

# Modelos de Armazenamento de dados

Prof. Guilherme Tomaschewski Netto  
guilherme.netto@gmail.com



# Legendas

- Nesta apresentação serão utilizadas algumas legendas:



Indica uma referência, para quem ficou curioso e quer aprofundar mais seus conhecimentos sobre o assunto



Indica uma referência importante, leitura obrigatória.

# Roteiro

- **Modelo de Banco de Dados.**
- **Construindo um Modelo de Banco de Dados.**
- **Modelagem de Banco de Dados.**
- **Projetando um banco de dados.**
- **Modelo Conceitual e Modelo Lógico.**
- **Modelo Relacional**

# Última encontro

- Conceitos de tipos de dados
- Persistência de dados

# Competências desejadas

Para compreensão dos conceitos abordados é desejado que os alunos já tenham apropriado as seguintes competências:

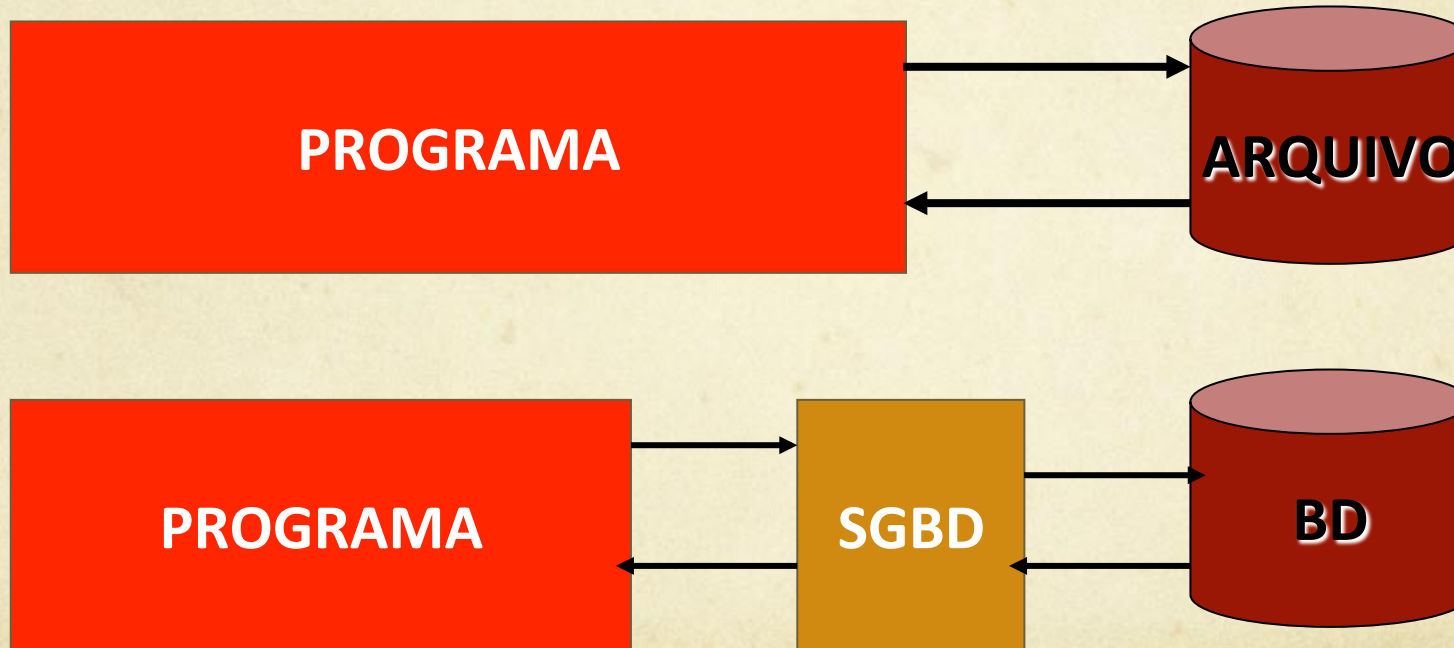
- Conhecimentos sobre algoritmos e programação de computadores
- Conhecimentos sobre as principais estruturas de dados

# Definindo o Modelo de Banco de Dados?

- Conhecido como a descrição dos tipos de informações que estão armazenadas em um banco de dados.
- Pode ser definido também como modelo de dados, descrevendo de maneira formal a estrutura de um banco de dados.

