

Introdução

Bancos de Dados

Luiz Celso Gomes-Jr
gomesjr@dainf.ct.utfpr.edu.br

Mundo de Dados



Mundo de Dados

- Curtidas em Redes Sociais
- Mensagens Instantâneas
- Páginas na Web
- Notas dos alunos
- Fotos
- Localização de Pokémons
- Sinais de televisão
- Saldo de contas correntes
- Produtos à venda

Dados X Aplicações

Curtidas



Notas dos alunos



Localização Pokémons



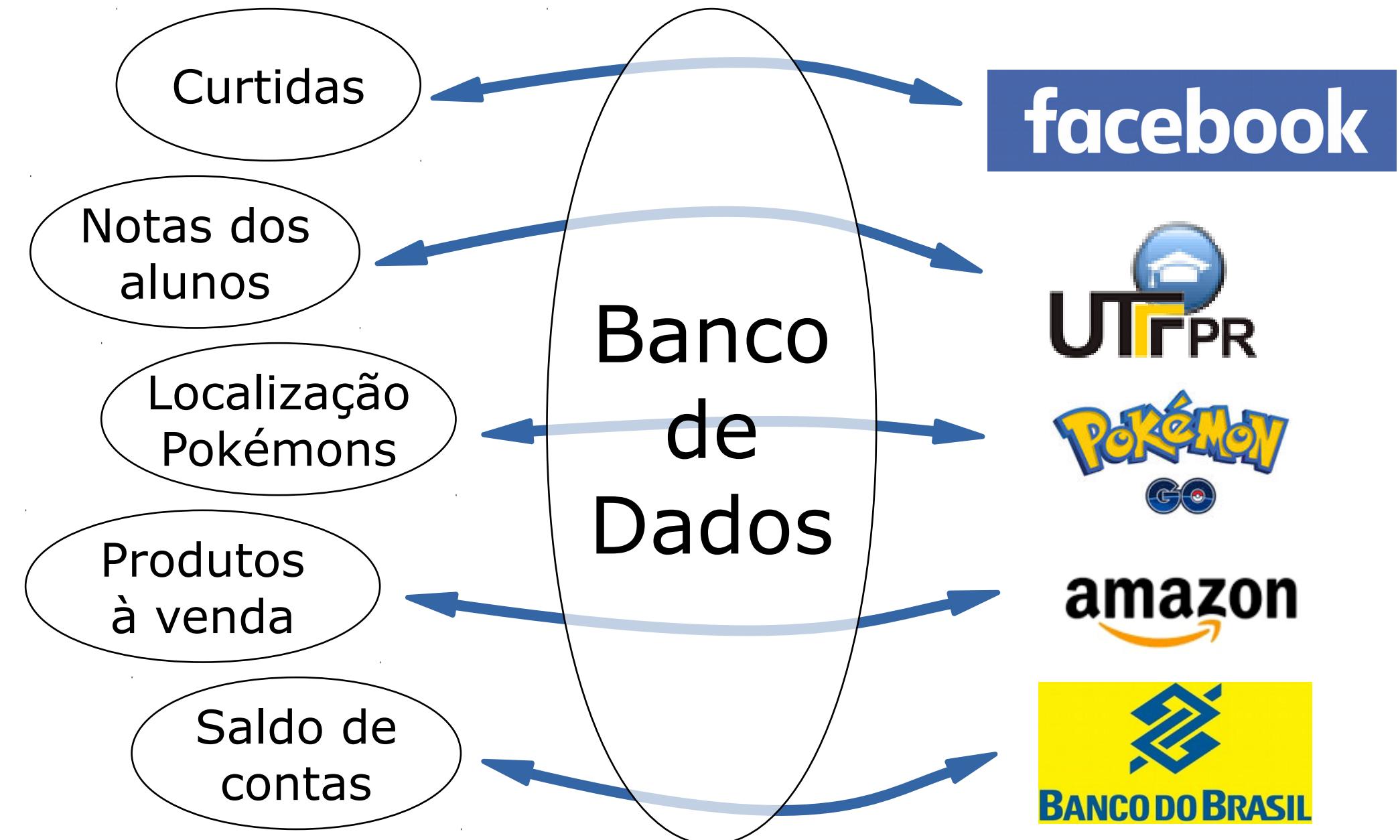
Produtos à venda



Saldo de contas



Dados X Aplicações



Dados e aplicações

- Diversas aplicações geram e lidam com uma **grande quantidade de dados**
- É importante garantir a **integridade** dos dados e disponibilizá-los da forma mais **eficiente** possível
- **Sistemas de Gerenciamento de Bancos de Dados (SGBDs)** são responsáveis pelo armazenamento, gerenciamento e disponibilização dos dados

Um Sistema de Gerenciamento de Bancos de Dados provê:

Armazenamento e acesso a uma
grande quantidade de dados
persistentes de forma **eficiente,**
confiável, conveniente, segura e
multi-usuário.

Super SGBD



Preciso implementar um sistema! Quem cuidará dos meus dados??

Eu,
o Super SGBD



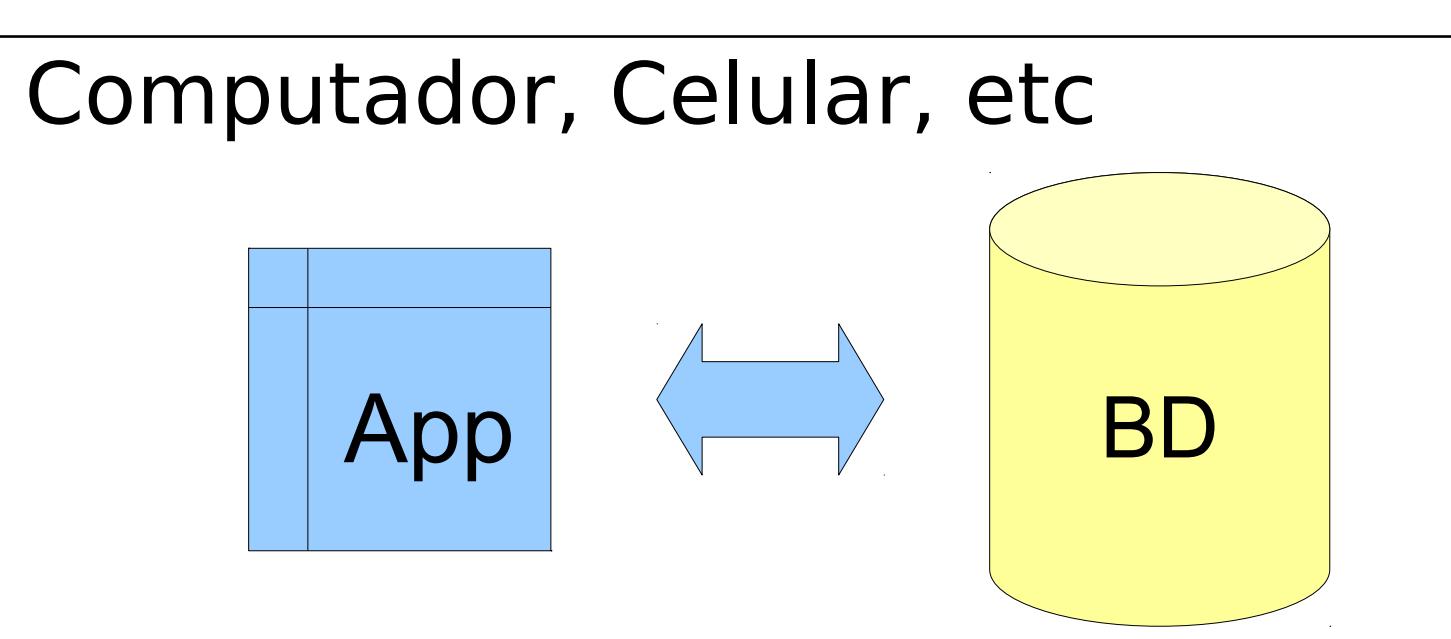
SGBDs desempenham muitas tarefas importantes que sobrecarregariam os programadores. Veremos ao longo do curso o quanto eles facilitam nosso trabalho.

