

Banco de Dados

Modelo entidade-relacionamento

FACOM – UFMS

Vanessa Borges

vanessa@facom.ufms.br

- Ramez Elmasri e Shamkant B. Navathe
- Capítulo 7: modelagem de dados utilizando o modelo Entidade-Relacionamento (ER)
- Carlos Alberto Heuser
- Capítulo 2: abordagem entidade-relacionamento

Projeto de banco de dados

- Objetivo da abordagem de BD
 - Oferecer **abstração dos dados**
 - Separar aplicações dos usuários dos detalhes de hardware
 - Ferramenta utilizada → **modelo de dados**



Modelos de Dados

- Uma característica fundamental do uso de banco de dados é a abstração dos dados.
 - Ocultando os detalhes de armazenamento
- Para isso, é criado um **Modelo de dados**:
 - Conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados
 - Fornece o significado necessário para permitir essa **abstração**

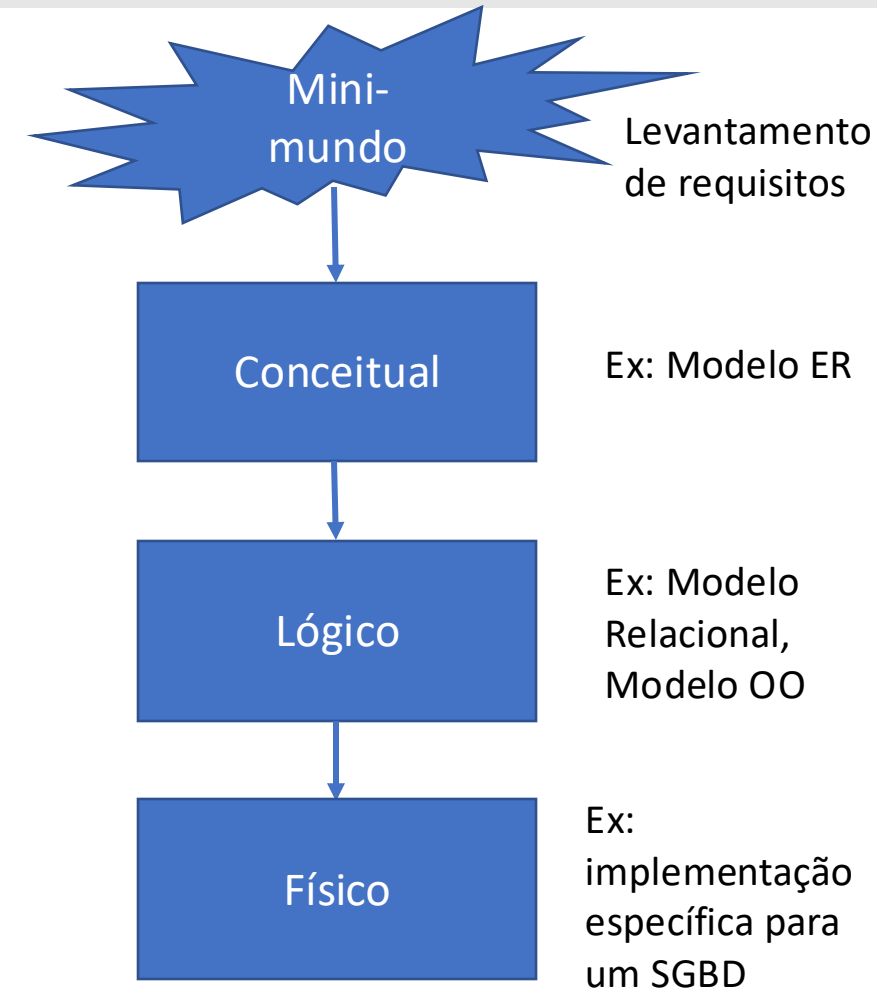


Tipos, relacionamentos e restrições de dados



Fases de um Projeto de Banco de Dados

- Existem várias propostas para representar um modelo de dados. Podemos classificá-los de acordo com os conceitos usados para descrever a estrutura do banco de dados:
- Modelos conceituais ou alto nível**
 - Descrevem a estrutura de um banco de dados de acordo com a percepção dos usuários **independentes de aspectos de implementação**
- Modelos representativos (lógicos)**
 - Descrevem a estrutura de um banco de dados da forma como será manipulado pelo SGBD **mais dependentes de aspectos de implementação**
- Modelos físicos ou baixo nível**
 - Descrevem a estrutura de um banco de dados da forma como os dados são fisicamente armazenados **totalmente dependentes de aspectos de implementação** (registros, blocos, índices, etc.)



Modelos de Dados - conceitual

- Descrição do banco de dados independente do SGBD

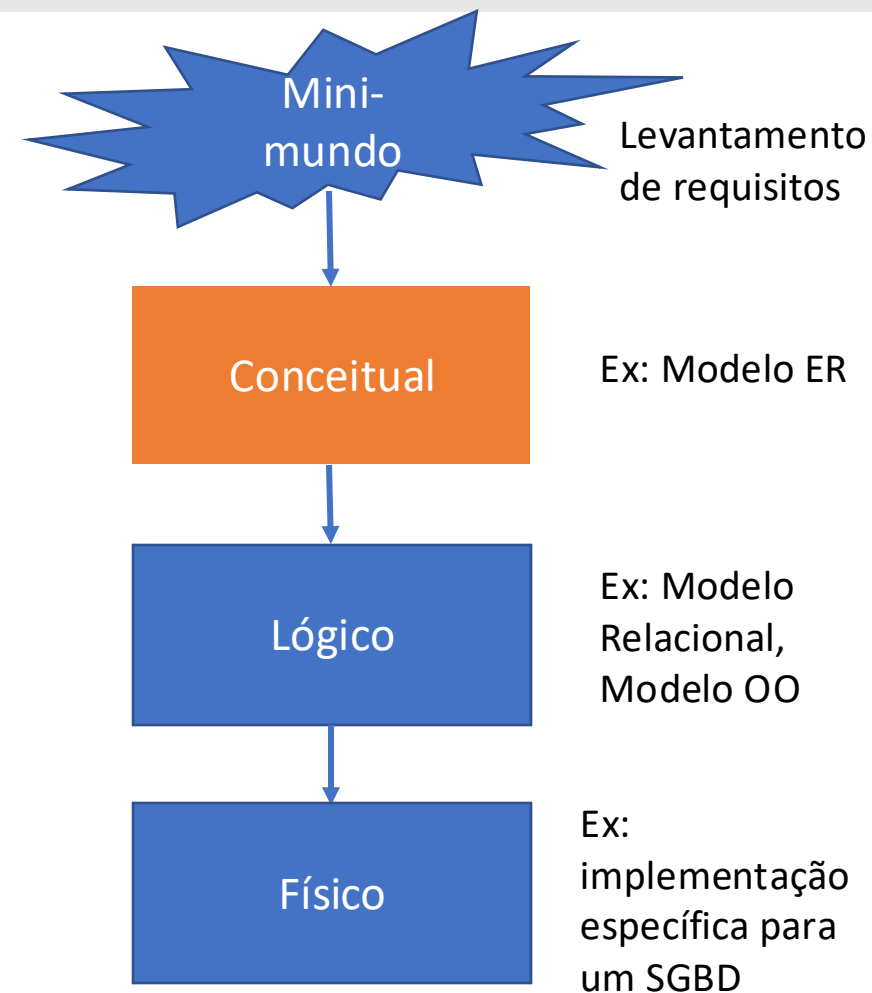
Entidade, atributo e relacionamentos

Representa um
objeto ou conceito
do mundo real

Representa alguma
propriedade de interesse.
Descreve melhor uma
entidade.

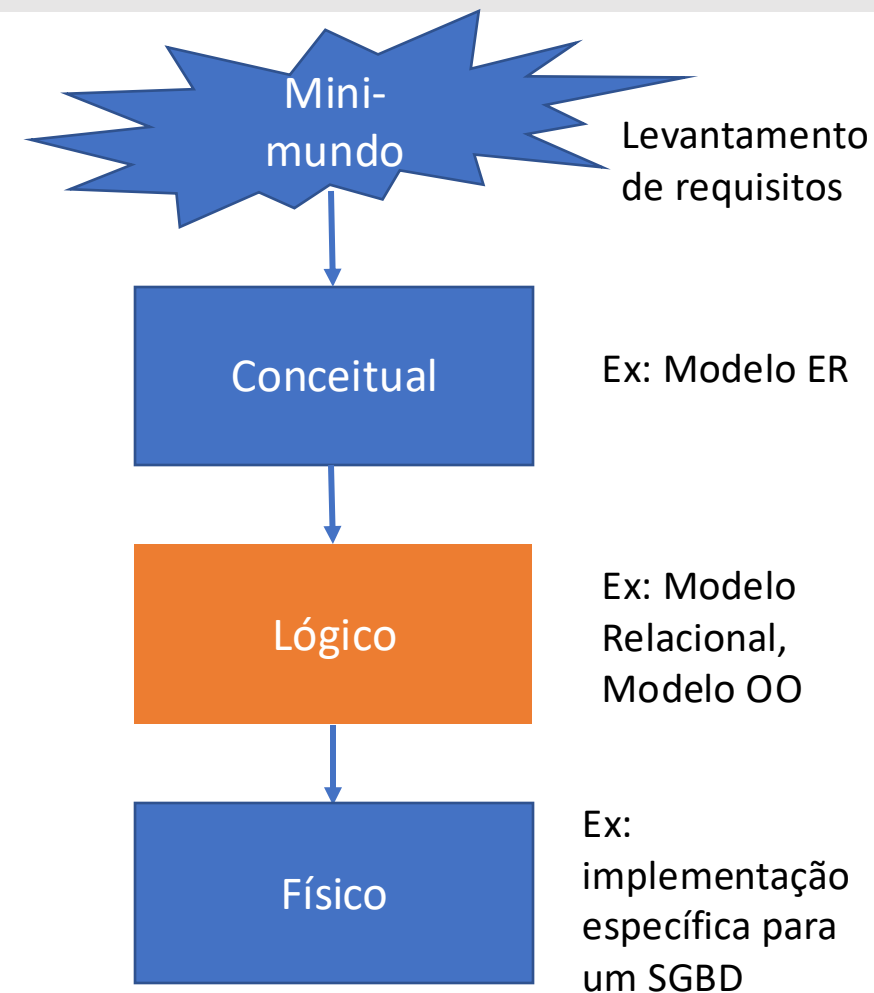
Representa uma
associação entre
as entidades

- **Modelo Entidade e Relacionamento (ER)**



Modelos de Dados - lógico

- O modelo de dados representacional (ou lógico) são baseados em registros.
 - **Modelo de Dados Relacional:** o modelo mais difundido usado atualmente
- Modelo de Dados: modelos de dados de **rede** e Modelo de dados **hierárquicos**.
- Modelo de dados **Orientados a Objetos:** são uma nova família de modelos de dados de implementação conceitual.



