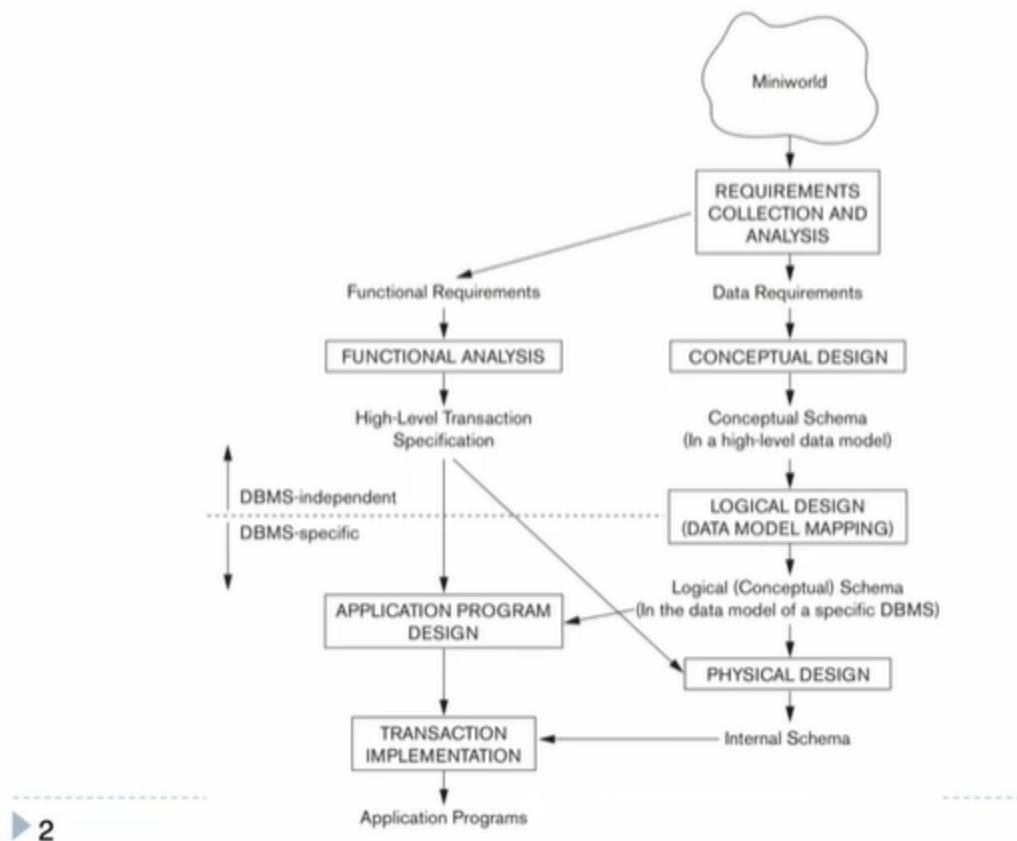


# Modelo ER

## ER: Entidade-Relacionamento

- ▶ **Entidades**
  - ▶ Coisas/objetos pertencentes ao mini-mundo
- ▶ **Relacionamentos**
  - ▶ Relacionam uma ou mais entidades

## Processo de Design de BD



# Modelo ER - Conceitos

---

- ▶ Atributos de Entidades
  - ▶ Propriedades usadas para descrever uma entidade

---
- ▶ Uma entidade específica vai ter um valor para seus atributos;
- ▶ Cada atributo tem um *value set* (conjunto de valores) Associado a ele – e.g. integer, string, date, ...

## Tipos de atributos

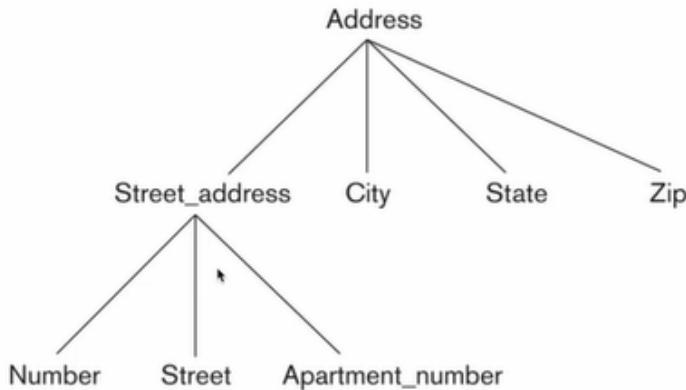
---

- ▶ Simples
  - ▶ Ex: SSN, Sex.
- ▶ Compostos
  - ▶ Ex:
    - ▶ Address(Apt#, House#, Street, City, State, ZipCode, Country)
    - ▶ Name(FirstName, MiddleName, LastName).
    - ▶ Podem formar uma hierarquia
- ▶ Multi-valorados
  - ▶ Ex: Cor de um ROUPA
    - ▶ Representado por {Cor}

Multi-valorados -> Vários valores diferentes, como a cor de uma roupa {Branco, vermelho amarelo}

## Exemplos

**Composto (Hierarquia):**



**Composto e multi-valorado:**

{PreviousDegrees (College, Year, Degree, Field)}

## Tipos de entidades e atributos-chave

### ► Tipos de Entidade

- ▶ Correspondem a um mesmo tipo de objeto/coisa do mini-mundo; possuem mesmo conjunto de atributos.
- ▶ Ex: EMPLOYEE e PROJECT.

### ► Chaves

- ▶ Atributos com a propriedade de unicidade
- ▶ Ex: SSN de EMPLOYEE.

## Chaves

---

- ▶ Chave pode ser composta
  - ▶ Placa (Estado, Numero).
- ▶ Uma entidade pode ter mais de uma chave
  - ▶ Carro:
    - ▶ Renavan
    - ▶ Placa
- ▶ Cada chave é sublinhada (diferente do esquema relacional onde só a chave primária é sublinhada).

## Tipo de Entidade VS Conjunto de Entidade

---

- ▶ Cada tipo de entidade terá uma coleção de entidades armazenada no banco de dados
  - ▶ Chamado de **conjunto** de entidades ou **coleção** de entidades;
  - ▶ O **mesmo nome** pode ser usado para referir ao **tipo** de entidade ou ao **conjunto de entidades**;
  - ▶ Conjunto de entidades representa o estado atual de um banco de dados;

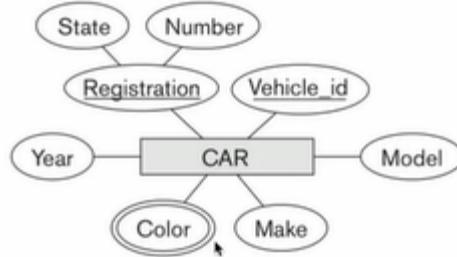
## Domínios de atributos

---

- ▶ Cada atributo simples é associado a um conjunto valor
  - ▶ Ex: Lastname: string de comprimento até 15 caracteres
- ▶ Um **conjunto valor** especifica o conjunto de valores associado a um atributo
- ▶ Matematicamente, define-se um atributo A de uma entidade E cujo conjunto valor é V como uma **função**:

$$A : E \rightarrow P(V)$$

onde  $P(V)$  indica a **conjunto potência** (todos os possíveis subconjuntos) de V.



Linha dupla -> Atributo multi-valorado

## Esquema COMPANY

- ▶ Entidades
  - ▶ DEPARTMENT
  - ▶ PROJECT
  - ▶ EMPLOYEE
  - ▶ DEPENDENT

Nem tudo visto como tabela é uma entidade, uma tabela pode representar relacionamentos de entidades também.

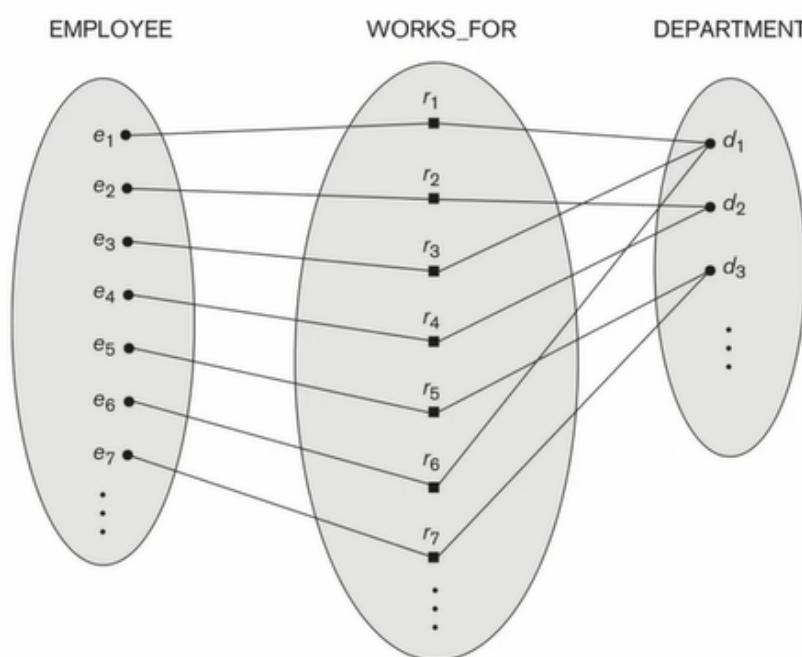
## Refinando o design: **relacionamentos**

