

Modelo Entidade Relacionamento Estendido (ERE)

- A maioria dos BDs podem ser modelados por meio dos conceitos básicos do modelo ER, mas alguns aspectos podem ser expressos mais convenientemente por meio de algumas extensões do modelo básico.

Modelo Entidade Relacionamento Estendido (ERE)

- O modelo ER estendido engloba todos os conceitos do modelo ER mais os conceitos de generalização, especialização, subtipo, supertipo e conceito de herança de atributos.

Generalização e Especialização

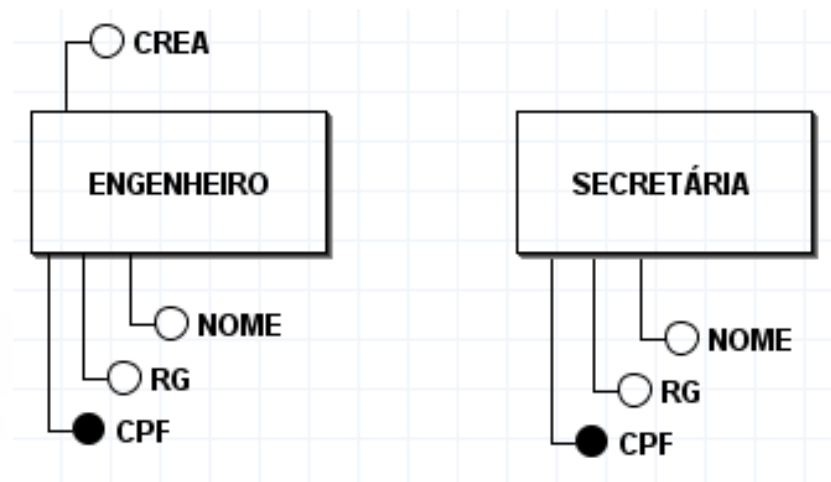
- A **especialização** corresponde a ação de especializar entidades:
 - **EMPREGADO** pode ser especializado como **ENGENHEIRO** ou **SECRETÁRIA**.
- A **generalização** corresponde a ação de generalizar entidades:
 - Uma entidade **ENGENHEIRO** e **SECRETÁRIA** são generalizadas como uma entidade **FUNCIONÁRIO**.

Generalização e Especialização

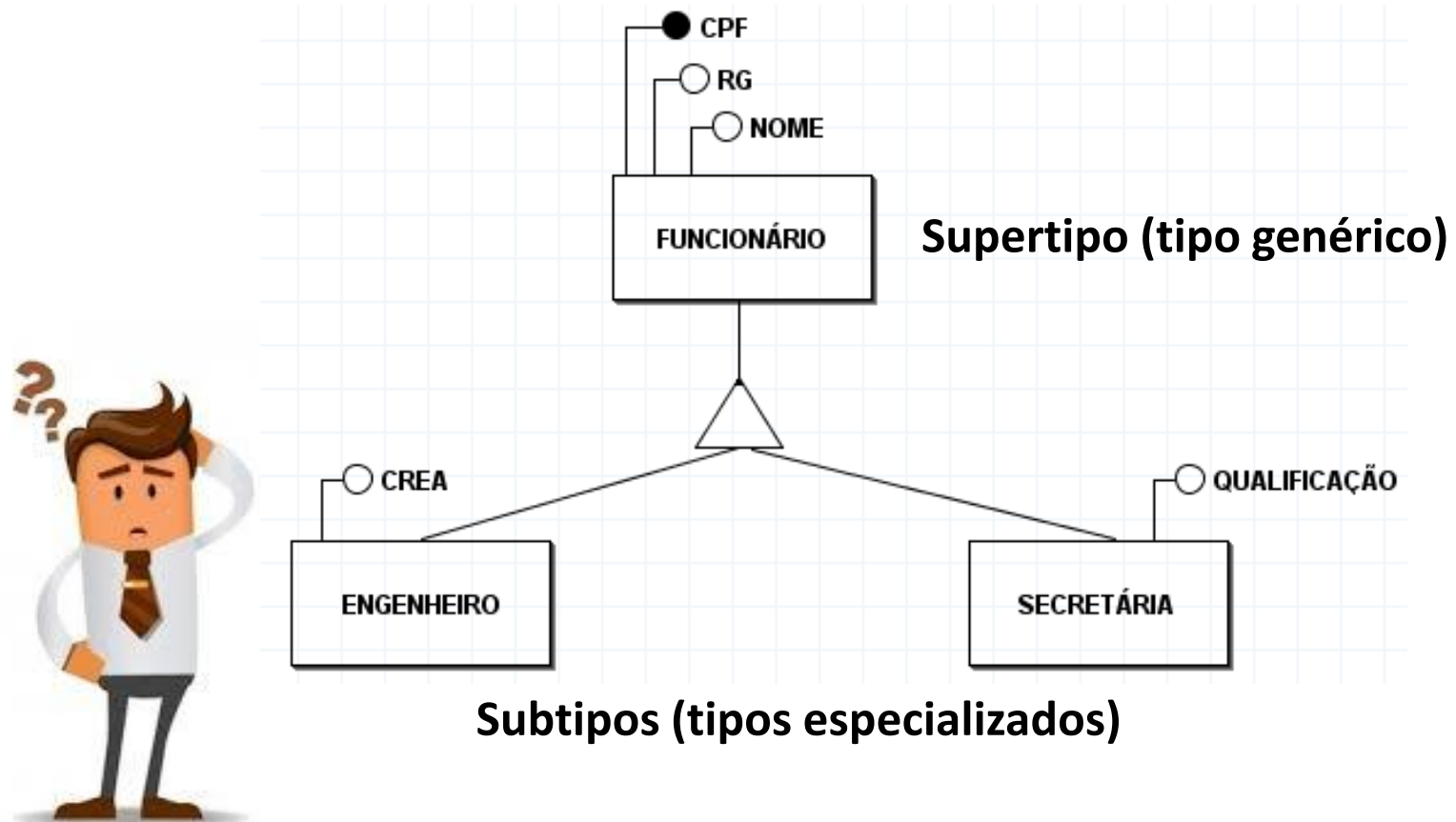
- Os subtipos da entidade “pai” são as suas especializações:
 - **ENGENHEIRO** e **SECRETÁRIA** são subtipos de **FUNCIONÁRIO**.
- O supertipo de um subtipo é a entidade “pai”:
 - **FUNCIONÁRIO** é o supertipo de **ENGENHEIRO** e **SECRETÁRIA**.

