



Universidade Federal  
de Campina Grande



# Banco de Dados I

## Unidade 2: Modelos e Modelo Conceitual

### Entidades e Relacionamentos - MER

Prof. Cláudio de Souza Baptista, Ph.D.  
Laboratório de Sistemas de Informação – LSI  
UFCG

# Modelos de Dados

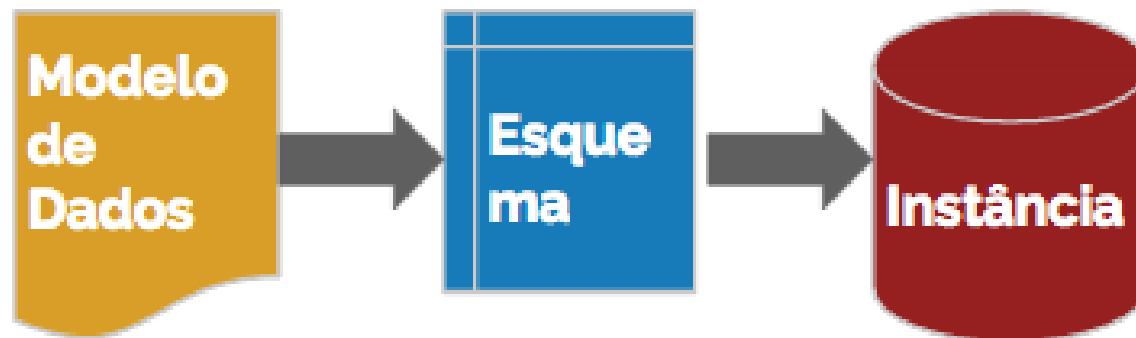
## ■ Modelo de Dados:

- Conjunto de conceitos para descrever a **estrutura** de um BD, as **operações** para manipular estas estruturas e as **restrições** que o BD deve obedecer.

## ■ Estrutura e Restrições de um Modelo de Dados:

- Construtores são usados para definir a estrutura do BD
- Construtores incluem elementos (com seus tipos de dados) e grupos de elementos (e.g. entidades, registro, tabela) e relacionamentos entre tais grupos
- Restrições (Constraints) aplicam-se a dados válidos, podendo ser validada todo tempo.

# Modelo, Esquema, Instância



Modelo de dados: Conjunto de conceitos descrevem a estrutura de um BD

Abstração de dados

Estrutura = tipos de dados + relacionamentos + restrições (+operações recuperação e atualização)

Esquema: Descrição (textual ou gráfica) da estrutura de um BD de acordo com um determinado modelo de dados

Instância: Conjunto de dados armazenados em um BD em um determinado instante de tempo

# Categorias de Modelos de Dados

- **Conceitual (alto-nível, semântico)**

- ☐ Provê conceitos que são próximos à maneira como os usuários percebem os dados, independente de aspectos de implementação. Ex. Entidades\_Relacionamentos (ER)

- **Físico (baixo-nível, interno)**

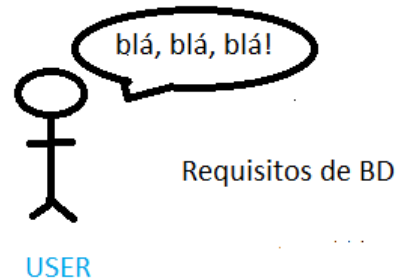
- ☐ Descrevem os detalhes de como os dados são armazenados no computador, estratégias de índices, paths, arquivos etc. Ex. Modelo Físico do Oracle

- **Lógico (representacional)**

- ☐ Provê conceitos que estão entre os dois modelos acima. Descrevem a estrutura do BD da forma como será manipulado pelo SGBD (e.g. modelo relacional)

# Modelagem de BD: Níveis de Abstração

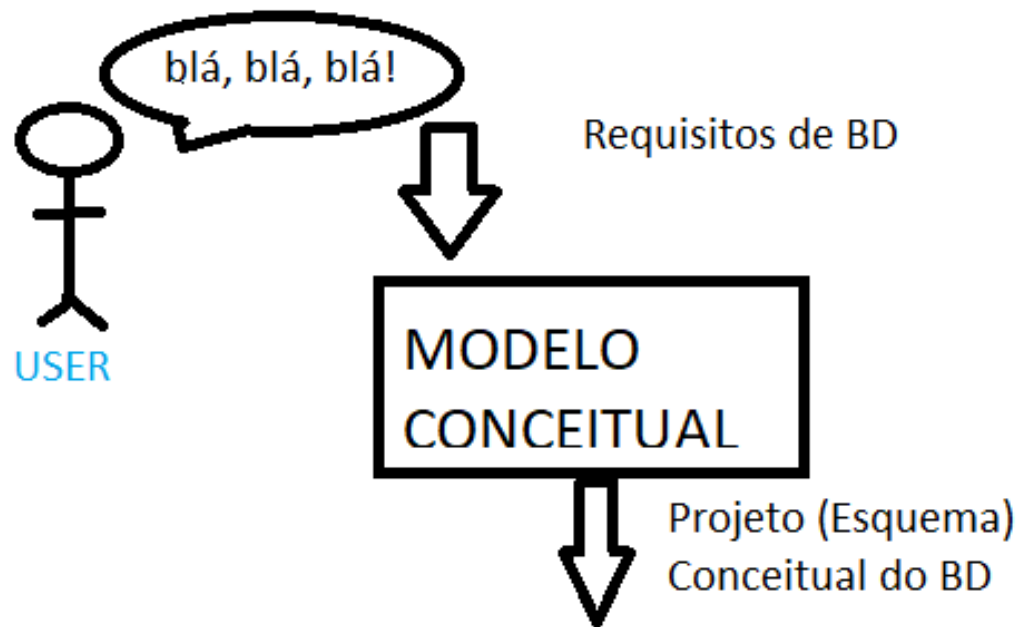
**Primeira Etapa:** Levantamento de Requisitos de Banco de Dados



# Modelagem de BD: Níveis de Abstração



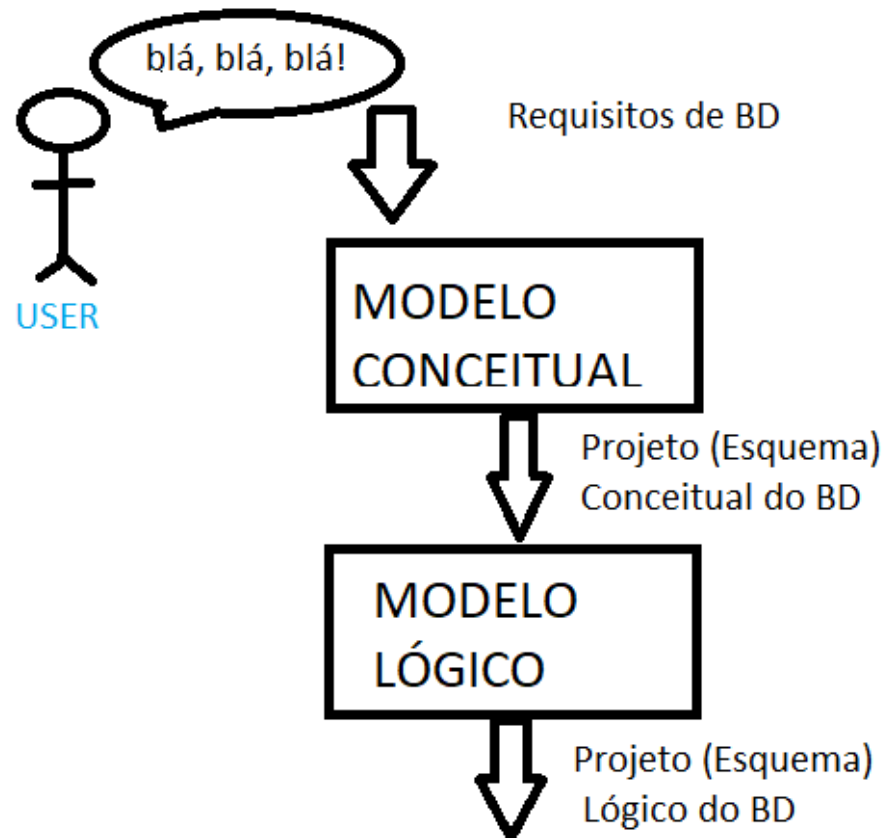
## Segunda Etapa: Modelagem Conceitual do BD