- ▶ O design inicial não está completo
- ▶ Alguns requisitos devem ser representados através de **relacionamentos**
- ▶ O modelo ER possui 3 principais conceitos:
  - ▶ Entidade (com tipos de entidades e conjuntos de entidades)
  - ▶ Atributo (simples, composto, multivalorado)
  - ▶ Relacionamento (com tipos de relacionamentos e conjuntos de relacionamentos)

# Relacionamentos

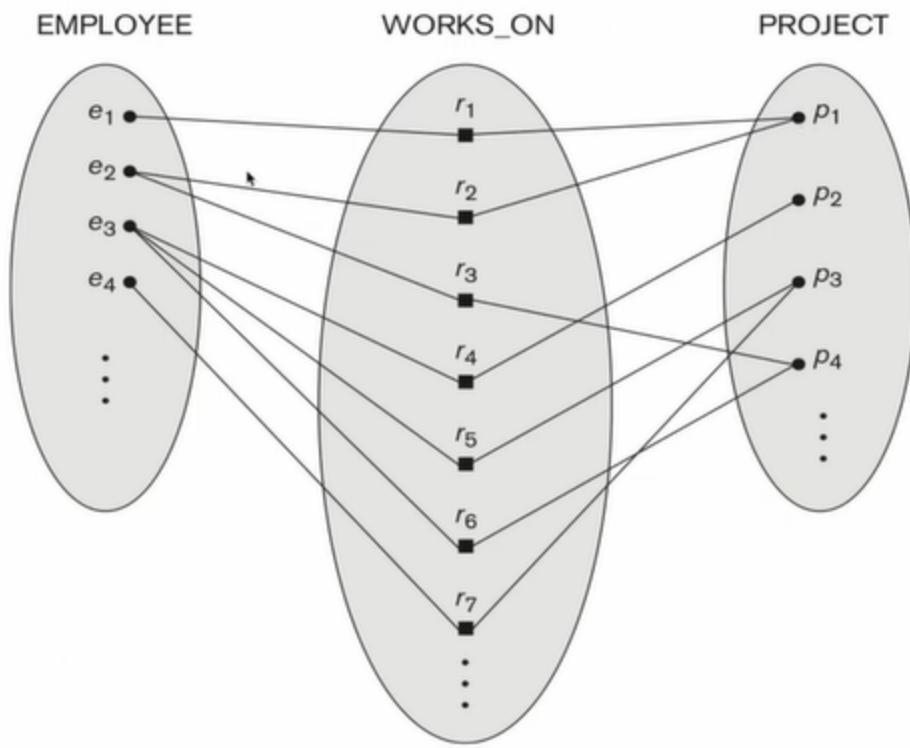
- ▶ Um **relacionamento** relaciona duas ou mais entidades.
  - ▶ Ex: EMPLOYEE John Smith *works on* the ProductX PROJECT
  - ▶ Ex: EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.
- ▶ Relacionamentos que relacionam entidades de mesmo tipo são agrupados e chamados de um **tipo de relacionamento**
  - ▶ Ex: WORKS\_ON: onde EMPLOYEES e PROJECTs **participam**;
  - ▶ Ex: MANAGES: onde EMPLOYEES e DEPARTMENTS **participam**.
- ▶ **Grau** de um relacionamento: número de entidades participantes:
  - ▶ MANAGES e WORKS\_ON são relacionamentos **binários**.

## Instâncias do relacionamento WORKS\_FOR



Empregado -> Departamento

## Instâncias do relacionamento WORKS\_ON



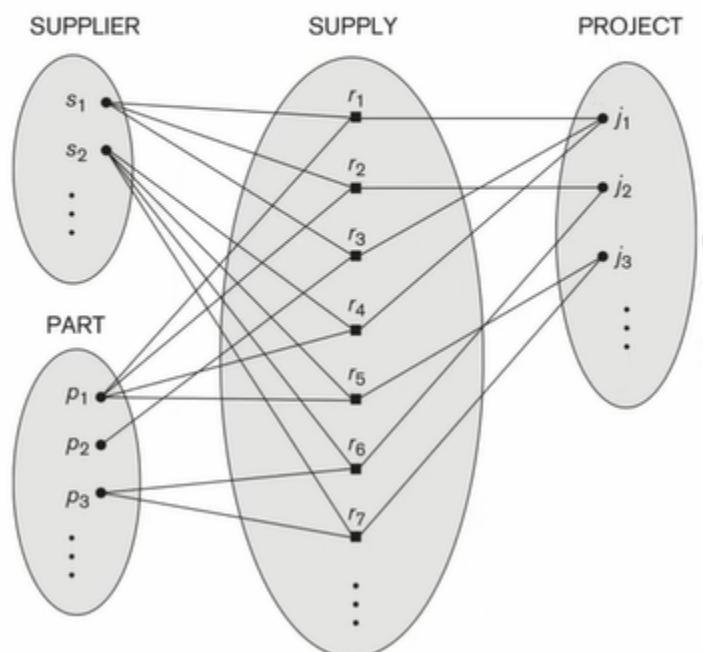
## Tipo de Relacionamento VS Conjunto de Relacionamentos

- ▶ **Tipo de relacionamento:**
  - ▶ Descrição esquemática do relacionamento
  - ▶ Identifica o nome do relacionamento e as entidades participantes
  - ▶ Também pode identificar certas constraints
- ▶ **Conjunto de relacionamentos:**
  - ▶ Atual conjunto de instâncias de relacionamentos existentes no BD
  - ▶ Estado atual de um tipo de relacionamento

## Relacionamento do Esquema COMPANY

- ▶ Ao analisar os requisitos, os seguintes relacionamentos são identificados (todos binários / grau 2):
  - ▶ WORKS\_FOR (entre EMPLOYEE, DEPARTMENT)
  - ▶ MANAGES (entre EMPLOYEE, DEPARTMENT)
  - ▶ CONTROLS (entre DEPARTMENT, PROJECT)
  - ▶ WORKS\_ON (entre EMPLOYEE, PROJECT)
  - ▶ SUPERVISION (entre EMPLOYEE (como subordinado), EMPLOYEE (como supervisor))
  - ▶ DEPENDENTS\_OF (entre EMPLOYEE, DEPENDENT)

## Exemplo de relacionamento **ternário**



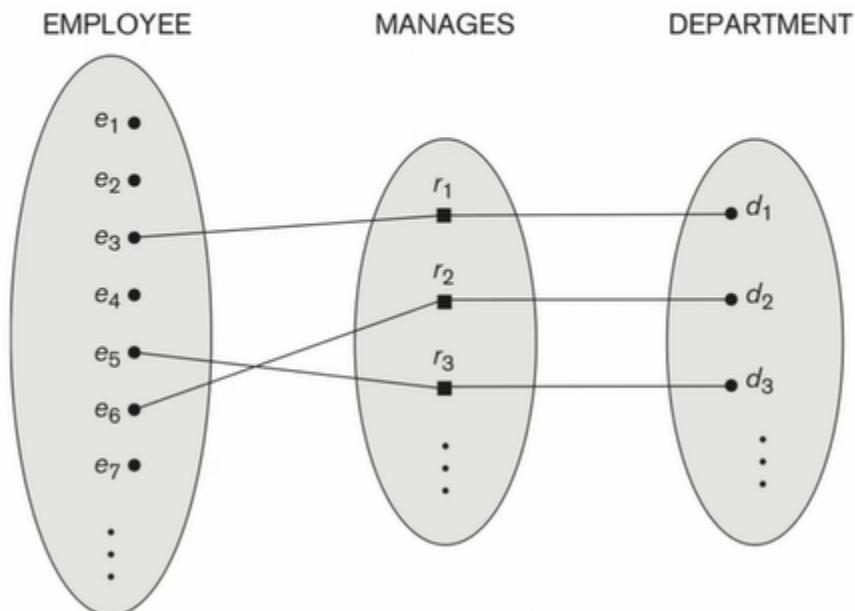
# Restrições em Relacionamentos

## ► Restrições (Constraints) em Relacionamentos:

- ▶ Razão de cardinalidade (especifica a participação máxima)
  - ▶ Um-para-um (1:1)
  - ▶ Um-para-muitos (1:N) ou Muitos-para-um (N:1)
  - ▶ Vários-para-vários (M:N)
- ▶ Dependências da existência (especifica a participação mínima) (também chamado de restrição de participação)
  - ▶ zero (participação opcional, não dependente da existência)
  - ▶ um ou mais (participação obrigatória dependente da existência)

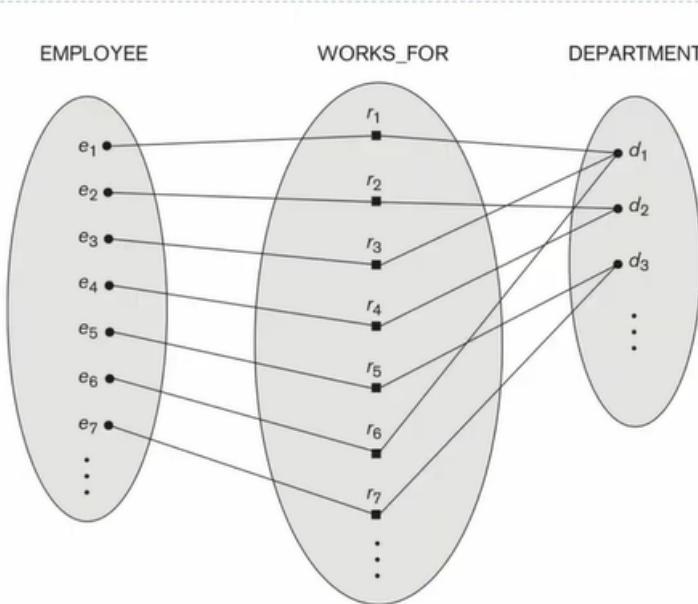
Critério de participação das entidades entre o relacionamento.