# Evoluindo em direção ao SGBD

**PROGRAMA COM DADOS  
ARMAZENADOS**

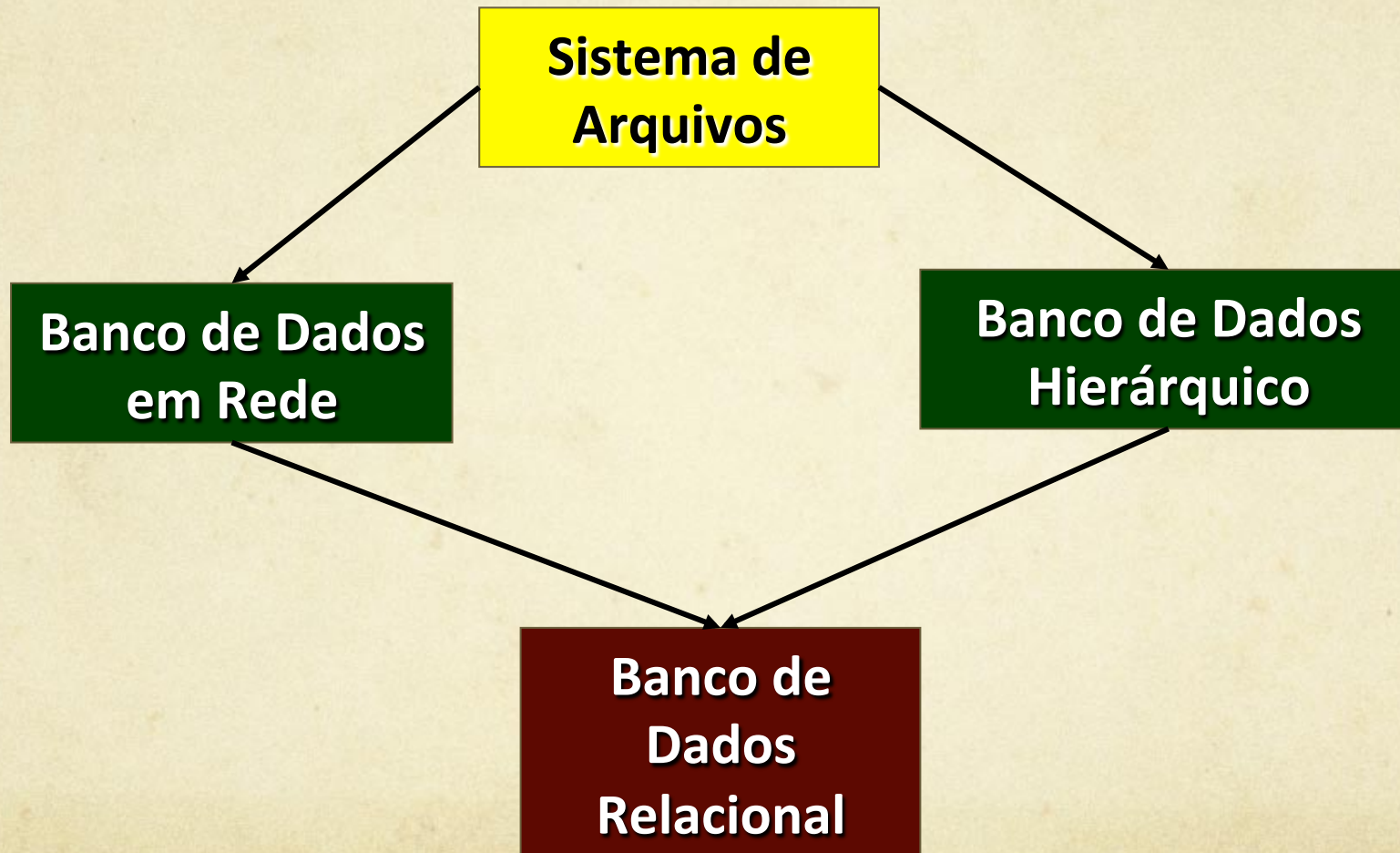




# Evoluindo em direção ao SGBD

- **Até 1960:** Sistema de Arquivos integrados ISAM, VSAM;
- **Final da década de 60:** Modelo Hierárquico IMS(IBM);
- **Década de 70:** Modelo de Redes (CODASYL) IDMS, DMS-II(Unisys);
- **Meados da década de 80:** Modelo Relacional (Codd) DB-2, SQL-DS (IBM), Oracle, Ingres;
- **Final da década de 80:** Modelo Orientado a Objetos e Relacional Estendido (Objeto-Relacional) BDOO: Vbase, O2, Orion, Gemstone, Jasmine, ObjectStore BDRE: Postgres, Illustra/Informix Universal Server, Oracle 8i, IBM DB2 Universal Server;
- **Década de 90:** BD Inteligentes, Espaciais e Temporais;

# Evolução - Modelo de Banco de Dados



# Desvantagens do Sistema de Arquivos

- **Redundância de dados** - Arquivos e programas são criados por diferentes programadores,
- **Inconsistência** - uma mesma informação, quando duplicada em diversos arquivos, eleva o custo de armazenamento e compromete a sua consistência;
- **Dificuldade do acesso a dados** - não permite acesso e manipulação dos dados de maneira eficaz;
- **Carência de padronização** - dados são espalhados em vários arquivos de diferentes formatos.
- **Carência de controle de segurança** – resulta no acesso indevido aos dados armazenados.

# Modelo de Banco de Dados

- **Modelo de dados** – uma coleção de conceitos que são usados para descrever a estrutura de um banco de dados, isto é, dados, relacionamentos, semântica e restrições;
- **Esquema** – Descrição de um BD;
- **Metadados** – Conjunto de esquemas e restrições de um BD;
- **Instância** - É o conjunto de dados armazenados no BD em um determinado instante (base de dados).

# **Construindo um Modelo de Banco de Dados**

# Como construir o Modelo de Banco de Dados?

- A construção de um modelo de dados, usa-se uma *linguagem de modelagem de dados*.
- Linguagens de modelagem de dados podem ser classificadas de acordo com a forma de apresentar modelos, em linguagens *textuais* ou linguagens *gráficas*.

# Modelagem de Banco de Dados

- A modelagem de banco de dados, consiste em apresentar como está internamente desenhado o banco de dados, facilitando na compreensão e entendimento para qualquer usuário.
- Esta modelagem de banco de dados realizada através da linguagem de banco de dados, denominada *esquema de banco de dados*.

# Projetando um banco de dados

O projeto de um novo banco de dados é dividido em três fases diferentes, fazendo uso do:

- modelo conceitual;
- modelo lógico; e
- modelo físico.



# Modelo Conceitual e Modelo Lógico



MILLER 2009, Capítulo 3

# Modelo Conceitual

- Representa a descrição de um banco de dados de forma independente de implementação em um SGBD, este modelo registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD.
- Pode ser definido também como modelo semântico.
  - Exemplos: **Modelo E-R** , diagrama de classes UML.

# Modelo Lógico

- Representa a descrição de um banco de dados no nível de abstração visto pelo usuário de SGBD, sendo totalmente dependente do SGBD que está utilizado. Sua principal finalidade é a representação da estrutura de um banco de dados conforme visto pelo usuário.
- Pode ser definido também como modelo de implementação, possuindo um nível de abstração inferior ao modelo conceitual.
- Podem ser classificados como:
  - modelos lógicos baseados em objetos;
  - modelos lógicos baseados em registros.