Arquiteturas típicas para SGBDs

- O SGBD intermedia a manipulação dos dados para as aplicações
- Como esta intermediação é feita e quais elementos estão envolvidos depende do contexto da aplicação
- Aplicações podem demandar diversas configurações de arquiteturas locais ou distribuídas (ou híbridas)

Arquitetura local

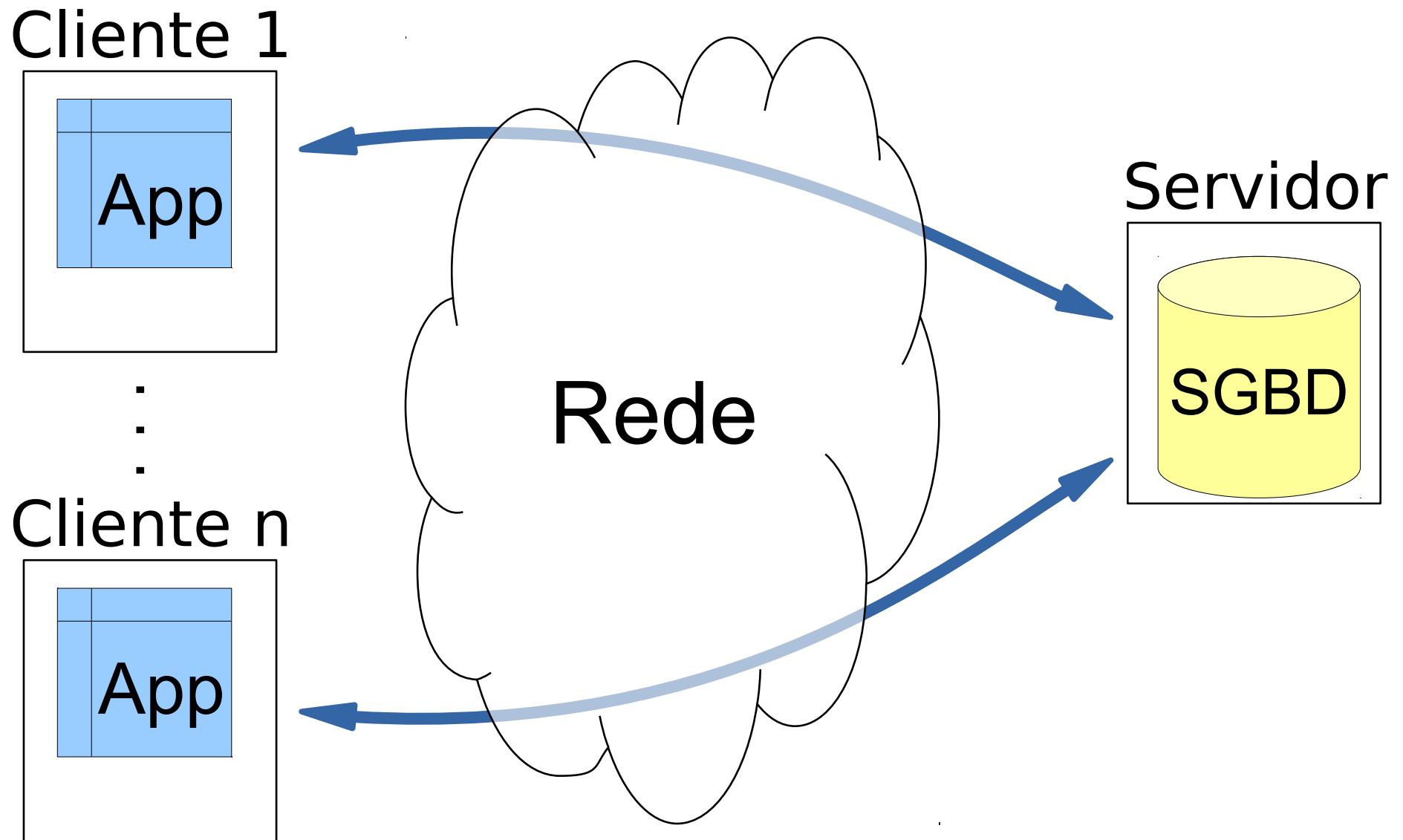
- Não costumam ser classificados como SGBDs
- O banco é um software ou biblioteca executado no mesmo dispositivo da aplicação



Arquitetura Cliente-Servidor

- Usada na maioria das instituições
- Usuário acessa a aplicação por um dispositivo Cliente (desktop, laptop, celular...)
- Aplicação envia consultas para obter dados do SGBD (Servidor)
- SGBD processa consulta e retorna dados para serem exibidos no Cliente
- Exemplos: Folha de pagamentos, iTunes

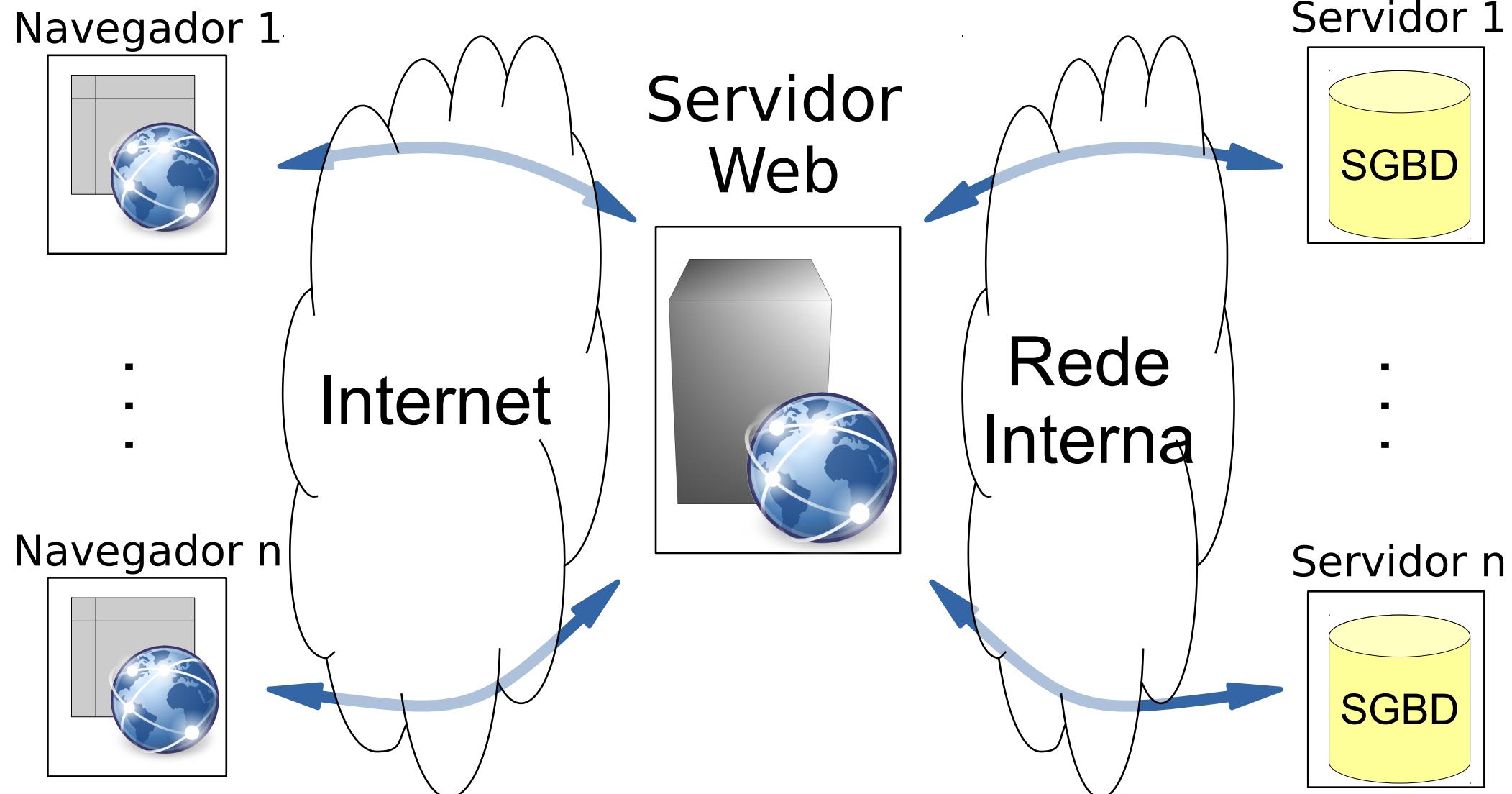
Arquitetura Cliente-Servidor



Arquitetura Web 1.0

- Usada na maioria dos sites “normais”
- Usuário usa o navegador para requisitar páginas para um Servidor Web
- Servidor Web envia consultas a um ou mais SGBDs para obter dados e montar a página
- Exemplos: bancos online, sites de empresas
- Muitas apps e sites como Facebook, Google precisam de arquiteturas mais complexas. Veremos estes casos no fim do curso.

Arquitetura Web 1.0



Exemplo: Facebook

1. Usuário abre o navegador e entra em facebook.com
2. Servidor Web do facebook recebe a requisição do usuário
3. Servidor Web do facebook obtém dados do mural de um SGBD interno
4. Servidor Web do facebook obtém dados de propaganda de um outro SGBD interno
5. Servidor Web do facebook monta a página e envia para o navegador exibir

Conceitos importantes

- **Modelo de Dados** – Como representar os dados (listas, tabelas, árvores, grafos...)
- **Esquema X Dados** – Separação entre a definição (tipo) e o conteúdo (instâncias, variáveis)
- **Linguagem de Consulta** – Como requisitar dados para o banco de forma conveniente (exemplo: como dizer para o SGBD “obtenha o salário médio de todos os funcionários que têm mais de 30 anos”)

Domadores de SGBDs

- **Implementador:** quem constrói o sistema (em geral grandes empresas como MS, IBM, Oracle)
- **Projetista:** Faz análise e cria o esquema (define os tipos ou classes)
- **Desenvolvedor** de aplicações: programador que usa o banco de dados (faz consultas, modifica dados)
- **Administrador** (DBA): Gerencia o carregamento de dados, segurança, backups, eficiência, etc.