## Um-para-um (1:1)



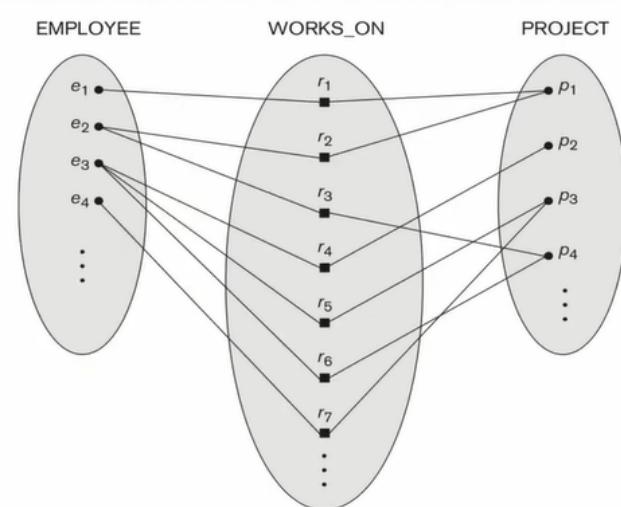
**Todo departamento tem no máximo 1 gerente e um funcionamento gerencia no máximo um departamento (1:1)**

## Muitos-para-um (N:1)



**Um funcionário trabalha para apenas um departamento, mas um departamento pode ter muitos funcionários.**

## Muitos-para-muitos (M:N)

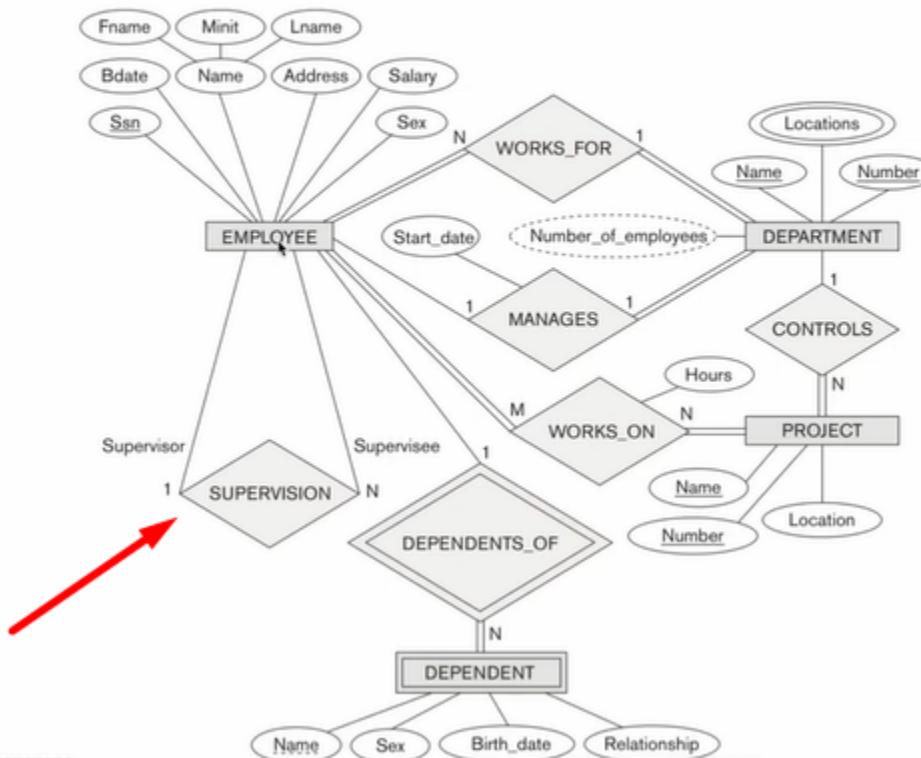


**Um empregado pode tá em vários projetos, e um projeto pode ter vários funcionários.**

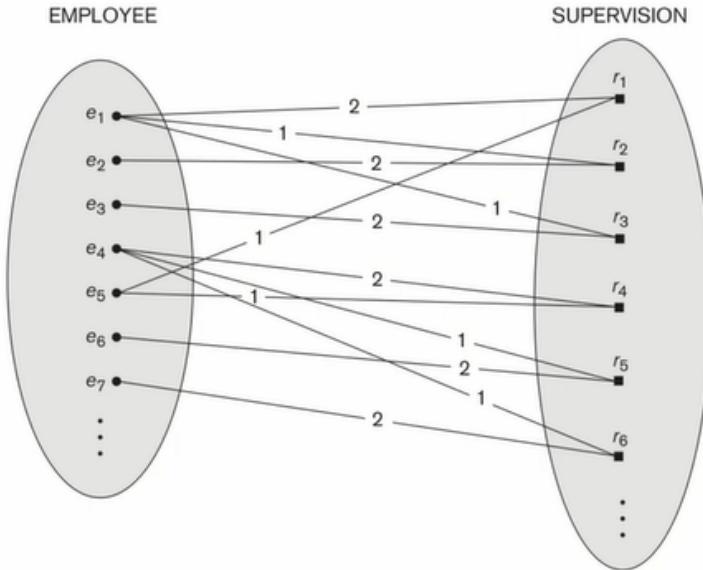
## Relacionamento Recursivo

- ▶ Um tipo de relacionamento entre o mesmo tipo de entidade participante em **papéis distintos**
- ▶ Também chamado de um tipo de relacionamento de **auto-referência**.
- ▶ Ex: relacionamento **SUPERVISION**
- ▶ **EMPLOYEE** participa com dois papéis:
  - ▶ Supervisor (ou chefe)
  - ▶ Supervisionado (ou subordinado)

## Diagrama ER



# Relacionamento Recursivo



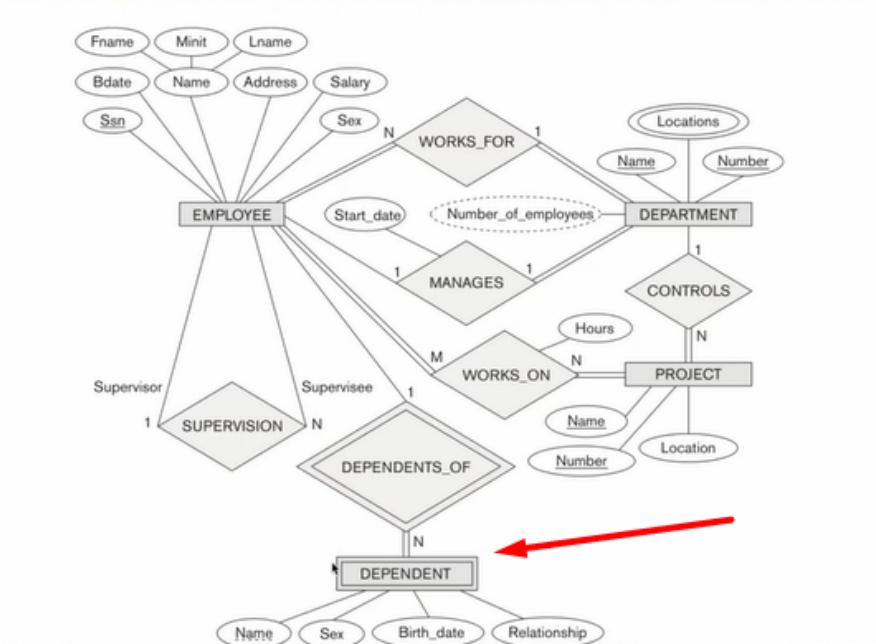
## Entidades Fracas

- ▶ **Entidade fraca:** um tipo de entidade que **não tem um atributo de chave** e é “**dependente de identificação**” por outro tipo de entidade.
- ▶ Uma entidade fraca **deve participar** de um tipo de relacionamento de identificação com uma “entidade de identificação” (chamada de entidade dona/proprietária);
- ▶ Entidades são identificadas pela combinação de:
  - ▶ Uma chave parcial da entidade fraca
  - ▶ A entidade em particular que estão relacionadas através do “relacionamento de identificação”
- ▶ **Exemplo:**
  - ▶ DEPENDENT é uma **entidade fraca**
  - ▶ Nome de DEPENDENT é uma **chave parcial**
  - ▶ EMPLOYEE é sua entidade de identificação através do relacionamento de identificação DEPENDENT\_OF

**Depende de outra para existir.**

Não tem uma chave para se identificar. Por si só fica “solto” “perdido” Precisa juntar atributos que estão na outra tabela, é a identidade de IDENTIFICAÇÃO, e a chave da entidade fraca é uma chave parcial. E quando juntamos a entidade fraca com a outra que faz a identificação fazemos um relacionamento de identificação.