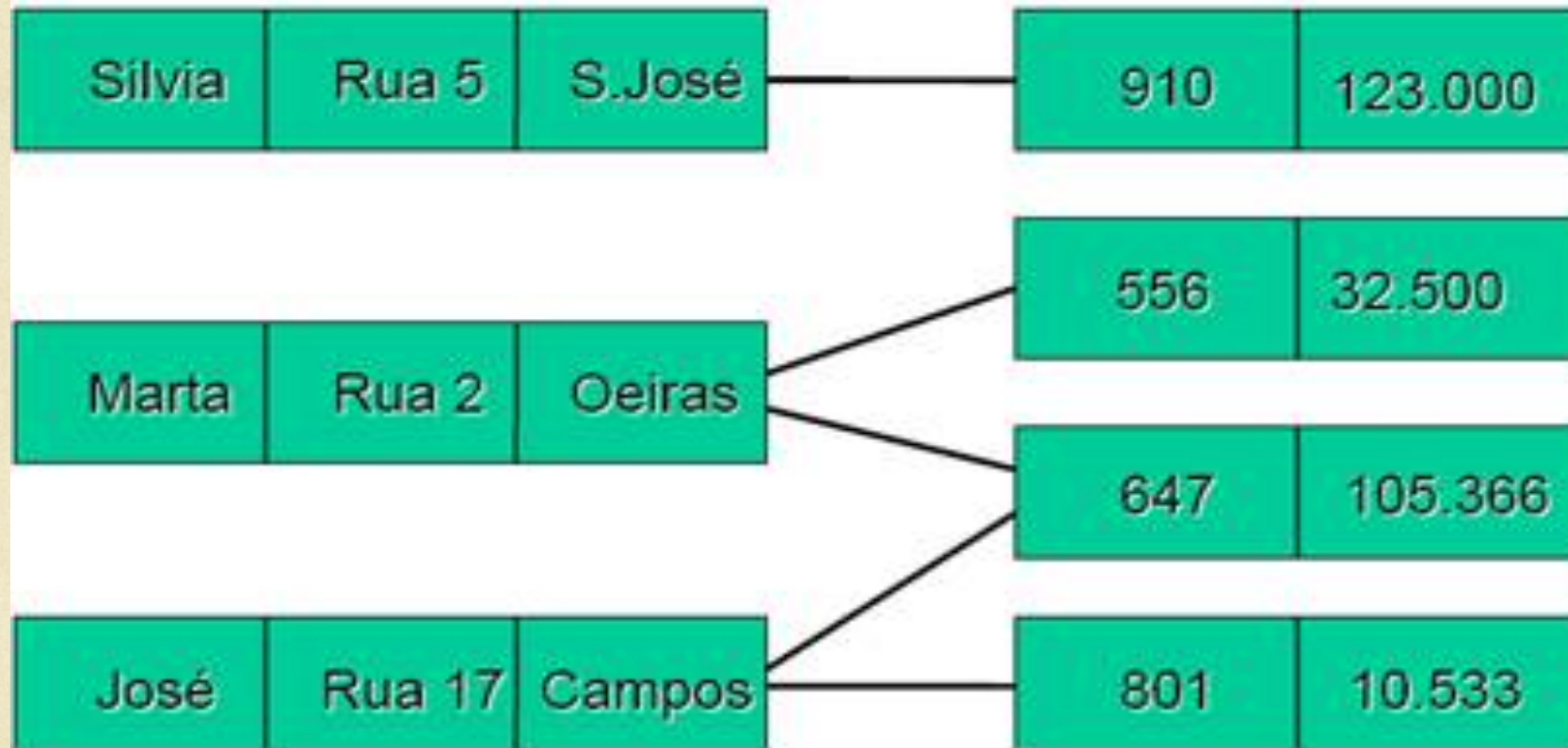
# Diferentes Modelos Lógicos

- **Modelo de rede** – Baseado em links de conexão;
- **Modelo hierárquico** – Baseado na estrutura de árvore;
- **Modelo Relacional** - Introduzido por Codd em 1970. A estrutura de dados é uma relação Modelo baseado em listas invertidas.
  - Exemplo: ADABAS
- **Modelos OO** – Exemplo: O2 e Jasmine;
- **Modelos Objeto-Relacional** - Exemplo: Oracle 8i.

# Modelo de Rede

- Coleção de registros conectados uns aos outros por meio de *links*(ponteiros).
- Fortemente dependente da implementação. Muitas vezes é necessário criar registros artificiais para implementar relacionamentos.
- Registros no BD são organizados como coleções arbitrárias de grafos, seu esquema pode ser representado por um diagrama de estrutura de dados constituído por caixas e linhas.

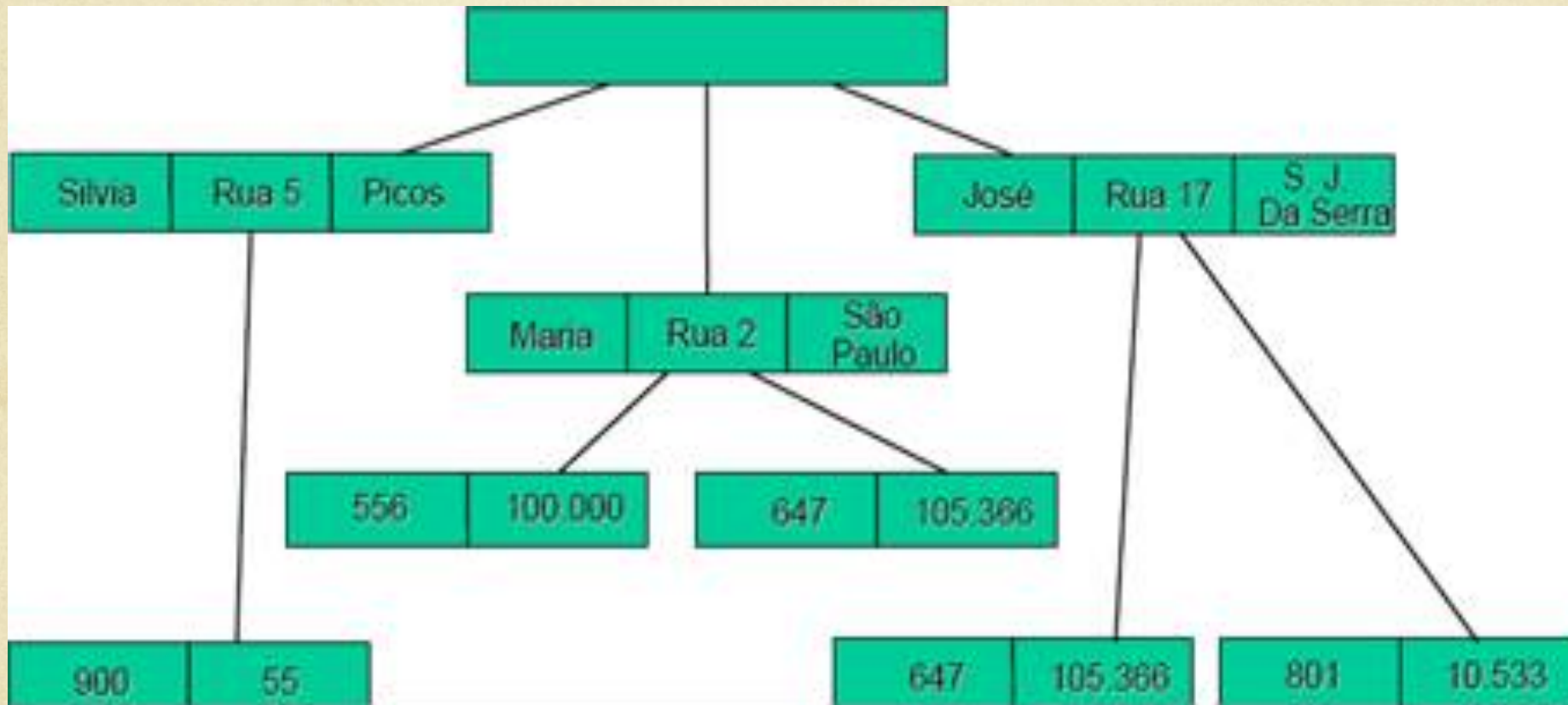
# Representação Gráfica Modelo de Rede



# Modelo Hierárquico

- Todos os registros no modelo Hierárquico são organizados na forma de uma árvore, este modelo sofre dos mesmos problemas do modelo de rede.
- Por muito tempo, os modelos hierárquico e de rede se mantiveram à frente do modelo relacional, atualmente estes modelos vêm perdendo a importância.
- Diferentemente do modelo relacional, os modelo hierárquico e de redes dependem de mecanismos externos de estruturação de dados.
- Os registros são organizados como coleções arbitrárias de árvores em vez de grafos.

# Representação Gráfica Modelo Hierárquico





# Modelo Relacional

- Dados e suas associações são representados por uma coleção de tabelas.
- Não requer mecanismos de estruturação para interrelacionar seus dados.
- Fortemente fundamentado na Álgebra Relacional e no cálculo relacional.
- Mais difundido dentre todos os modelos de dados.



