemplo:

para o tipo de entidade Empregado,

gere a relação:

Empregado (cpf, primeironome, sobrenome, salario)



Mapeamento ERE para Relacional

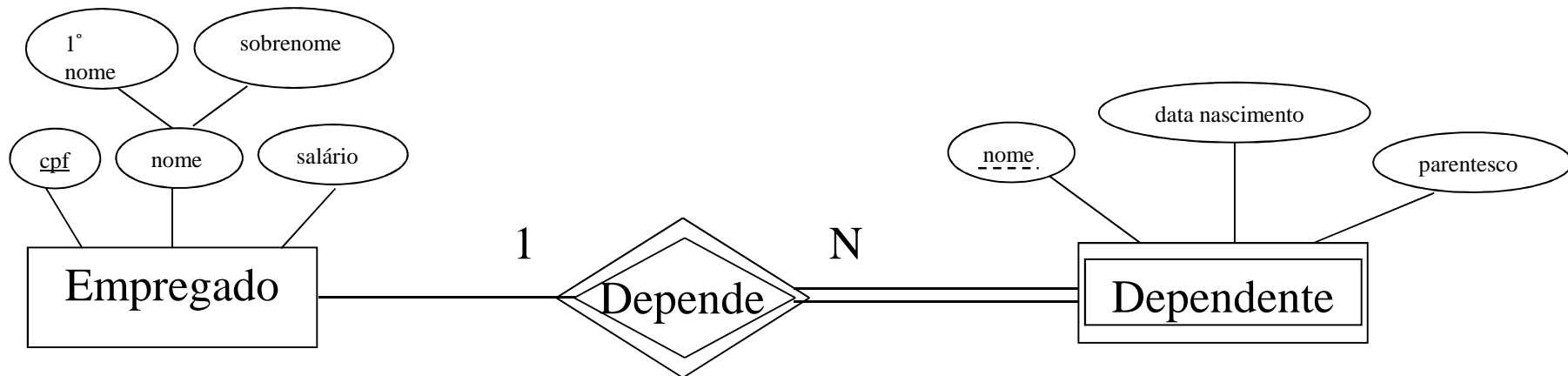
- **Entidade Fraca**

- Para cada tipo de entidade fraca W no esquema ERE, crie uma relação R e inclua todos os atributos simples (ou componentes simples de atributos compostos) de W como atributos de R .
- Inclua também como atributos de R todos os atributos componentes das chaves primárias de cada uma das entidades fortes de W . Cada uma dessas inclusões corresponde a uma chave estrangeira de R .
- A chave primária de R é a combinação dos atributos das chaves primárias das entidades fortes de W mais a chave parcial de W .

Mapeamento ERE para Relacional

Exemplo:

para o tipo de entidade fraca Dependente,



gere a relação:

Dependente (cpfEmp, nome, dataNascimento, parentesco)

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

Mapeamento ERE para Relacional

- **Atributo Multi-valorado**

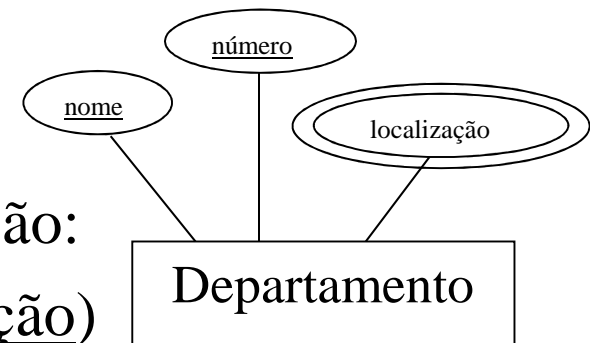
- Para cada atributo multi-valorado A, crie uma nova relação R que inclua o atributo A mais a chave primária K (como chave estrangeira em R) da relação que representa o tipo de entidade ou o tipo de relacionamento que tem A como atributo. Se o atributo multi-valorado é composto, inclua seus componentes simples.
- A chave primária de R é a combinação de K e A.