## Diagrama ER

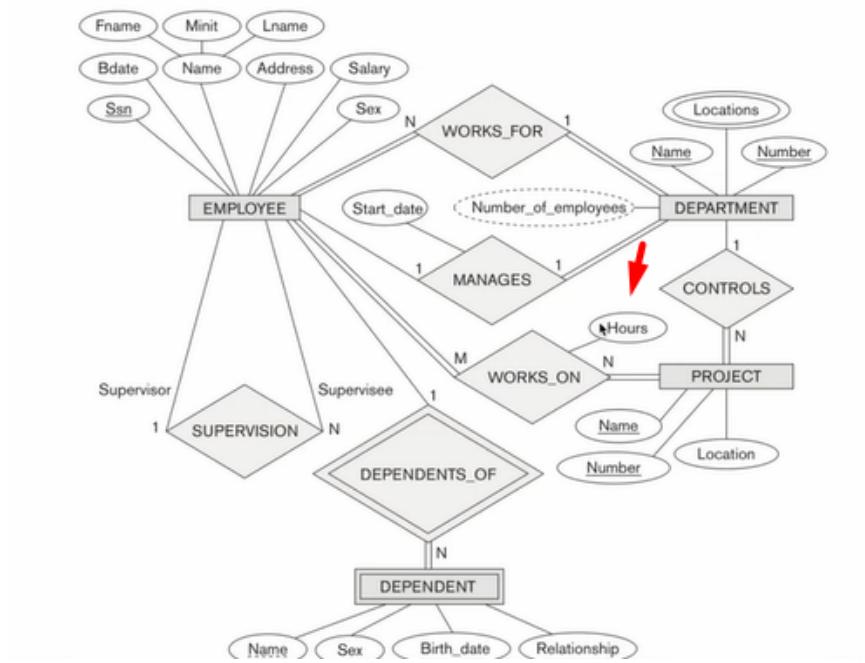


## Atributos em Relacionamentos

- ▶ Relacionamentos podem conter atributos
  - ▶ Ex: HoursPerWeek em WORKS\_ON
- ▶ Cada instância pode ter um valor diferente associado
- ▶ A maioria dos atributos aparecem em relacionamentos M:N
  - ▶ Em relacionamentos 1:N, eles podem ser associados ao lado N

Pode ter atributos no relacionamento.

# Diagrama ER

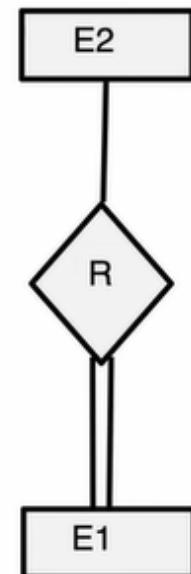


Linha dupla → Participação Total

Linha simples → Participação Parcial

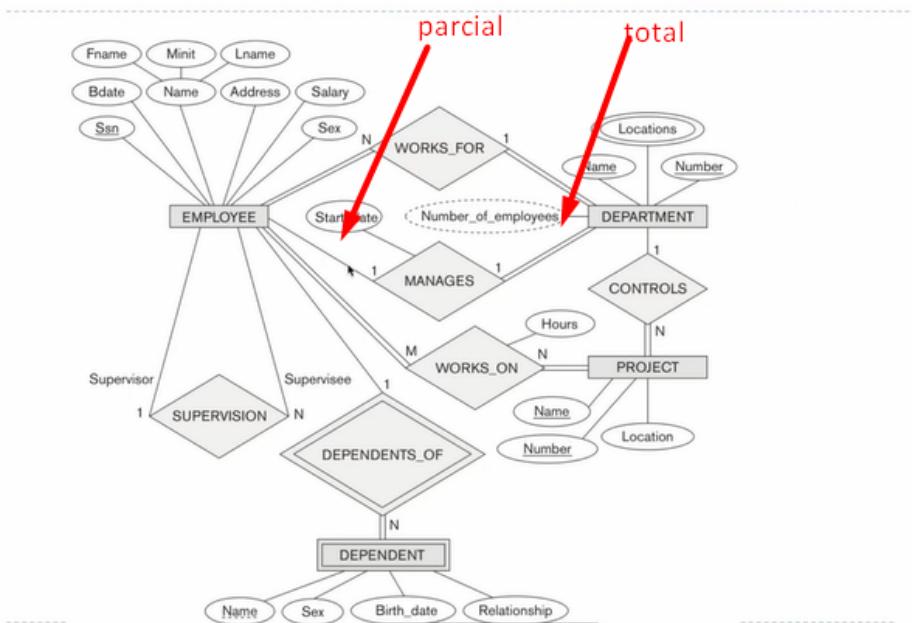
# Notação para constraints em relacionamentos

- ▶ Razão de cardinalidade: 1:1, 1:N, N:1, M:N
  - ▶ Posicionados nas arestas (linhas de ligação do diagrama).
- ▶ Restrição de participação (Participation Constraint)
  - ▶ **Total**
    - ▶ Mínimo de 1 (um)
    - ▶ “Dependência de existência”
    - ▶ Toda instância de E1 ocorre em R
    - ▶ **Linha dupla**
  - ▶ **Parcial**
    - ▶ Nenhum mínimo
    - ▶ Podem haver instâncias de E2 que não ocorrem em R
    - ▶ **Linha simples**



Total → Pelo menos uma participação. “Dependência de existência”  
Parcial → Não existe um mínimo.

## Diagrama ER

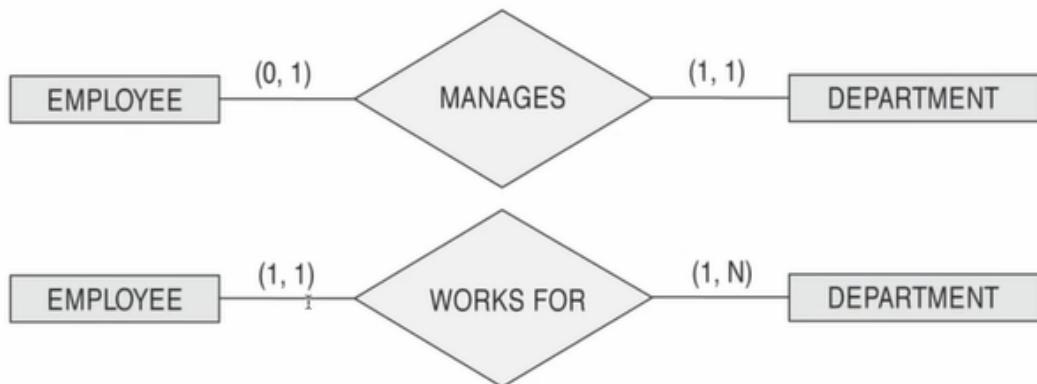


## Notação (min, max)

- ▶ Especificado em cada participação de um tipo de entidade E em um tipo de relacionamento R
- ▶ Especifica que cada entidade E participa em pelo menos *min* e no máximo *max* instâncias do relacionamento R
- ▶ Default(sem constraint): *min*=0, *max*=n (ou seja, nenhum limite)
- ▶  $\text{min} \leq \text{max}$ ,  $\text{min} \geq 0$ ,  $\text{max} \geq 1$
- ▶ Derivado das restrições do mini-mundo
- ▶ Exemplos:
  - ▶ Um departamento tem exatamente um gerente e um funcionário pode gerenciar no máximo um departamento .
    - ▶ Especificar (0,1) para a participação de EMPLOYEE em MANAGES
    - ▶ Especificar (1,1) para a participação de DEPARTMENT em MANAGES
  - ▶ Um funcionário pode trabalhar para exatamente um departamento mas um departamento pode ter qualquer número de funcionários.
    - ▶ Especificar (1,1) para a participação de EMPLOYEE em WORKS\_FOR
    - ▶ Especificar (0,n) para a participação de DEPARTMENT em WORKS\_FOR

**WORKS\_FOR -> Empregado trabalha em um departamento.**

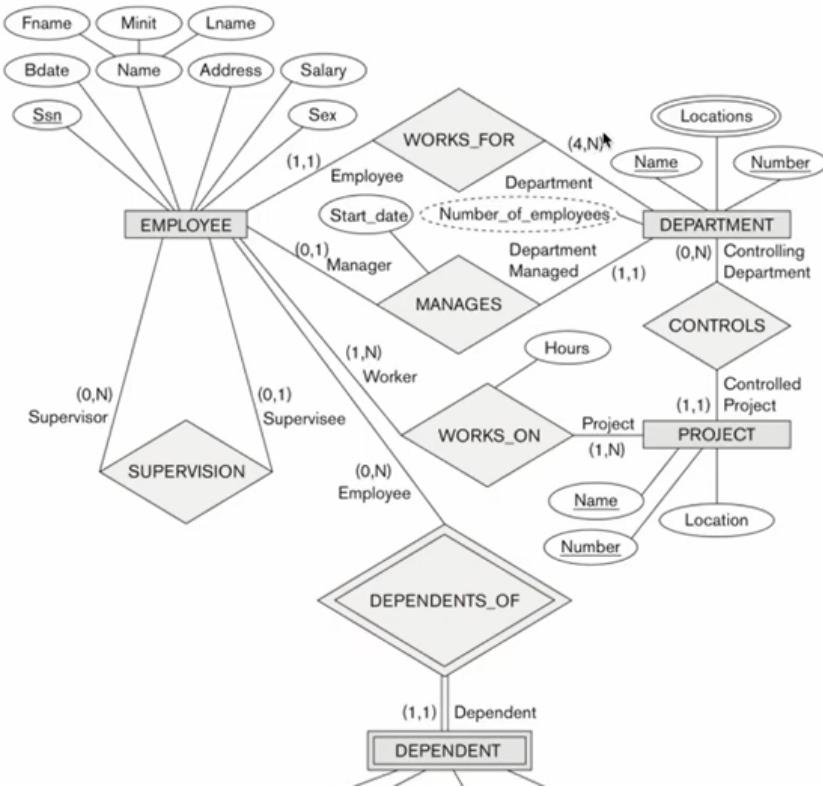
## Notação (min,max)



**Atenção:** nesta notação clássica, (min, max) corresponde à entidade mais distante!

Porém, a maioria das notações alternativas utilizam a representação invertida. Utilizaremos a notação invertida em nossas atividades.