Generalização e Especialização

- O que pode melhorar neste modelo ER?

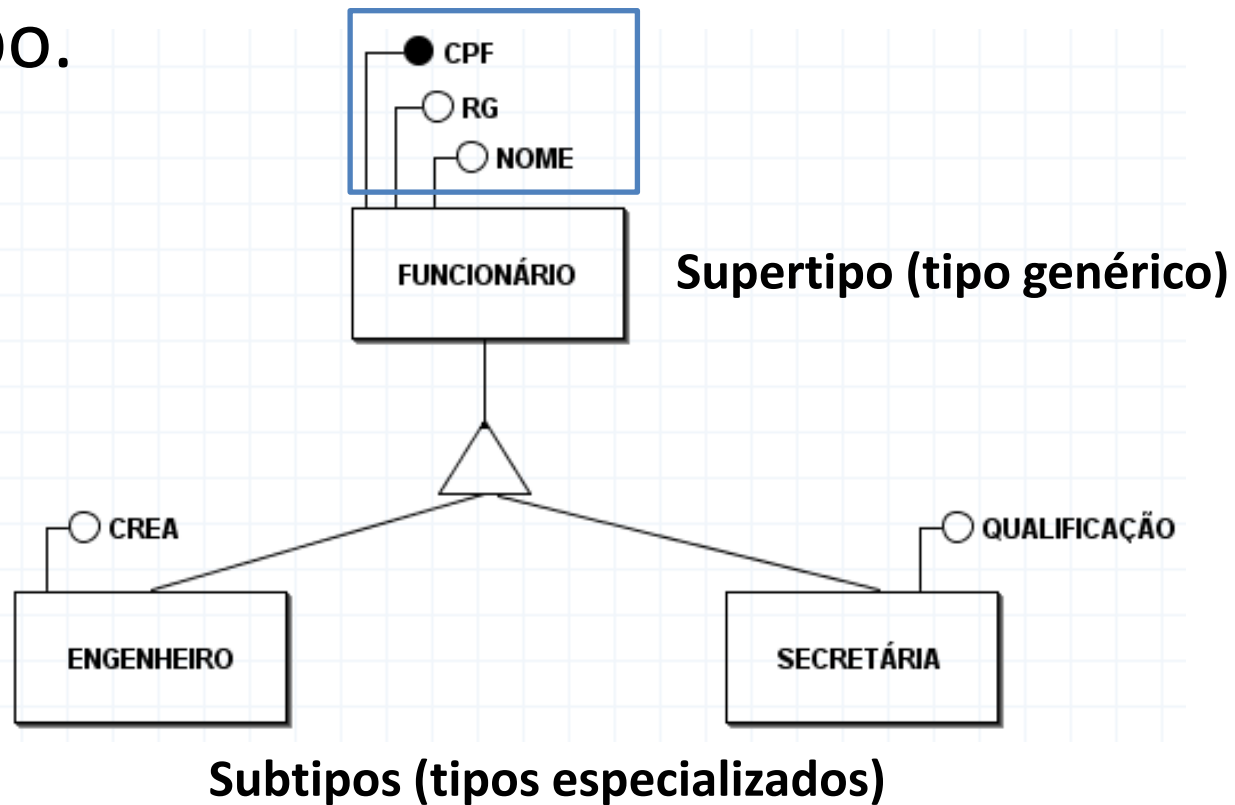


Generalização e Especialização



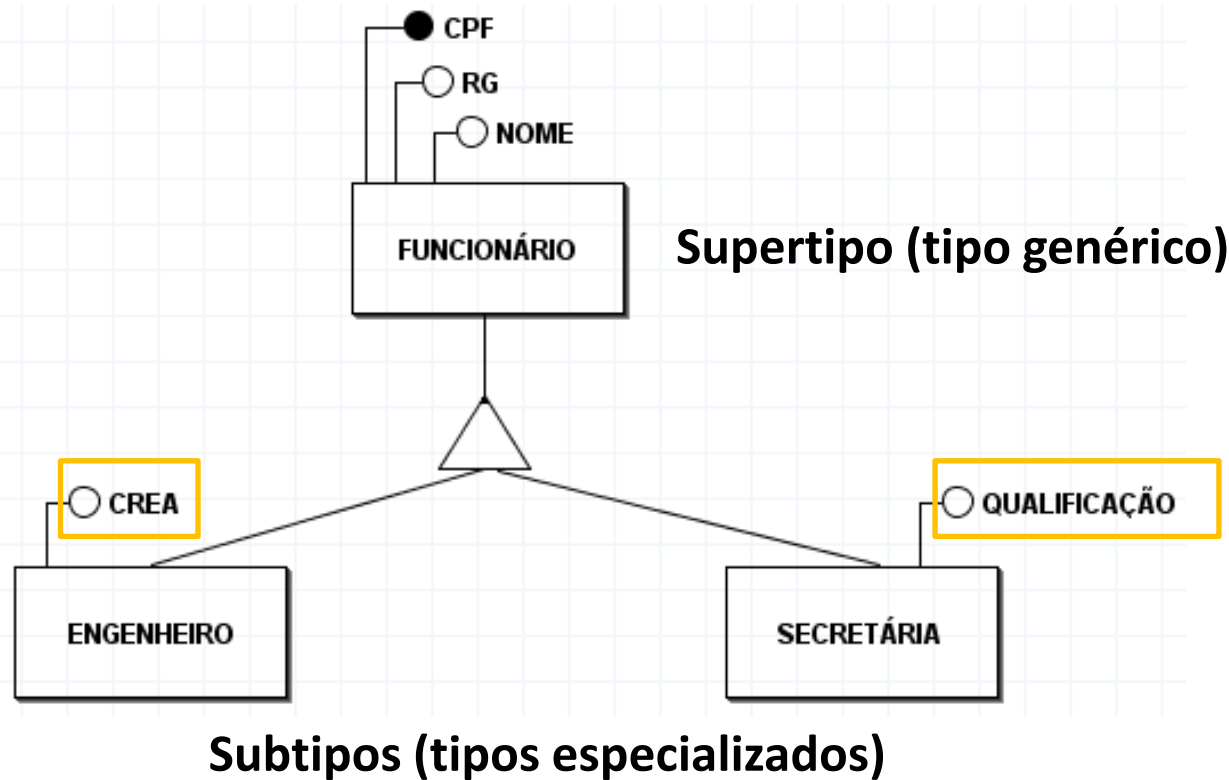