Modelos de Dados - lógico

- **Modelo de Dados Relacional:**

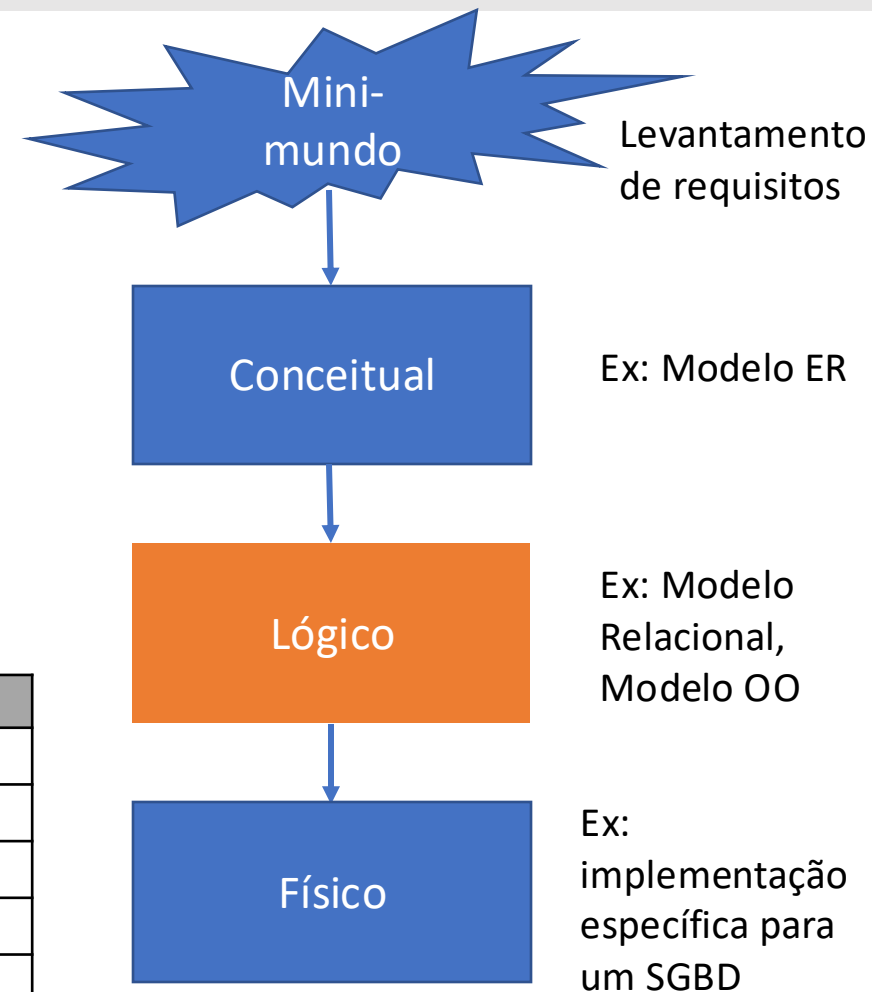
- Modelo criado por E.F. Codd na década de 70
- Baseado em fortes conceitos matemáticos e teoria dos conjuntos
- É o modelo utilizado atualmente pela maioria dos SGBDs relacionais
- Composto por coleção de tabelas representando os dados e seu relacionamento

DEPARTAMENTO

<u>CodDepartamento</u>	Nome
1	Administrativo
2	Financeiro
3	Jurídico
4	Comercial

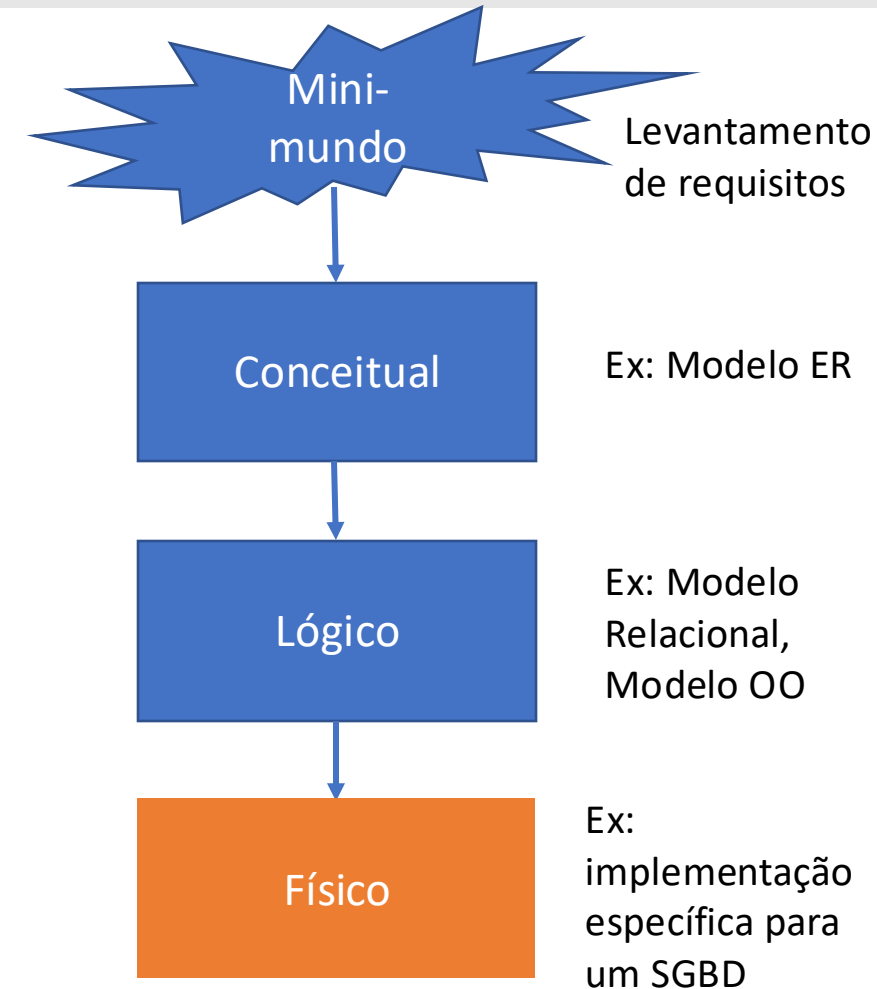
FUNCIONARIO

<u>CodFuncionario</u>	Nome	CodDepartamento	DtLotacao
1	Carlos	1	12/01/17
2	Ana	1	NULL
3	Arthur	1	20/06/18
4	Renato	2	NULL
5	Thaís	3	20/06/19



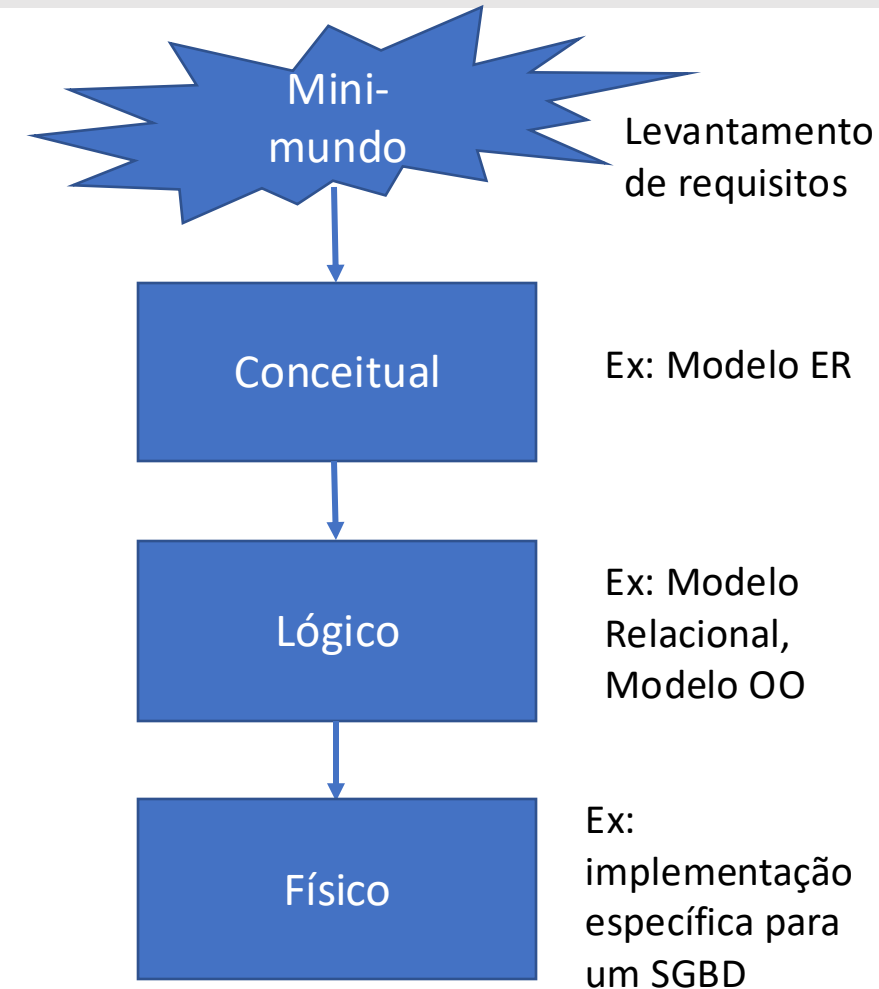
Modelos de Dados - físico

- Descreve a base de dados **internamente** (ajuste de performance).
 - Indexação e estruturas de arquivos
 - Transações e controle de concorrência
 - Otimização
 - Recuperação em casos de falhas
 - Mecanismos de proteção (segurança)
- Tendência em produtos modernos é cada vez mais **esconder o modelo físico**.
- **Implementação para um SGBD específico**
- Levará em conta os tipos de dados, restrições, tipos de objetos e outros objetos proprietários de cada SGBD



Fases de um Projeto de Banco de Dados

- Existem várias propostas para representar um modelo de dados. Podemos classificá-los de acordo com os conceitos usados para descrever a estrutura do banco de dados:
- **Modelos conceituais ou alto nível**
 - Descrevem a estrutura de um banco de dados de acordo com a percepção dos usuários independentes de aspectos de implementação
- **Modelos representativos (lógicos)**
 - Descrevem a estrutura de um banco de dados da forma como será manipulado pelo SGBD mais dependentes de aspectos de implementação
- **Modelos físicos ou baixo nível**
 - Descrevem a estrutura de um banco de dados da forma como os dados são fisicamente armazenados totalmente dependentes de aspectos de implementação (registros, blocos, índices, etc.)