# Diagrama ER com notação (min, max)



Linha tracejada → representa um atributo derivado de outros atributos, não necessariamente está representando no modelo lógico como uma coluna, é uma coluna que pode ser considerada no modelo conceitual, mas no lógico não está representado.

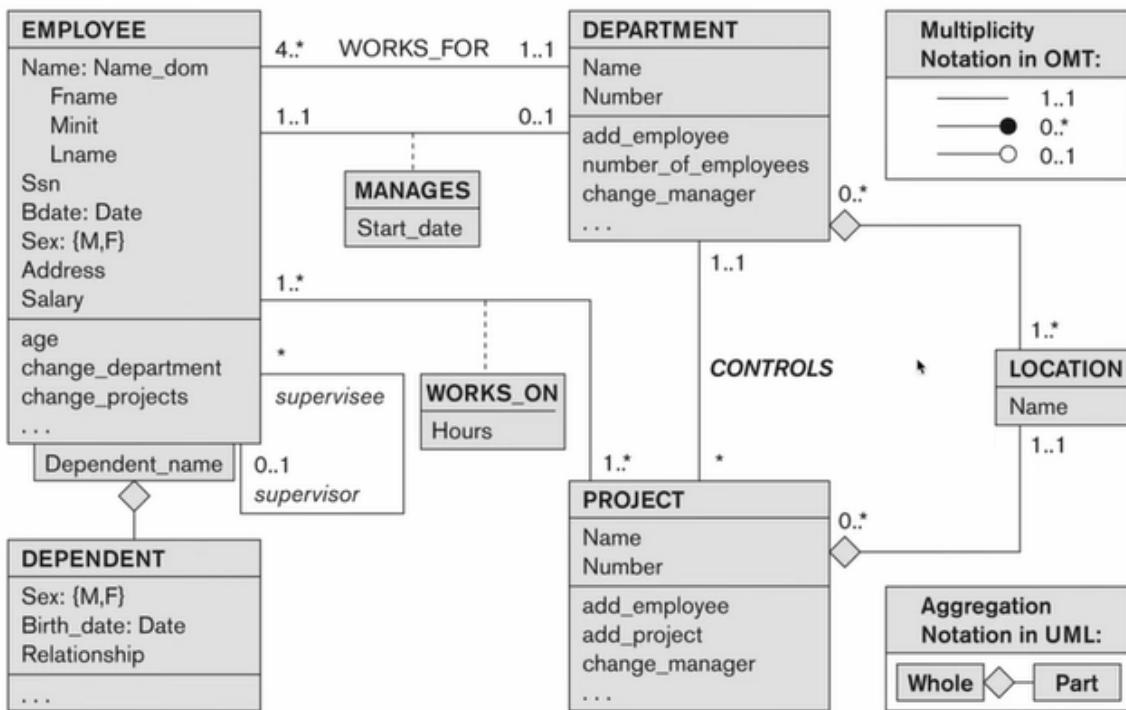
# Sumário da notação de diagramas ER

Symbol	Meaning	
	Entity	 Composite Attribute
	Weak Entity	 Derived Attribute
	Relationship	 Total Participation of E2 in R
	Identifying Relationship	 Cardinality Ratio 1: N for E1 : E2 in R
	Attribute	
	Key Attribute	
	Multivalued Attribute	

## Notações diagramáticas alternativas

- ▶ **Diagrama ER** é um exemplo **popular** para a exibição de esquemas de banco de dados;
- ▶ Muitas **outras notações** existem na **literatura** e em várias **ferramentas** de design e modelagem de banco de dados;
- ▶ **Diagrama de Classe UML** é uma outra maneira de exibir conceitos ER que é adotada em várias ferramentas comerciais;

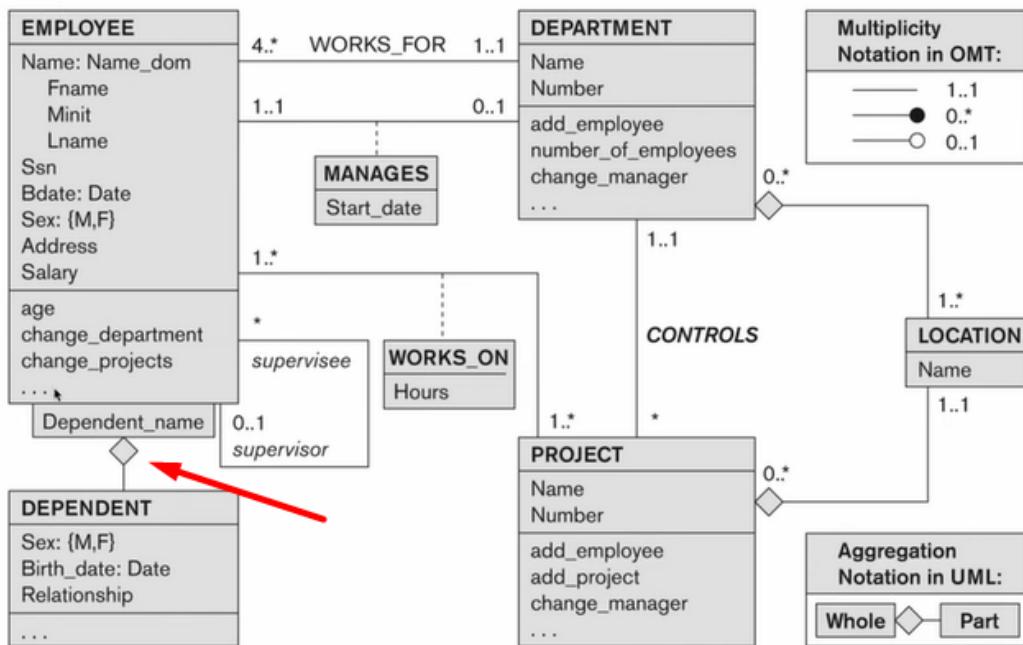
# Diagrama de classe UML para representação ER



## Entidades Fracas em UML

- ▶ Entidades fracas podem ser representadas utilizando a notação de **associação qualificada** (ou **agregação qualificada**);
- ▶ Isto representa tanto a **relação de identificação** quanto a **chave parcial** (que é colocada junto à “classe proprietária”);
- ▶ A chave parcial (no exemplo, `Dependent_name`) é chamada de **discriminadora**

# Diagrama de classe UML para representação ER



Símbolo de agregação para representar a entidade fraca. Chamado de chave discriminadora.

## ER em UML - Atributos como entidade

- ▶ Em algumas situações é conveniente representar como entidade:
  - ▶ Atributos multivalorados;
  - ▶ Atributos que podem ter muitos valores diferentes, mas que casualmente se repetem
    - ▶ Exemplo: nome de uma cidade;
  - ▶ Isto ocorre especialmente quando estes valores podem sofrer ajustes frequentes (sem modificar explicitamente nenhuma instância de entidade associada);
    - ▶ Uma nova cidade que surge;
    - ▶ Correção no nome de uma cidade que foi especificado incorretamente;

Cidade pode ter uma atributo cidade em Pessoa. Mas também pode ter um atributo multi-valorado Cidade em Departamento (Onde departamento atua...)  
Pode tratar cidade como sendo uma entidade e fazer relacionamento dessa entidade com outras coisas.

## Representação da Cardinalidade em UML

---

- ▶ Valores de mínimo e máximo fica próximo da entidade (diferente da notação clássica)
- ▶ Exemplos na figura:
  - ▶ Um funcionário (EMPLOYEE) trabalha para (no mínimo e no máximo) 1 departamento
    - ▶ todo funcionário trabalha em algum departamento; e
    - ▶ um funcionário trabalha em apenas um departamento;
    - ▶ 1..1 está próximo à departamento
  - ▶ Em um departamento, trabalham no mínimo 4 funcionários
    - ▶ Não há limite máximo de funcionário em um departamento
    - ▶ 4..\* está próximo de funcionário (EMPLOYEE)

UML → Mínimo e Máximo fica próximo à Entidade.