Uma breve história dos SGBDs

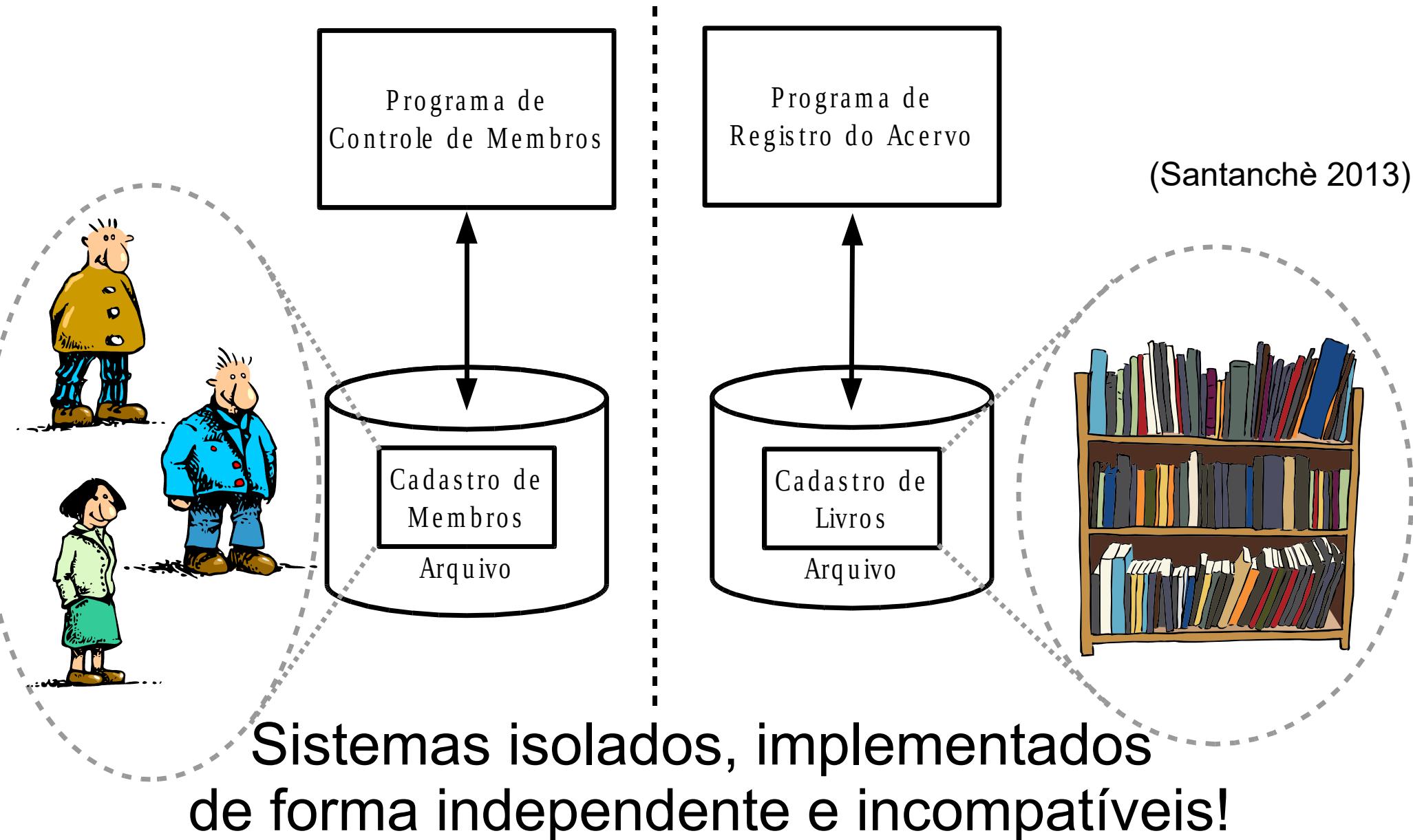
- Caos (sem SGBDs)
- Bancos de dados relacionais surgem, dão ordem ao caos, dominam indústria, fazem bilionários
- Web explode e inicia um novo caos!!!

Caos (até ~1980)

- Avanços em discos e fitas magnéticas permitem armazenamento de grandes quantidades de dados
- Ninguém sabe como armazenar os dados
- Cada um faz do seu jeito
- Métodos populares: hierárquico, rede
- Manutenção cara, difícil integração entre sistemas, bugs, inconsistência...



Exemplo: Biblioteca



Sistemas/Arquivos Isolados

- Redundância não controlada
 - Repetição
 - Inconsistência
- Barreiras para referenciar dados em outros sistemas
- Dificuldades com:
 - acesso concorrente
 - integridade e recuperação em caso de falha
 - segurança e controle de acesso

SGBDs (80's)

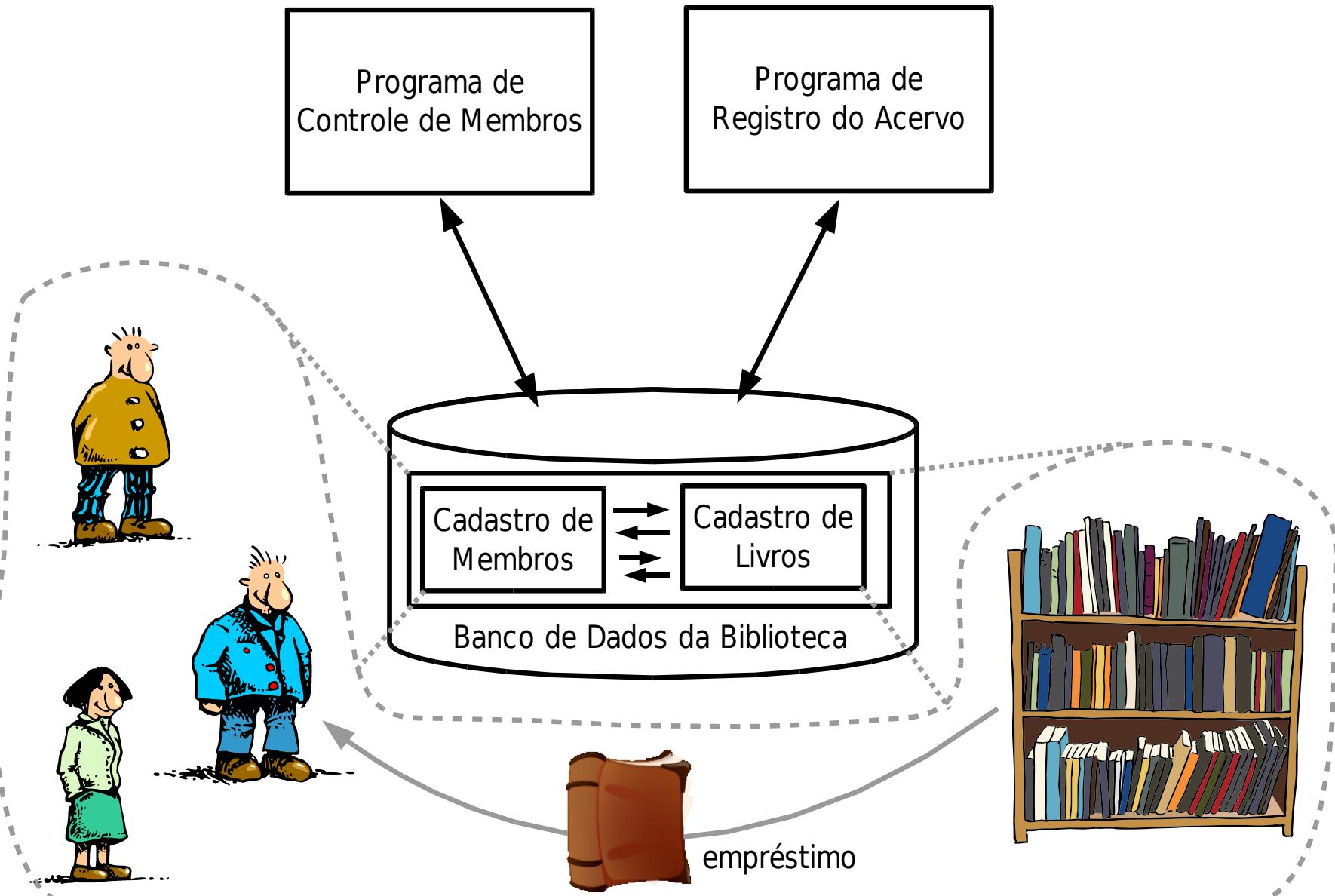
- Modelos conceituais sólidos (Modelo Relacional)
- Separação entre visão lógica e implementação física
- Linguagens de consulta poderosas (SQL)
- Implementações comerciais eficientes