Fases de um Projeto de Banco de Dados

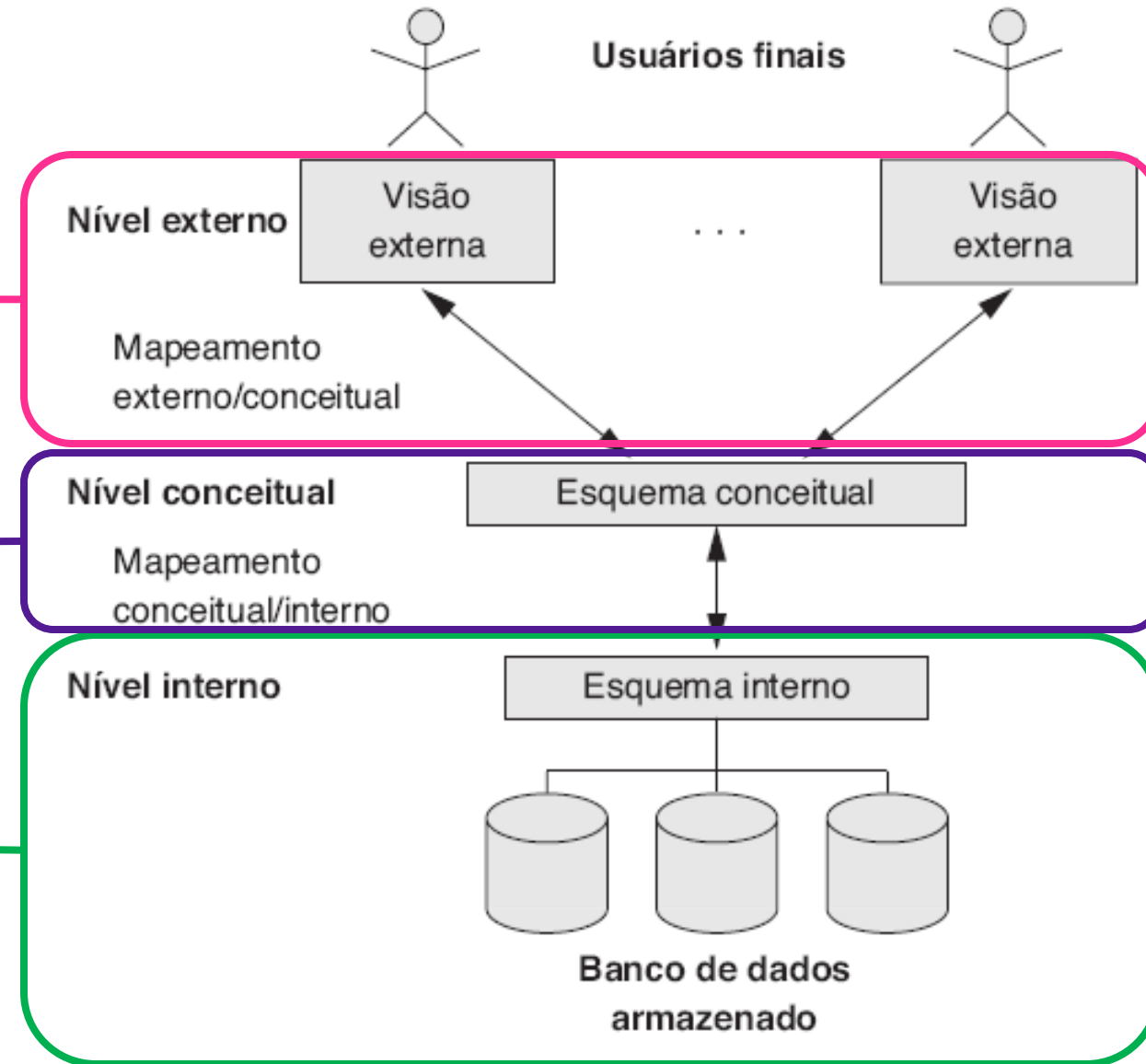
Arquitetura de três esquemas (ANSI/SPARC)

- **Objetivo:** separar as aplicações do usuário do banco de dados físico

Descreve a parte do banco de dados em que um grupo de usuários (ou aplicações) em particular está interessado

Descreve a estrutura do banco de dados inteiro para uma comunidade de usuários (entidades, relacionamentos, restrições, etc)

Descreve a estrutura do armazenamento físico do banco de dados.

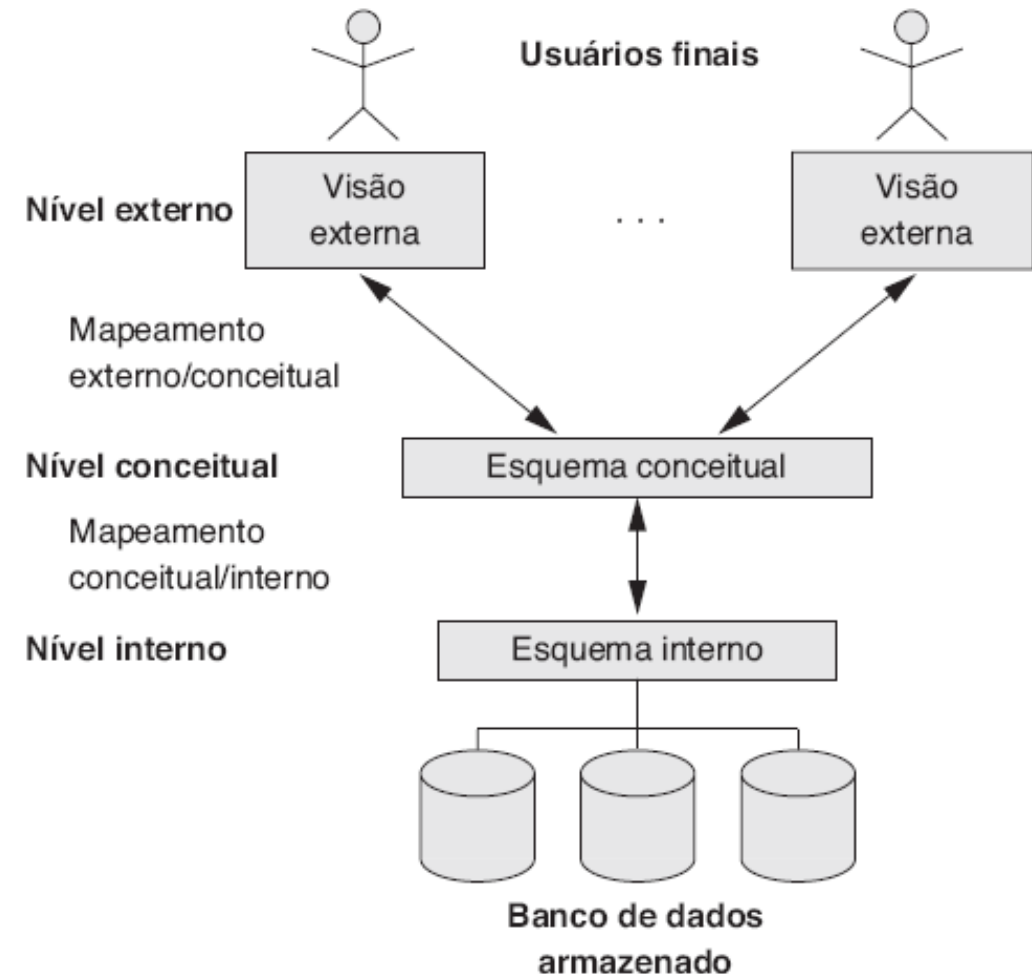


Fases de um Projeto de Banco de Dados

Arquitetura de três esquemas (ANSI/SPARC)

Segunda Navathe: relação entre os modelos e os níveis da arquitetura três esquemas

Níveis	Modelos			
		Modelo Físico	Modelo ER	Modelo Relacional
	Nível interno	X		
	Nível conceitual		Baseado em um projeto conceitual	X
	Nível externo		Baseado em um projeto conceitual	X



Fases de um Projeto de Banco de Dados

Arquitetura de três esquemas (ANSI/SPARC)

- **Independência de dados:**

- Pode ser definida como a capacidade de se **alterar um esquema** em um nível em um banco de dados **sem ter que alterar um nível superior**.

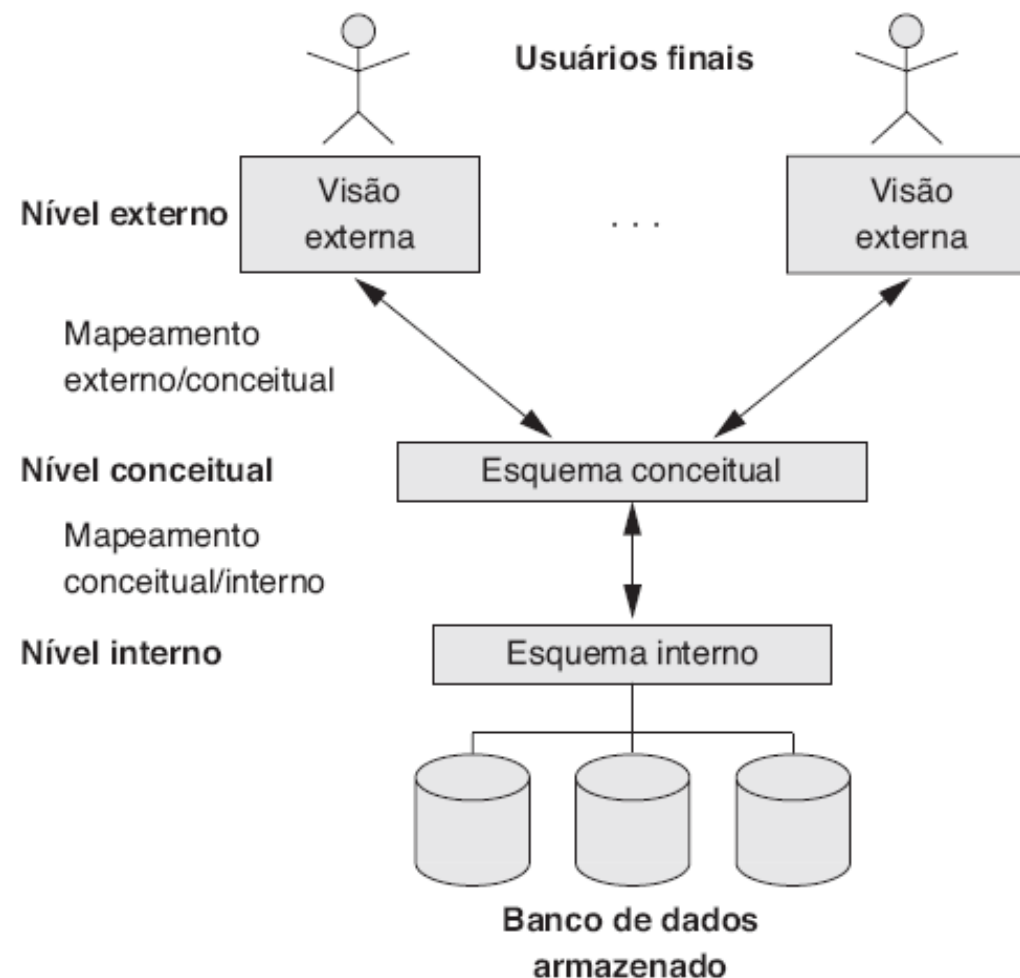
- Existem dois tipos de independência de dados:

- **Independência de dados lógica:**

- É a capacidade de alterar o **esquema conceitual** sem ter que alterar o **esquema externo** ou as aplicações do usuário;

- **Independência de dados física:**

- É a capacidade de alterar o **esquema interno** sem ter que alterar o **esquema conceitual**, o esquema externo ou as aplicações do usuário.





Banco de Dados

Modelo entidade-relacionamento



FACOM – UFMS

Vanessa Borges

vanessa@facom.ufms.br

- Ramez Elmasri e Shamkant B. Navathe
- Capítulo 7: modelagem de dados utilizando o modelo Entidade-Relacionamento (ER)
- Carlos Alberto Heuser
- Capítulo 2: abordagem entidade-relacionamento

Categorias de Modelo de Dados

- Existem várias propostas para representar um modelo de dados. Podemos classificá-los de acordo com os conceitos usados para descrever a estrutura do banco de dados:

- Modelos conceituais ou alto nível**

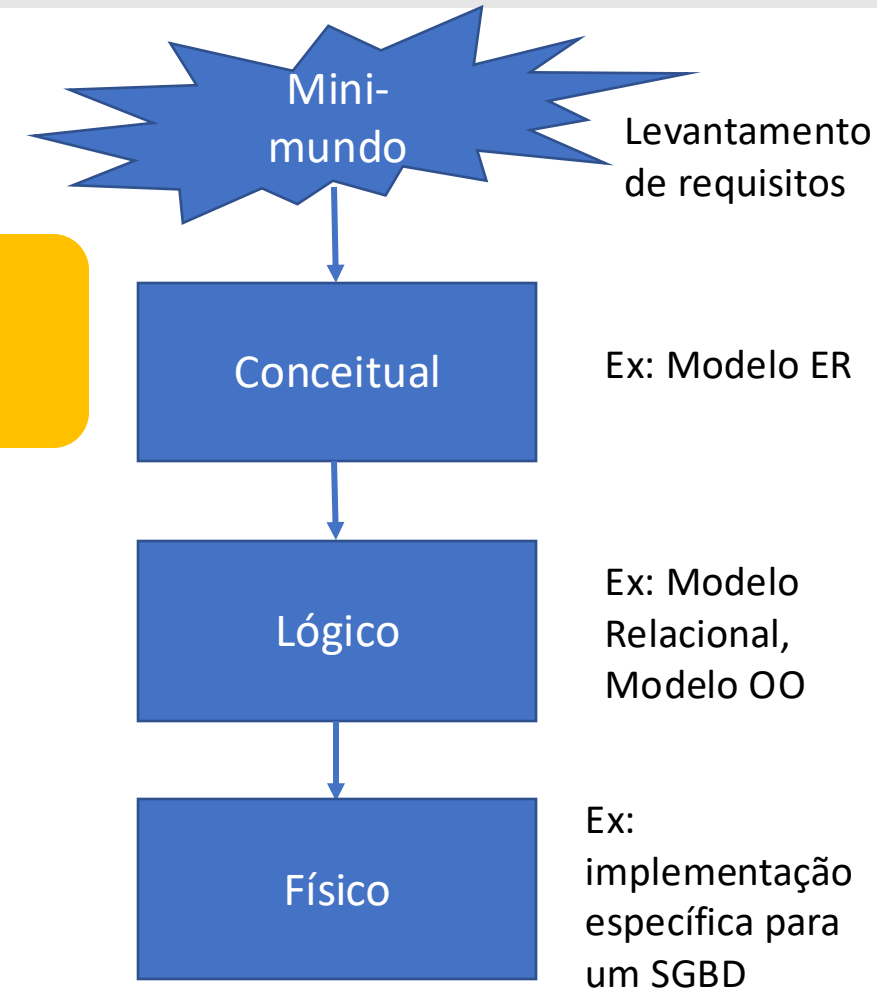
- Descrevem a estrutura de um banco de dados de acordo com a percepção dos usuários independentes de aspectos de implementação

- Modelos representativos (lógicos)**

- Descrevem a estrutura de um banco de dados da forma como será manipulado pelo SGBD mais dependentes de aspectos de implementação

- Modelos físicos ou baixo nível**

- Descrevem a estrutura de um banco de dados da forma como os dados são fisicamente armazenados totalmente dependentes de aspectos de implementação (registros, blocos, índices, etc.)



Introdução

- **Utilizando modelo de dados conceituais de alto nível para o projeto do banco de dados**
 - Esquema conceitual
 - Projeto conceitual
 - Descrição concisa dos requisitos de dados
 - Inclui detalhes dos tipos de entidade, relacionamentos e restrições
 - Transformado do modelo de dados de alto nível para o modelo de dados da implementação



Introdução

- O **Modelo Entidade-Relacionamento** (MER) é um modelo de dados de alto-nível criado com o objetivo de representar a semântica associada aos dados do minimundo
- **Características**
 - Proposto por Peter Chen em 1976
 - Foi desenvolvido para **facilitar o projeto lógico do BD**
 - Permite a representação da estrutura lógica global do BD
 - É um dos modelos de dados com **maior capacidade semântica**
 - Representa um problema como um conjunto de entidades e relacionamentos entre estas entidades



Introdução

- O esquema conceitual criado utilizando-se o MER é chamado de **Diagrama Entidade-Relacionamento (DER)**
 - *Unified Modeling Language (UML)*

- MER: conjunto de conceitos e elementos de modelagem que o projetista de banco de dados precisa conhecer.
- DER: resultado do processo de modelagem executado pelo projetista de dados que conhece o MER.



Modelos de Dados - conceitual

Controle acadêmico

- Deseja-se manter informações sobre **alunos**, **cursos**, **disciplinas** e **departamentos**.
- Uma disciplina pode possuir diversos **pré-requisitos**.
- Uma disciplina pode ser pré-requisito de muitas outras disciplinas.
- Uma disciplina pode aparecer no currículo de muitos cursos e um curso pode possuir muitas disciplinas em seu currículo.
- Um aluno está inscrito em exatamente um curso e um curso pode ter nele inscritos muitos alunos.
- Um aluno possui um RGA único, nome e endereço.
- Uma disciplina possui um código único, um nome e um número de créditos.
-

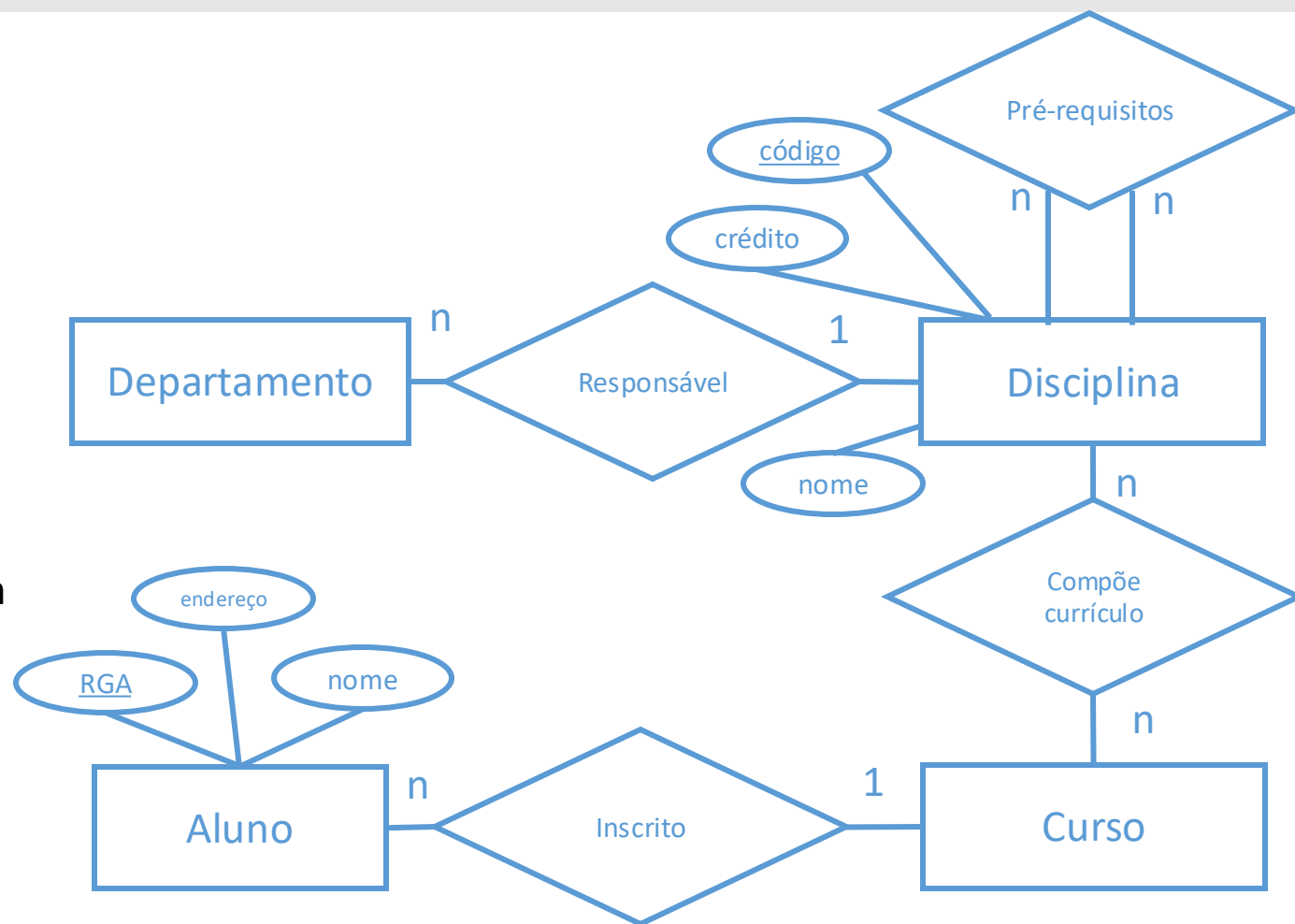
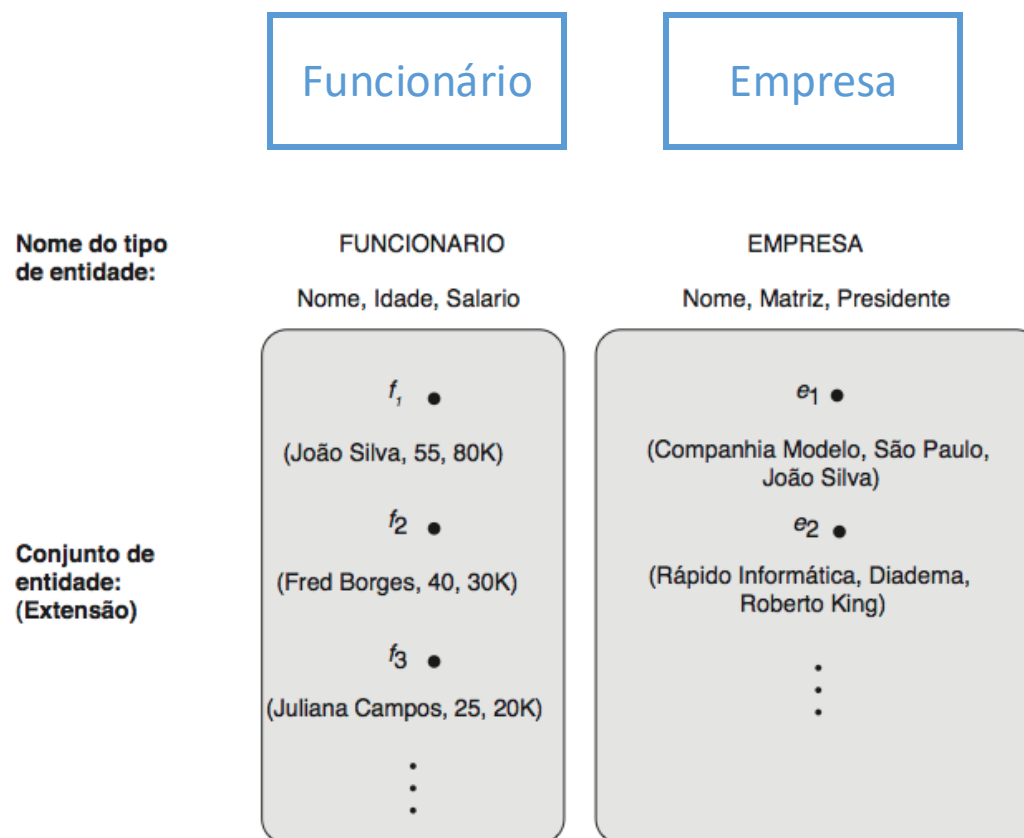


Diagrama entidade-relacionamento



Modelo ER – Tipo entidade

- Qualquer coisa do **mundo real** envolvida no problema
- **Coleção ou conjunto de entidades que possuem os mesmos atributos**



Modelo ER – Entidade

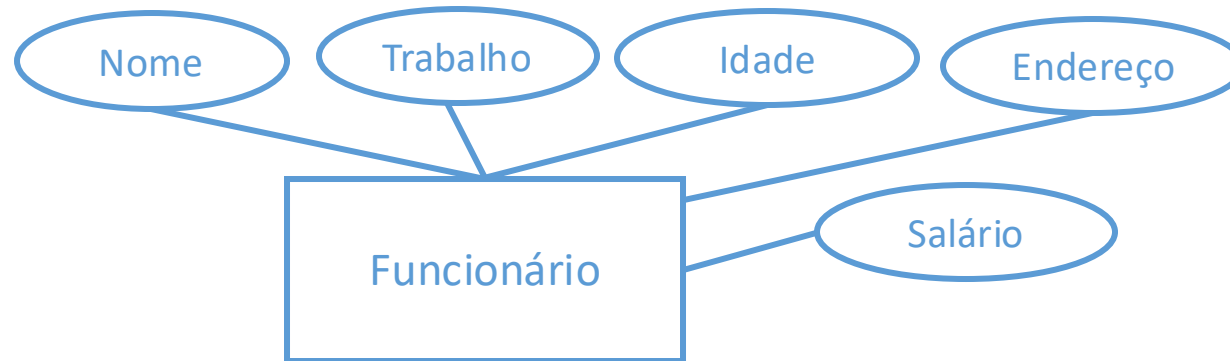
- Exemplo de entidade – banco de dados empresa

A empresa é organizada em **departamentos**. Cada **departamento** tem um nome exclusivo, um número exclusivo e um **funcionário** em particular que o gerencia. Um **departamento** pode ter vários locais. Um **departamento** controla uma série de **projetos**, cada um deles com um nome exclusivo, um número exclusivo e um local exclusivo.