# Modelagem de BD: Níveis de Abstração

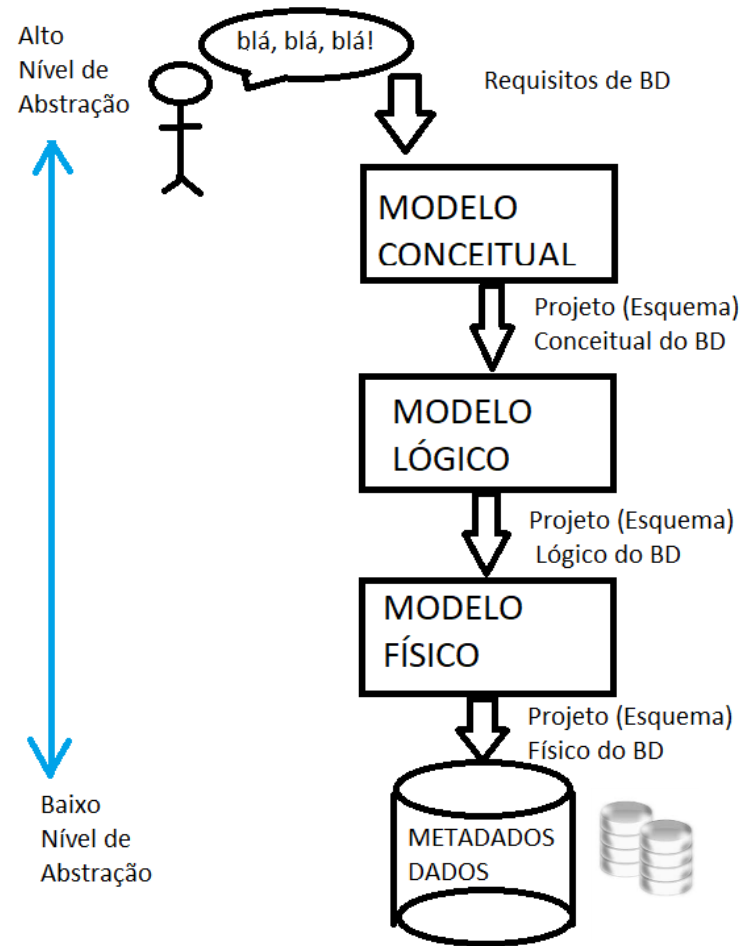


## Terceira Etapa: Modelagem Lógica do BD



# Modelagem de BD: Níveis de Abstração

## Quarta Etapa: Modelagem Física do BD





# Esquemas

- **Esquemas** são a descrição de um banco de dados específico em diversos níveis: conceitual, lógico e físico.
- São projetos de BD utilizando-se de modelos de dados.
- Ex. Esquemas conceitual ER e lógico relacional do banco de dados de uma empresa XYZ.

# Ex. Esquema conceitual - ER



# Ex. Esquema Lógico - Relacional

## STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

## COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

## PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

## SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

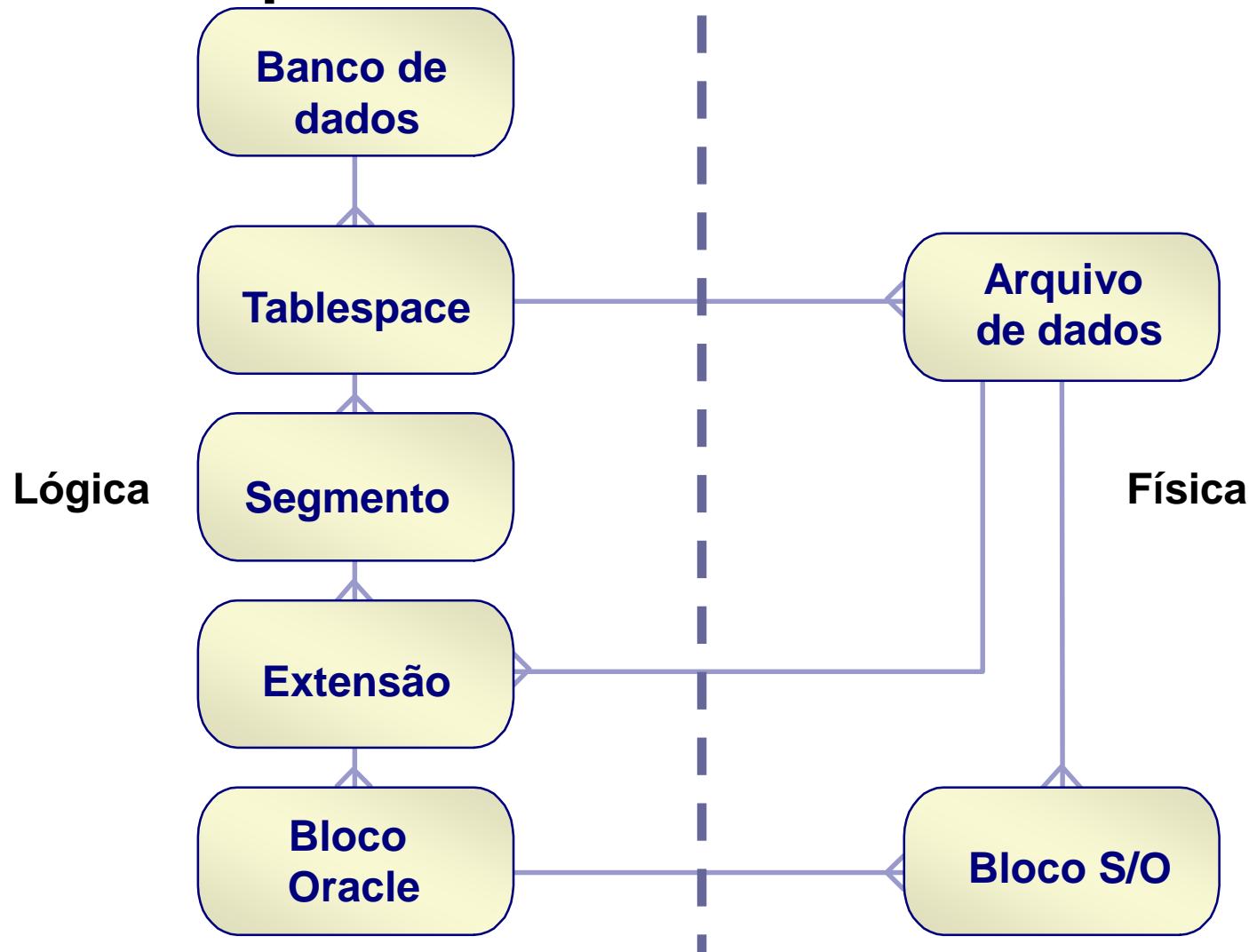
## GRADE\_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

**Figure 2.1**

Schema diagram for the database in Figure 1.2.

# Ex. Esquema Físico – Oracle



# Ex. Esquema Físico – Oracle

- Tablespace: system, sysaux, temp, data, undo, users
- Arquivos de Dados: dat1, dat 2
- Arquivos de log
- Arquivos de controle

# Instância de um BD

- Também chamado de **Estado** de um BD
- Conjunto de dados armazenados num dado instante de tempo (shapshot).
- É a coleção de todos os dados de um BD num determinado instante de tempo.

# Ex. Estado de um BD

## COURSE

Course_name	Course_number	Credit_Hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

## SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

## GRADE REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

## PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

**Figure 1.2**  
A database that stores student and course information.

# Esquema versus Estado

## ■ Distinção

- O esquema do BD muda muito infrequentemente
- O estado do BD muda toda vez que o BD é atualizado