Generalização e Especialização

- Os subtipos podem herdar atributos do supertipo.



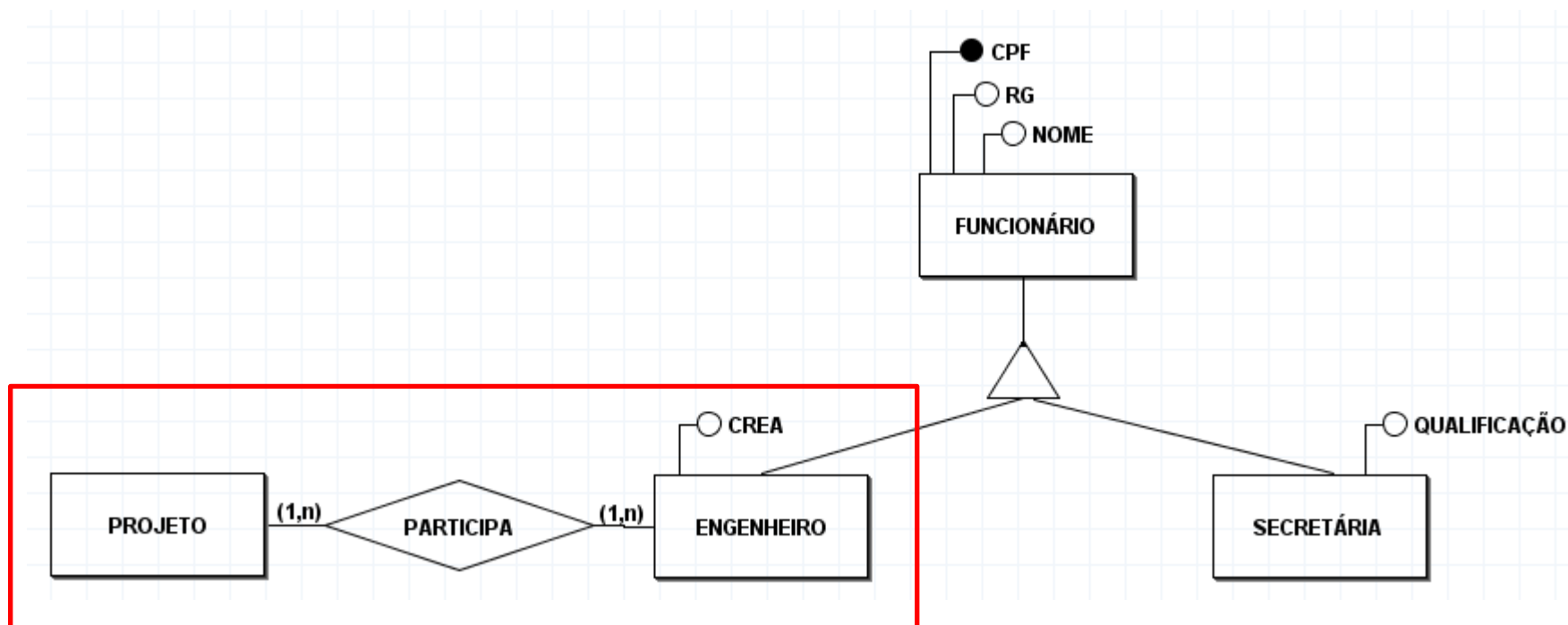
Generalização e Especialização

- Certos atributos aparecem somente em algum subtipo:



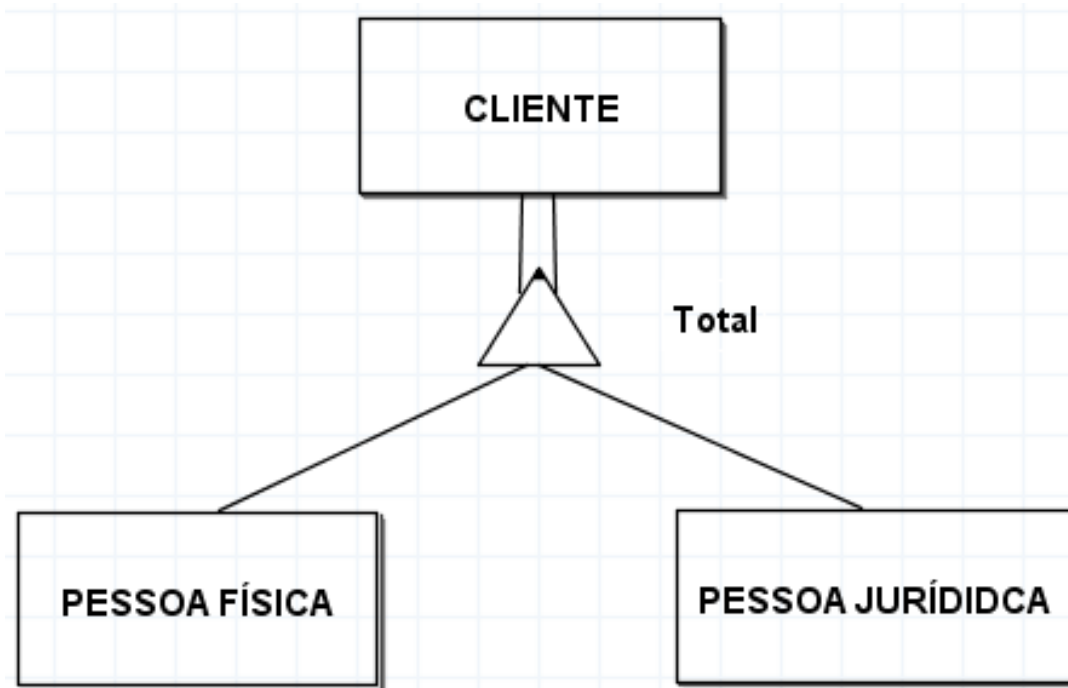
Generalização e Especialização

- Alguns relacionamentos se aplicam apenas a determinado subtipo:



Generalização/Especialização

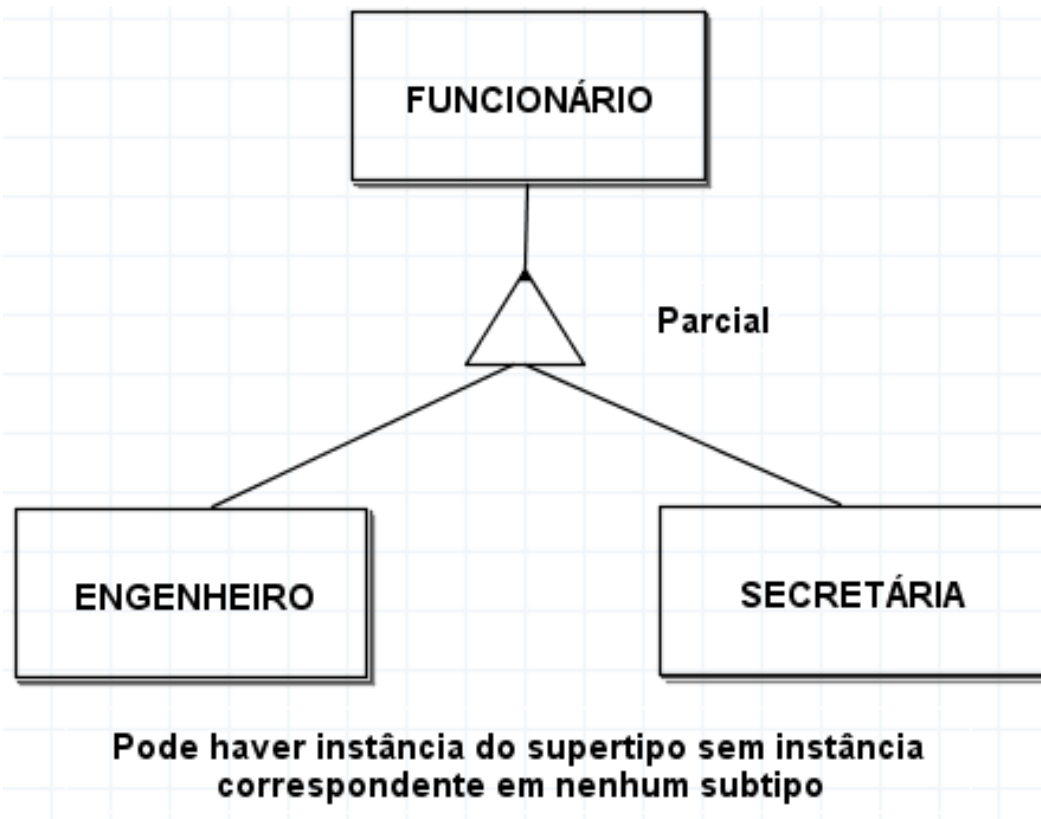
Restrição Total



Toda instância do supertipo precisa estar associado a uma instância correspondente de algum subtipo