# Representação Gráfica

## Modelo Relacional

nome	rua	cidade	Nconta
Pedro	Av. 2	Fpolis	1452
João	R. 51	S.José	521
Tiago	Largo 2	Fpolis	53256
Marcos	Av. 43	Biguaçú	53256
Arthur	Rua 23	Fpolis	14532

Nconta	saldo
1452	2452.25
521	6565.74
53256	465.52
4651	456.25
14532	8767.02

# Revisão

- **Modelo de Banco de Dados.**
- **Construindo um Modelo de Banco de Dados.**
- **Modelagem de Banco de Dados.**
- **Projetando um banco de dados.**
- **Modelo Conceitual e Modelo Lógico.**
- **Modelos Lógicos:**
  - **Rede, Hierárquico e Relacional.**

# Modelo Entidade-Relacionamento

Módulo desenvolvido por Chen em 1976

Provê ao usuário um alto nível de abstração, e por conseguinte facilita a construção de um esquema de BD

A estrutura lógica do BD pode ser expressa graficamente pelo diagrama E-R

Popular - Simplicidade e Expressividade

Um banco de dados representado por um modelo E-R, pode ser representado por uma coleção de tabelas

O mapeamento entre os modelos E-R e Relacional é relativamente simples.

Existem várias ferramentas destinadas a mapear o Modelo E-R para Relacional

É também chamado de esquema E-R ou diagrama E-R

# Modelo Entidade-Relacionamento

Diversas extensões e notações foram definidas ao longo do tempo

Os dados do mundo real são representados por meio de *conjuntos de entidades*, *relacionamentos* entre esses e *atributos* que os caracterizam

*Conjunto de Entidades*: representa um conjunto de elementos do mundo real que têm a mesma “estrutura” e o mesmo “significado”

- Estrutural e semanticamente iguais
- Ex: Pessoas

*Entidade*: elemento do *conjunto de entidades* identificado por características individuais definidas por meio do conceito de *atributos*

- Ex: “coisas”, objetos, pessoas (Murilo)

*Atributos*: Propriedades que descrevem a *entidade* ou o *relacionamento* entre *entidades*

*Relacionamento*: conjunto de associações entre *conjunto de entidades*; podem ser caracterizados por *atributos*

# Conjunto de Entidades

As instâncias de uma entidade não são representadas no diagrama de Entidades e Relacionamento, mas são semanticamente interpretadas no mesmo

- MER não trata Entidades individuais, apenas Conjuntos de Entidades
  - Notação DER: retângulo

Funcionário

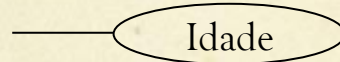
Departamento

# Atributos

Valores que representam propriedades das entidades e relacionamentos no mundo real

➤ Tipos:

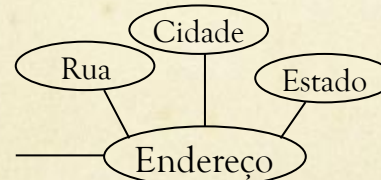
- **Atributo Monovalorado:** possui um valor para cada *entidade* que caracteriza



- **Atributo Multivalorado:** possui mais de um valor para cada *entidade* que caracteriza



- **Atributo Composto:** quando o *atributo* tem vários subcampos



- **Atributo Derivado:** quando o valor de um *atributo* é obtido por meio de valores de outros *atributos*. Ex: *tempo\_de\_casa* pode ser derivado do valor da *data\_contratação*



# Atributos

- **Chave** = *Atributo* ou um conjunto de *atributos* que, com seus valores, consiga identificar uma única *entidade* dentro do *conjunto de entidades*
- Uma chave deve ser **mínima** no sentido de que se a chave for composta, nenhum *atributo* que a compõe poderá ser retirado, e ainda sim, a composição resultante continuar sendo chave
- É o principal meio de acesso a uma entidade
- Outras possíveis chaves não são indicadas no diagrama, e não são contempladas pelo MER, mas podem ser anotadas separadamente, para efeito de documentação
- **Chave Composta:** mais de um atributo compõe a chave de um conjunto de entidades. A concatenação de todos eles indica a chave única
- Notação DER: grifar atributo chave





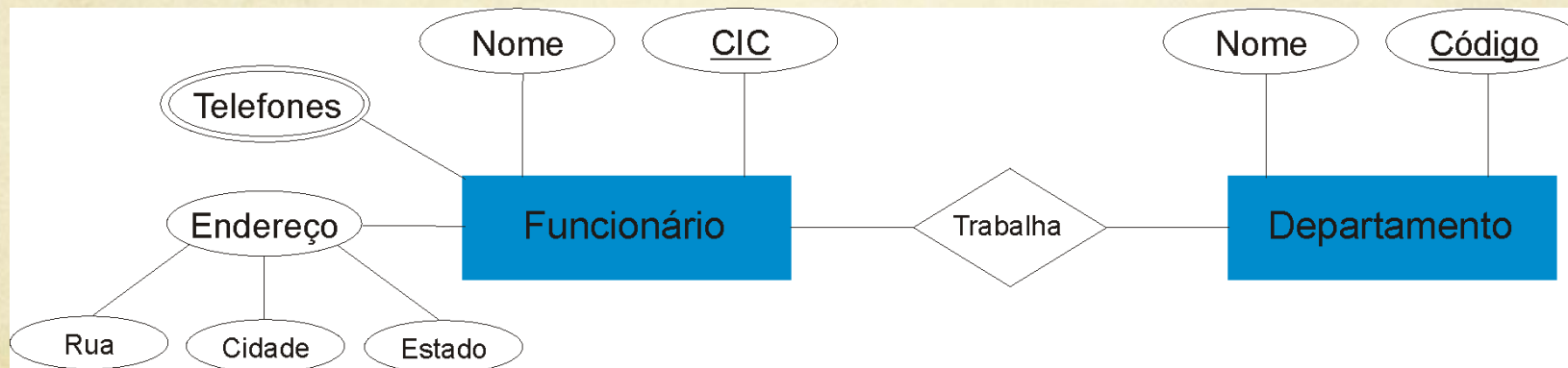
# Conjunto de Relacionamentos

- **Relacionamento** é uma associação entre uma ou várias entidades
- **Conjunto de Relacionamentos** é um conjunto de relacionamentos de mesmo tipo
- Expressam uma rica semântica entre os *conjuntos de entidades* por meio dos conceitos como:
  - **Cardinalidade**
  - **Restrição de participação** (total ou parcial)
  - **Grau de Relacionamento**
- Esses conceitos impõem restrições aos dados que alimentarão o banco de dados
- Notação DER: losango



# Conjunto de Relacionamentos

- O grau de relacionamento: é o número de entidades participantes
  - Binário, ternário, etc.
- Dependendo do conjunto de entidades associadas entre si, é necessário adicionar atributos em um relacionamento
  - Ex: Horas em Trabalha\_Em entre Funcionário e Projeto
- Exemplo MER:



# Cardinalidade

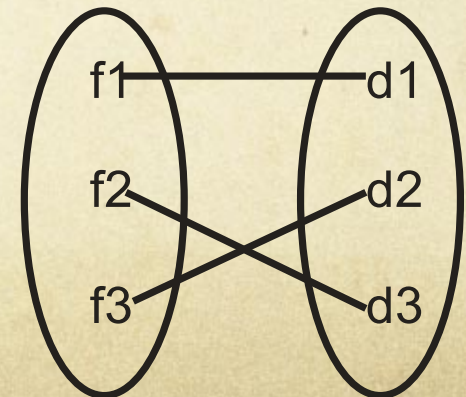
A cardinalidade expressa o número de entidades as quais outra entidade pode estar associada em um relacionamento