■ Esquema é chamado de **intenção**

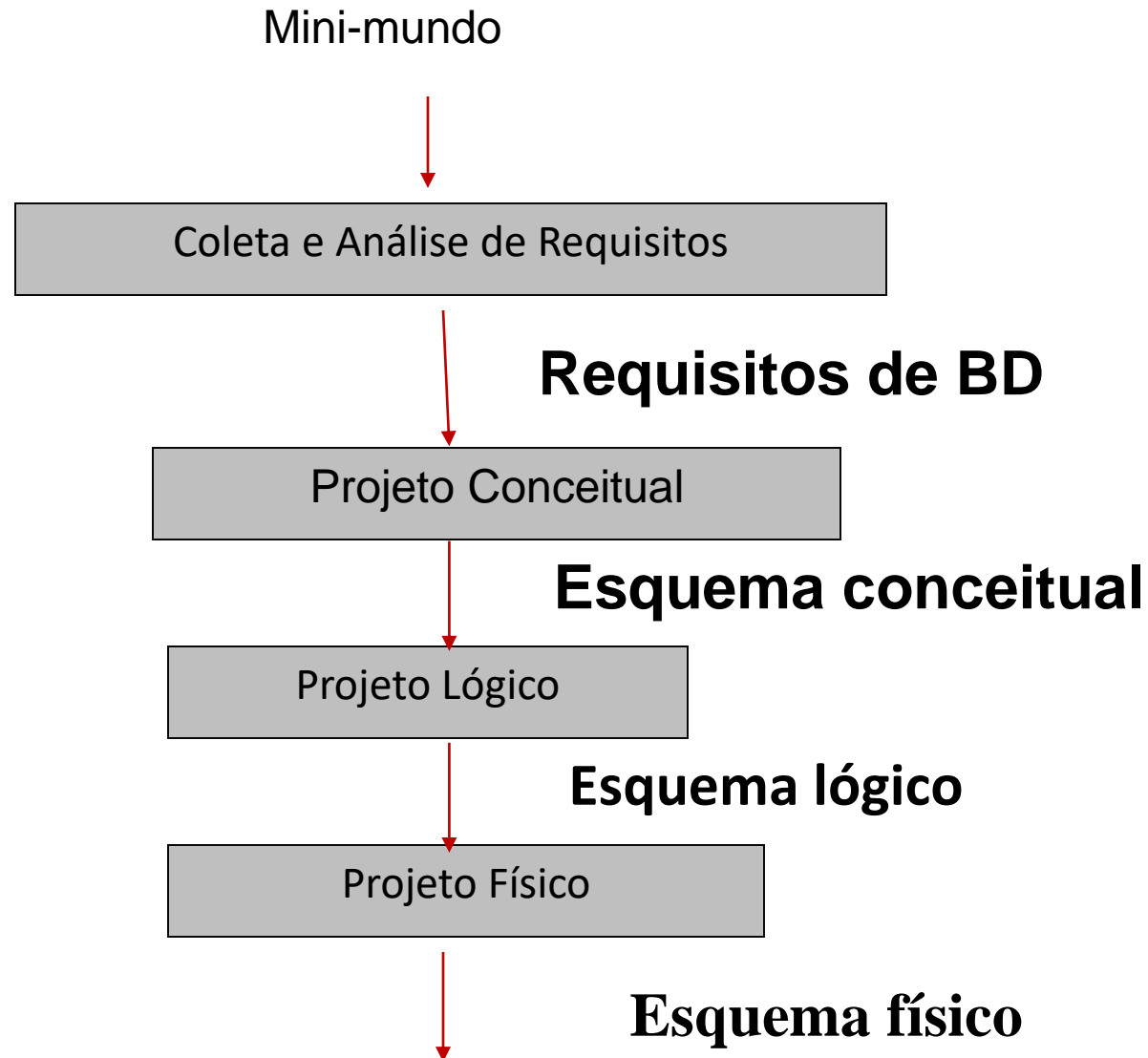
■ Estado é chamado de **extensão**.



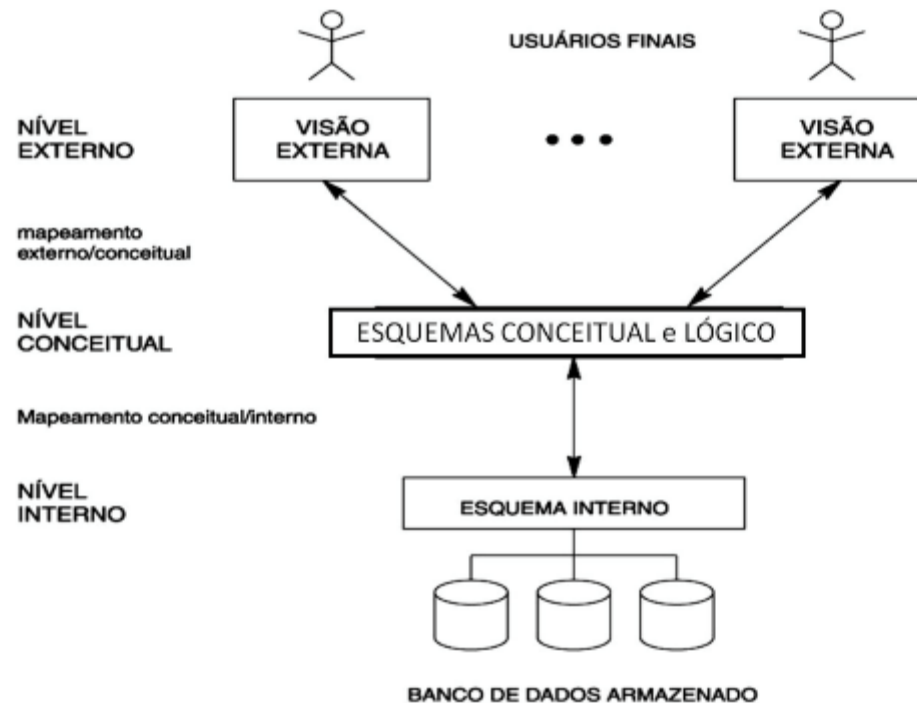
# Projeto de Banco de Dados

- Porquê?
  - ☐ Decidir sobre a estrutura do BD antes de partir para uma implementação
- Considerar questões como:
  - ☐ Quais entidades modelar?
  - ☐ Como as entidades serão modeladas?
  - ☐ Quais restrições existem no domínio?
  - ☐ Como obter um BOM DESIGN?
- Vários formalismos existem (usaremos ER)

# Fases de um projeto de BD



# Arquitetura ANSI/Sparq



*Arquitetura de um Sistema de Bancos de Dados*

# Independência de Dados

- Quando um esquema num nível mais baixo é mudado, apenas os mapeamentos entre este esquema e os esquemas em níveis mais altos precisam ser mudados
- As mudanças um nível não alteram os níveis superiores

# Independência de Dados

## ■ Independência Lógica

- ☐ Capacidade de mudar o esquema lógico sem mudar o esquema conceitual
- ☐ Capacidade de mudar o esquema conceitual sem mudar o nível externo, referente às visões dos usuários

## ■ Independência Física

- ☐ Capacidade de mudar o esquema físico sem mudar o esquema lógico / conceitual

# Processo de Design de um BD

1. Requirements Analysis

2. Conceptual Design

3. Logical, Physical,  
Security, etc.

## 1. Requirement Analysis (Análise de Requisitos)

- ☐ O que vai ser armazenado?
- ☐ Como serão usados os dados?
- ☐ O que vamos fazer com os dados?
- ☐ Quem deveria ter acesso aos dados?

# Processo de Design de um BD



## 2. Conceptual Design (Modelagem Conceitual)

- ☐ Uma descrição em alto nível do BD
- ☐ Suficientemente preciso para pessoas técnicas poderem entender o modelo
- ☐ Porém, não tão preciso para detalhes técnicos, independente de implementação

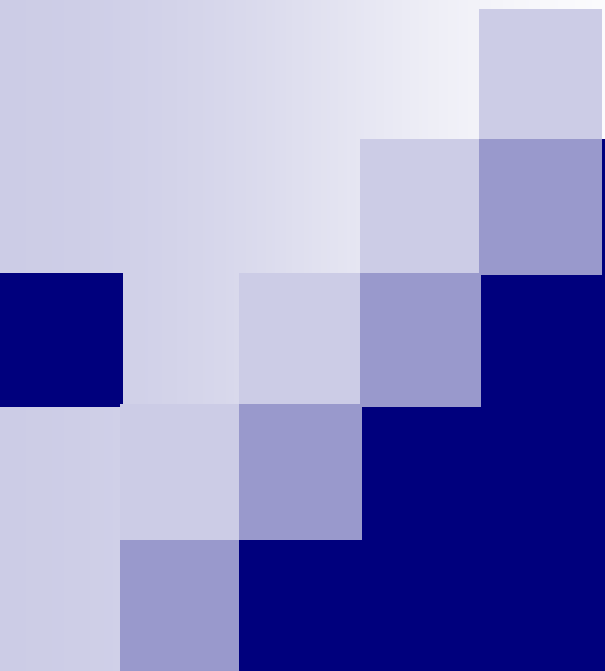
# Processo de Design de um BD



## 3. Mais Modelos com considerações sobre implementação:

- ☐ Logical Database Design
- ☐ Physical Database Design
- ☐ Security Design



A decorative graphic on the left side of the slide, consisting of a grid of squares in various shades of blue and purple, arranged in a stepped pattern.