MODELO ER – entidades e atributos

- Cada **Entidade** tem propriedades particulares que são chamadas de **Atributos**
 - Uma entidade **FUNCIONÁRIO** pode ser descrita pelo seu **nome**, o **trabalho que realiza**, **idade**, **endereço** e **salário**



Atributos

■ Tipos de atributos:

- *Simples versus composto*
- *Valor único versus multivalorados*
- *Armazenado versus derivado*
- Valores NULL
- Atributos complexos



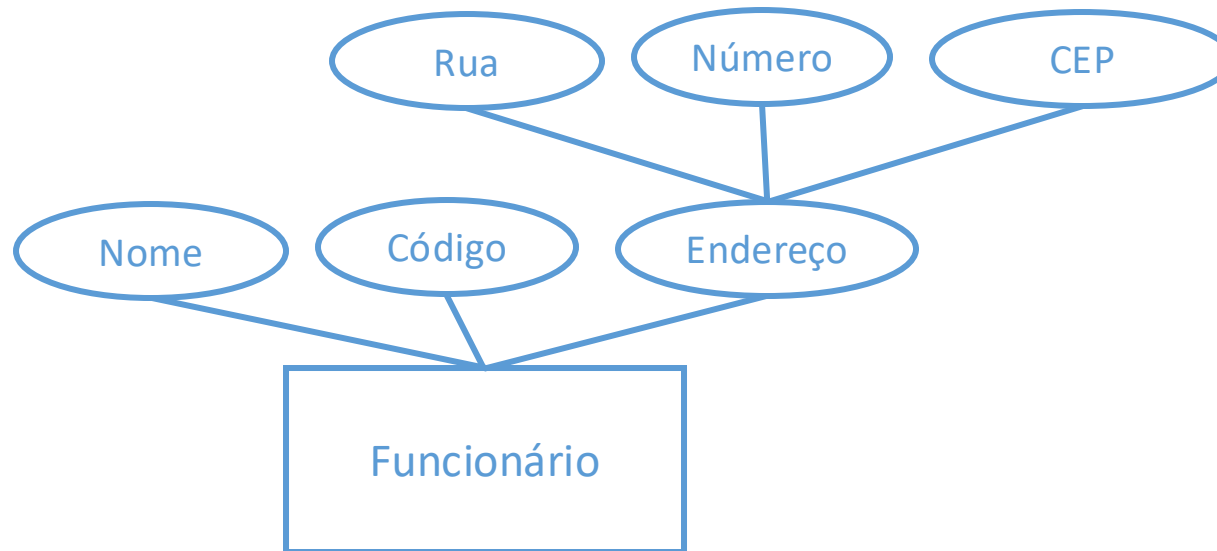
Atributos simples X composto

- Alguns atributos podem ser divididos em sub-partes com significados independentes.
- **Simples**
 - Cada entidade tem um único valor atômico para o atributos
 - São atômicos (não podem ser decompostos)
- **Composto**
 - O atributo pode ser composto por vários componentes
 - Podem formar uma hierarquia



Atributos simples X composto

- Notação gráfica:



Atributos multivalorado X monovalorado

- Muitos atributos têm apenas um valor (*monovalorados*). Porém existem atributos que podem ter um conjunto de valores (***Multivalorados***)

Monovalorados

- Um único valor para cada entidade (Ex: **nome**)

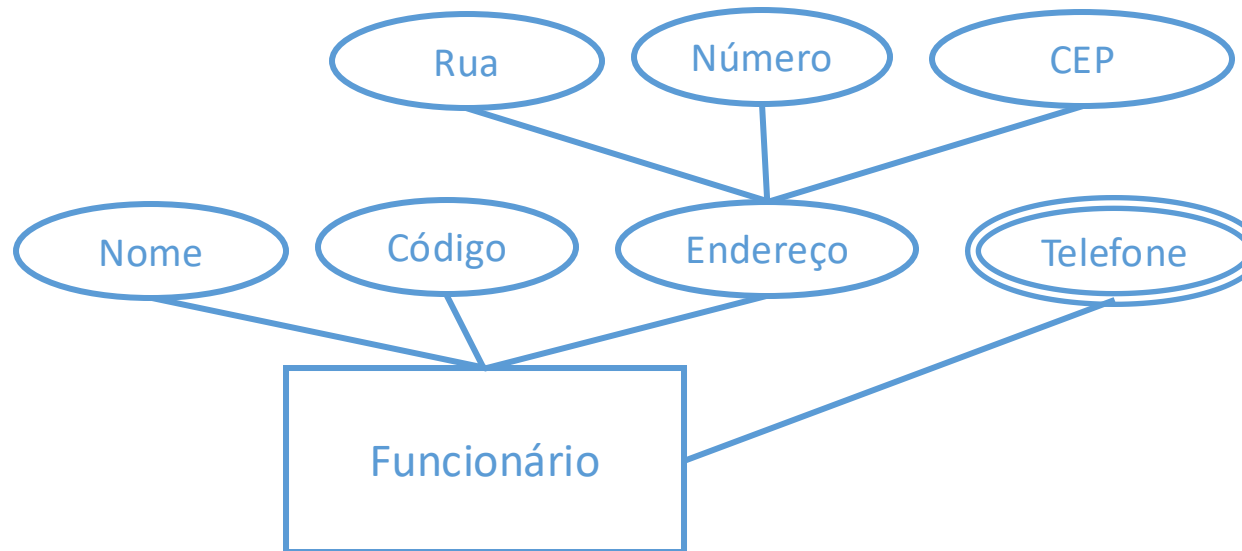
Multivalorados

- Múltiplos valores para cada entidade (Ex: uma pessoa pode ter mais de um **telefone**)



Atributos multivalorado X monovalorado

- Notação gráfica



Atributos armazenado x derivado

- **Armazenados**

- Está de fato armazenado em um DB

- **Derivados**

- Pode ser determinado através de outros atributos ou através de entidades relacionadas
 - Pode ou não ser armazenado no BD

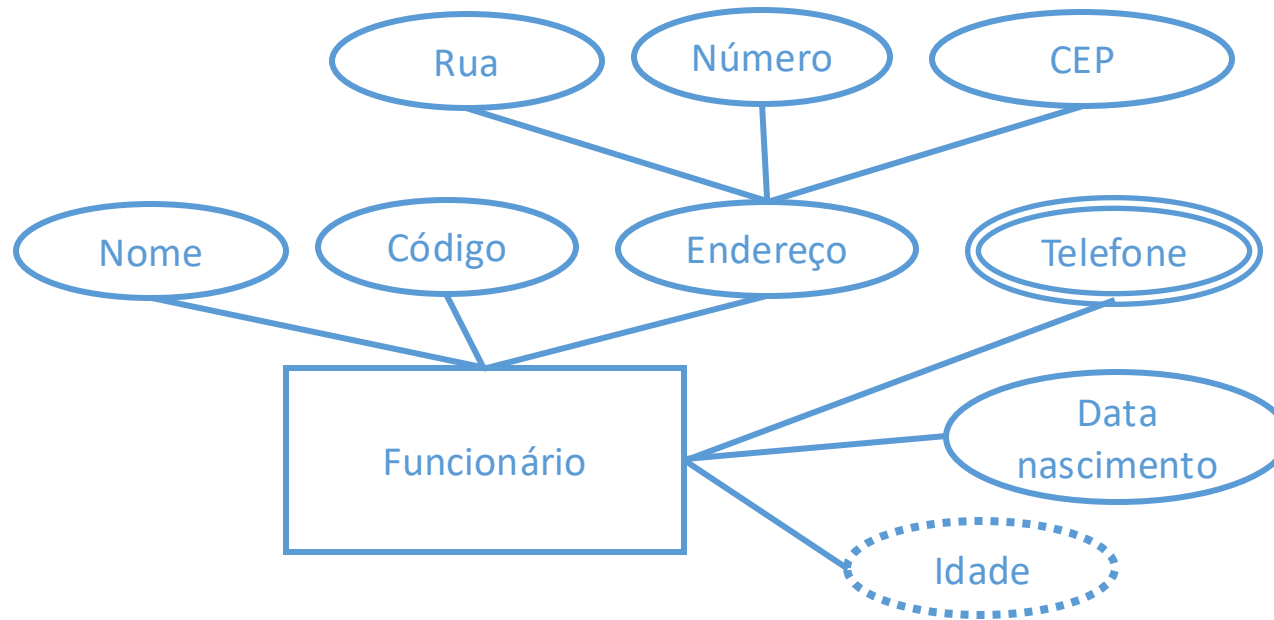
- **Exemplo:**

- **Idade = Data_Atual - Data_Nascimento**
 - **Número de empregados de um determinado departamento**



Atributos armazenado x derivado

- Notação gráfica:



Valores NULOS de atributos

- Algumas vezes pode acontecer de um atributo **não possuir valor**
- Nesses casos, atribui-se um valor nulo (*null*) para esse atributo
 - Apartamento = *null* para aqueles funcionários que não residam em um prédio. (***não aplicável***)
- O valor *null* pode ser aplicado também para denotar que o valor é ***desconhecido***



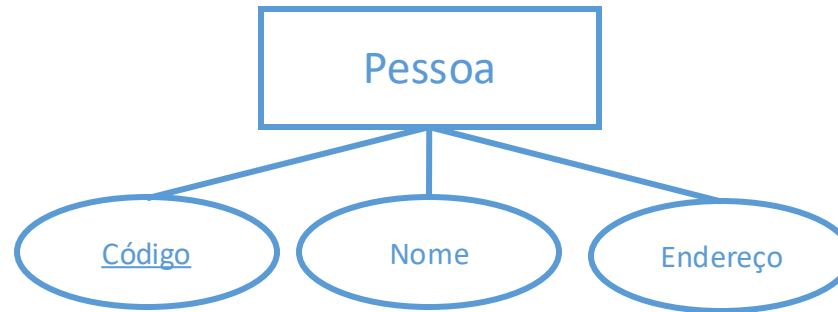
Atributo chave (identificadores de entidade)

- Uma restrição importante das entidades de um tipo de entidade é a **chave** ou **restrição de exclusividade sobre os atributos**.
- Em geral uma entidade possui **um ou mais atributos**
 - Esse atributo é chamado **atributo-chave**
- Quando vários atributos juntos formam uma chave, significando que a combinação dos valores de atributo deve ser distinta para cada entidade.
- Uma entidade também **pode não ter chave**; nesse caso, ele é chamado de tipo de **entidade fraca**



Atributo chave (identificadores de entidade)

- Na notação diagramática, se dois atributos forem sublinhados separadamente, então cada um é uma chave por si só (diferente do modelo relacional).
 - Não existe o conceito de chave primária no modelo ER, a chave primária será escolhida durante o mapeamento para um esquema relacional



Modelo ER – Relacionamento

- **Conjunto de associações entre duas ou mais entidades**

- Exemplo de relacionamento – banco de dados empresa

(...) Um funcionário **gerencia** um departamento (...). Os departamentos **possuem** vários funcionários.
Um departamento **controla** uma série de projetos (...).