- Um para um (1 para 1)
  - Uma entidade em *A* está associada no máximo a uma entidade em *B*, e uma entidade em *B* está associada a no máximo uma entidade em *A*
- Um para muitos (1 para N)
  - Uma entidade em *A* está associada a várias entidades em *B*. Uma entidade em *B*, entretanto, deve estar associada no máximo a uma entidade em *A*
- Muitos para muitos (N para N) - pode ser substituído por qualquer outra letra, como M, P, Q)
  - Uma entidade em *A* está associada a qualquer número de entidades em *B* e uma entidade em *B* está associada a um número qualquer de entidades em *A*

# Cardinalidade 1 para 1

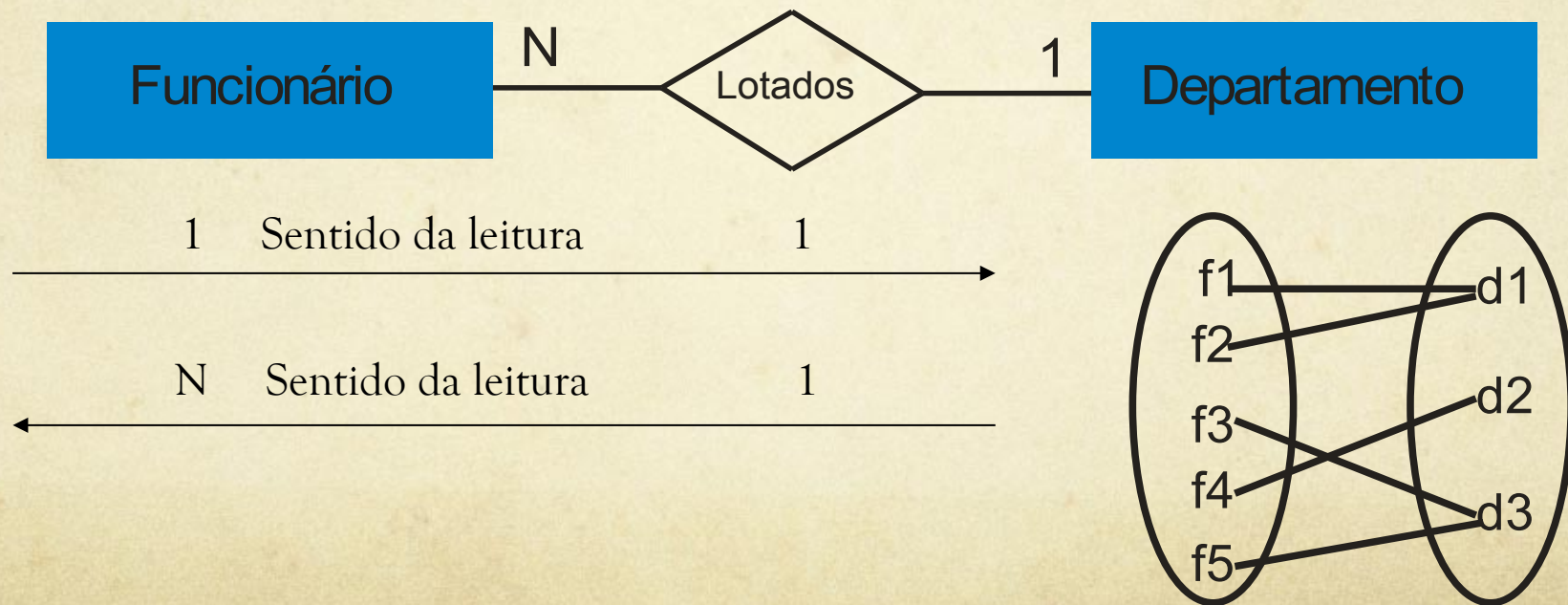
## ➤ Exemplos:

- Um funcionário gerencia no máximo 1 departamento. Um departamento é gerenciado por no máximo um funcionário



# Cardinalidade 1 para muitos

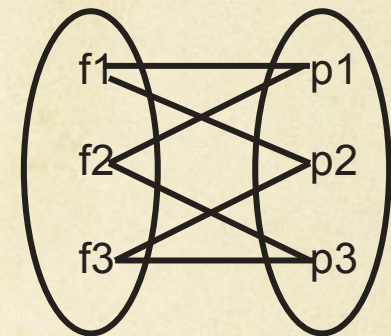
- Exemplos:
  - Um funcionário está lotado no máximo em 1 departamento.  
Um departamento tem até N funcionários lotados nele



# Cardinalidade muitos para muitos

➤ Exemplos:

- Um funcionário participa de vários projetos. Um projeto pode ter a participação de até N funcionários



# Cardinalidade máxima e mínima

- Indica se a participação das ocorrências de entidades no relacionamento é obrigatória ou opcional



*Um funcionário pode estar lotado no máximo em 1 departamento. Um departamento obrigatoriamente tem até N empregados lotados nele*

# Cardinalidade máxima e mínima



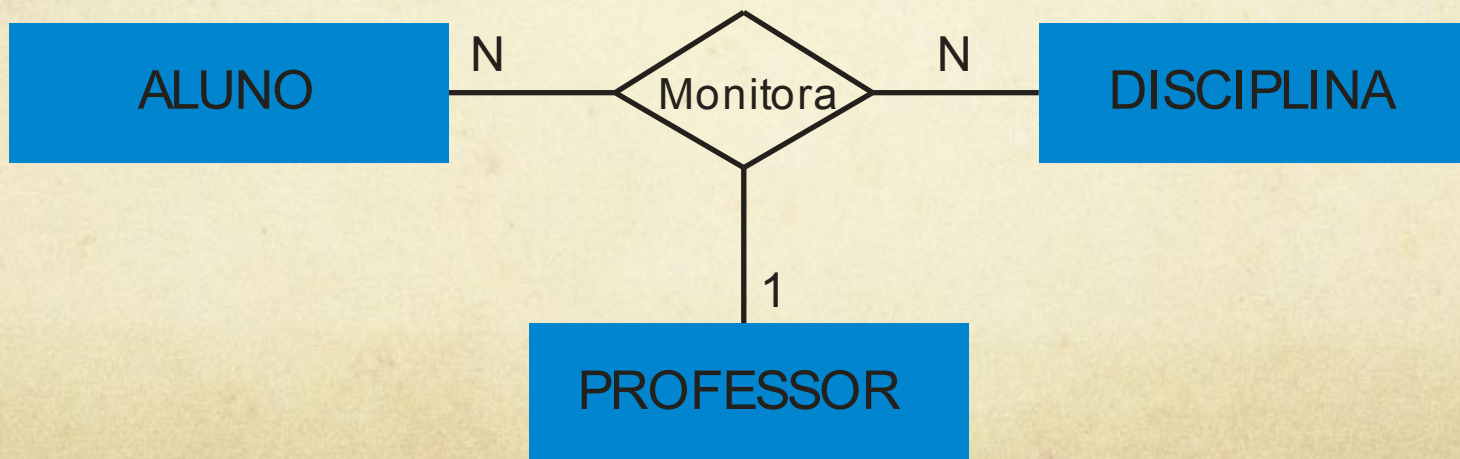


# Grau de Relacionamento

- Indica quantos *conjuntos de entidades* estão envolvidos em determinado *relacionamento*
- Os *relacionamentos* podem ter associado vários *conjuntos de entidades*, caracterizando:
  - relacionamentos binários (grau de relacionamento 2)
  - relacionamentos ternários (grau 3)
  - relacionamentos quaternários (grau 4), entre outros
- É importante observar que um *relacionamento* com grau  $N > 2$  só é justificável se não puder ser decomposto em *relacionamentos* com graus menores e ainda manter a semântica desejada