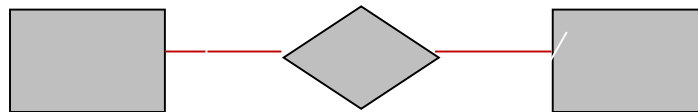
# Modelo Conceitual Entidades e Relacionamentos - MER

# Motivação

- **Modelagem semântica permite aproximar o modelo obtido do mundo real**
- **Exemplo de Modelos Conceituais:**
  - ✓ **MER - Modelo de Entidades e Relacionamentos**
  - ✓ **UML (linguagem de modelagem universal)**

# Modelo de Entidades e Relacionamentos

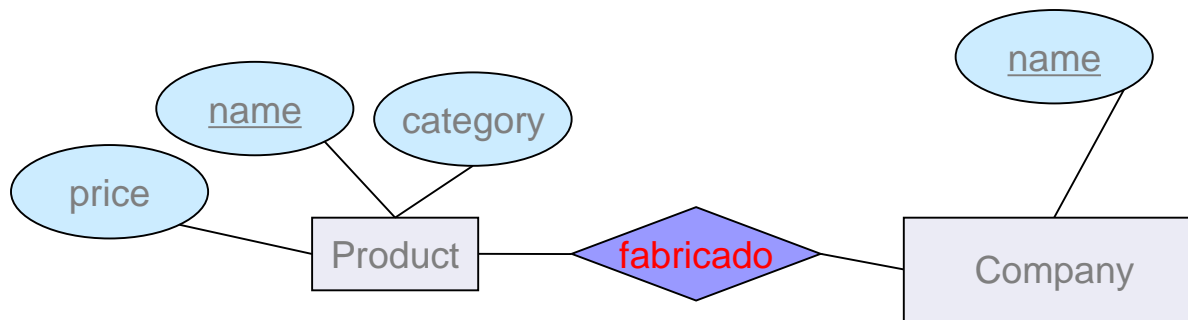
- Representação semântica das estruturas de dados mantidas num banco de dados
- Foi proposto por Peter Chen em 1976, um dos artigos mais citados da história da Ciência da Computação
- Usados por empresas grandes e pequenas, devido a sua simplicidade e expressividade.
- Possui várias notações:
  - Relacionamentos como objetos do Modelo (Chen)



- Relacionamentos apenas como simples ligações (Codd, Martin)



# Modelo de Entidades e Relacionamentos



E/R é uma sintaxe visual (diagrama) para um DB design com alto nível de abstração, não implementável, que serve para formalizar os requisitos que vêm em linguagem natural

# Entidades

Uma entidade é tudo aquilo sobre o qual se deseja manter informações.

- Também chamado de Entity Set
- São as classes ou tipos de objetos no modelo
- Podendo representar:
  - objetos concretos: pessoas, livros, carros, ...
  - conceitos abstratos: empresas, eventos, embarques, ...

# Entidades

Ex.:

- Conjunto de todas as contas correntes de um banco
  - Conjunto de todos os empregados de uma empresa
  - Conjunto de todos os filmes de um produtor
- 
- Representação de entidades no diagrama E-R (entidades e relacionamentos):

Empregado

Aluno

Empréstimo

# Entidades

- Entidades devem ser descritas num Dicionário de Dados

**Entidade: EMPREGADO**

**Descrição: Pessoa que mantém vínculo empregatício com a Empresa através de um contrato de trabalho de acordo com a legislação trabalhista**

# Entidades

- Entidades devem ser descritas num Dicionário de Dados

Entidade: ENCOMENDA

Descrição: Instrumento contratual de emissão unilateral pela empresa e aceitação, expressa ou tácita, pelo fornecedor do material.



# Instância

## **Instância (chamado de entidade ou objeto):**

Objeto de uma entidade com suas respectivas propriedades que é distinguível dos outros objetos.

São os objetos individuais que são membros de uma Entidade (Entity Set)

Não são explicitamente representados no MER

**Ex.:** A entidade Empregado poderia ter a seguinte instância: “Maria dos Anjos, 31 anos, Secretária, Solteira, R\$ 800,00”

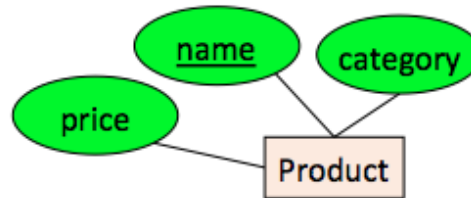
# Atributos

São as propriedades que caracterizam ou descrevem uma entidade ou um relacionamento.

- Ex.: A entidade CARRO poderia ter os seguintes atributos:
  - ❖ Placa, fabricante, modelo, ano de fabricação, cor, preço
  - ❖ O relacionamento TRABALHA entre EMPREGADO e PROJETO pode ter o atributo: horasTrabalhadas.

# Atributos

- São representados dentro de elipses ligadas por um traço à Entidade ou Relacionamento



- Cada atributo possui um domínio que identifica o conjunto de valores permitidos para aquele atributo.
- Ex.:        nome: domínio string(20)  
              salário: domínio numérico

# Atributos

Atributos devem também ser descritos no Dicionário de Dados:

**Entidade: EMPREGADO**

**Atributo: Data de Admissão**

**Descrição: data na qual foi assinado o contrato de trabalho entre a empresa e o empregado**

**Domínio: data posterior a 03/01/78 (data de criação da empresa) e a data de nascimento do empregado**

# Atributos

- ***Simples***: é atômico.

**Ex.** Idade: numérico  
Nome: cadeia de caracteres

# Atributos

- ***Simples***: é atômico.

Ex. Idade: numérico  
Nome: cadeia de caracteres

- ***Composto***: contém sub-atributos que compõem o atributo.

Ex. Endereço(rua, número, bairro, CEP, cidade)

# Atributos

- ***Monovalorados***: têm um único valor para uma instância de uma entidade.

Ex.: PESSOA: Idade

# Atributos

- ***Monovalorados***: têm um único valor para uma instância de uma entidade.

Ex.: PESSOA: Idade

- ***Multivalorados***: possuem vários valores numa instância de uma entidade.

Ex.: PESSOA:TitulaçãoSuperior(nenhum, Bel. MSc., PhD)



# Atributos

- ***Monovalorados***: têm um único valor para uma instância de uma entidade.

Ex.: PESSOA: Idade

- ***Multivalorados***: possuem vários valores numa instância de uma entidade.

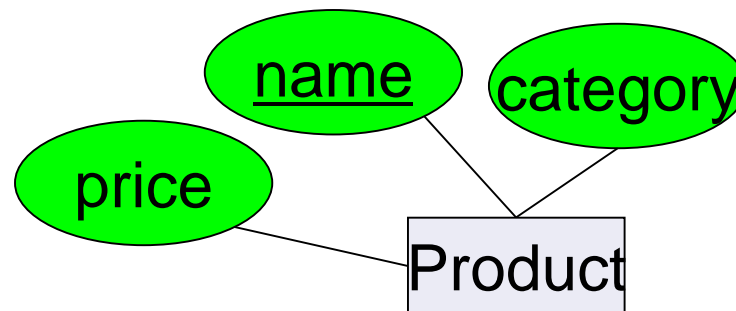
Ex.: PESSOA:TitulaçãoSuperior(nenhum, Bel. MSc., PhD)

- **Atributos derivados**: podem ser determinados a partir de outros atributos/entidades.

Ex.: Idade e dataAniversário

# Chaves

- Uma chave é um conjunto mínimo de atributos que identifica unicamente uma instância de uma entidade
- Costuma-se sublinhar a chave a ser escolhida dentre as chaves candidatas