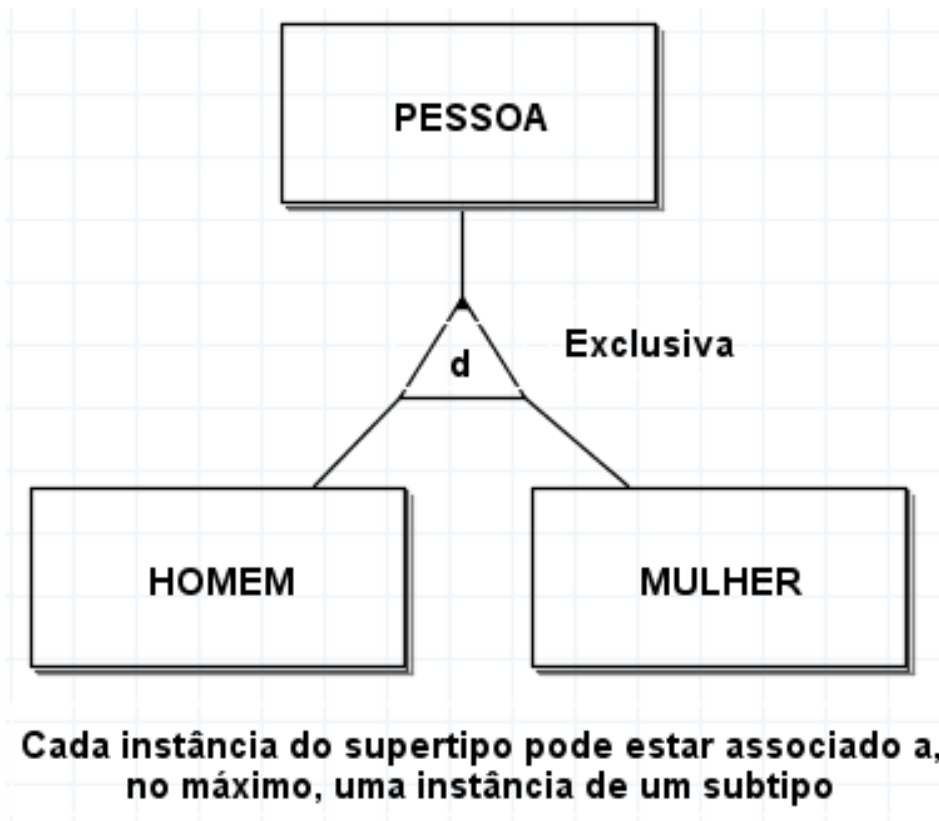
Generalização/Especialização

Restrição Parcial



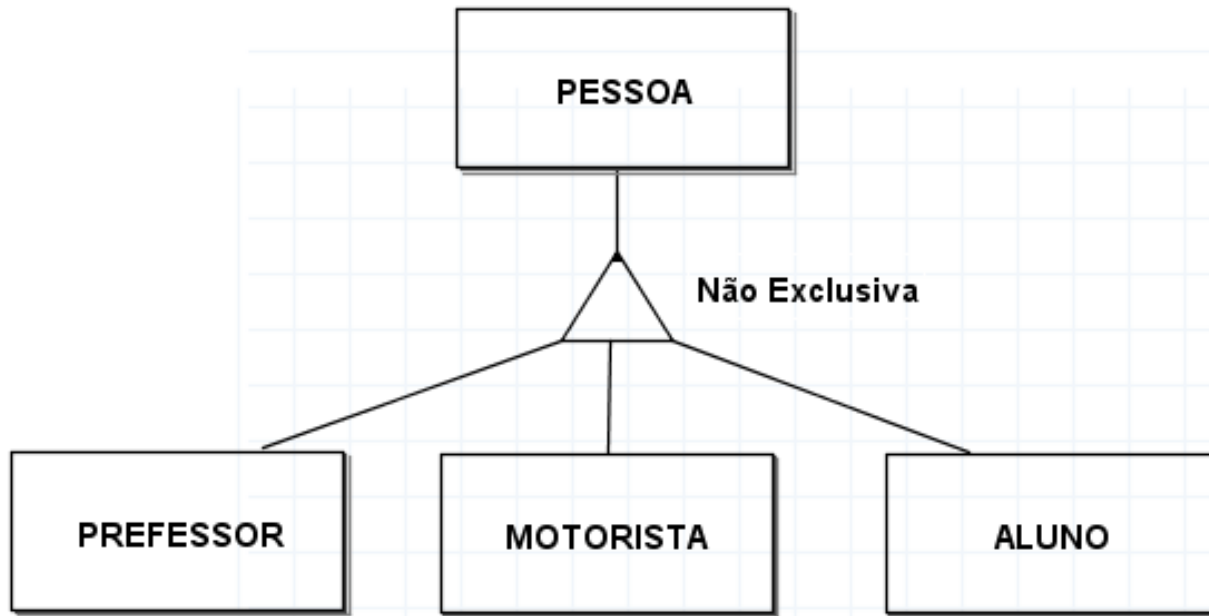
Generalização/Especialização

Restrição Exclusiva (disjunção)