# Relacionamento Ternário

- Os relacionamentos entre múltiplas entidades expressam um fato em que todas as entidades ocorrem simultaneamente, ou seja, todas as ocorrências do relacionamento possuem, sempre, ligações com todas as entidades envolvidas no relacionamento
- Não pode existir de um relacionamento triplo, em um determinado momento, se transformar em duplo

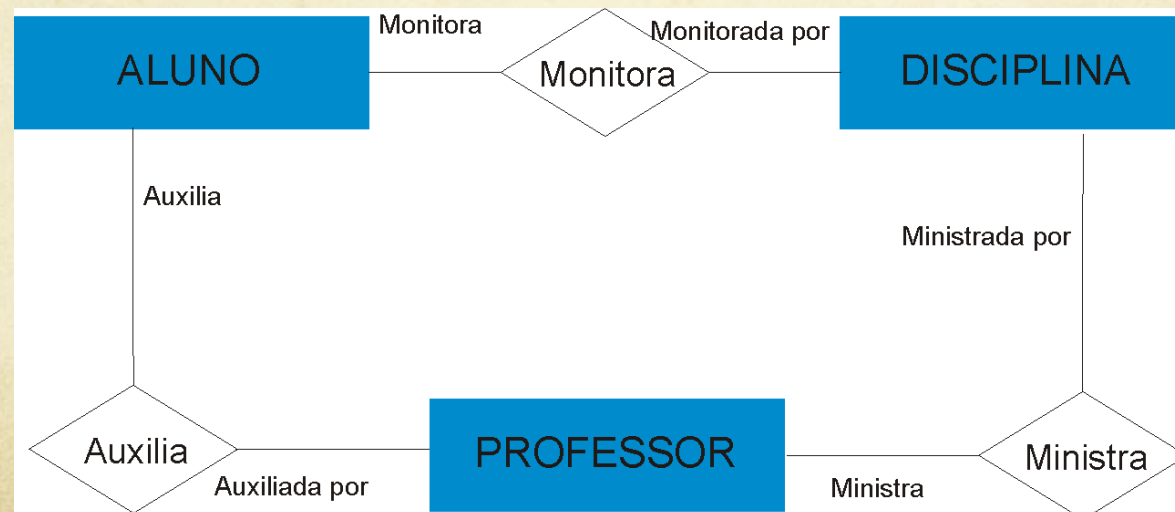


# Relacionamento Ternário

- Para descobrir a cardinalidade do relacionamento ternário, deve-se proceder da seguinte forma:
  - Separar a entidade ALUNO e analisar o par PROFESSOR, DISCIPLINA. Para cada par PROFESSOR / DISCIPLINA podemos ter de 1 até N ALUNOS relacionados
  - Separar a entidade PROFESSOR e analisar o par ALUNO, DISCIPLINA. Para cada par ALUNO / DISCIPLINA podemos ter 1 e somente 1 PROFESSOR relacionado
  - Separar a entidade DISCIPLINA e analisar o par PROFESSOR, ALUNO. Para cada par PROFESSOR / ALUNO podemos ter de 1 até N DISCIPLINAS relacionadas
- Sempre que existe uma ocorrência no relacionamento , esta apresenta referência às três entidades

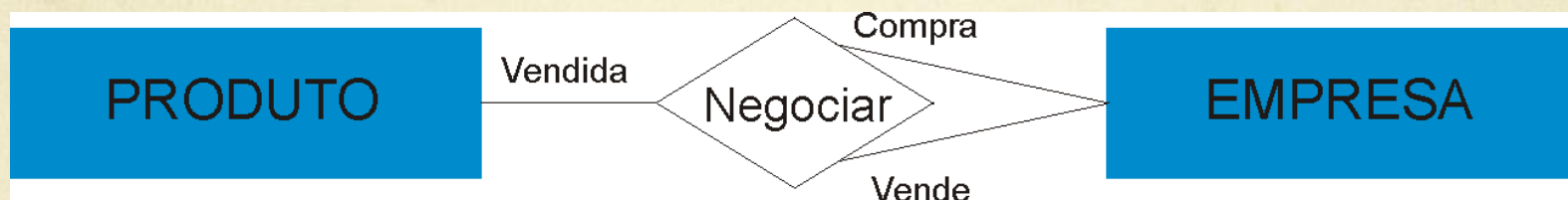
# Relacionamento Ternário

- Podemos tentar “quebrar” o relacionamento ternário em vários binários
- O problema é a perda de informações semânticas
  - A informação representada por um conjunto de relacionamentos ternário nem sempre pode ser obtida apenas com conjunto de relacionamentos binários
  - ex: como responder: Aluno A auxilia Professor P em qual Disciplina?



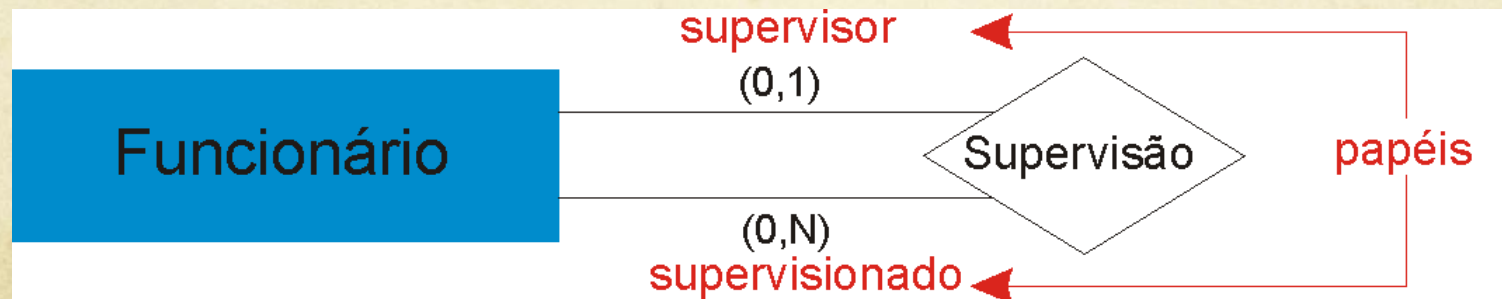
# Relacionamento Ternário

- Um mesmo Conjunto de Entidades pode desempenhar vários papéis num Conjunto de Relacionamentos
- Uma Empresa (vendedora) negocia Produtos com outra Empresa (compradora)



# Auto-Relacionamento

- Representa uma associação entre ocorrências de uma mesma entidade
- Requer a identificação de papéis
  - **Papel** do *relacionamento*: utilizado quando não é clara a participação de um determinado *conjunto de entidades* em um *relacionamento*
  - A representação do papel é extremamente importante quando se trata de um **relacionamento unário** ou *auto-relacionamento*
- Ex: “Um Funcionário pode ser supervisionado por no máximo 1 Funcionário. Um Funcionário pode supervisionar no máximo N Funcionários.”