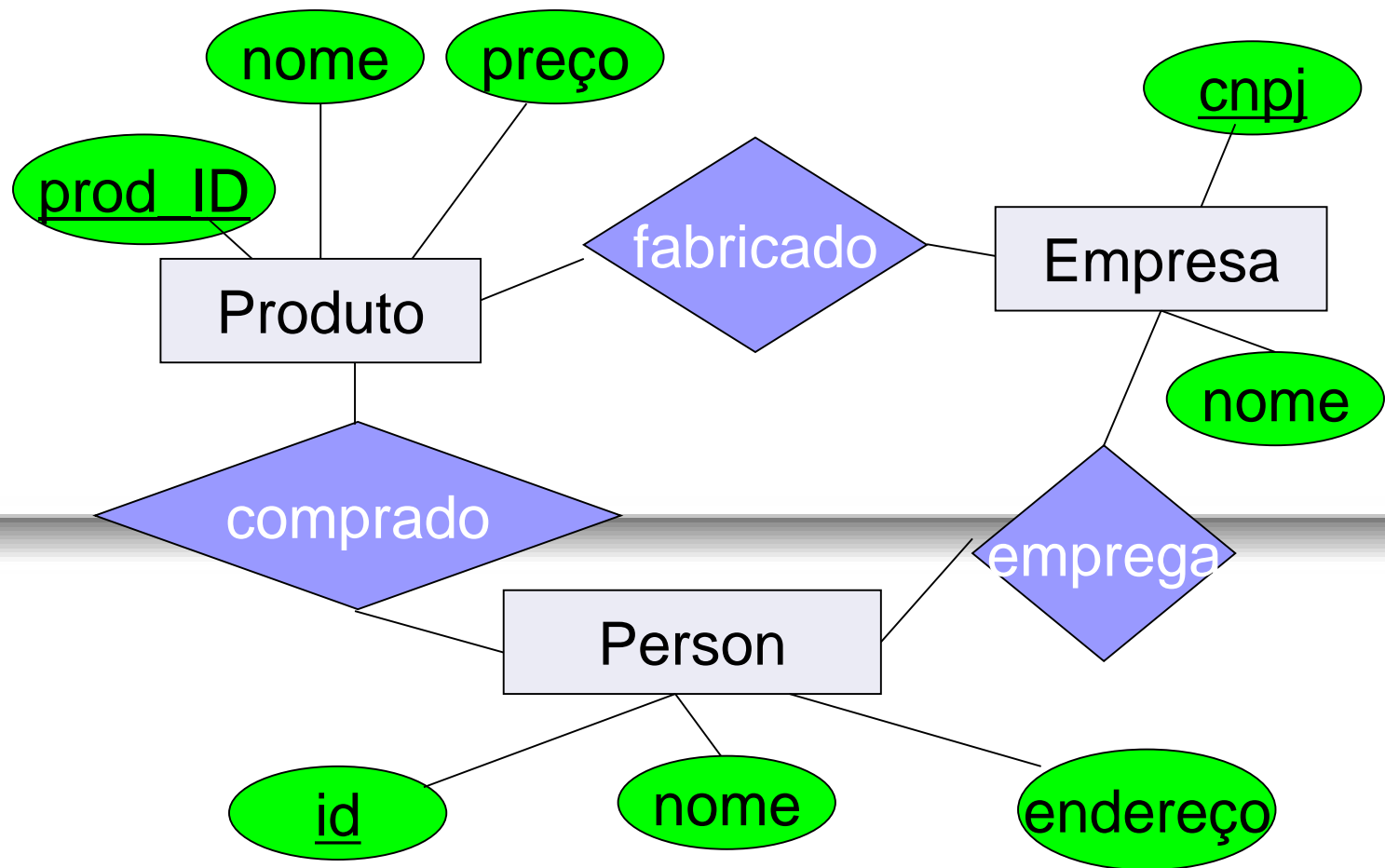


# Relacionamentos

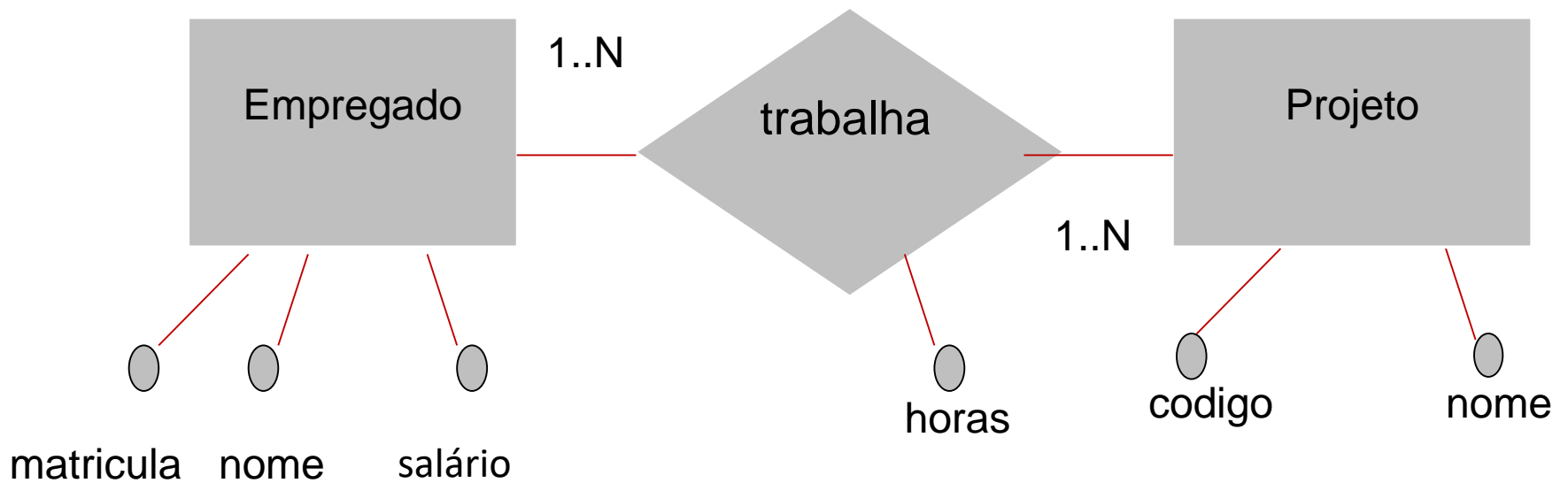
São funções que mapeiam um conjunto de instâncias de uma entidade em um outro conjunto de instâncias de outra entidade (ou da mesma entidade: “auto relacionamento”). Em outras palavras, são associações entre diversas entidades.

**Ex.:** “Um empregado trabalha num projeto”  
“Um cliente possui conta bancária”  
“Um filme possui vários atores”

# Relacionamentos

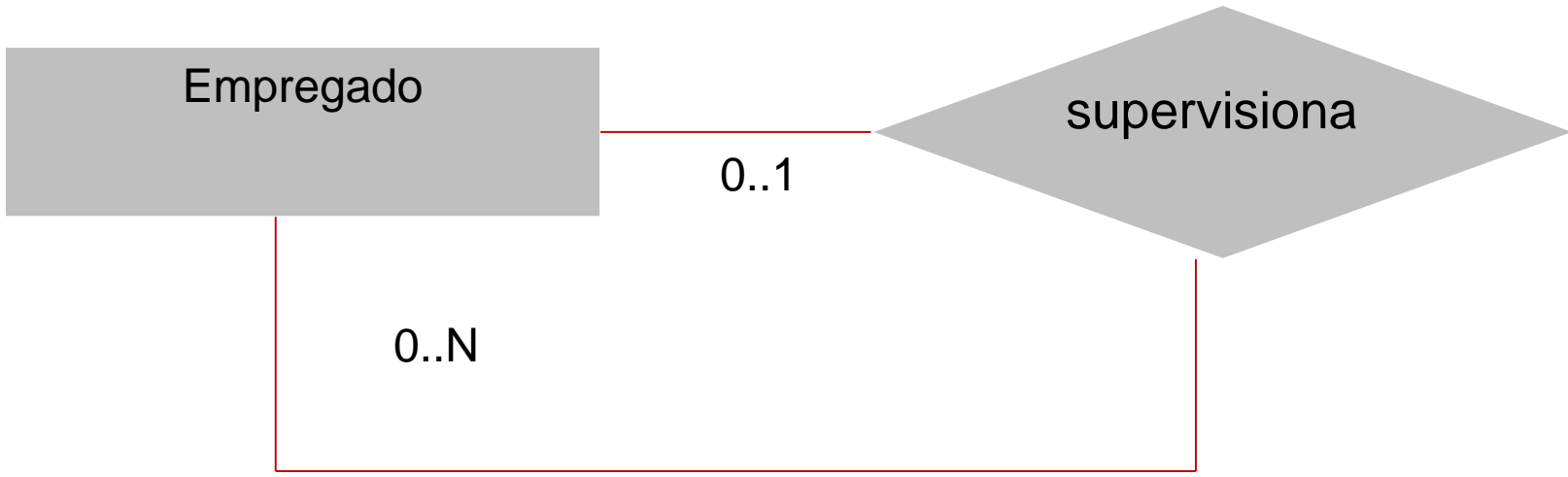


# Relacionamentos





# Auto-Relacionamientos



# Relacionamentos

**OBS.1:** No modelo UML, em que entidades são classes, os relacionamentos são implementados com o conceito de papel. Por exemplo, no diagrama de Empregado Trabalho em Departamento, trabalha é o papel do relacionamento.

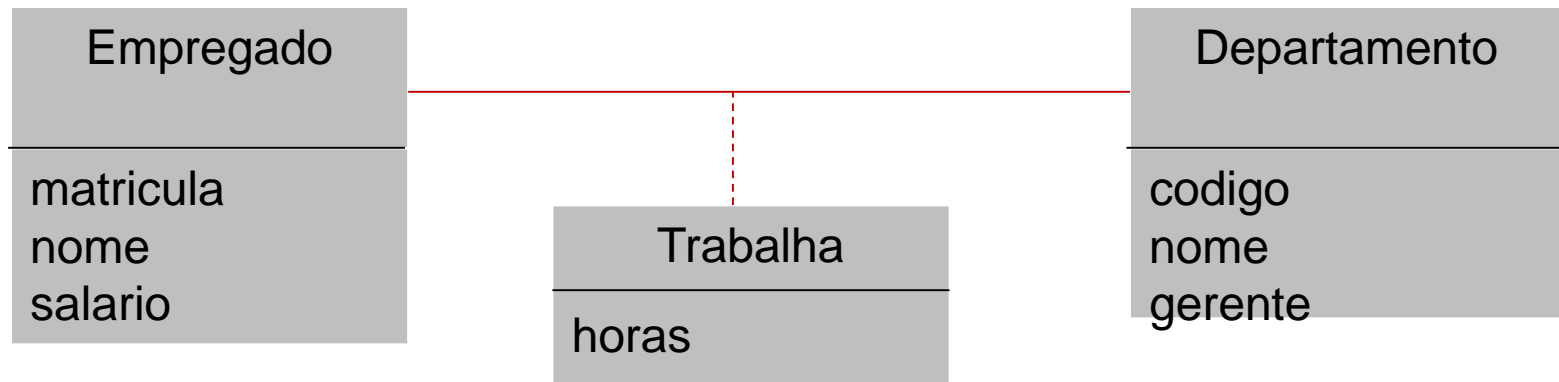
**Ex.**



# Relacionamentos

**OBS.1:** no modelo UML se precisarmos modelar atributos do relacionamento, usamos uma notação especial que cria uma classe do relacionamento.