---

## Relacionamento de Generalização

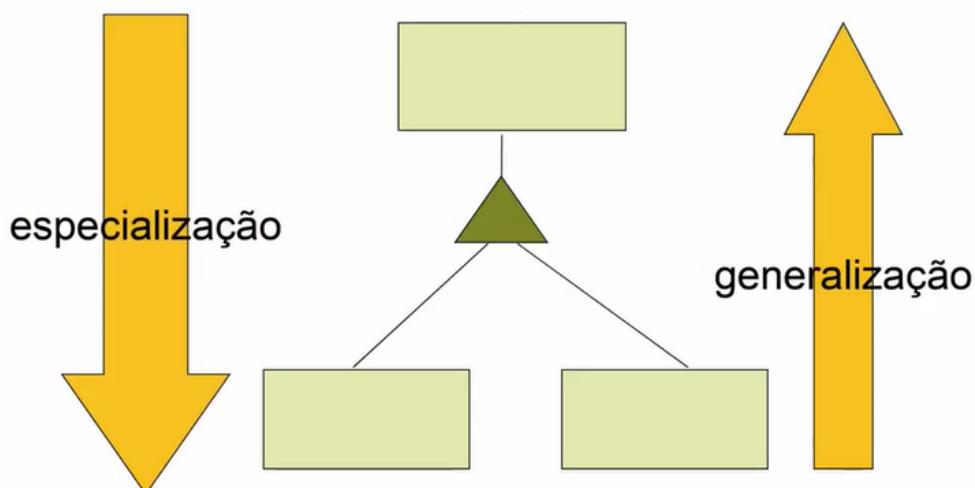
- ▶ Relacionamento entre um elemento **mais geral** e outro mais específico
- ▶ O elemento mais geral é denominado entidade de nível superior (**superclasse**) e o mais específico de entidade de nível inferior (**subclasse**)
- ▶ Relacionamento de **herança**:
  - ▶ O elemento mais geral tem todas as características (atributos) que são comuns aos elementos
  - ▶ As características do nível superior são herdadas no nível inferior

Entidades mais gerais → Nível Superior

Entidades mais específicas → Nível Inferior

Entidades Especializadas herdam carac. dos níveis superiores.

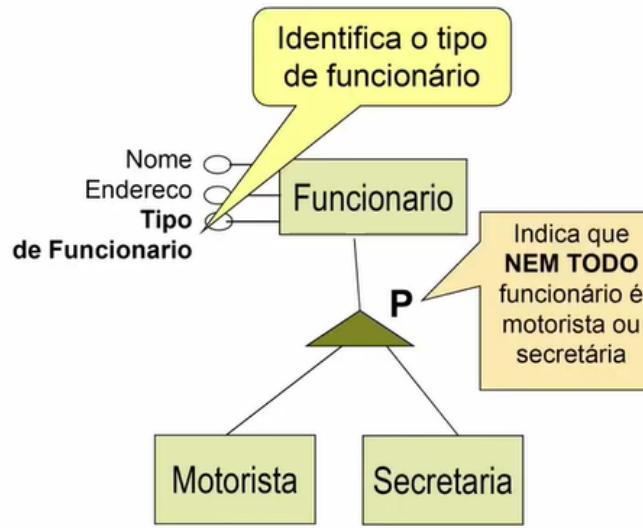
## Relacionamento de Generalização



Não há limites nos níveis da hierarquia

## Generalização Parcial

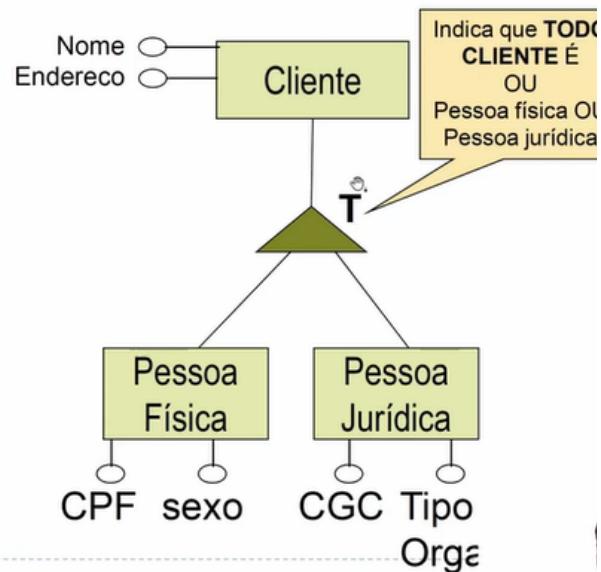
- ▶ Nem toda ocorrência da entidade genérica possui uma ocorrência correspondente em uma entidade especializada
- ▶ A ocorrência pode estar somente na classe genérica



Generalização parcial → Nem motorista nem secretário, simplesmente Funcionário.

## Generalização Total

- ▶ Para cada ocorrência da entidade genérica existe **sempre** uma entidade especializada
- ▶ A ocorrência está sempre na entidade especializada



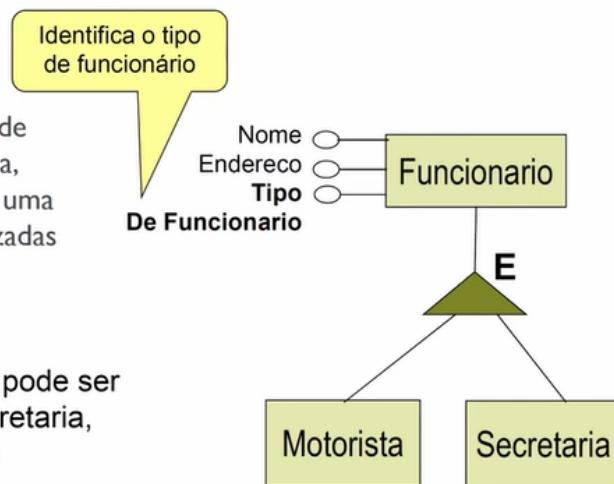
Generalização Total → Não podemos ter apenas Cliente cadastrado, tem que dizer se é pessoa Física ou jurídica.

## Generalização Compartilhada/Exclusiva

### ► Exclusiva

- ▶ A ocorrência da entidade especializada é exclusiva, aparecendo em apenas uma das entidades especializadas

O Funcionário somente pode ser  
OU Motorista OU Secretaria,  
**jamais ambos**

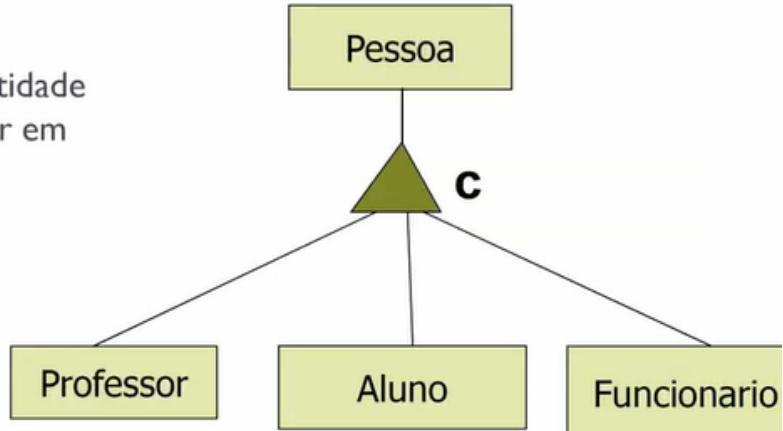


**Exclusiva** → Não pode ser um motorista e secretário ao mesmo tempo.

## Generalização Compartilhada/Exclusiva

### ► Compartilhada

- ▶ Uma ocorrência da entidade genérica pode aparecer em múltiplas entidades especializadas



A pessoa em uma universidade pode ser  
um professor (na graduação), ser um  
funcionário e ser um aluno (de doutorado)

**Compartilhada** → Uma pessoa pode ser professor e aluno ao mesmo tempo.

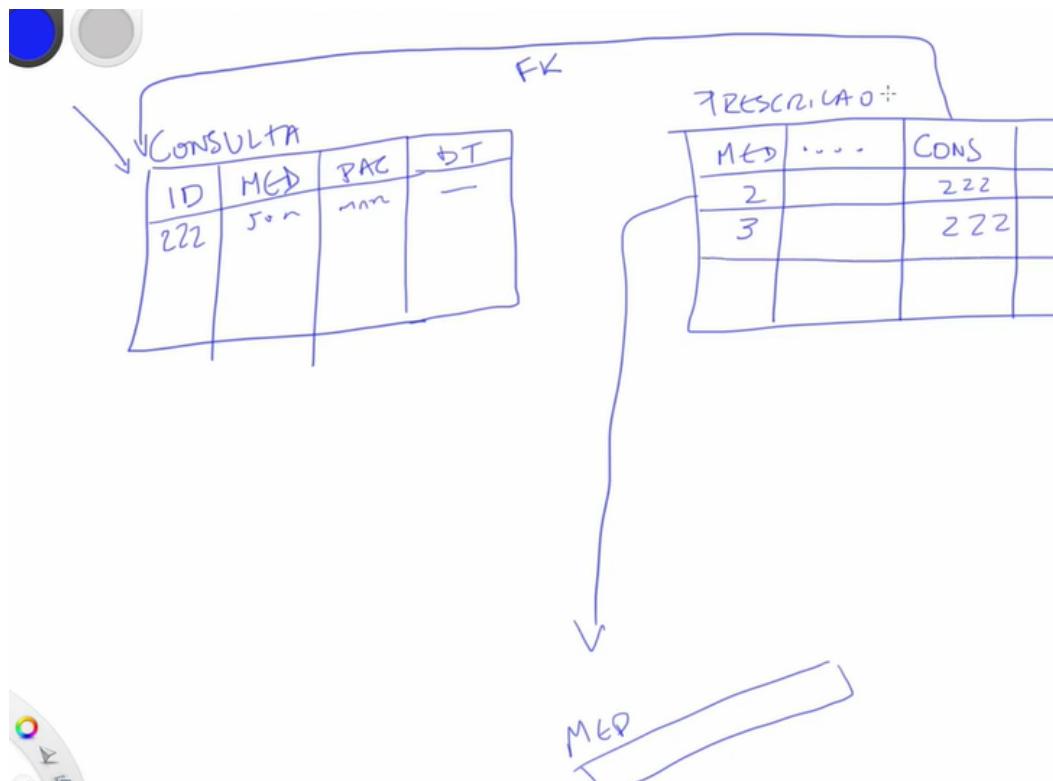
# Entidade Associativa

## ► Possíveis relações de herança:

- ▶ Total – Exclusiva
- ▶ Total – Compartilhada
- ▶ Parcial – Exclusiva
- ▶ Parcial – Compartilhada

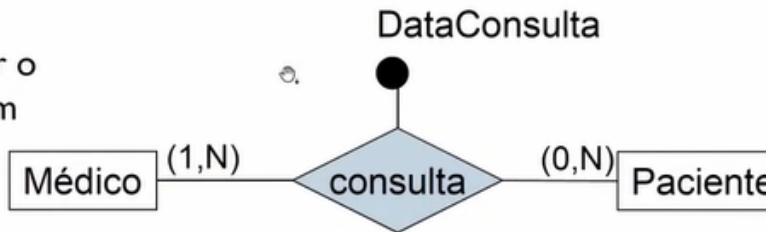
## ► Outras notações:

- ▶ Exclusiva (e/E) – Disjunta (d/D)
- ▶ Compartilhada (c/C) – Sobreposta (s/S) - Overlapping (o/O)



# Entidade Associativa

- ▶ Se for necessário registrar o medicamento prescrito em cada consulta??