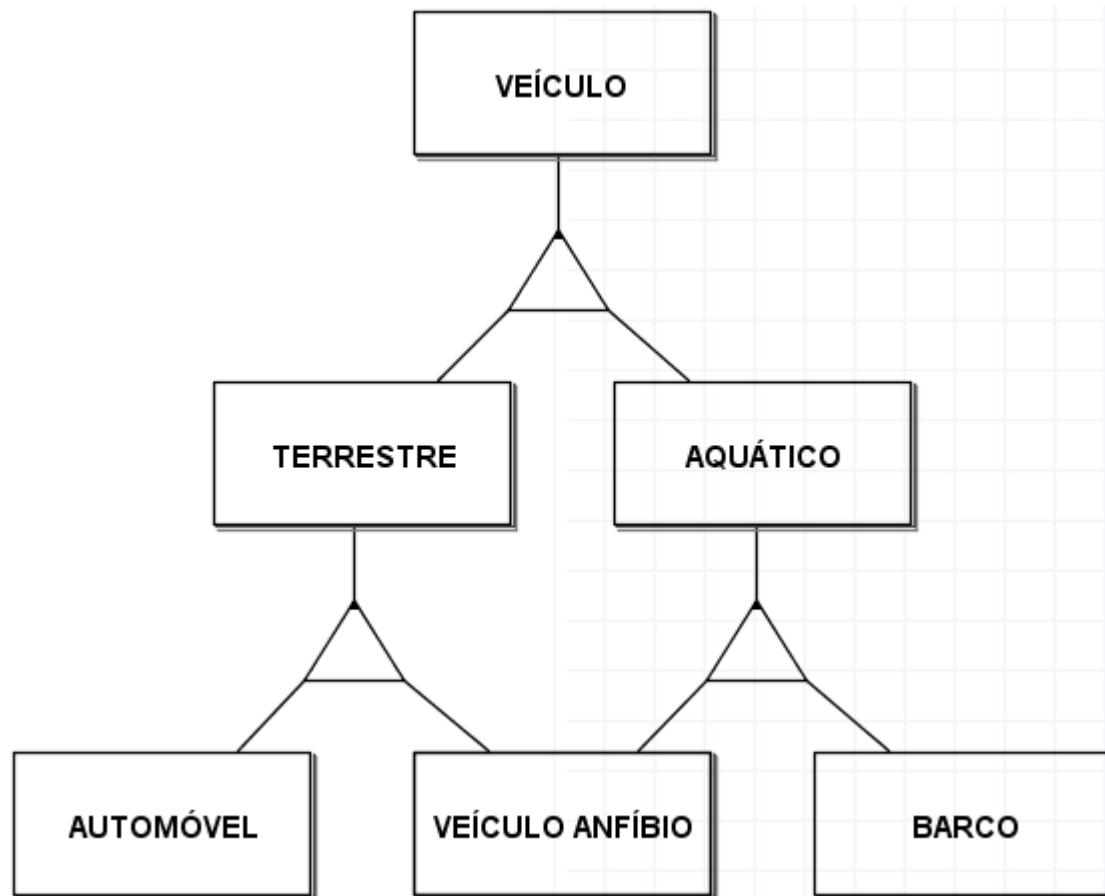
Generalização/Especialização

Restrição Não Exclusiva (sobreposição)



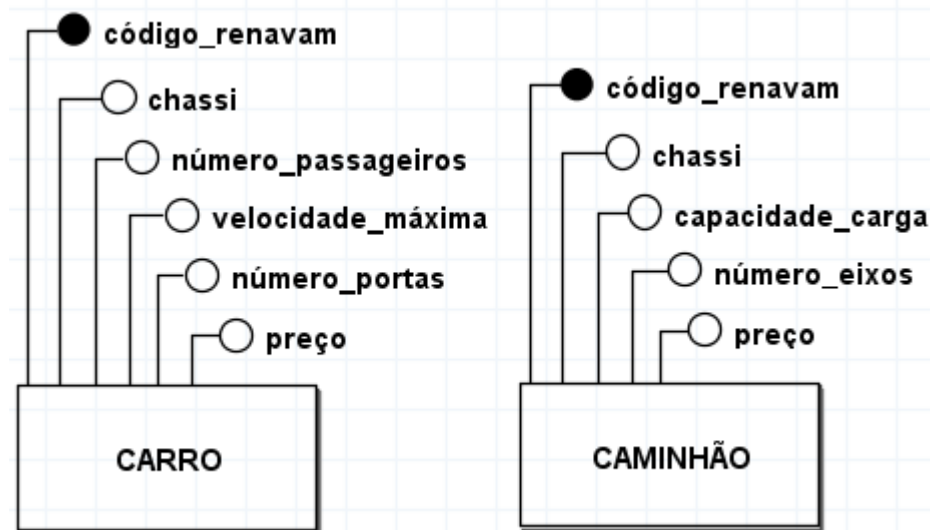
Uma instância do supertipo pode estar associada a até uma instância de cada um dos subtipos

Herança Múltipla



Exercício – Modelo ERE

- Modelo uma hierarquia de generalização e especialização para as entidades CARRO e CAMINHÃO. Defina as restrições para o modelo.



Modelo Relacional

- Modelo introduzido por Edgar F. Codd em 1970, tendo como finalidade representar os dados como uma coleção de *relações*, onde cada *relação* é representada por uma **tabela**.
- Modelo com sólida base formal:
 - Baseado na *teoria dos conjuntos* (álgebra relacional)

Modelo Relacional

- Cada linha da tabela representa uma coleção de dados relacionados:
 - Os dados podem ser interpretados como fatos que descrevem uma instância de uma entidade ou de um relacionamento.

Modelo Relacional

- De acordo com a sua terminologia:
 - Tabela:
 - É chamada de *relação*
 - Conjunto não ordenado de *tuplas*
 - Representam entidades ou relacionamentos
 - São definidas por esquemas do tipo $R(A_1, A_2, \dots, A_n)$, onde R é o nome da relação e A_1, A_2, \dots, A_n é a lista de atributos.