**Ex.**





# Restrições de Integridade

Caracterizam as restrições nas quais os relacionamentos entre entidades estão submetidos (regras do negócio).

Ex.:

“Todo empregado deve estar lotado num departamento”

“Existe Cliente que não foi recomendado por Cliente”

“Toda Nota Fiscal deve ter pelo menos um item discriminado”

“Toda multa deve estar associada a um carro”

“Existe carro sem multa associada”

# Restrições de Integridade

- Podemos caracterizar um relacionamento em termos de:
  1. **Cardinalidade:** quantidade de instâncias que podem participar do relacionamento
  2. **Totalidade:** obrigatoriedade da ocorrência do relacionamento entre as entidades envolvidas.

# Restrições de Integridade

## ■ Tipos de Cardinalidade

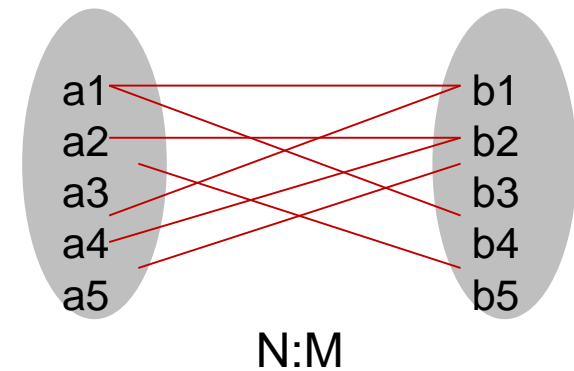
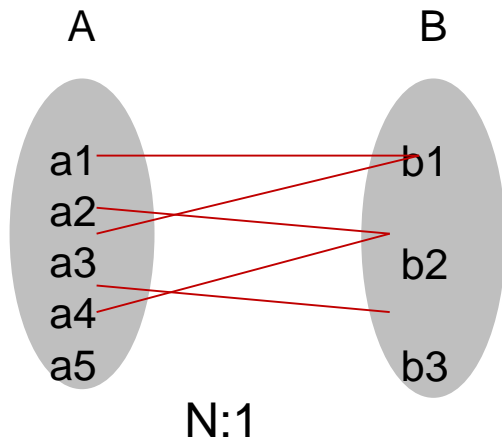
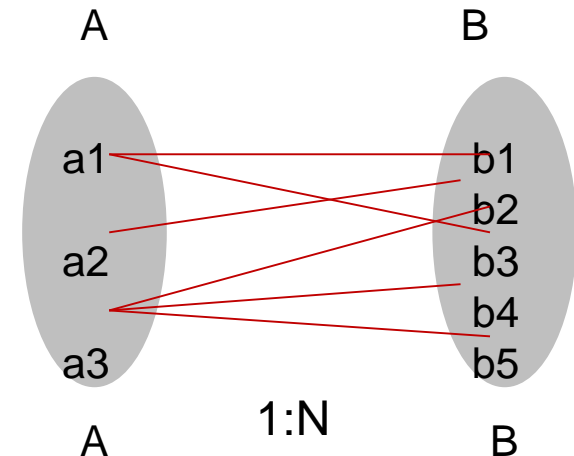
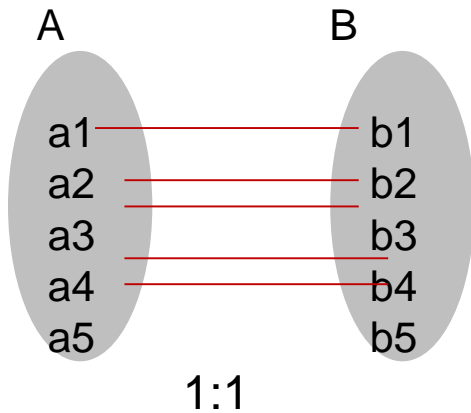
- **Um\_para\_Um (1:1):** uma instância de uma entidade A está associada a no máximo a uma instância de uma entidade B, e vice-versa.
- **Um\_para\_Muitos (1:N):** uma instância de uma entidade A está associada a qualquer número de instâncias da entidade B. Porém, uma instância da entidade B pode estar associada, no máximo, a uma instância da entidade A.

# Restrições de Integridade

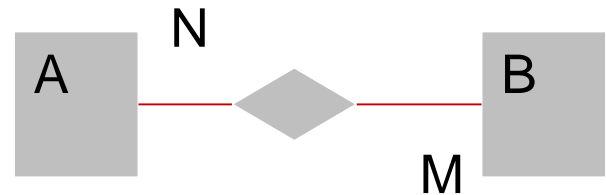
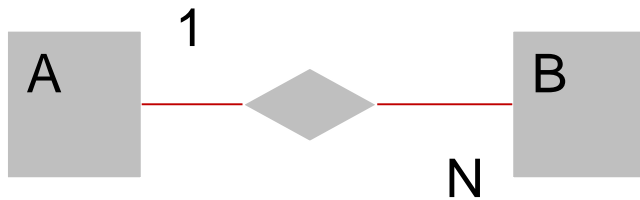
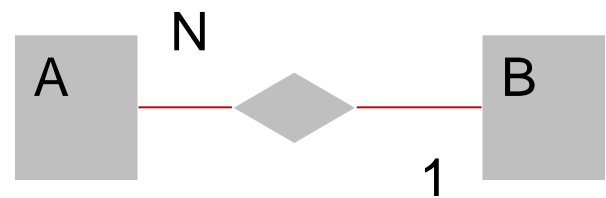
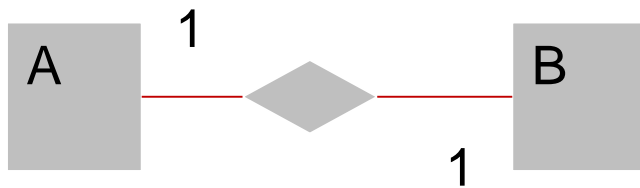
## ■ Tipos de Cardinalidade

- **Muitos\_para\_Um (N:1)**: uma instância da entidade A está associada a uma instância de B. Porém, uma instância de B pode estar associada a qualquer número de instâncias de A.
- **Muitos\_para\_Muitos(M:N)**: uma instância da entidade A está associada a qualquer número de instâncias da entidade B, e vice-versa.