Exemplo:

para o atributo Localização, gere a relação:

Localização_Depto (número, localização)

Localização_Depto[número] \xrightarrow{p} Departamento[número]



Mapeamento ERE para Relacional

- **Relacionamento Binário 1:1**

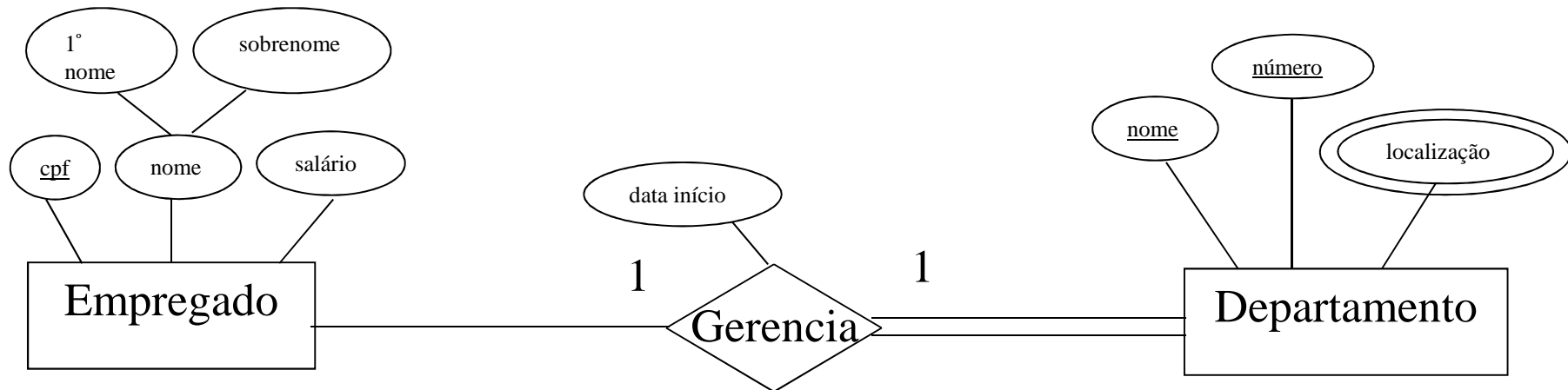
- Para cada tipo de relacionamento binário R 1:1 no esquema ERE, identifique as relações S e T que correspondem aos tipos de entidades participantes de R. Escolha uma das relações, S por exemplo, e inclua como chave estrangeira em S a chave primária de T. É melhor escolher um tipo de entidade com participação total em R no papel de S.
- Inclua todos os atributos simples (ou componentes simples de atributos compostos) do tipo de relacionamento R como atributo de S.

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.

Mapeamento ERE para Relacional

Exemplo:

para o tipo de relacionamento Gerencia,



adicione os atributos cpfGerente e dataInício à relação Departamento:

Departamento (número, nome, cpfGerente, dataInício)

Departamento[cpfGerente] \xrightarrow{b} Empregado[cpf]