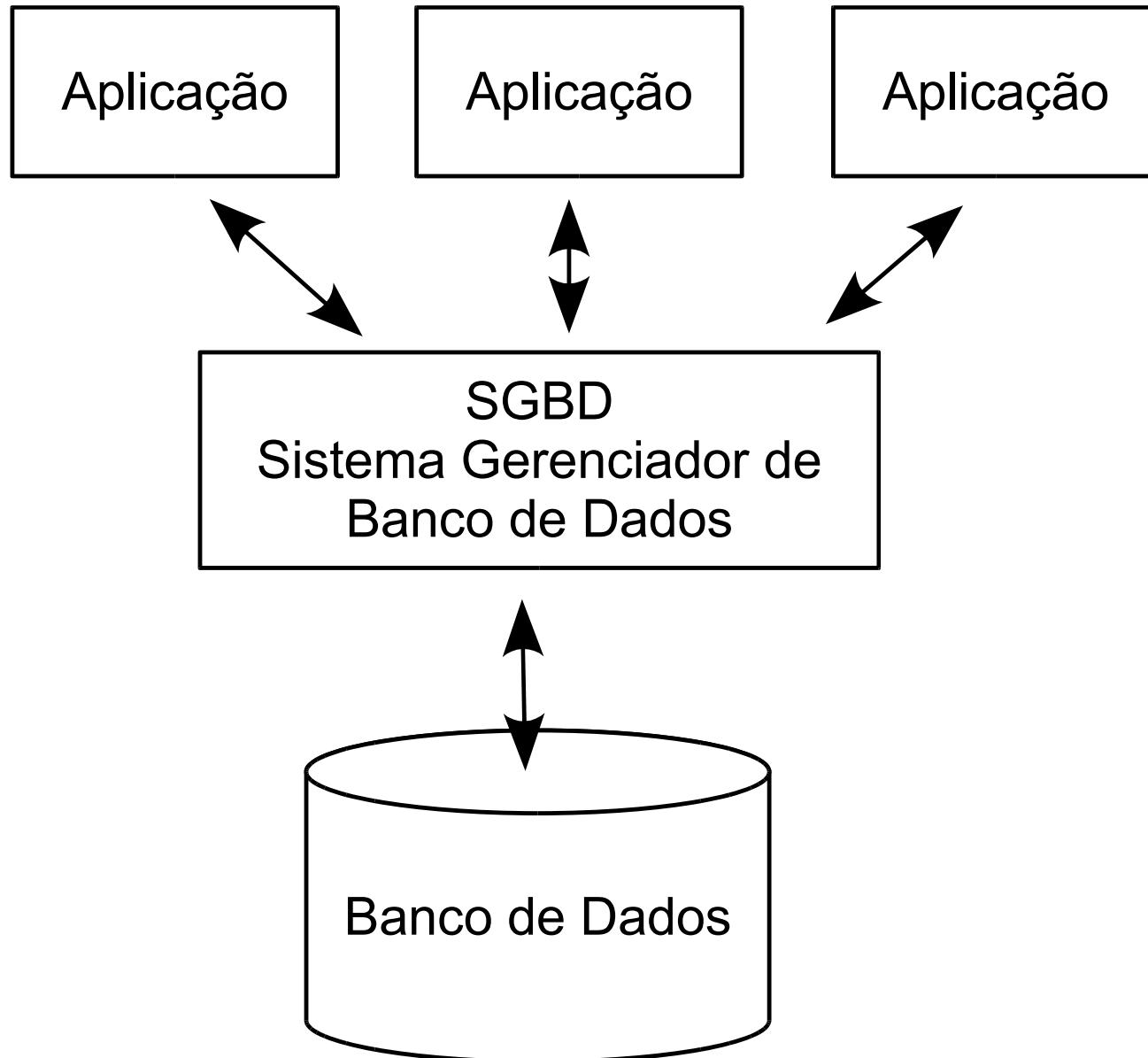
Benefícios dos SGBDs

- Independência de dados
- Acesso eficiente
- Tempo reduzido no desenvolvimento de aplicações
- Segurança e integridade de dados
- Administração de dados uniforme
- Acesso concorrente
- Recuperação de falhas

SGBD



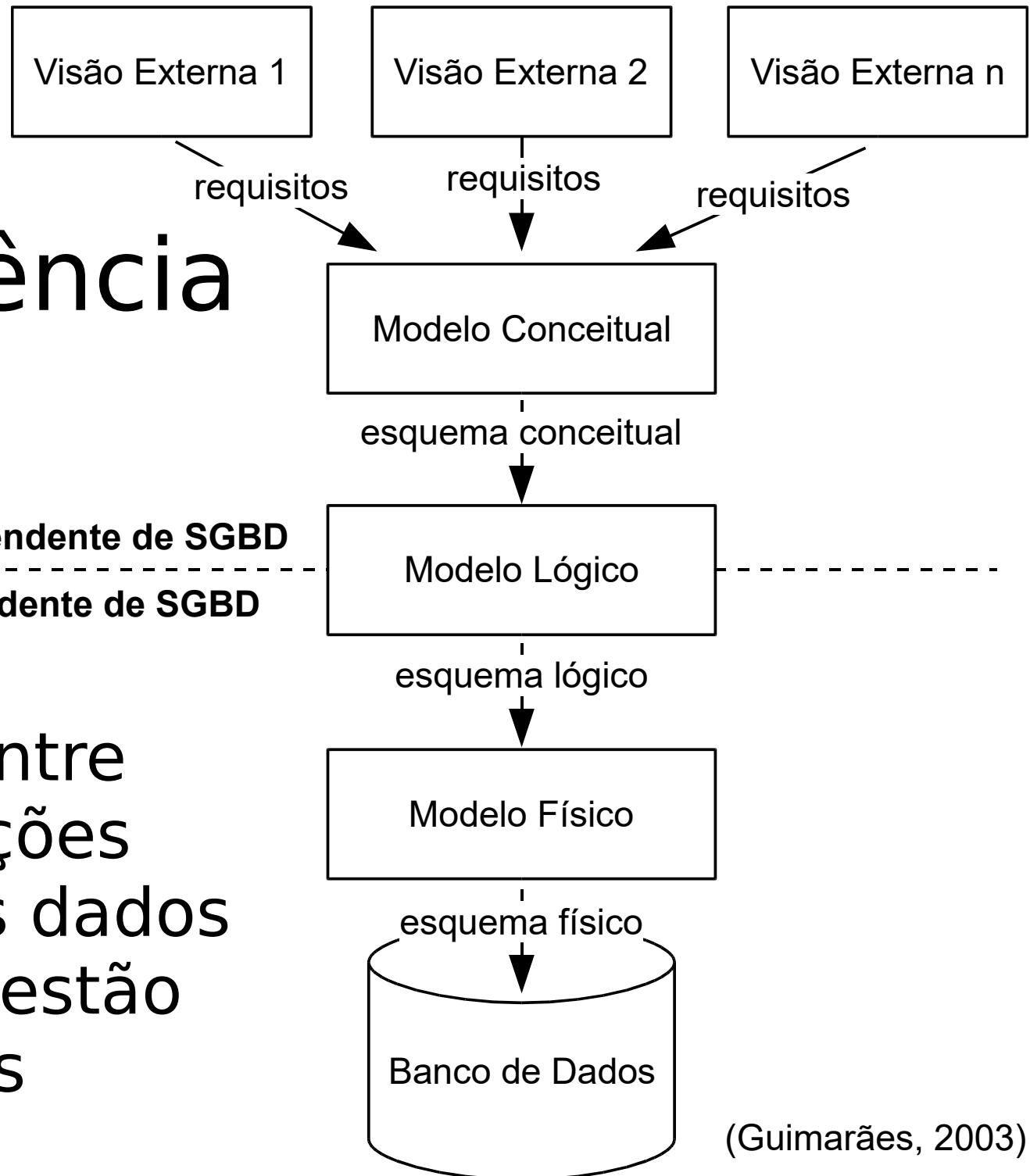
SGBD



Independência de Dados

Independente de SGBD
Dependente de SGBD

- Separação entre como aplicações enxergam os dados e como eles estão armazenados



(Guimarães, 2003)

Modelo Relacional

- Proposto por E. F. Codd em 1970 no artigo:
 - “A Relational Model for Large Shared Data Banks”
- Independência da estrutura interna
 - “Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed [...]”
(Codd, 1970)