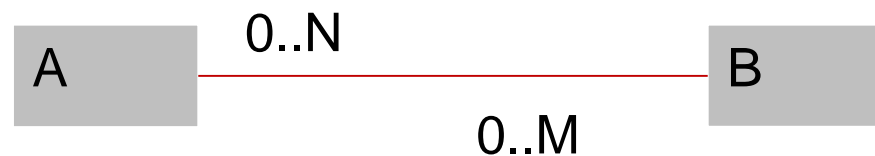
# Restrições de Integridade



# Representação clássica Chen



# Representação UML



# Entidades Fracas

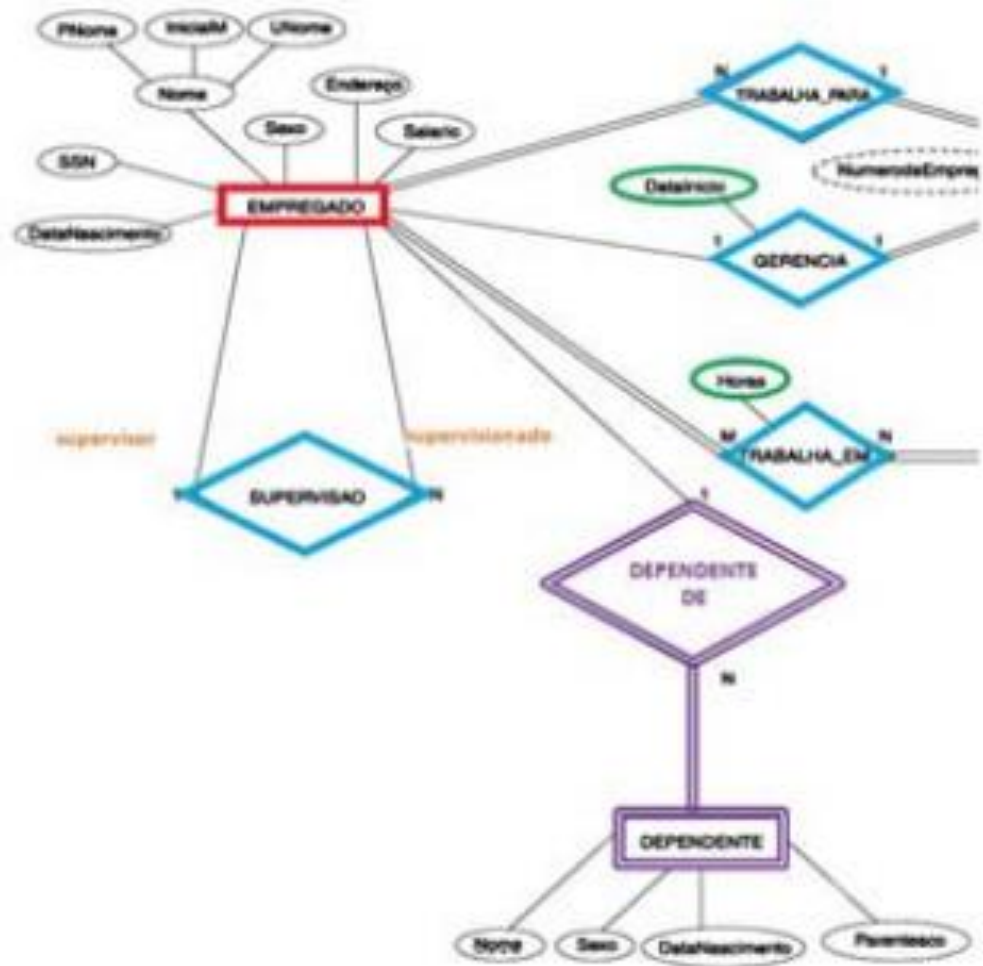
- São entidades que são dependentes de existência ou de **identificação** de um outra entidade.
- Se a existência de uma instância  $x$  depende da existência de uma outra instância  $y$ , então  $x$  (instância subordinada) é *dependente de existência* de  $y$  (instância dominante), e, portanto, a entidade que contém  $x$  é fraca com relação à entidade que contém  $y$ . Então, se  $y$  for removido,  $x$  também o será.
- As instâncias são identificadas através do relacionamento com a entidade forte (dominante)
- Possui chave parcial (sua chave advém da entidade da qual depende



# Entidades Fracas










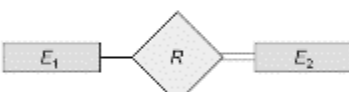


Denotadas por um duplo losango e duplo retângulo

Ex. Empregado e Dependente



# Resumo da notação ER – Elmasri & Navathe

**Figure 3.14**  
Summary of the  
notation for ER  
diagrams.

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of $E_2$ in $R$
	Cardinality Ratio 1: N for $E_1:E_2$ in $R$
	Structural Constraint (min, max) on Participation of $E$ in $R$

# Exemplo

- Modelagem Conceitual de um BD para uma Universidade, com informações sobre
  - Estudantes
  - Disciplinas
  - Departamentos
  - Professores

# Exemplo: BD Universidade

## ■ Estudantes:

- Armazenar o número de matrícula que os identifica, nome, endereço completo (logradouro, número, complemento, cidade, estado), sexo.
- Os estudantes matriculam-se em disciplinas em semestre e turma específicos, recebem uma nota ao final para formar o histórico

## ■ Professores:

- Ministram disciplinas
- Têm uma matrícula, cpf, nome e vários contatos
- Pertencem a um único departamento
- São avaliados por estudantes (anônimo), e cada avaliação contém data-hora, nota (0-10) e comentários

# Exemplo: BD Universidade

## ■ Disciplinas:

- ☐ Possuem código, nome e número de créditos
- ☐ São oferecidas por departamento e ministradas por professores
- ☐ Algumas possuem pré-requisitos

## ■ Departamentos

- ☐ Ofertam disciplinas
- ☐ Possuem código, nome e localização
- ☐ Têm um professor como Chefe, para o qual a data de início de mandato deve ser armazenada

# Exemplo: BD Universidade

## ■ Disciplinas:

- ☐ Possuem código, nome e número de créditos
- ☐ São oferecidas por departamento e ministradas por professores
- ☐ Algumas possuem pré-requisitos

## ■ Departamentos

- ☐ Ofertam disciplinas
- ☐ Possuem código, nome e localização
- ☐ Têm um professor como Chefe, para o qual a data de início de mandato deve ser armazenada

# Exemplo: BD Universidade

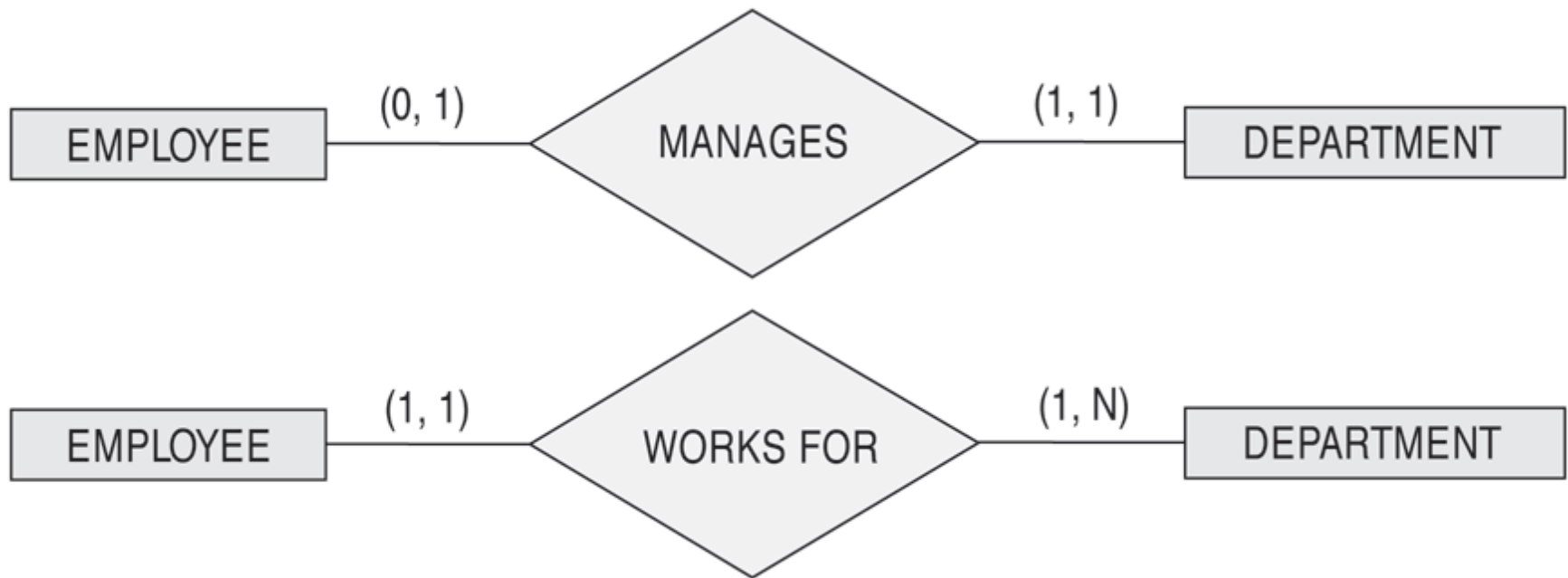


# Alternative (min, max) notation for relationship structural constraints:

- Especificado em cada participação de uma entidade E em um relacionamento R
- Especifica que cada instância de E participa em ao menos *min* e no máximo *max* instâncias do relacionamento R
- Default(no constraint): min=0, max=n
- Deve obedecer:  $\text{min} \leq \text{max}$ ,  $\text{min} \geq 0$ ,  $\text{max} \geq 1$
- Exemplo:
  - Um departamento tem exatamente um gerente um empregado pode gerenciar no máximo um departamento:
    - Especificar (0,1) para participação de EMPREGADO em GERENCIA
    - Especificar (1,1) para participação de DEPARTAMENTO em GERENCIA



# Notação com (min,max) para restrições de relacionamentos



# MER Estendido

- Estende o modelo ER com a incorporação de Herança de Entidades (Especialização e Generalização)
- Subclasses (Subentidades)
  - ✓ Estas subentidades são subconjuntos da entidade Empregado, ou seja, cada instância de uma subentidade é também uma instância da entidade Empregado. Então dizemos que Secretária **é uma** Empregada, Engenheiro **é um** Empregado e Técnico **é um** Empregado.
  - ✓ Este relacionamento **é um** caracteriza a **herança**. Ou seja, a subentidade (subclasse) herda todos os atributos e relacionamentos da superentidade (superclasse).