Relacionamentos geralmente são expressos por ações que envolvem entidades.



Relacionamento

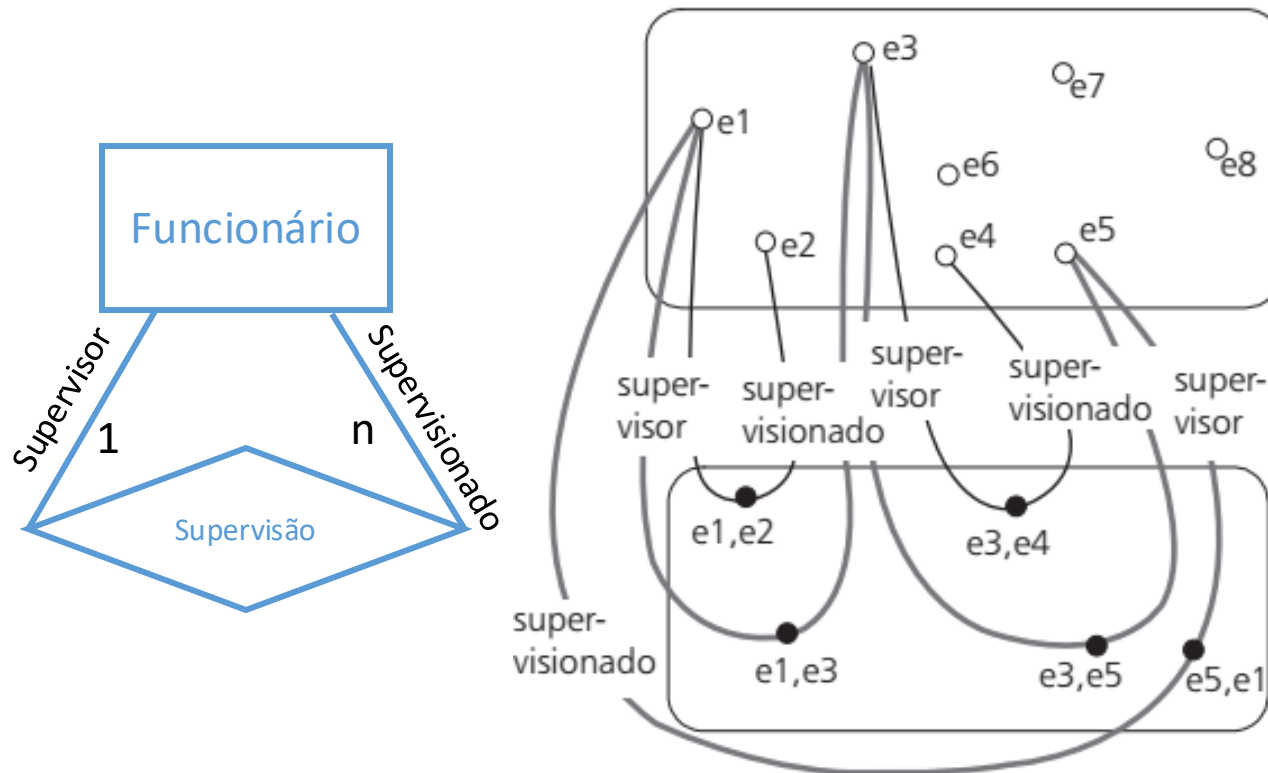
- Notação gráfica:



Relacionamento recursivo

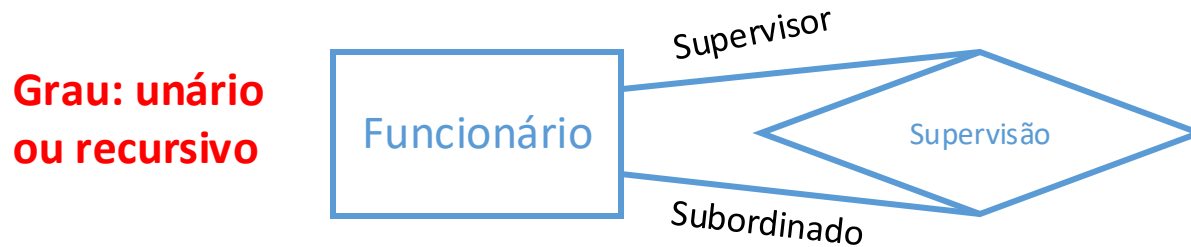
- Não necessariamente um relacionamento associa entidades diferentes
- Cada entidade que participa de um relacionamento possui um **papel** específico

Auto relacionamento
ou relacionamentos
recursivos



Grau do relacionamento

- O **Grau de um Tipo de Relacionamento** é número de entidades envolvidas

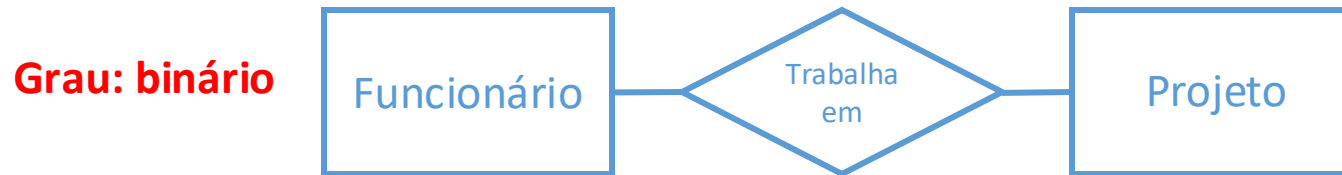


- Unário (ou recursivo) – relaciona um tipo-entidade com ela mesma – indicado utilizar nomes de papéis



Grau do relacionamento

- O **Grau de um Tipo de Relacionamento** é número de entidades envolvidas



- Relaciona duas entidades
 - grau de relacionamento mais utilizado



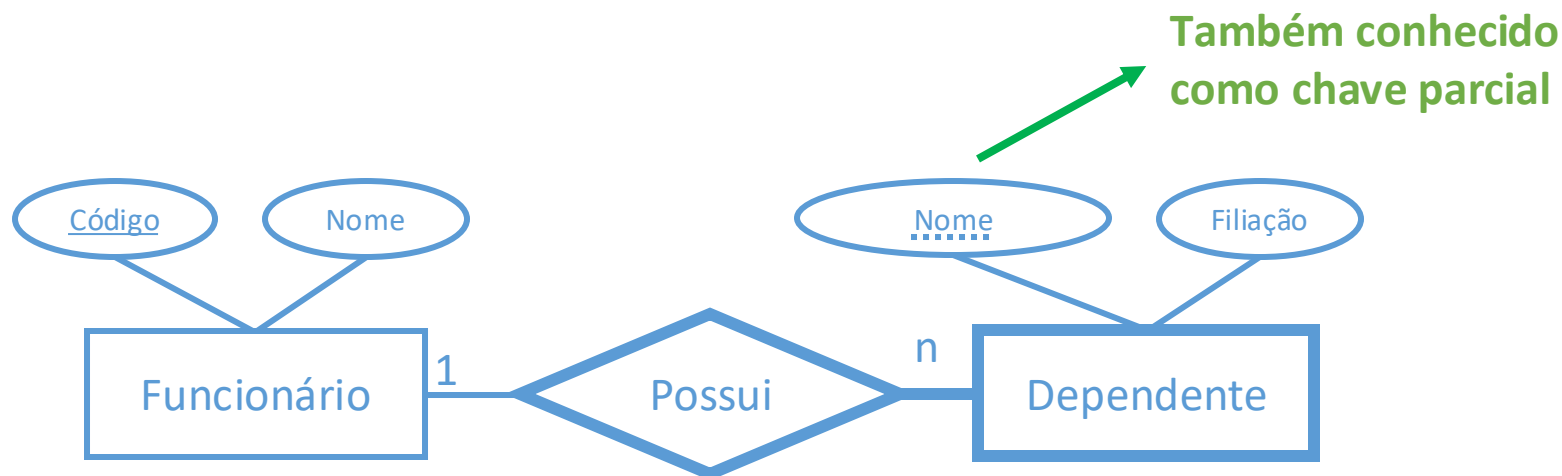
Entidade fraca

- Tipos de entidade que **não possuem atributos-chave próprios** são chamados tipos de entidade fraca.
- Uma entidade fraca normalmente tem uma chave parcial, que é o **atributo que pode identificar exclusivamente as entidades fracas** que estão relacionadas à mesma entidade proprietária.



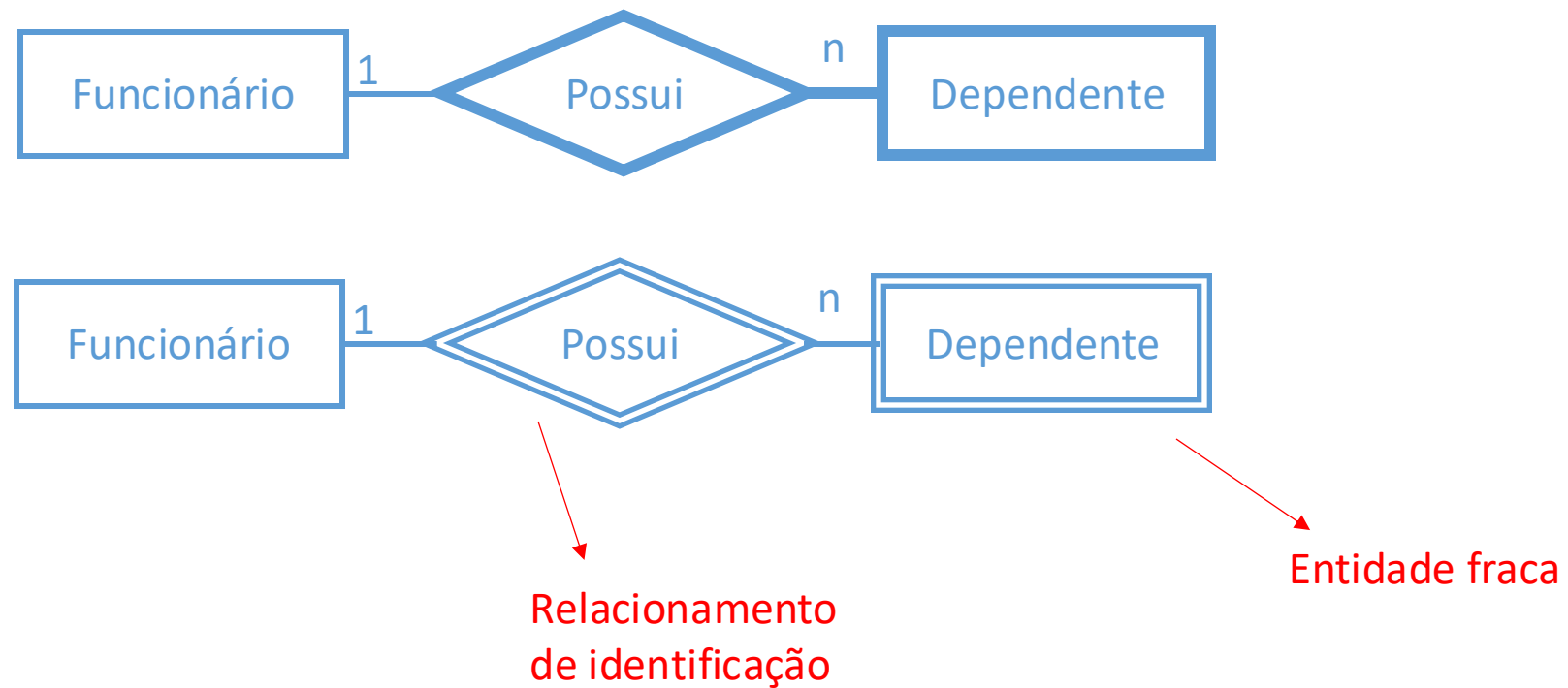
Entidade fraca

- Exemplo:
 - Empregado é um **entidade proprietária**
 - O relacionamento entre empregado de dependente é um **relacionamento identidade**
 - Se considerarmos que dois dependentes do mesmo funcionário não poderão ter o mesmo nome, o atributo Nome de DEPENDENTE é a chave parcial