Modelo Relacional

- Modelo mais amplamente utilizado por SGBDs
- Maiores empresas de informática oferecem soluções: IBM, Microsoft, Oracle, SAP
- Grandes projetos Open Source: MySQL, PostgreSQL, SQLite
- De celulares à data centers
- Mercado de U\$24bi (2012)

Larry Ellison, ORACLE

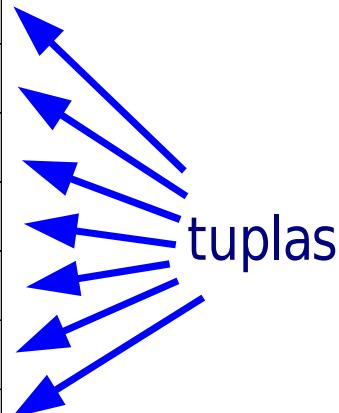


Relação (Tabela)

- Tupla = conjunto ordenado de atributos
- Valores de atributos são atômicos e monovalorados

Livro  relação

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565



Consultas (SQL)

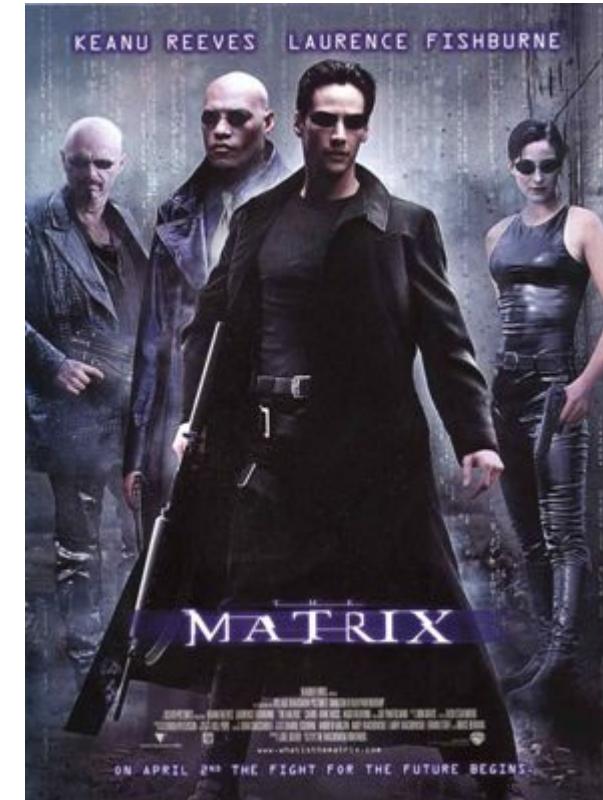
- Obtenha todos os livros de categoria “drama”
- `SELECT * FROM Livro
WHERE Categoria = “Drama”`

Livro

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565

SGBDs (90's em diante)

- Modelo relacional continua dominando o mercado
- BDs relacionais são adequados para mais de 90% das aplicações
- Mas a Internet, Celulares, Redes Sociais, Sensores... Têm causado uma revolução!



BigData – Dilúvio de Informação



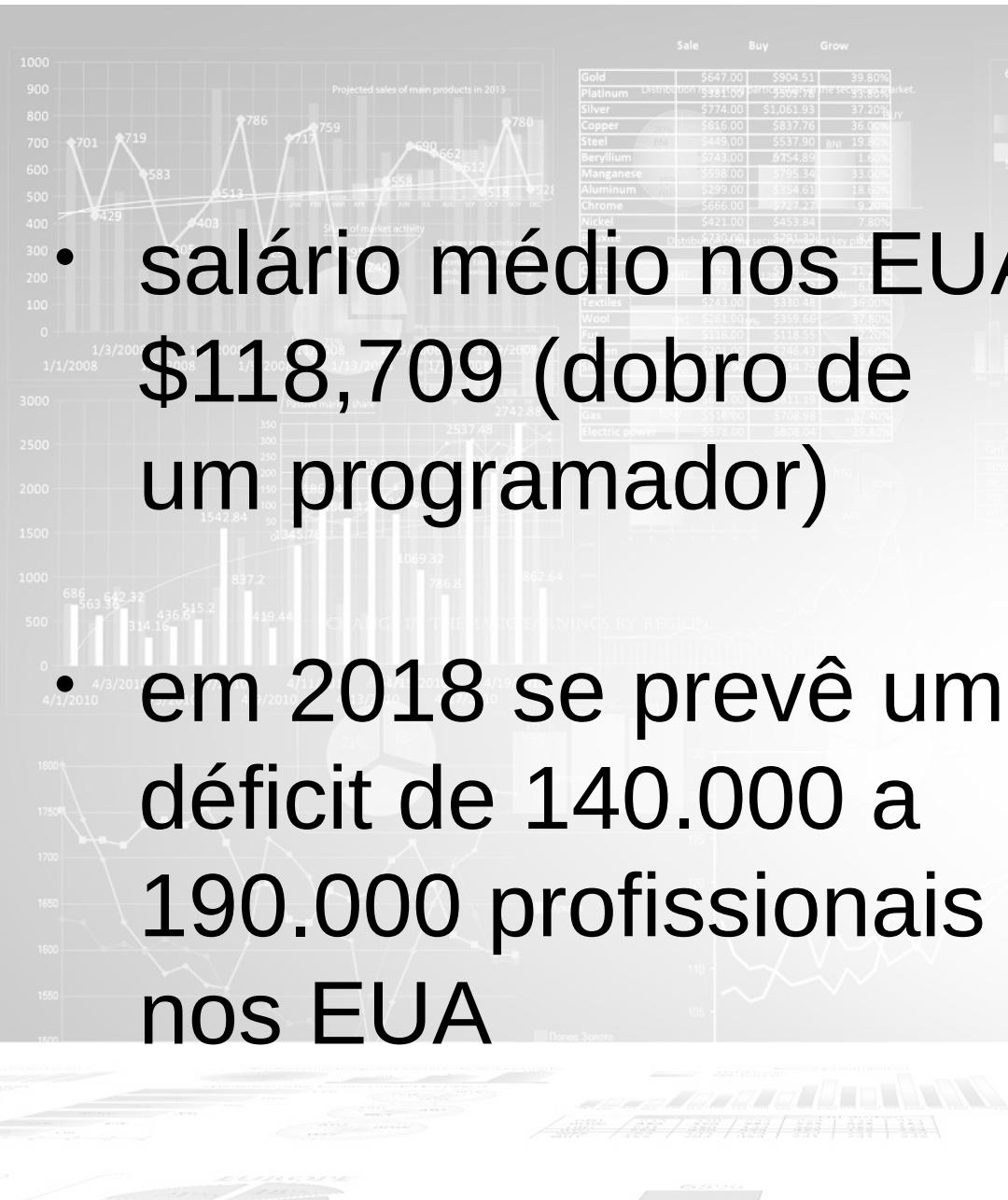
Dilúvio de Informação

- 1bi usuários conectados no facebook
(23/08/2015)
- 2bi smartphones no mundo
- 1b sites web
- 300 horas de vídeo no YouTube a cada minuto
- Google, Amazon, Microsoft and Facebook =
1,200 petabytes = 1.200.000.000.000.000
bytes = 5 pilhas de CDs até a Estação Espacial
Internacional



Cientista de Dados

- salário médio nos EUA \$118,709 (dobro de um programador)
- em 2018 se prevê um déficit de 140.000 a 190.000 profissionais nos EUA



Para os cientistas de dados não há desemprego

Germano Lüders / EXAME



Sinhigno, do Itaú: em três anos, o número de cientistas de dados no banco passou de 15 para mais de 100 profissionais

Revista Exame

- Cerca de 2,5 bilhões de gigabytes de dados são criados diariamente, o suficiente para encher 156 milhões de iPads.
- ... É aí que entra a figura do cientista de dados, capaz de extrair informações úteis de onde impera o caos.
- “Esse profissional é alguém que consegue unir dados desconexos, apontar tendências e ajudar na tomada de decisões” (IBM Brasil)

Revista Exame

- A atual ênfase dada a esse profissional marca uma nova fase da indústria de TI.
- No Brasil, um profissional recém-formado pode ganhar até 6000 reais, enquanto os gerentes chegam a receber cerca de 20000 reais.
- “Hoje, a disputa por esses profissionais é global e deve persistir por pelo menos uma década”
- o déficit de profissionais pode chegar a 380000 vagas no mundo em 2018.

O que vamos ver

- Como obter e armazenar dados
- Como modelar os dados
- Como gerenciar os dados
- Como consultar e manipular os dados
- Como otimizar o desempenho do banco para atender às necessidades da aplicação
- E como resolver qualquer problema relacionado com dados no mundo e região!