Modelo Relacional

- De acordo com a sua terminologia:
 - Linha da tabela:
 - É chamada de *tupla*
 - Representa uma coleção de dados relacionados
 - Todas a *tuplas* de uma relação são distintas

Modelo Relacional

- De acordo com a sua terminologia:
 - Coluna:
 - É denominada *atributo*
 - Nome que identifica uma propriedade da relação

Exemplo:

- A_1 representa o atributo A_1 da relação R

Modelo Relacional

- De acordo com a sua terminologia:
 - Tipo de dado:
 - É chamado *domínio*
 - É um conjunto de valores atômicos que caracterizam um atributo
 - Grau da relação:
 - Corresponde ao número de atributos da relação

Modelo Relacional

R	A₁	A₂	A₃	A₄
ESTUDANTE	Nome	Matrícula	Classe	Departamento
Tuplas	José	3217	1	DCT
	Ana Maria	2325	2	DCT
	Carla	4112	1	ENG

- **ESTUDANTE** é o nome da relação **R**, que tem 4 atributos (**R** de grau 4)
- $\text{dom}(\text{Nome}) = \text{Nome do aluno};$
 $\text{dom}(\text{Matrícula}) = \text{Número da matrícula};$
 $\text{dom}(\text{Classe}) = \text{Número da turma};$
 $\text{dom}(\text{Departamento}) = \text{Código do departamento a que os cursos estão vinculados}.$

Modelo Relacional

- Chaves:
 - Refere-se ao atributo ou conjunto de atributos que permite distinguir uma tupla das demais dentro da relação.
- Chave Candidata:
 - Atributo ou agrupamento de atributos que identifica unicamente uma ocorrência na relação.

Modelo Relacional

- Chave Primária (PK – Primary Key):
 - É a chave candidata escolhida pelo projetista do BD para identificar univocamente as tuplas de uma relação.
 - Ela deve ser preenchida obrigatoriamente, ou seja, **seu valor não pode ser nulo**.

Exemplo:

***Clientes** = {Codigo, CPE, RG, Nome, Endereco, LimCredito}*

Modelo Relacional

- Chave Primária Composta:
 - É a chave primária (PK) criada a partir da junção de dois ou mais atributos de uma relação.

Exemplos:

Contatos={*Nome*, *Telefone*}

Contas={*Agencia*, *Numero*, *Saldo*,
CodCliente, *DtAbertura*}

Importante: poderá haver valores repetidos para cada um dos atributos-chave, porém, a junção dos atributos deve prover valor único.

Modelo Relacional

- Chave Estrangeira (FK – Foreign Key):
 - É a quando o atributo de uma relação é a chave primária de outra relação, ou seja, pode armazenar somente valores de uma chave primária de outra relação.

Exemplos:

Clientes={Codigo, Nome, Telefone}

Vendas={NotaFiscal, **CodCliente**, Produto}

Restrições do Modelo Relacional

- Restrições de cardinalidade:
 - Indica o número de relacionamentos dos quais uma entidade pode participar:
 - um para um (1:1)
 - um para muitos (1:N)
 - muitos para muitos (M:N)

Restrições do Modelo Relacional

- Restrições de integridade de domínio:
 - Os valores de um atributo devem pertencer ao domínio do atributo
- Restrições de integridade da chave:
 - Não podem existir duas ou mais tuplas de uma mesma relação com valores iguais na chave primária (PK)

Restrições do Modelo Relacional

- Restrições de integridade de entidade:
 - Os valores da chave primária não podem ser NULOS (*NULL*).
- Restrições de integridade referencial:
 - Estabelece que uma tupla de uma relação que se refere à outra relação deve se referir a uma tupla existente naquela relação.

Dependência Funcional

- Dependência funcional é uma restrição entre 2 conjuntos de atributos:
 - Seja $R(A_1, A_2, \dots, A_n)$ um esquema relacional e X e Y sejam dois subconjuntos de atributos de R . Diz-se que X determina funcionalmente Y (ou que Y depende funcionalmente de X), representado por $X \rightarrow Y$, se quaisquer duas tuplas de R têm os mesmos valores em X também têm os mesmos valores em Y :

$t1[X] = t2[X]$, é obrigado a existir $t1[Y] = t2[Y]$

Dependência Funcional

- **Exemplos:**

Cliente={Código, Nome}

Código → Nome

Projeto={NumProjeto, Nome, Localização}

NumProjeto → Nome,Localização

Dependência Funcional Total

- Numa relação **R** o atributo **Y** é funcionalmente dependente total de **X**, no caso de **X** ser um atributo composto, se, e apenas se, é funcionalmente dependente de **X** e não é funcionalmente dependente de qualquer subconjunto dos atributos de **X**.

Dependência Funcional Total

- **Exemplos:**

Pedido={CodPedido, Cliente, DtPedido}

Produto = {CodProduto, Nome, Descrição}

ItemPedido={CodPedido, CodProduto, Quantidade}

CodPedido, CodProduto → Quantidade

Dependência Funcional Parcial

- Uma dependência funcional parcial é uma dependência funcional $X \rightarrow Y$ se existir um atributo A qualquer do componente X que pode ser removido e a dependência funcional $X \rightarrow Y$ não deixa de existir.

Dependência Funcional Parcial

- **Exemplos:**

Notas={NumMatricula, CodDisciplina, Período,
NomeDisciplina, Nota}

NomeDisciplina depende apenas do **CodDisciplina**.

Dependência Funcional Transitiva

- Ocorre quando atributos não chave da relação não dependem funcionalmente do atributo chave (ou de parte dela), mas de outros atributos da relação.

- **Exemplos:**

Funcionário={Matricula, Nome, CodCargo, Cargo, SalárioCargo}

Cargo e SalárioCargo dependem funcionalmente do atributo não chave CodCargo.