# Entidade Fraca

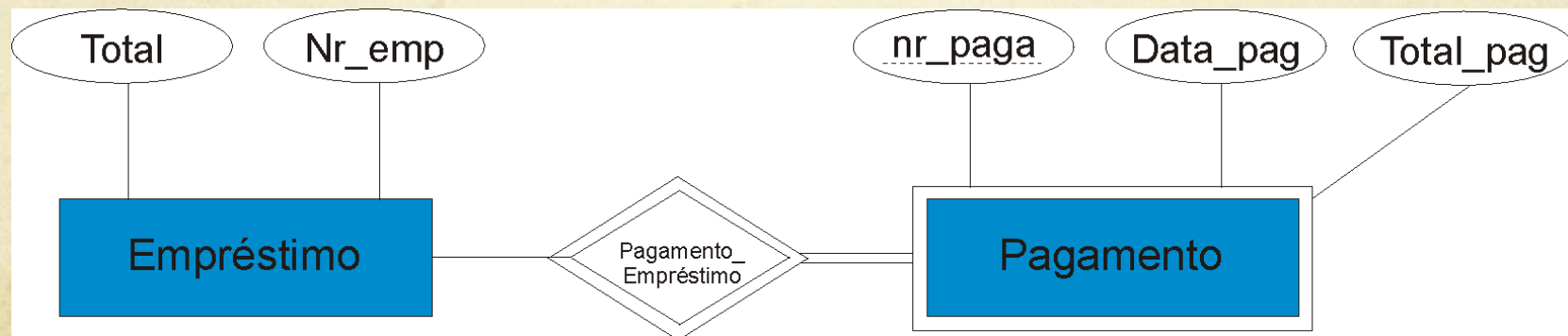
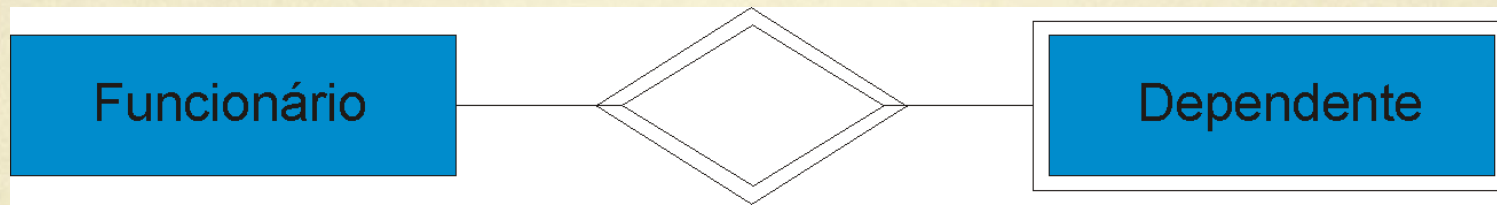
- Conjunto de entidades que não possui identificação própria
- Entidade que não tem atributos que possam identificá-la univocamente
  - sua identificação depende de um relacionamento com uma entidade de outro conjunto
  - O identificador de um conjunto de entidades fracas é também chamado de **chave parcial** de um conjunto de entidades
  - A chave primária de um conjunto de entidades fracas é formada pela chave primária do conjunto de entidades fortes vinculada mais o identificador do conjunto de entidades fracas

Ex:

- Num *relacionamento* entre as *entidades* Funcionário e Dependente, o Dependente só existe se houver um Funcionário a ele associado
- Notação DER:
  - linha dupla no retângulo e no losango do relacionamento
  - sublinhar com linha tracejada a chave parcial

# Entidade Fraca

## ➤ Exemplos:



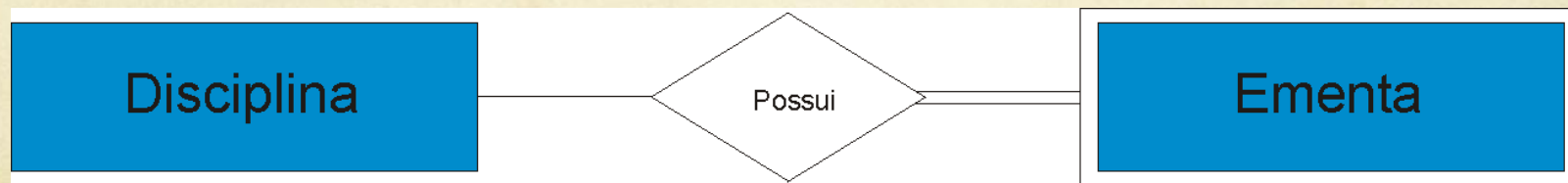


# Restrição de Participação

- Mostra se a existência de uma *entidade* depende de outra *entidade* por meio de um *relacionamento*
- Há dois tipos de restrição de participação:
  - **participação total:** toda *entidade* em um *conjunto de entidades* precisa ser relacionada com a *entidade* correspondente no *relacionamento* em questão
  - **participação parcial:** somente parte do *conjunto de entidades* é relacionada à *entidade* correspondente no *relacionamento*

# Restrição de Participação

- Participação total ou Dependência Existencial:
  - Uma entidade só existe se estiver associada a outra entidade por meio de um relacionamento
  - Ex: uma entidade *Ementa* tem que participar de um relacionamento *Possui*, ou seja, deve estar associada a uma entidade *Disciplina*



## Reconhecendo Entidades / Relacionamentos

- **Lista de perguntas úteis para identificar entidades em um contexto:**
  - Que coisas são trabalhadas?
  - O que pode ser identificado por número, código?
  - Essa coisa tem atributos? Esses atributos são relevantes, pertinentes?
  - Essa coisa pode assumir forma de uma tabela?
  - É um documento externo (recibo, fatura, nota fiscal)? Se sim, é forte candidato à entidade.
  - Tem significado próprio?
  - Qual a entidade principal do contexto?
- **Dicas:**
  - Substantivos que não possuem atributos podem ser atributos de outras entidades
  - Adjetivos colocados pelos usuários indicam normalmente atributos de uma entidade