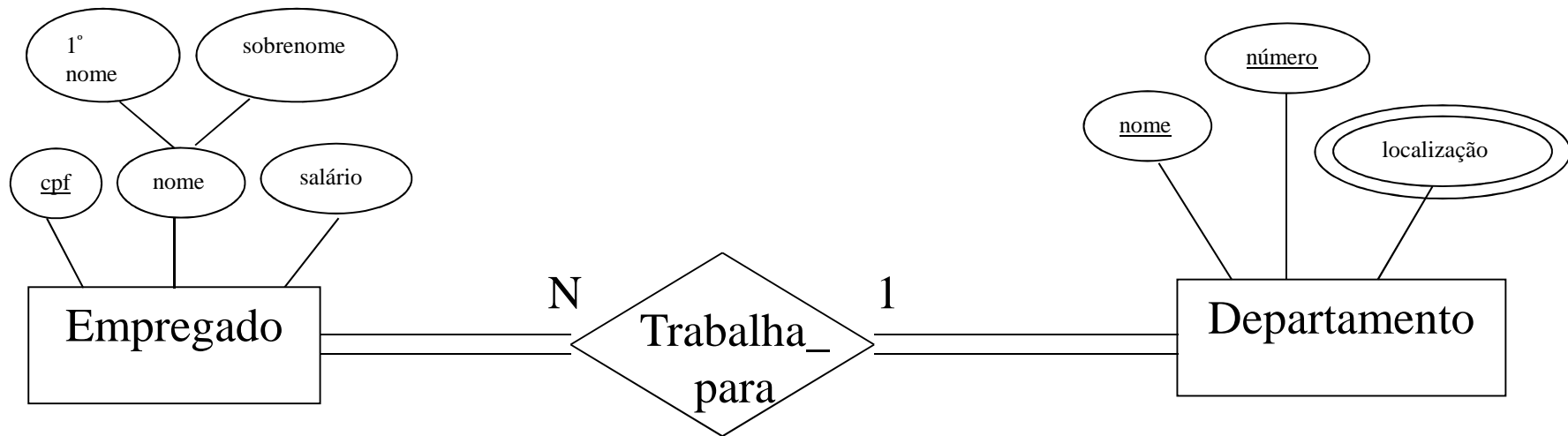
Mapeamento ERE para Relacional

- **Relacionamento Binário 1:N**
 - Para cada tipo de relacionamento binário R 1:N regular (não identificador) no esquema ERE, identifique a relação S que representa o tipo de entidade participante do lado N do tipo de relacionamento. Inclua como chave estrangeira em S a chave primária da relação T que representa o outro tipo de entidade participante de R. Isso ocorre porque cada instância do lado N está relacionada a, no máximo, uma instância do lado 1 do tipo de relacionamento.
 - Inclua todos os atributos simples (ou componentes simples de atributos compostos) do tipo de relacionamento R como atributos de S.

Mapeamento ERE para Relacional

Exemplo:

para o tipo de relacionamento Trabalha_para,



adicione o atributo númeroDepto à relação Empregado:

Empregado (cpf, primeiroNome, sobrenome, salario, númeroDepto)

Empregado[númeroDepto] \xrightarrow{b} Departamento[número]

Mapeamento ERE para Relacional

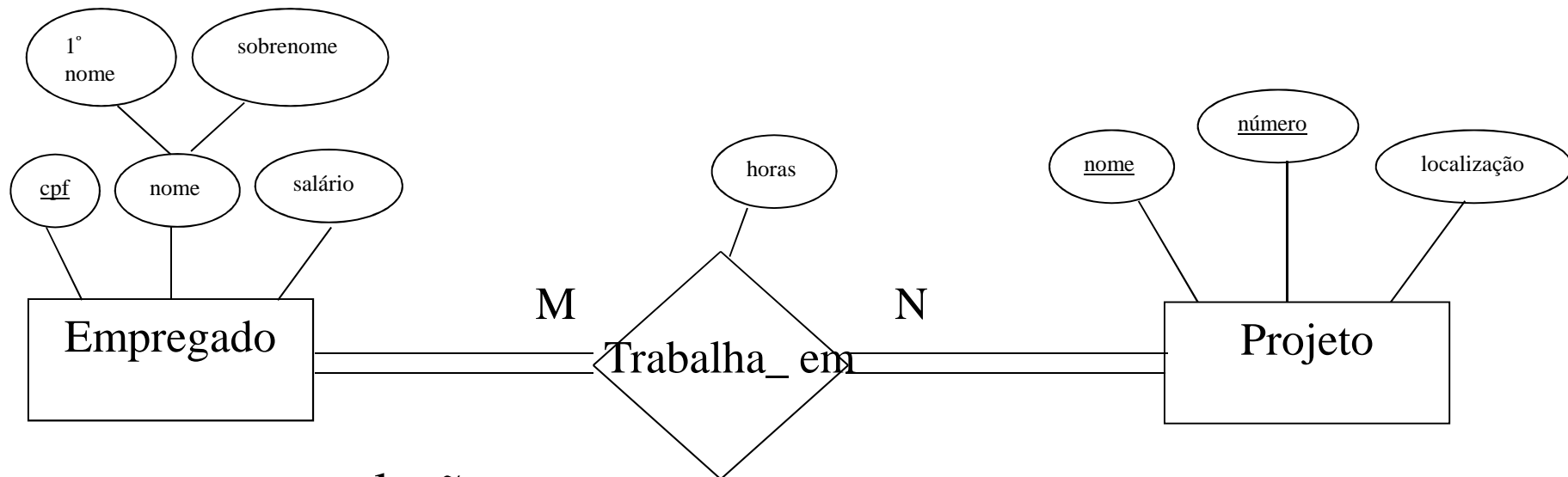
- **Relacionamento Binário M:N**

- Para cada tipo de relacionamento binário R M:N no esquema ERE, crie uma nova relação S para representar R. Inclua como chaves estrangeiras em S as chaves primárias das relações que representam os tipos de entidades participantes.
- Também inclua 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 a combinação de suas chaves estrangeiras.