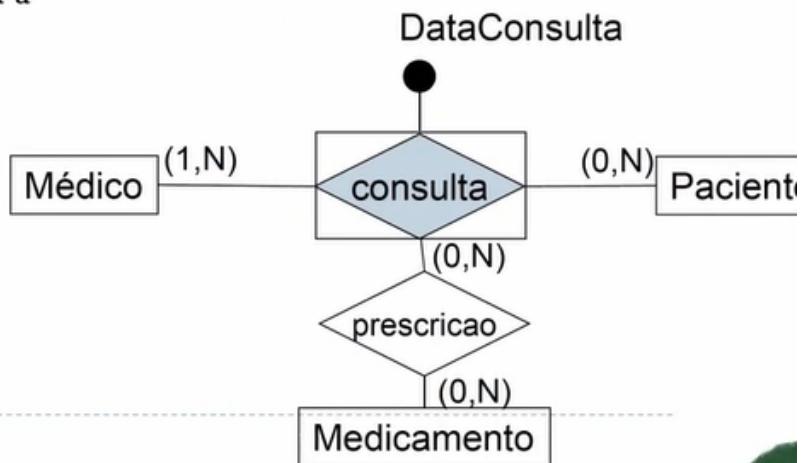


- ▶ O relacionamento passa a ser tratado como uma entidade

- ▶ Símbolo



- ▶ Pode ser relacionada a outras entidades

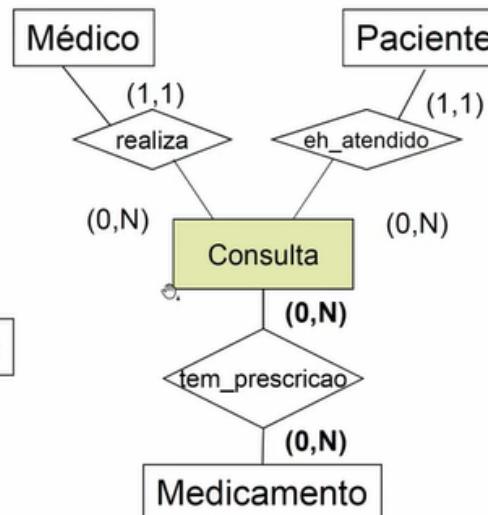
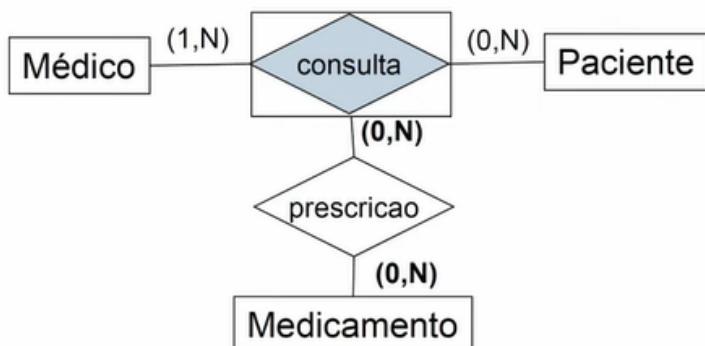


► 12

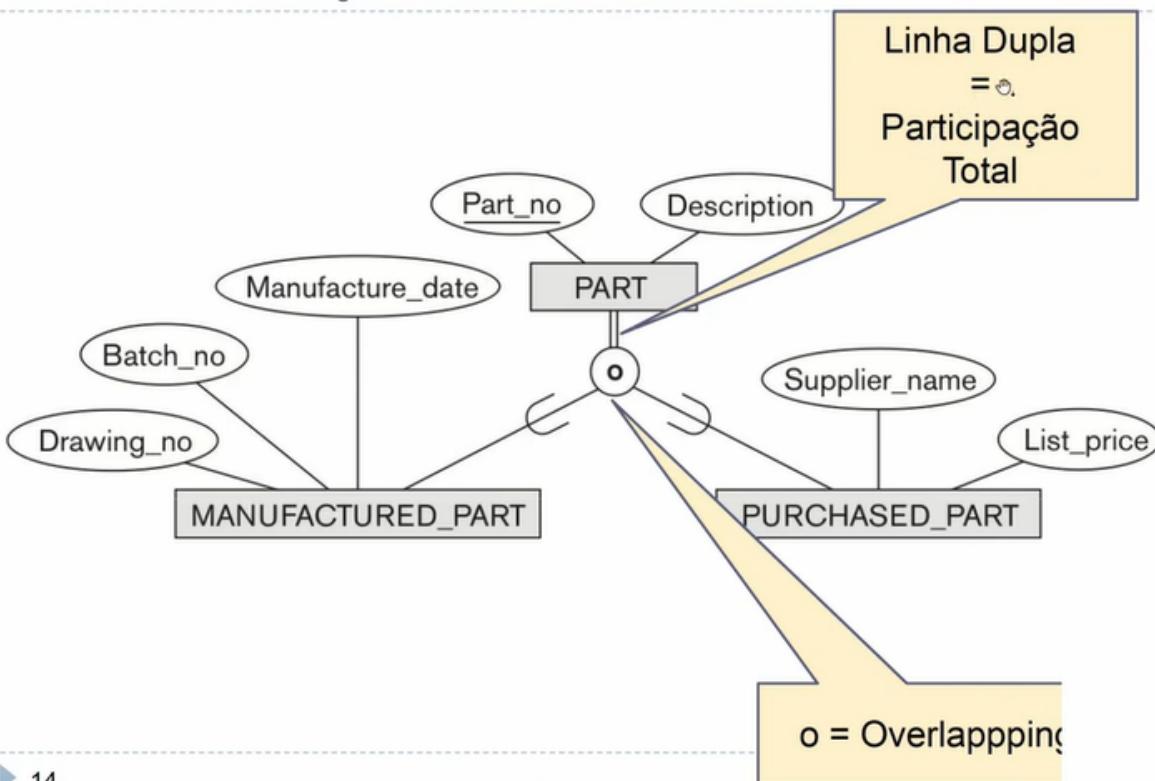
**Entidade Associativa** → Conceitualmente/essencialmente quando fez a modelagem conceitual pensou nela como um relacionamento, mas por uma necessidade ficou mais conveniente representar como uma entidade.

# Entidade Associativa

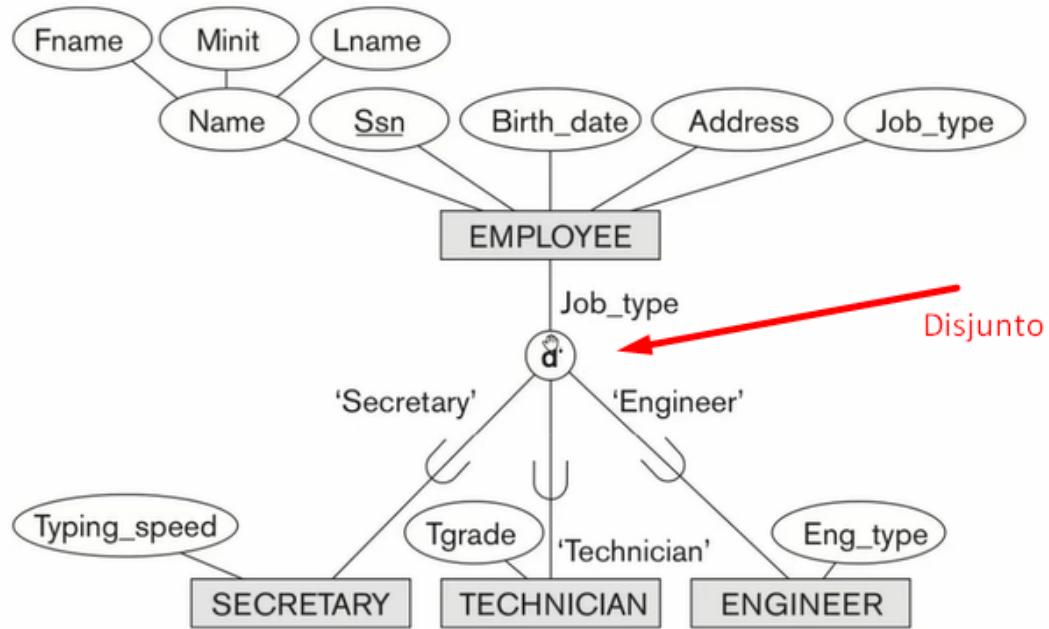
- Uma entidade associativa faz o papel de 2 relacionamentos (substituição do relacionamento consulta por entidade)



## Outra Notação



# Outra Notação

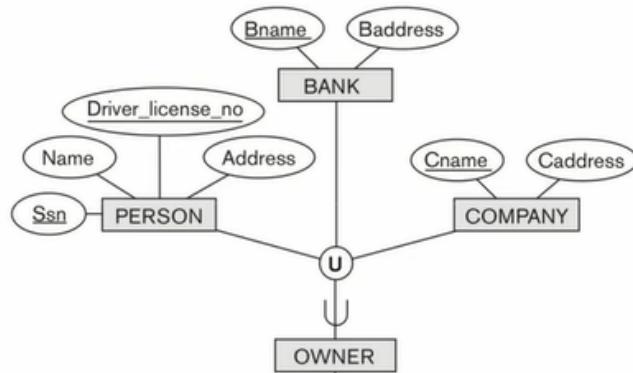


Disjunta ⇔ Exclusiva

C → Vira para a entidade mais geral.