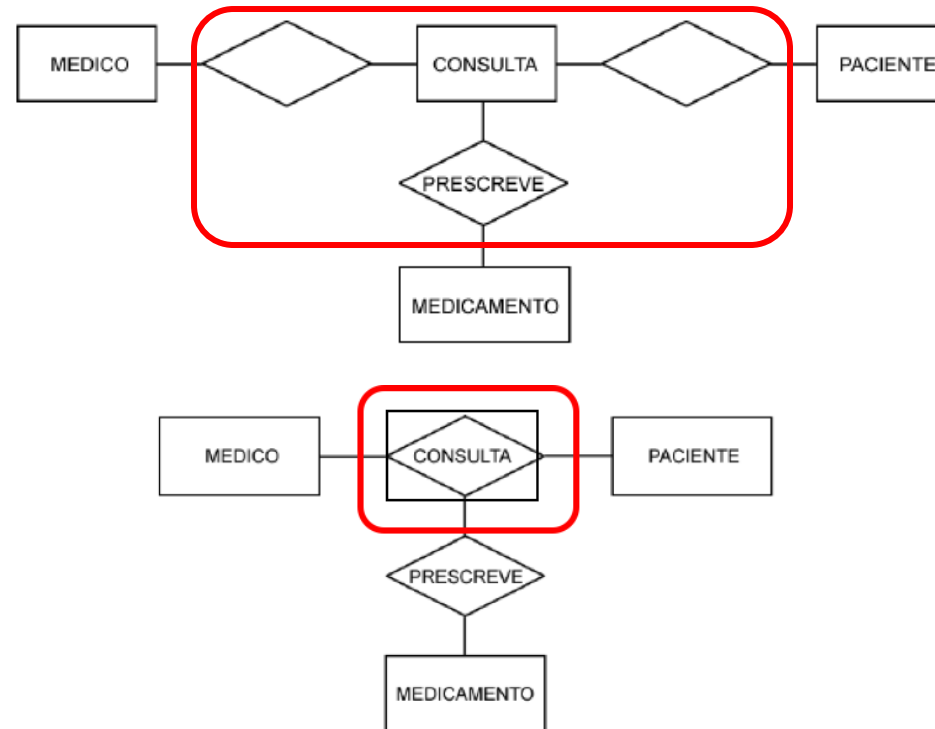
Entidade fraca

- Notações alternativas para entidade fraca



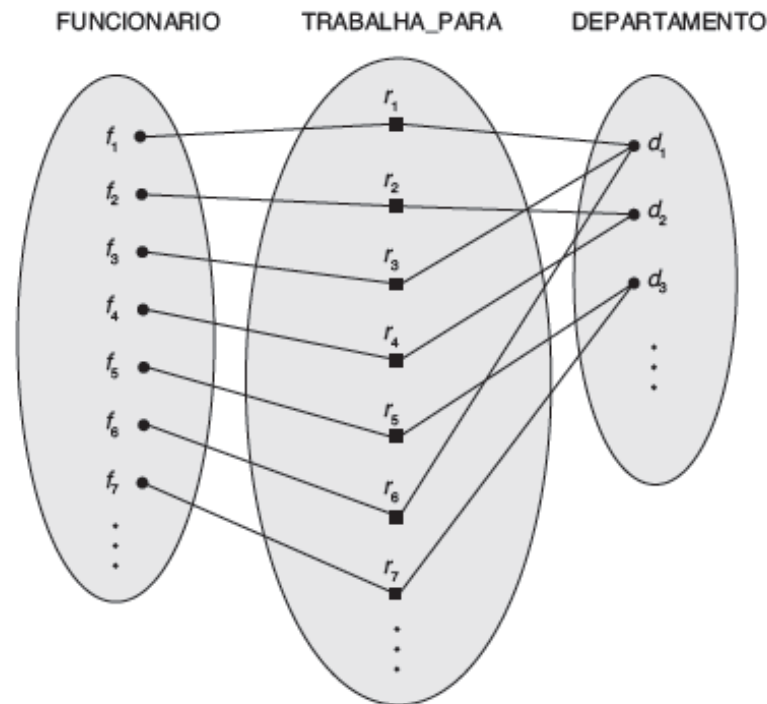
Entidade associativa

- Na modelagem ER não é possível realizar a associação entre relacionamentos
- Uma entidade associativa é a redefinição de um relacionamento que passa a ser tratado como se fosse também uma entidade



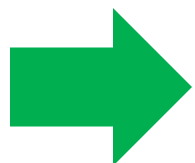
Cardinalidade

Especifica o número máximo de instâncias de relacionamento nos quais uma instância de entidade pode participar.



Restrição de participação na relação

Notação alternativa de cardinalidade



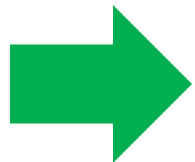
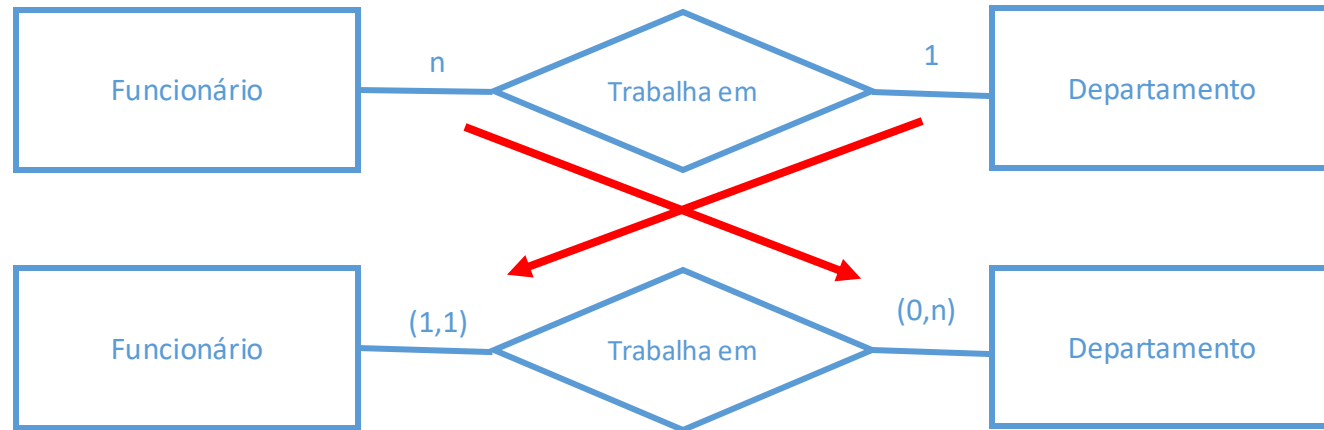
**Notação min, max
segundo Carlos Alberto
Heuser (Peter Chen)**



Restrição de participação na relação

Notação alternativa de cardinalidade

**Notação min, max
segundo Ramez Elmasri e
Shamkant B. Navathe**



**Notação min, max
segundo Carlos Alberto
Heuser (Peter Chen)**

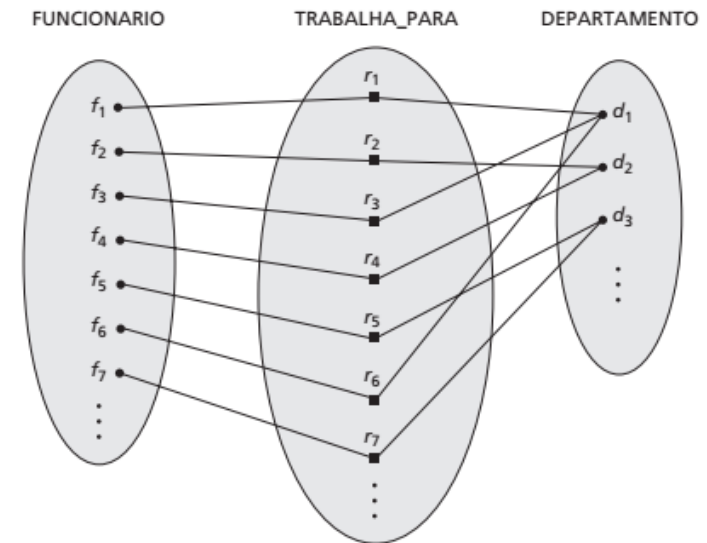
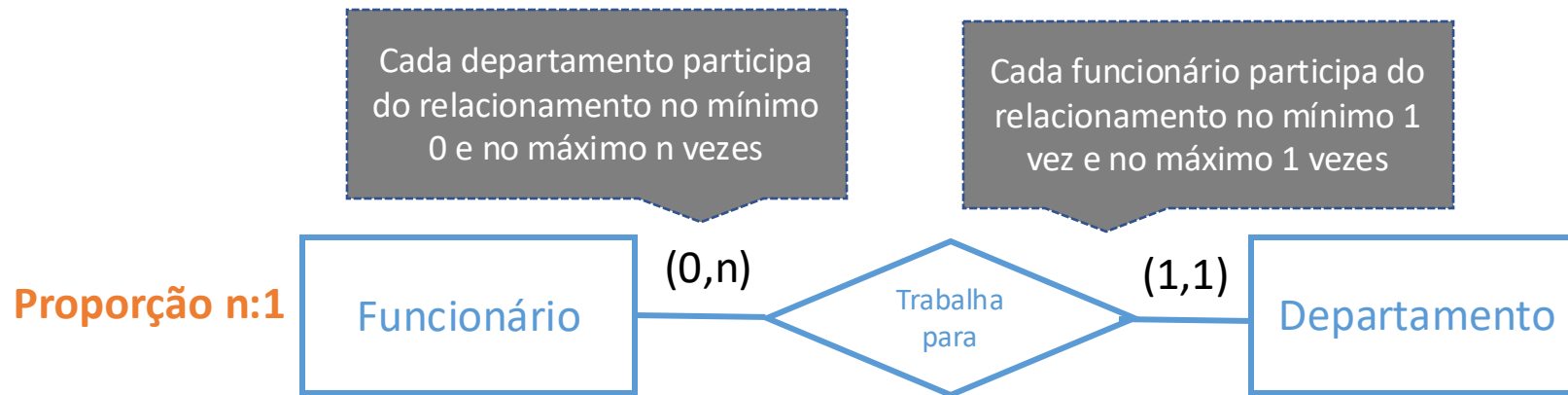


Restrição de participação na relação

Notação alternativa de cardinalidade

Notação min, max segundo Carlos Alberto Heuser (Peter Chen)

- Indica a restrição mínima e máxima (min, max) da participação de cada entidade no relacionamento