Representação de Dados e XML

Bancos de Dados

Luiz Celso Gomes-Jr
gomesjr@dainf.ct.utfpr.edu.br

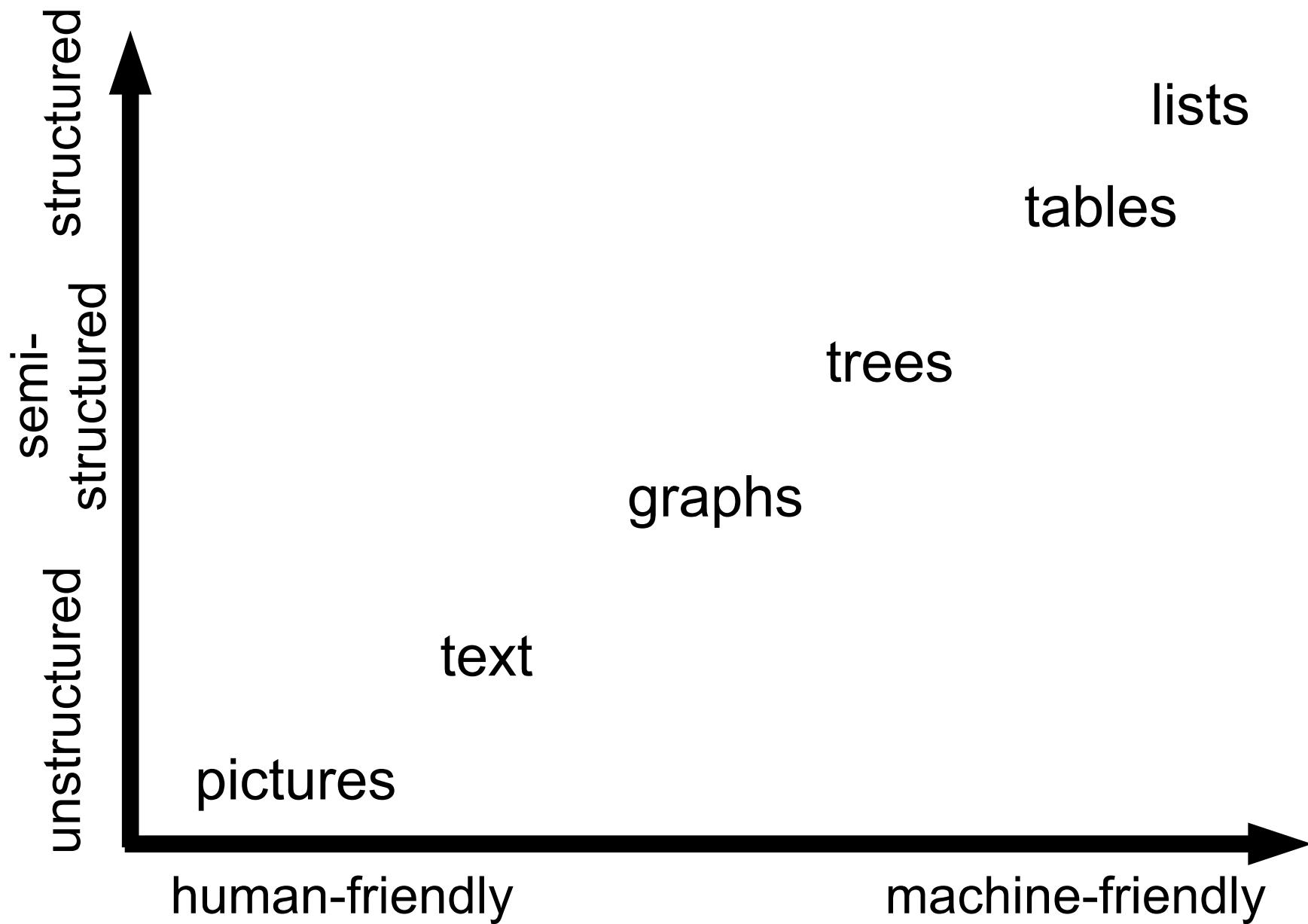
Representando Dados

- Antes da invenção do computador, a grande maioria dos dados eram representados em papel
- No papel, **texto, figuras, tabelas**, etc. são representados com tinta e cabe ao leitor identificar os diferentes elementos e entender o conteúdo
- No computador, precisamos saber de antemão com que **tipo de dados** estamos lidando para representá-los adequadamente

Processando Dados

- Quanto mais “organizados” (estruturados) os dados, mais simples é o processamento
- Exemplo de dados **estruturados**: listas, tabelas, matrizes
- Exemplo de dados **não-estruturados**: texto, imagens, sons
- Exemplo de dados **semiestruturados**: árvores, grafos

Processando Dados



Principais modelos de dados

- Modelo Relacional (tabelas - foco do curso)
- Modelo hierárquico (árvores - foco desta aula)
- Modelo de grafos (veremos no fim do curso)

Importância dos modelos

- Permitem a separação entre representação dos dados e a implementação física das estruturas
- Exemplo de ED1: um programador pode implementar um programa que usa uma lista encadeada e no futuro mudar a implementação das bibliotecas para uma lista duplamente encadeada sem precisar alterar o programa principal

Importância dos modelos

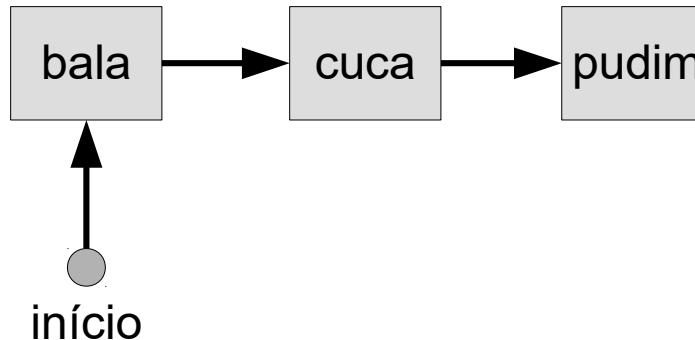
Aplicação

```
item = "chocolate";  
adicionaItem(item);  
...
```

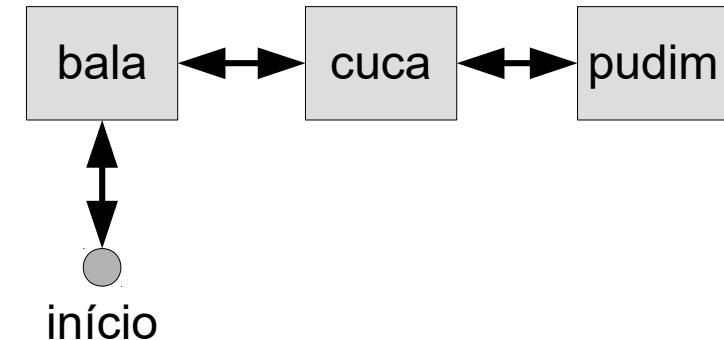
```
item = "chocolate";  
adicionaItem(item);  
...
```

Modelo: Lista (biblioteca ou API)

Implementação
das Estruturas



Lista Encadeada



Lista duplamente Encadeada

Importância dos modelos

- Em grandes empresas, os mecanismos de armazenamento mudam frequentemente para atender às demandas e isto não pode afetar as aplicações
- Exemplos: atualização de versão do SGBD, mudança de fornecedor de SGBD, upgrade de SGBD centralizado para SGBD distribuído

Modelo Hierárquico

- Usado para representar diversos tipos de dados
- Exemplos: Documentos, Categorias de Produtos, Modelos Orientados a Objeto, Disciplinas (tópicos, conteúdo, tarefas)
- Relativamente **flexível** e processamento relativamente **eficiente** (no meio do caminho entre tabelas e grafos)
- **XML** é o principal padrão para representação e compartilhamento de dados