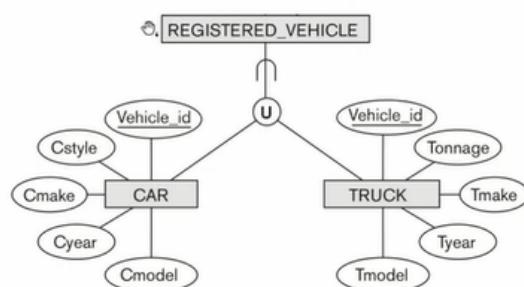
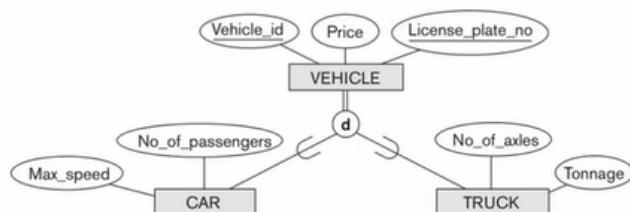
# Tipos de União (Categorias)

- ▶ Relacionamento entre várias superclasses e uma subclasse, onde as superclasses são **entidades diferentes** (com **chaves diferentes**).



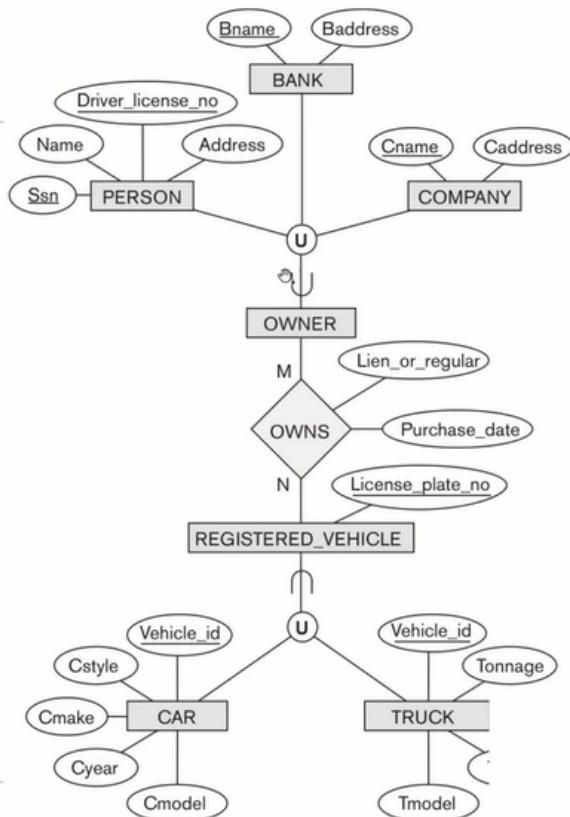
Grupos de entidades que pode ser “dona” de coisas...  
Agrupar para facilitar na hora de fazer os relacionamentos.

## Categorias VS Herança



## Categorias diferentes relacionadas

► 18



## Categorias - Propriedades

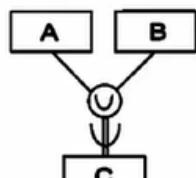
- ▶ Uma entidade **não** pode existir no BD pertencendo **apenas a uma categoria**; tem que pertencer também a uma (e somente uma) das superclasses.
- ▶ Uma entidade de uma categoria **herda apenas** os atributos da superclasse a que pertence.
- ▶ As **categorias podem ter atributos específicos** (loais à categoria) e **podem participar em relacionamentos** com outras entidades e categorias.

Pertencer a só e somente uma superclasse.

## Categorias - Propriedades

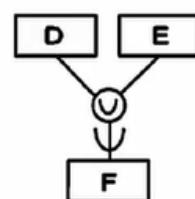
As instâncias da subclasse representam um **subconjunto da união** das instâncias das superclasses.

Categorização **total** (todas as instâncias das entidades das superclasses pertencem à categoria).



$$C = A \cup B$$

Categorização **parcial** (apenas um subconjunto das instâncias das entidades das superclasses pertencem à categoria).



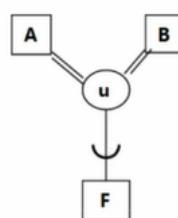
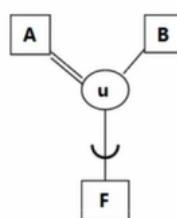
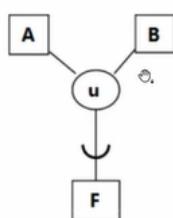
$$F \subseteq (D \cup E)$$

Total → Todas as Pessoas/Bancos/Empresas são Owner e podem se relacionar com coisas que Owner se relaciona, por exemplo como o veiculo registrado.

Parcial → Nem todas as instâncias daquelas entidades de pessoas/bancos.. Fazem parte da categoria, pode ter alguma delas que não se relaciona.

## Categorias - Propriedades

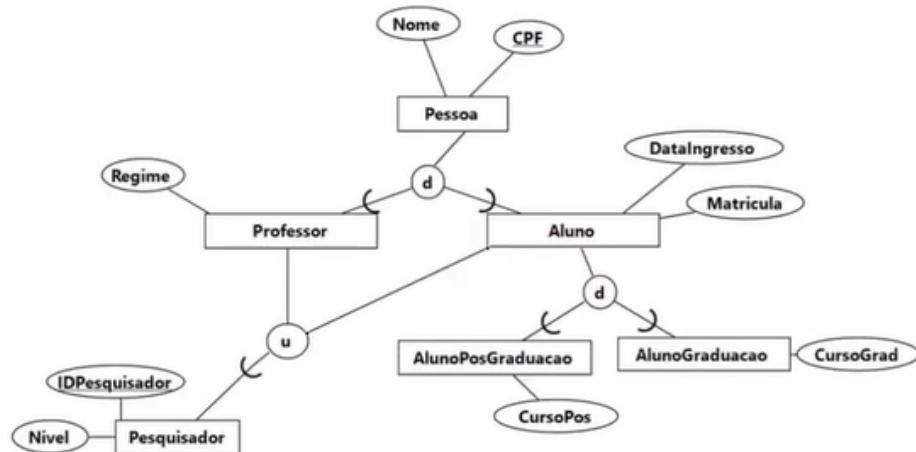
Outra forma de representar Categorização **Total** e **Parcial**



Total → linhas duplas.

## Categorias - Propriedades

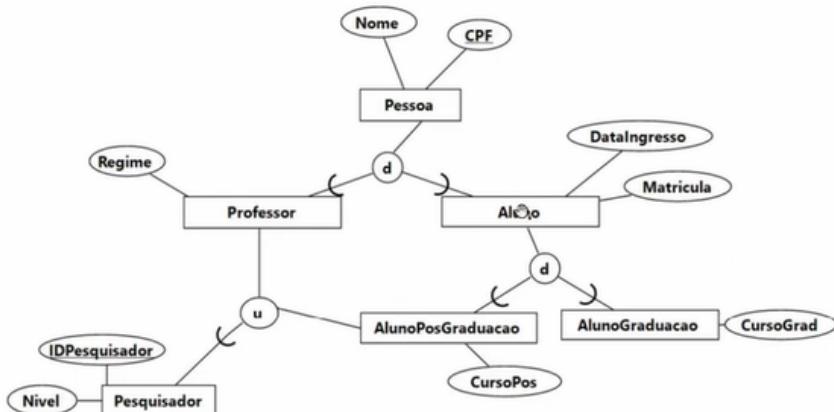
Uma superclasse (de categoria) pode conter subclasses (por herança) e/ou ser uma subclasse (por herança).



No exemplo: **Professores** e **Alunos** (quaisquer alunos) podem atuar como pesquisadores. Essa categorização pode ainda ser definida como **Parcial** ou **Total**.

## Categorias - Propriedades

Uma superclasse (de categoria) pode conter subclasses (por herança) e/ou ser uma subclasse (por herança).



No exemplo: **Professores** e **Alunos de Pós-Graduação** (e somente esses alunos) podem atuar como pesquisadores. Essa categorização pode ainda ser definida como **Parcial** ou **Total**.