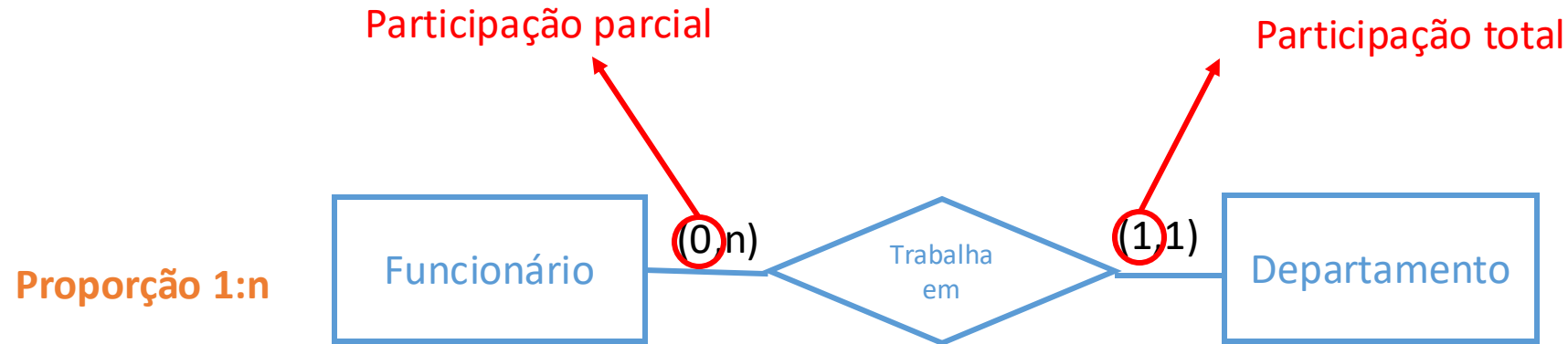
- Um funcionário não pode não trabalhar em um departamento (mínimo 1)
- Um funcionário pode trabalhar em no máximo 1 departamento (máximo 1)
- Um departamento pode não ter funcionários trabalhando (mínimo 0)
- Um departamento pode possuir diversos funcionários trabalhando (máximo n)

Restrição de participação na relação

Notação alternativa de cardinalidade

Notação min, max segundo Carlos Alberto Heuser (Peter Chen)

- Indica a restrição mínima e máxima (min,max) da participação de cada entidade no relacionamento



- Um funcionário não pode não trabalhar em um departamento (mínimo 1)
- Um funcionário pode trabalhar em no máximo 1 departamento (máximo 1)
- Um departamento pode não ter funcionários trabalhando (mínimo 0)
- Um departamento pode possuir diversos funcionários trabalhando (máximo n)

Cardinalidade dos relacionamentos

1:1
um-para-um



1:n
um-para-muitos

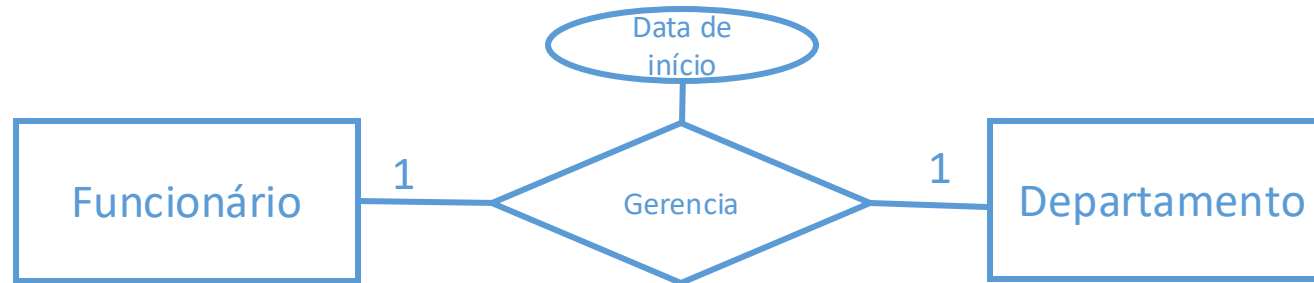


n:n
muitos-para-muitos



Atributo de relacionamento

- Relacionamentos também podem ter atributos
 - Os atributos dos tipos de relacionamento **1:1** ou **1:n** podem ser migrados para um dos tipos de entidade participantes
- Relacionamento 1:1

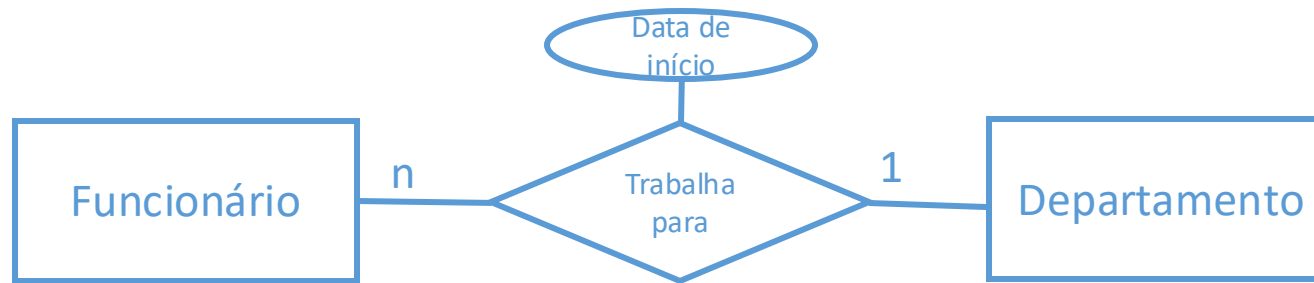


Data de início pode ser atributo tanto de FUNCIONÁRIO quando de DEPARTAMENTO (1:1)



Atributo de relacionamento

- Os atributos dos tipos de relacionamento **1:1** ou **1:n** podem ser migrados para um dos tipos de entidade participantes
- Relacionamento 1:n
 - O atributo do relacionamento pode ser migrado apenas para a entidade do lado n

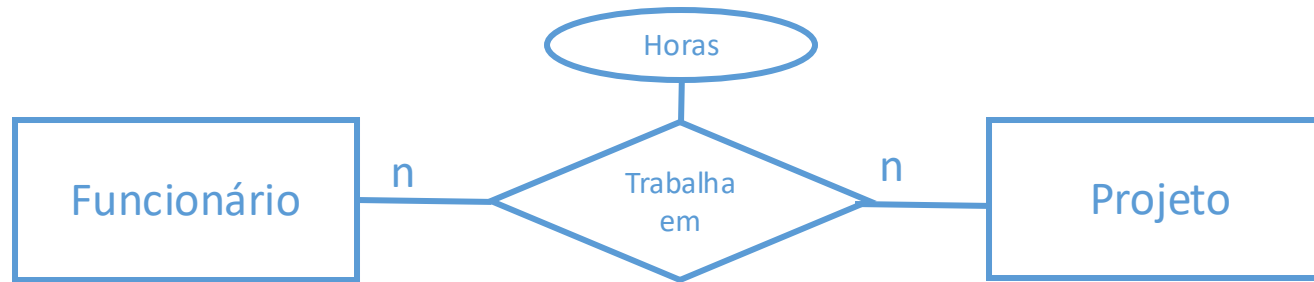


Data de início poderá ser atributo de FUNCIONÁRIO (1:n)



Atributo de relacionamento

- Para tipos de relacionamento **n:n**, alguns atributos podem ser determinados pela combinação de entidades participantes em uma instância de relacionamento, e não por qualquer entidade isolada
- **Esses atributos precisam ser especificados como atributos de relacionamento**

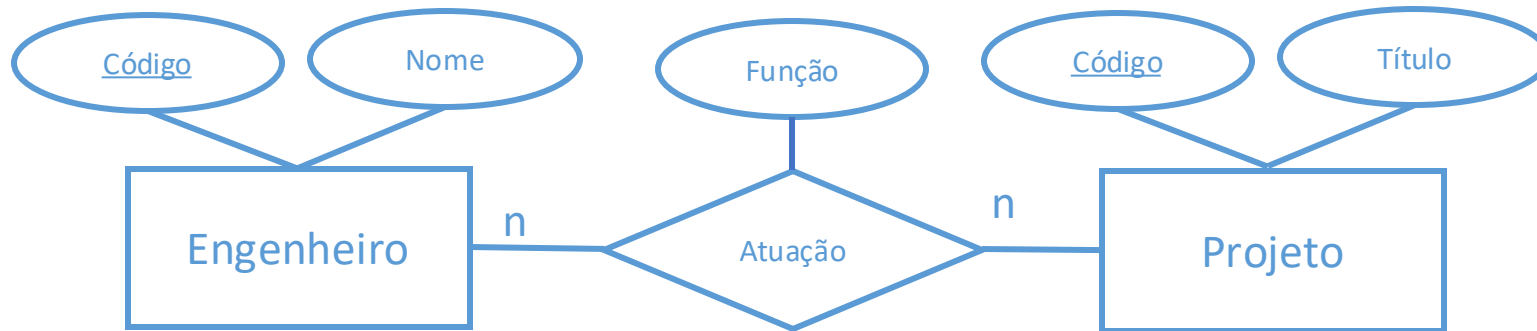
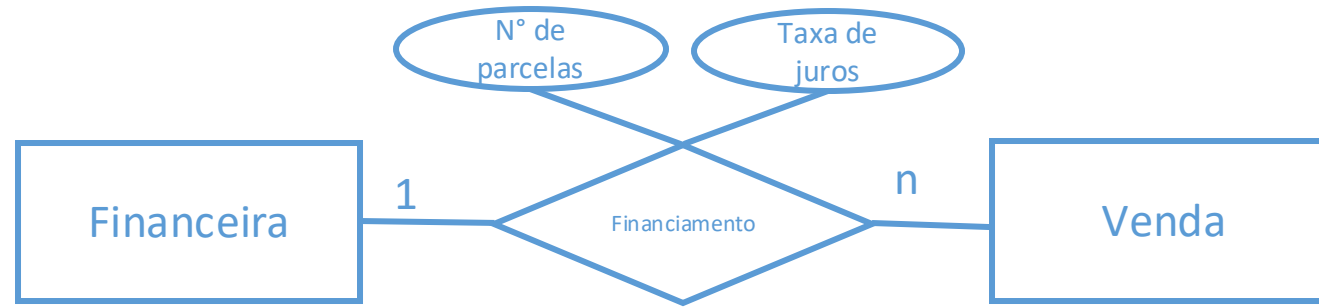


Horas é obrigatoriamente atributo do relacionamento (n:n)

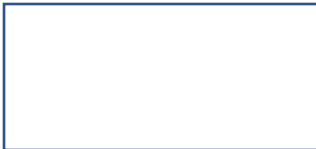




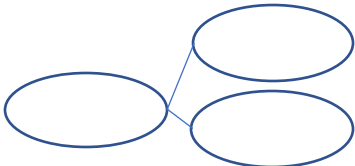


Atributo de relacionamento

- Exemplos

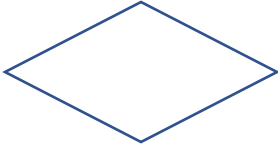

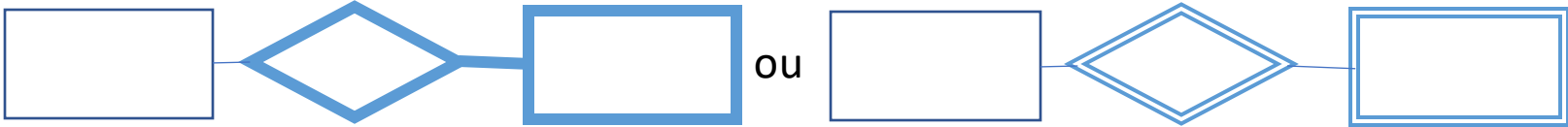


Resumo - notação

Conceito	Símbolo
Entidade	
Atributo	
Atributo identificador	
Atributo derivado	
Atributo multivalorado	
Atributo composto	



Resumo - notação

Conceito	Símbolo
Relacionamento	
Razão de cardinalidade 1:n de $E_1:E_2$ em R	
Relacionamento identificado / entidade fraca	
Entidade associativa	