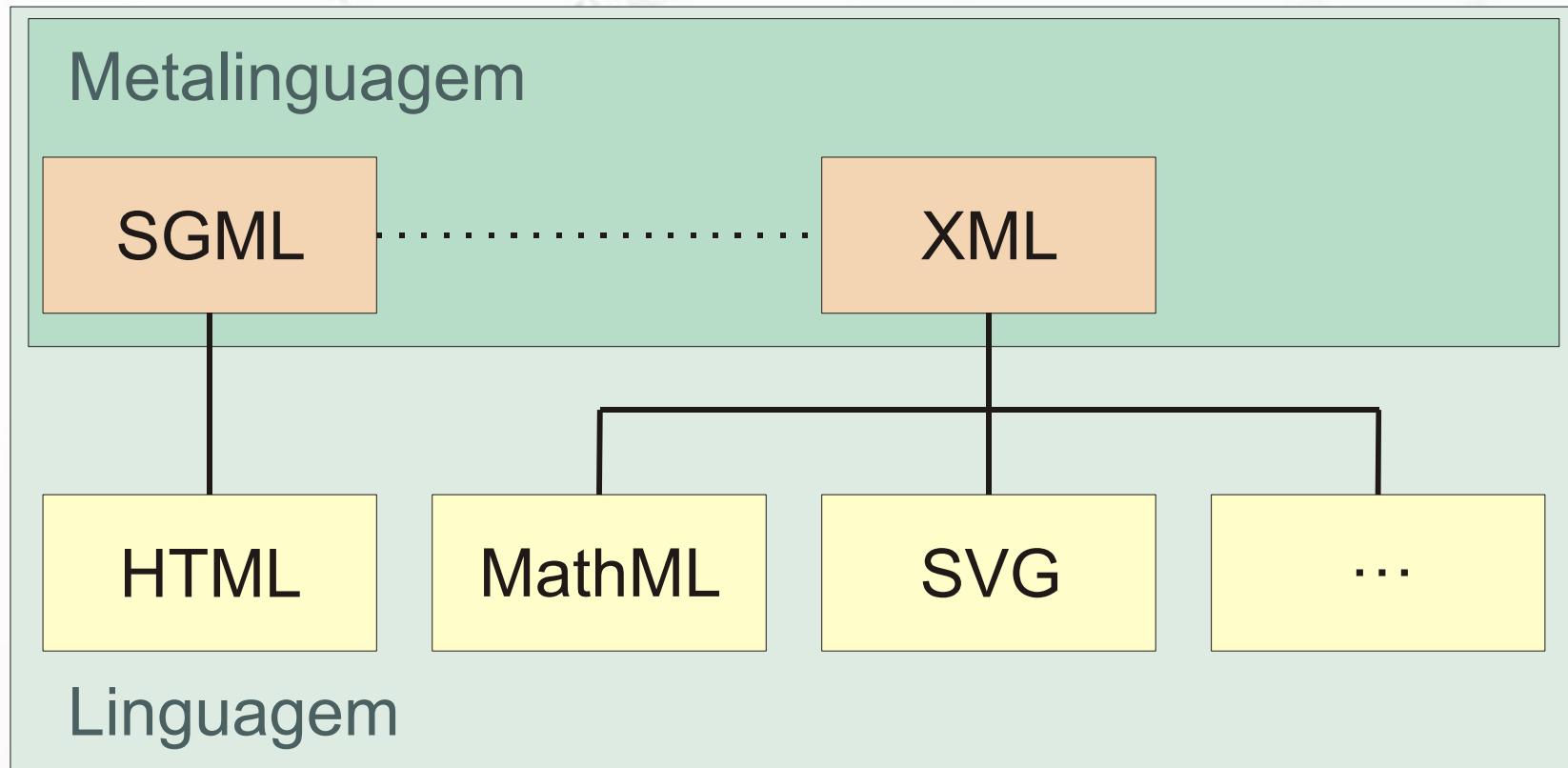
XML - eXtensible Markup Language

XML

- Lançada em 1996 como uma versão simplificada da SGML (*Standard Generalized Markup Language*), para ser utilizada na Web.

Metalinguagem

- Tal como SGML, XML é uma metalinguagem.
- HTML ao contrário, foi escrita em SGML.



Linguagem de Marcação

- Utiliza marcadores para agregar informações adicionais a documentos.
- Tomemos como exemplo a seguinte frase:
Horácio escreveu o livro Vida dos Dinossauros.
- Desejamos agregar informações que identifiquem quem é o autor e qual a ação realizada.

Linguagem de Marcação

- Os marcadores se diferenciam do conteúdo pelos símbolos “<” e “>” (seguem o mesmo princípio de HTML):

```
<autor>Horácio</autor> <ação>escreveu o livro Vida dos Dinossauros</ação>
```

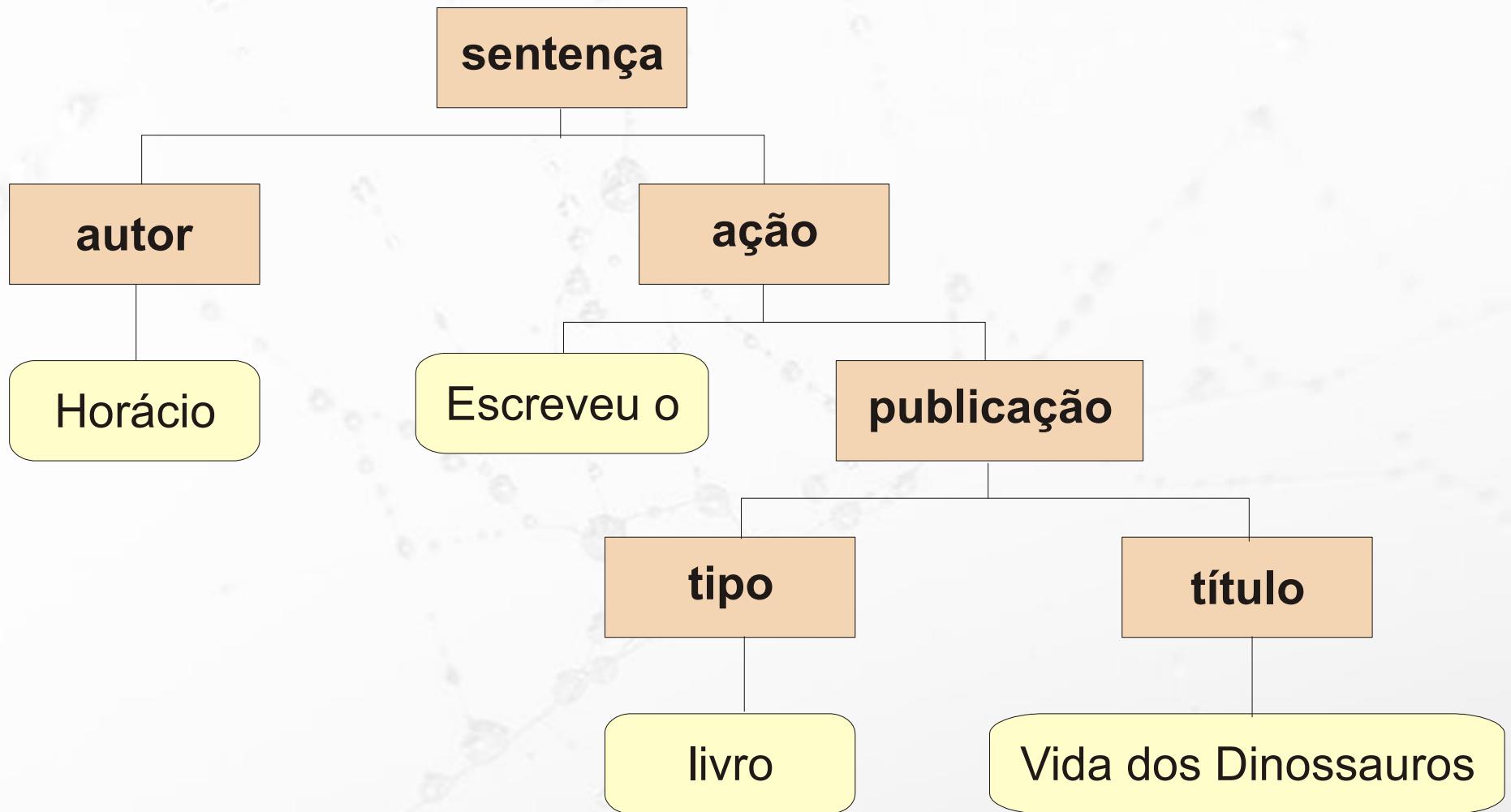
- Os marcadores delimitam unidades estruturais denominadas **elementos**.

Estrutura Hierárquica

- Marcações podem ser agrupadas hierarquicamente.
- A interpretação de cada marcador está subordinada a seu contexto.

```
<sentença>
  <autor>Horácio</autor>
  <ação>escreveu o
    <publicação>
      <tipo>livro</tipo>
      <título>Vida dos Dinossauros</título>
    </publicação>
  </ação>
</sentença>
```

Modelo de Dados XML



Elementos e Atributos

- Atributos:

```
<autor cpf="487.526.548-74" nascimento="12/5/1960">
```

Horácio

```
</autor>
```

- Elementos vazios:

```
<esgotado/>
```

Validação de Documentos

- Documento bem formado:
 - atende às regras de construção XML
- Documento válido:
 - bem formado
 - atende a um esquema
 - DTD
 - XML Schema

DTD

- O documento XML pode se basear em uma gramática definida através de uma DTD (*Document Type Definition*).

```
<!ELEMENT documento (topico+)>
<!ELEMENT topico (titulo, subtopico*)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT subtopico (titulo, #PCDATA)>
```