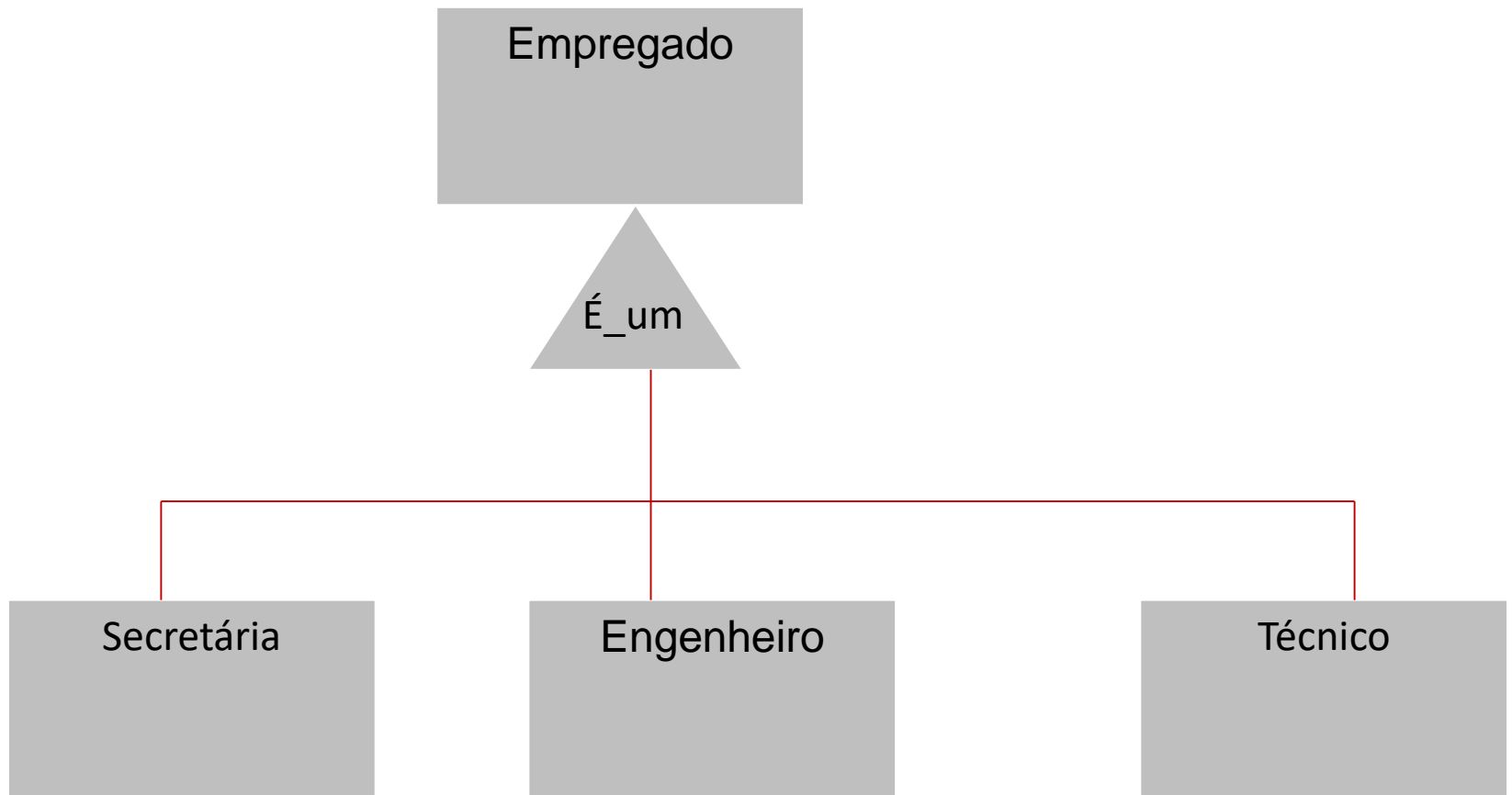
# MER Estendido

## ■ Superclasses e Subclasses

- É importante notar, que nem toda instância da superentidade é membro de uma subentidade.

**Ex.:** podemos ter empregados que não são nem secretária, nem engenheiro, nem técnico.

# MER Estendido



# Especialização

É o processo de definir um conjunto de subclasses de uma entidade (superentidade). Podem-se ter várias especializações de uma mesma entidade, baseado em Características distintas.

**Ex.:** a entidade **Empregado** pode também ser especializada nas subentidades **Assalariado** e **Horista**.

# Especialização

- Existem pelo menos duas razões para usar especialização num modelo de dados:

- ☐ Certos atributos podem ser aplicados somente a algumas instâncias de uma entidade (subclasse), mas não a todas.

Ex.:      Secretária: línguas, velocidade de digitação      Engenheiro: Especialidade, CREA      Motorista: número da carteira de habilitação, categoria

- ☐ Alguns relacionamentos só se aplicam a algumas instâncias que pertencem a uma subclasse.

Ex.:      Horistas pertencem\_a Empreiteiras

# Generalização

É o processo inverso à Especialização, isto é, é um processo de síntese em que suprimimos as diferenças entre várias entidades (subclasses), identificamos suas características comuns e as generalizamos numa superclasse.

# MER Estendido – Restrições

- **Cobertura Total:** cada instância da superentidade deve ser uma instância de alguma subentidade.

**Ex.:** Todo Empregado deve ser Engenheiro, Secretária ou Motorista

- **Cobertura Parcial:** uma instância de uma superentidade pode não ser membro de nenhuma subclasse.

**Ex.:** Pode existir empregado que não seja Engenheiro, Secretária nem Motorista.



# MER Estendido – Restrições

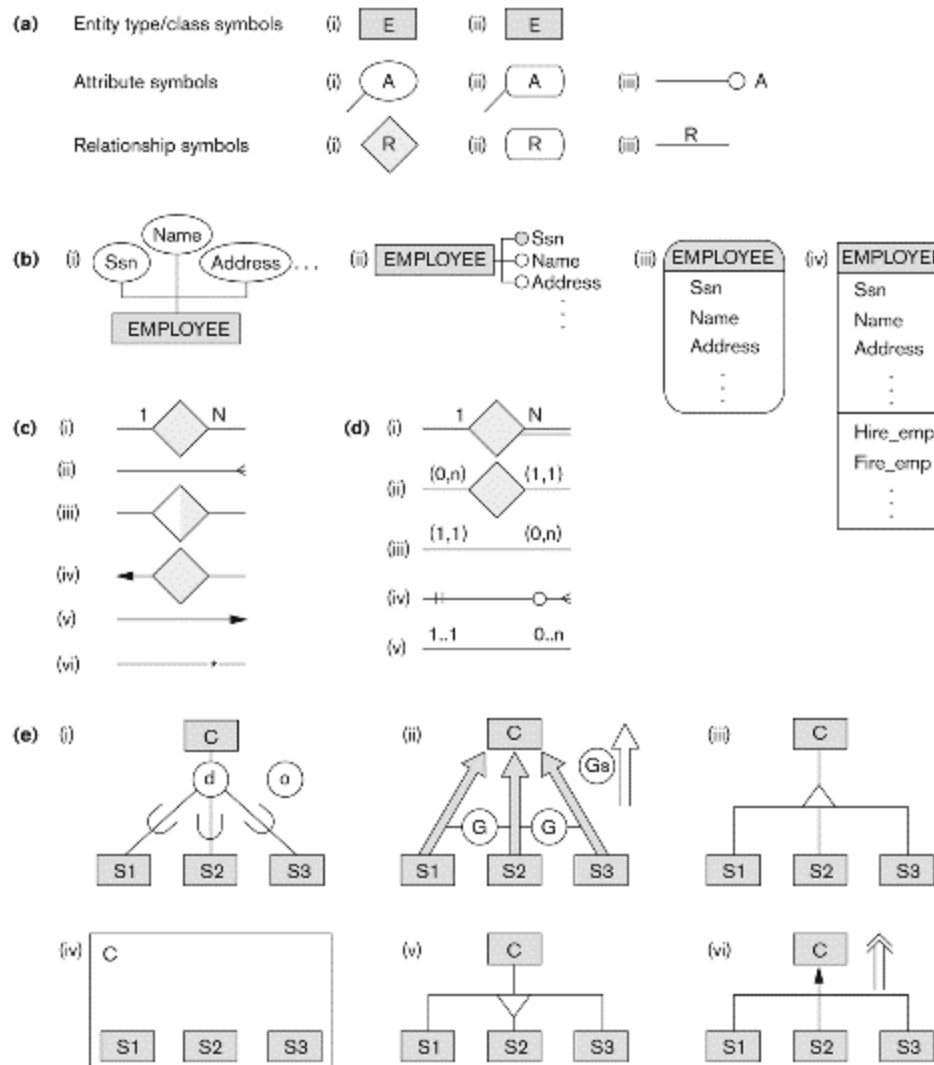
- **Disjunção:** uma dada instância pode ser membro de no máximo uma subentidade.

**Ex.:** Empregado ou é secretária, engenheiro ou técnico.

- **Sobreposição:** uma mesma instância pode ser membro de mais de uma subentidade

**Ex.:** Empregado pode ser engenheiro e técnico.

# Notações Alternativas



**Figure A.1**

Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.