Mapeamento ERE para Relacional

Exemplo:

para o tipo de relacionamento Trabalha_em,



gere a relação:

Trabalha_em (cpf, númeroProjeto, horas)

Trabalha_em[cpf] \xrightarrow{p} Empregado[cpf]

Trabalha_em[númeroProjeto] \xrightarrow{p} Projeto[número]

Mapeamento ERE para Relacional

- Os tipos de relacionamentos 1:1 e 1:N podem ser mapeados de forma similar ao tipo de relacionamento M:N.
 - Essa alternativa é particularmente 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); no caso, a outra chave estrangeira deve ser definida como chave secundária.

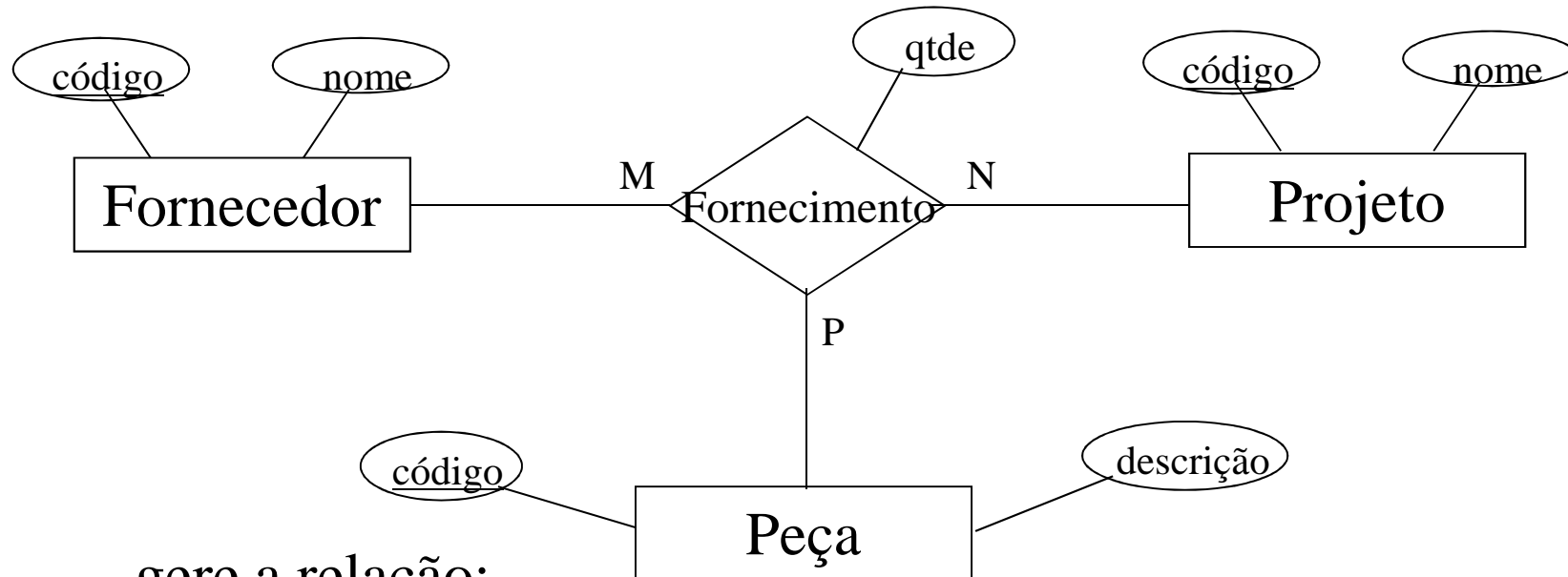
Mapeamento ERE para Relacional

- **Relacionamento n-ário ($n > 2$)**
 - Para cada tipo de relacionamento n-ário R, onde $n > 2$, no esquema ERE, crie uma nova relação S para representar R. Inclua como atributos da chave estrangeira em S as chaves primárias das relações que representam os tipos de entidades participantes.
 - Também inclua 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.