CodCargo \rightarrow Cargo, SalárioCargo

Normalização

- É um conjunto de regras que visa, principalmente, a organização de um projeto de banco de dados para reduzir a redundância de dados, aumentar a integridade dos dados e o desempenho.
- Por que normalizar?
 - Evitar anomalias de inserção, alteração e remoção
 - Facilitar a manutenção
 - Manter a integridade dos dados

Normalização

- Principais Formas Normais:
 - 1FN – Primeira Forma Normal
 - 2FN – Segunda Forma Normal
 - 3FN – Terceira Forma Normal

Primeira Forma Normal – 1FN

- Para estar na primeira forma normal (1FN), todos os atributos da relação devem ser atômicos e monovalorados.

Cliente={CÓDIGO, NOME, TELEFONE, ENDEREÇO}

<u>CÓDIGO</u>	NOME	TELEFONE	ENDEREÇO
1	EMANUEL	1111-1111 1010-1010	RUA ARARÍPE, Nº 10, CENTRO, SALVADOR
2	JOSÉ	1234-5678	AV PADRE AMÁRO, Nº 201, LESTE, SÃO PAULO

Primeira Forma Normal – 1FN

- É necessário reestruturar a relação para que ela atenda à 1FN.

Cliente={CÓDIGO, LOGRADOURO, NÚMERO, BAIRRO, CIDADE}

<u>CÓDIGO</u>	NOME	LOGRADOURO	NÚMERO	BAIRRO	CIDADE
1	EMANUEL	RUA ARARÍPE	10	CENTRO	SALVADOR
2	JOSÉ	AV PADRE AMÁRO	201	LESTE	SÃO PAULO

Primeira Forma Normal – 1FN

Cliente_Telefone={CÓDIGO CLIENTE,
TELEFONE CLIENTE}

<u>CÓDIGO_CLIENTE</u>	<u>TELEFONE_CLIENTE</u>
1	1111-1111
1	1010-1010
2	1234-5678

Segunda Forma Normal – 2FN

- Para estar na segunda forma normal (2FN), a relação deve, obrigatoriamente, estar na 1FN e todos os atributos não chave devem depender funcionalmente de toda a chave primária (composta).

Cliente={CÓD, CLIENTE}

Filme={CÓD, FILME}

Locação={CÓD-CLIENTE, CÓD-FILME, SALDO}

Segunda Forma Normal – 2FN

Cliente={CÓD, CLIENTE}

<u>CÓD</u>	CLIENTE
1	EMANUEL
2	JOSÉ

Filme={CÓD, FILME}

<u>CÓD</u>	FILME
1	CAPITÃO AMÉRICA
2	THOR

Locação={CÓD-CLIENTE, CÓD-FILME, SALDO}

<u>CÓD-CLIENTE</u>	<u>CÓD-FILME</u>	SALDO
1	1	2,50
2	2	0,00
1	2	0,00

Segunda Forma Normal – 2FN

- Na relação **Locação**, o atributo SALDO depende funcionalmente apenas de parte da chave composta da relação.

CÓD-CLIENTE, Cód-FILME → SALDO
CÓD-FILME → SALDO

<u>CÓD-CLIENTE</u>	<u>CÓD-FILME</u>	SALDO
1	1	2,50
2	2	0,00
1	2	0,00

Segunda Forma Normal – 2FN

- Temos que transpor o atributo SALDO para relação **CLIENTE** para que a relação **Locação** atenda à 2FN:

Cliente={CÓD, CLIENTE, SALDO}

<u>CÓD</u>	<u>CLIENTE</u>	SALDO
1	EMANUEL	0,00
2	JOSÉ	0,00

Segunda Forma Normal – 2FN

Locação={CÓD-CLIENTE, CÓD-FILME}

<u>CÓD-CLIENTE</u>	<u>CÓD-FILME</u>
1	1
2	2
1	2

Terceira Forma Normal – 3FN

- Para estar na terceira forma normal (3FN), a relação deve, obrigatoriamente, estar na 2FN e não haver dependência transitiva entre os atributos não chave da relação.

Funcionário={CÓD, FUNCIONÁRIO, CÓD-CARGO, CARGO, SALÁRIO-CARGO}

<u>CÓD</u>	FUNCIONÁRIO	CÓD-CARGO	CARGO	SALÁRIO-CARGO
1	EMANUEL	1	MÉDICO	15000
2	JOSÉ	2	DENTISTA	6000

Terceira Forma Normal – 3FN

- Os atributos CARGO e SALÁRIO-CARGO dependem funcionalmente apenas do atributo não chave CÓD-CARGO:
CÓD-CARGO → CARGO, SALÁRIO-CARGO
- É necessário retirar os atributos CARGO e SALÁRIO-CARGO da relação Funcionário e transferi-los para uma nova relação:

Terceira Forma Normal – 3FN

Funcionário={CÓD, FUNCIONÁRIO, CÓD-CARGO}

<u>CÓD</u>	FUNCIONÁRIO	CÓD-CARGO
1	EMANUEL	1
2	JOSÉ	2

Cargo={CÓD, CARGO, SALÁRIO-CARGO}

<u>CÓD</u>	CARGO	SALÁRIO-CARGO
1	MÉDICO	15000
2	DENTISTA	6000

Resumo

- A normalização é utilizada para evitar anomalias de inserção, alteração e remoção, facilitar a manutenção e manter a integridade dos dados, mas deve ser utilizada com cautela, pois há citações em que uma certa “desnormalização” é requerida para melhorar o desempenho.