## Reconhecendo Entidades / Relacionamentos

### ➤ **Dicas: (cont)**

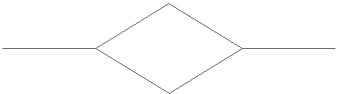

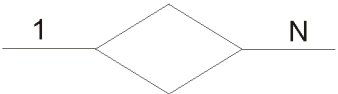

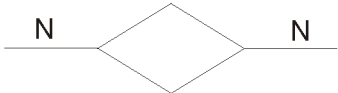

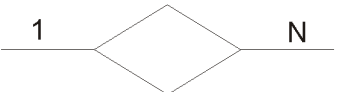

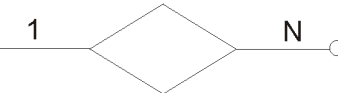

- Verbos indicam prováveis relacionamentos
- Advérbios temporais indicam prováveis atributos de um relacionamento
- Procure sempre visualizar a entidade principal do contexto sob análise

### ➤ **Dicas para reconhecer e inserir relacionamentos no modelo:**

- O relacionamento é necessário?
- Ele é útil?
- É redundante? Se redundante, retirar?
- Qual sua finalidade? (Documentar)
- Verbos indicam possíveis relacionamentos
- Analisar sempre as entidades aos pares

## Notação James Martin

- Notação de Peter Chen é interessante e bastante expressiva, porém para grandes modelos torna sensivelmente confuso, com muitos cruzamentos e complicada de ser lido
- As ferramentas Case utilizam a notação da Engenharia de informações ou notação de James Martin

Conectividade	Peter Chen	James Martin
1:1		
1:N		
N:N		
Existência		
Obrigatório		
Opcional		

## Notação James Martin



Exemplo de Modelagem, onde:

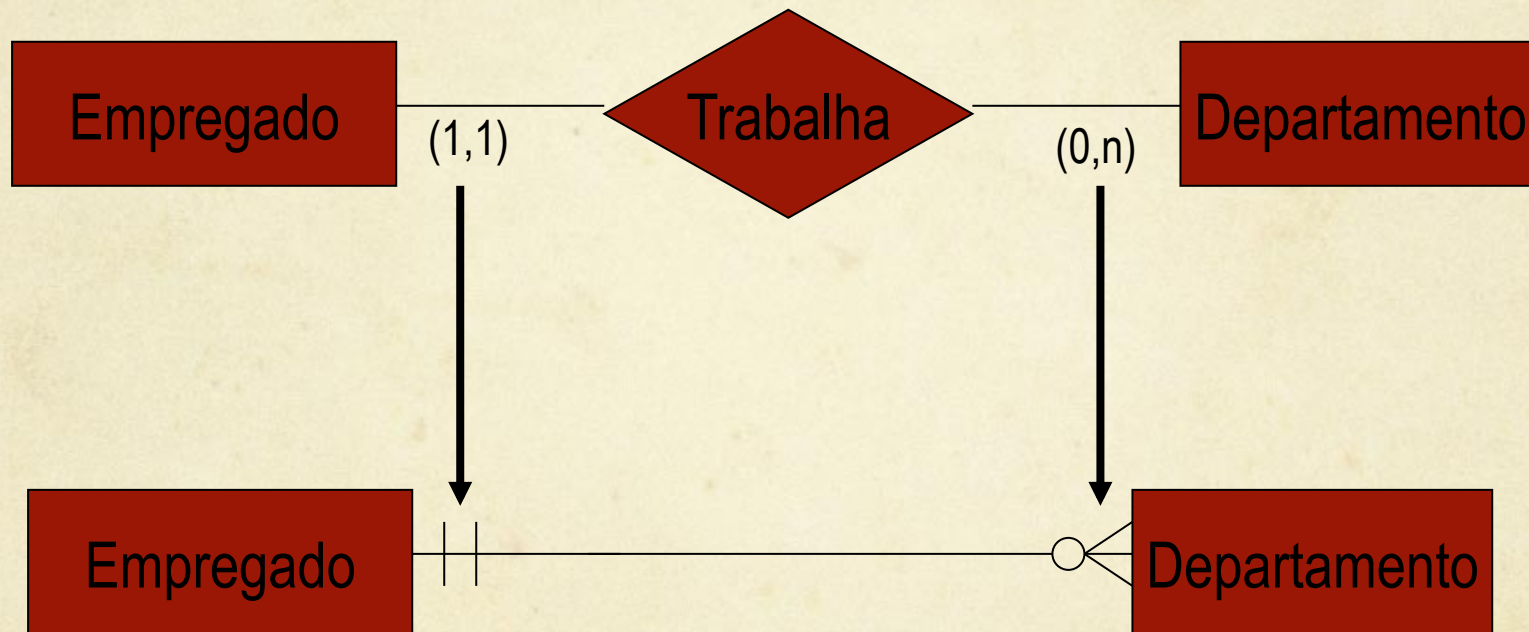
≠ = muitos

| = um

○ = a ocorrência do relacionamento é opcional;

| = a ocorrência do relacionamento é obrigatória;

## Notação James Martin



## Notação James Martin

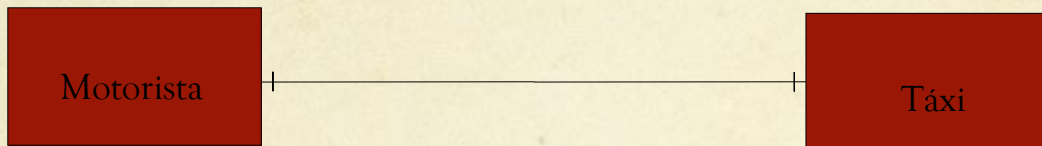
As principais diferenças entre as notações:

- Os relacionamentos são representados apenas por uma linha que une as duas entidades;
- Somente relacionamentos binários;
- A notação de cardinalidade máxima e mínima é gráfica, sendo assim: o símbolo mais próximo ao retângulo é a representação da cardinalidade máxima e o mais distante a cardinalidade mínima.



# Notação James Martin

## Associações de “Um para Um”



## Associações de “Um para Muitos”



## Associações de “Muitos para Muitos”



## Exercício

- Uma empresa é organizada em departamentos, cada um com um nome único, uma sigla e um funcionário responsável por gerenciá-lo. Uma data determina quando o funcionário iniciou suas atividades de gerência no departamento.
- Um departamento da empresa controla vários projetos, cada um com um código único e um nome.
- Um funcionário da empresa está vinculado a um departamento, mas pode trabalhar em vários projetos, sendo determinado o número de horas semanais dedicadas a cada um. Além disso, todo funcionário tem um supervisor direto.
- Para cada funcionário são armazenadas informações como: nome, CPF, endereço, telefone(s) de contato, salário e dependentes (primeiroNome, idade, parentesco)

# Bibliografia

- ALVES, W.P. 2009. Banco de Dados - Teoria e Desenvolvimento. Editora Érica, 288 p.
- DATE, C. 2000. Introdução a sistemas de banco de dados. Editora Campus, Rio de Janeiro (Brasil), 803 p.
- MILLER, F. 2009. Introdução à Gerência de Banco de Dados - Manual de Projeto. Editora LTC, 228 p.
- ROB, P. & CORONEL, C. 2010. Sistemas de Banco de Dados - Projeto, Implementação a Administração. 8<sup>a</sup>. Ed., Editora Cengage Learning, 744 p.

that's all folks!