Mapeamento ERE para Relacional

Exemplo:

para o tipo de relacionamento Fornecimento,



gere a relação:

fornecimento (codForn, codProj, codPeça, qtde)

fornecimento[codForn] \xrightarrow{p} Fornecedor[código]

fornecimento[codProj] \xrightarrow{p} Projeto[código]

fornecimento[codPeça] \xrightarrow{p} Peça[código]

Mapeamento ERE para Relacional

- **Especialização / Generalização**
 - **Opção 1:** crie uma relação L para a superclasse C no esquema ERE com os atributos de C . A chave primária de L é uma chave de C . Crie também uma relação L_i para cada subclasse S_i . 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 . Essa opção funciona para qualquer restrição na especialização: disjunta/sobreposta, total/parcial.
 - **Opção 2:** crie uma relação L_i para cada subclasse S_i com os atributos da subclasse mais os atributos da superclasse. A chave primária de L_i é uma chave da superclasse. Essa opção deve ser usada para restrições total e disjunta.

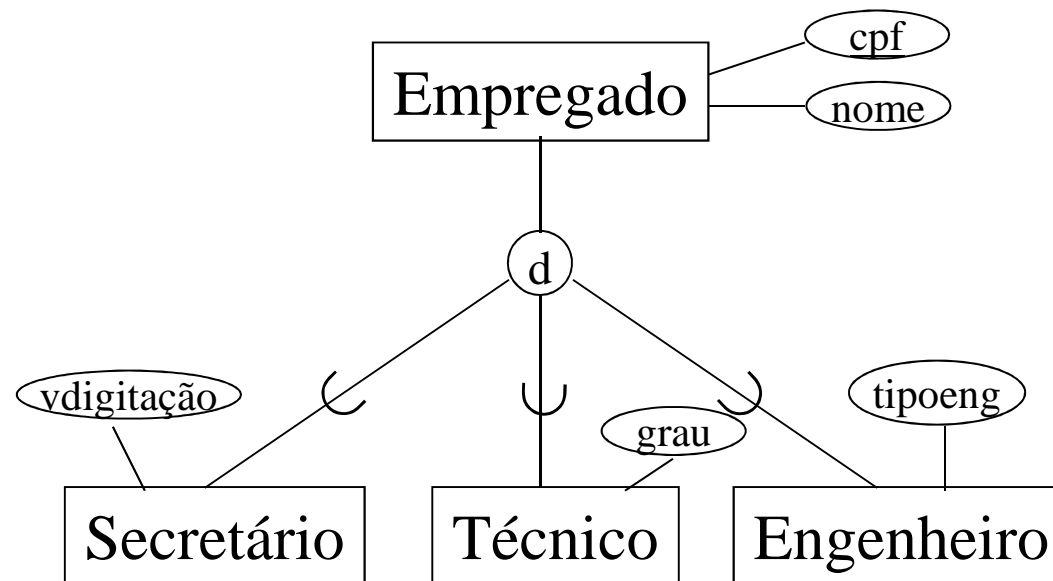
Mapeamento ERE para Relacional

- **Opção 3:** crie uma única relação L com todos os atributos da superclasse C e das subclasses S_i , 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. Essa opção pode gerar um grande número de valores nulos.
- **Opção 4:** crie uma única relação L com todos os atributos da superclasse C e das subclasses S_i , mais 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).

Mapeamento ERE para Relacional

Exemplos:

para a especialização abaixo.



Mapeamento ERE para Relacional

usando a **opção 1**, gere as relações:

Empregado (cpf, nome)

Secretário (cpf, vdigitação)

Secretário[cpf] \xrightarrow{p} Empregado[cpf]

Técnico (cpf, grau)

Técnico[cpf] \xrightarrow{p} Empregado[cpf]

Engenheiro (cpf, tipoeng)

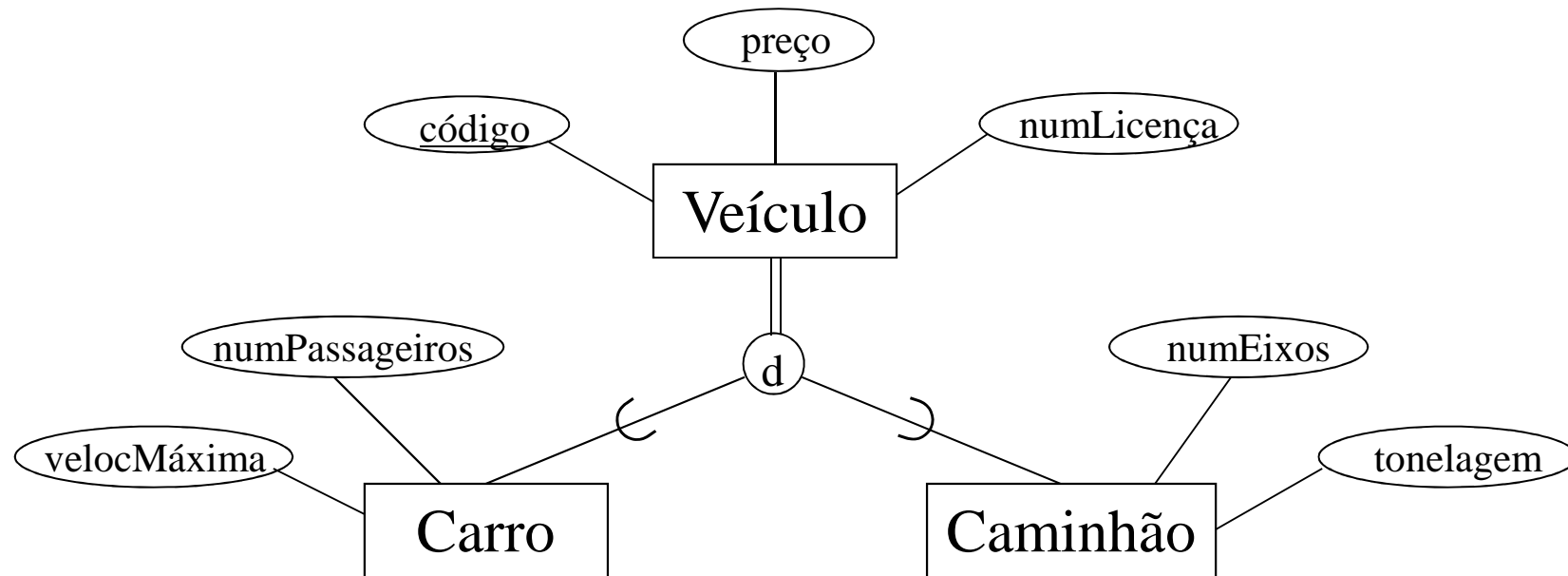
Engenheiro[cpf] \xrightarrow{p} Empregado[cpf]

usando a **opção 3**, gere a relação:

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

Mapeamento ERE para Relacional

para a generalização abaixo:



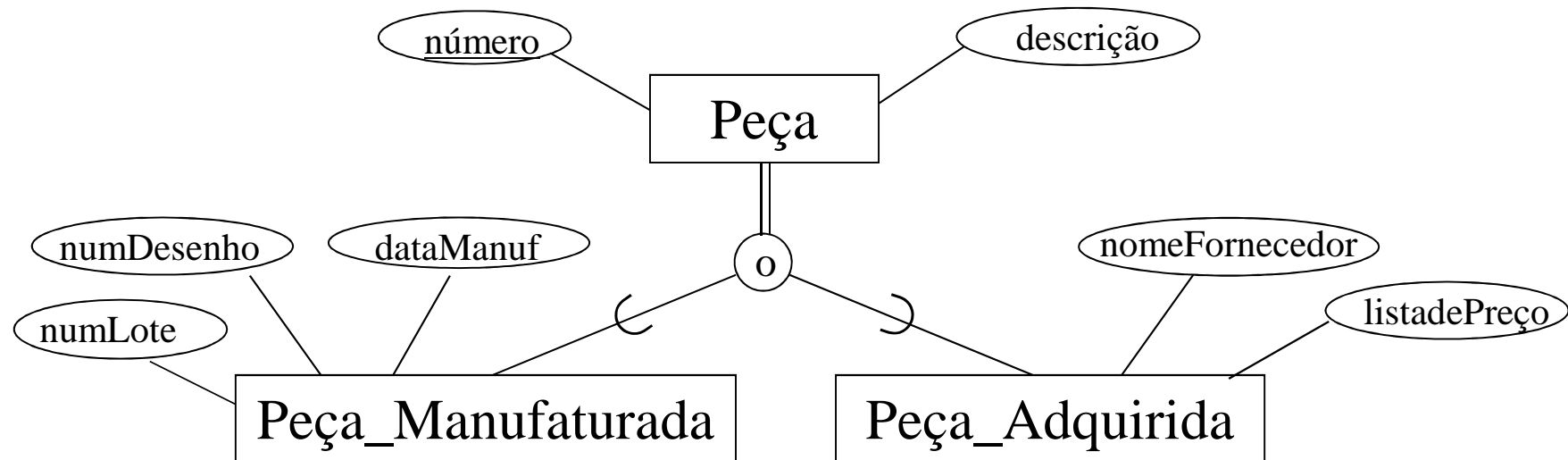
usando a **opção 2**, gere as relações:

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

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

Mapeamento ERE para Relacional

para a especialização abaixo:



usando a **opção 4**, gere a relação:

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

Mapeamento ERE para Relacional

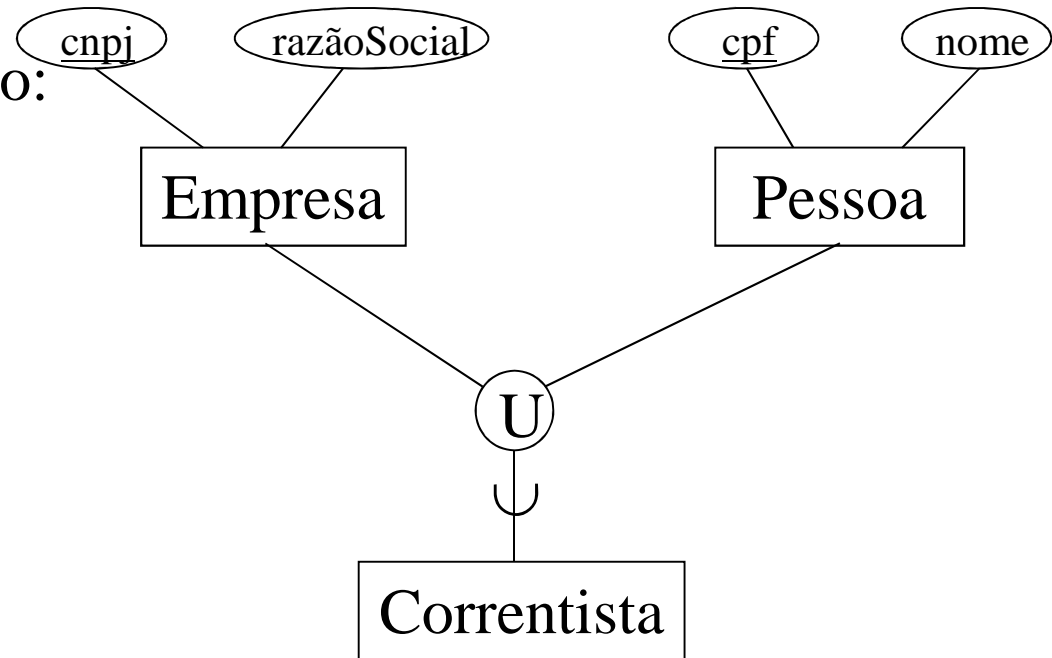
- **Tipo União ou Categoria**

- Crie uma relação para representar a categoria e inclua todos os seus atributos.
 - Para uma categoria cujas superclasses têm chaves diferentes, adicione um novo atributo chave, chamado “chave substituta”, para ser a chave primária da relação. Adicione 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. Adicione 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.

Mapeamento ERE para Relacional

Exemplo:

para a categoria abaixo:



gere as relações:

Empresa (cnpj, razãosocial, numCorrentista)

Empresa[numCorrentista] \xrightarrow{n} Correntista[numCorrentista]

Pessoa (cpf, nome, numCorrentista)

Pessoa[numCorrentista] \xrightarrow{n} Correntista[numCorrentista]

Correntista (numCorrentista)