XML Schema

- Padrão para definição de esquemas XML
- Mais poderoso

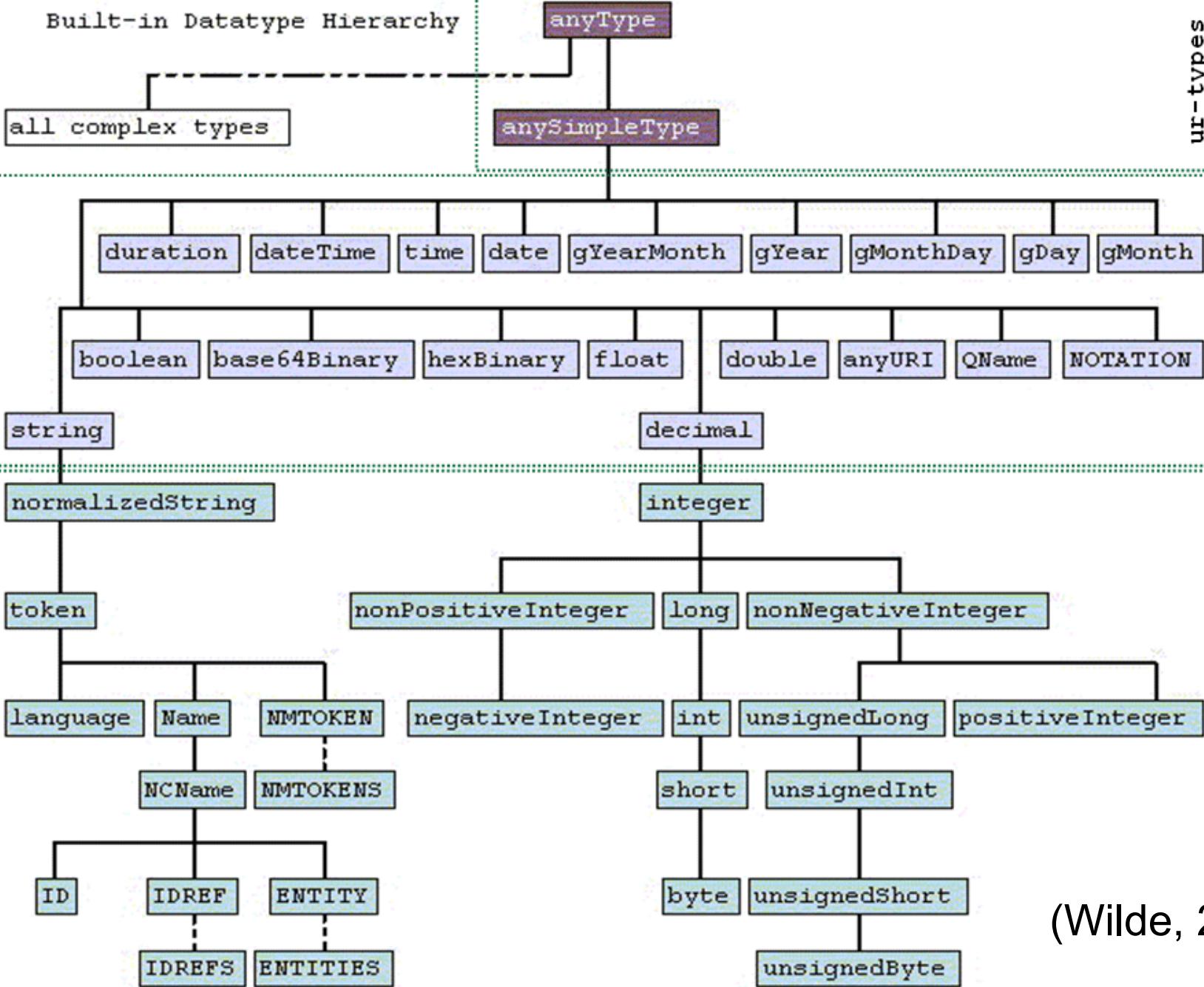
Tipos Simples

```
<xs:element name="business">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:maxLength value="30"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

(Wilde, 2006)

Built-in Datatype Hierarchy

primitive types



(Wilde, 2006)

derived by restriction

built-in primitive types

ur types

derived by list

derived by extension or

derived by extension or

Tipo Composto

```
<xs:schema>
  <xs:element name="billingAddress" type="addressType"/>
  <xs:element name="shippingAddress" type="addressType"/>
  <xs:complexType name="addressType">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="street" type="xs:string"/>
      <xs:element name="city" type="xs:string"/>
      <xs:element name="state" type="xs:string" minOccurs="0"/>
      <xs:element name="zip" type="xs:decimal"/>
    </xs:sequence>
    <xs:attribute name="country" type="xs:NMTOKEN"/>
  </xs:complexType>
</xs:schema>
```

(Wilde, 2006)

Query

- **XPath**

- Especifica expressões na forma de caminhos que atendem padrões para alcançar nós específicos (elementos ou atributos)

- **XQuery**

- Queries para XML (usam XPath)



URI

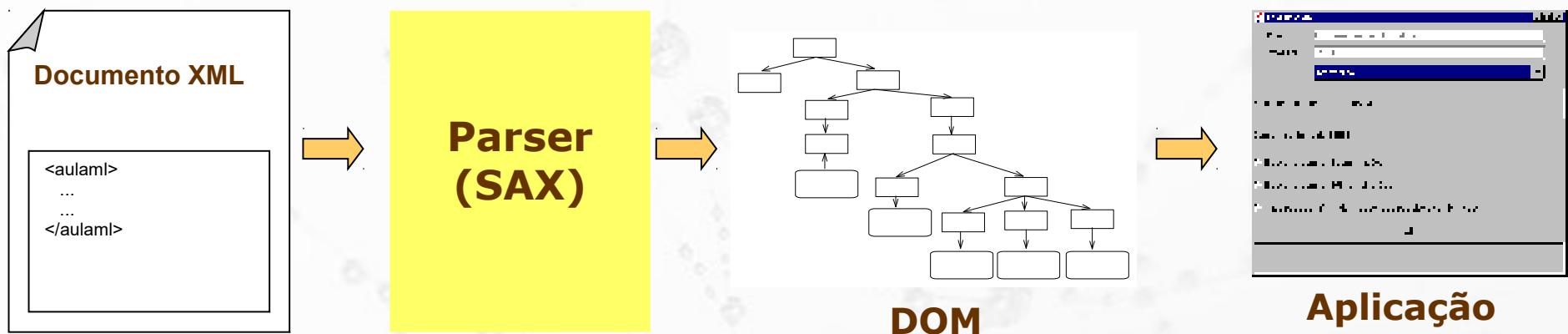
- A identificação de um recurso é feita através de um URI - Uniform Resource Identifier.
- URI = URL ou URN



- URL (*Uniform Resource Locator*): identifica recursos por meio de sua localização física na Internet.
Ex.: <http://www.paleo.org>
<ftp://ftp.unicamp.br>
<mailto:horacio@paleo.org>
- URN (*Uniform Resource Names*): identificador é relacionado indiretamente com sua localização física na rede (exige um resolver).

Ex.: <urn:ogc:def:uom:celsius>
<urn:mpeggra:mpeg21:di:iswc:T-041.220.506-1>

Parser XML

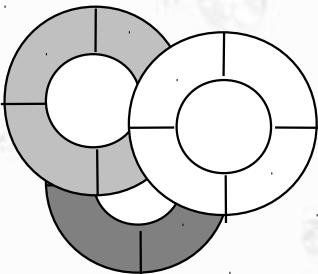


Introdução

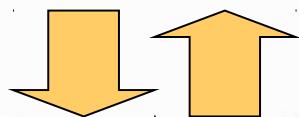
Diversas tecnologias têm sido criadas para o processamento de documentos XML.

Aplicação

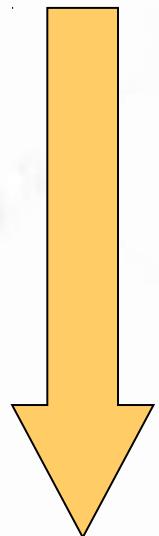
Server
Pages



Classes



Data-Binding



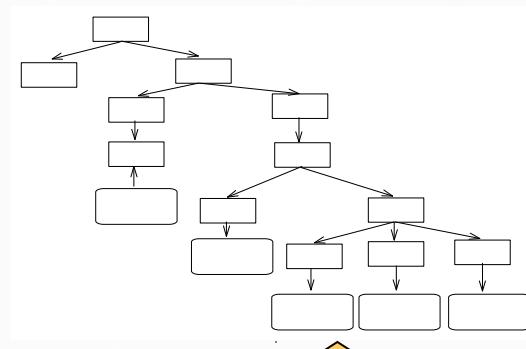
```
<aulaml>
  <curso>
    ...
  </curso>
  <quadro>
    <texto>
      ...
    </texto>
    <teste>
      ...
    </teste>
  </quadro>
</aulaml>
```

```
<aulaml>
  <curso>
    ...
  </curso>
  <quadro>
    <texto>
      ...
    </texto>
    <teste>
      ...
    </teste>
  </quadro>
</aulaml>
```

Eventos



SAX



DOM

```
<aulaml>
  <curso>
    ...
  </curso>
  <quadro>
    <texto>
      ...
    </texto>
    <teste>
      ...
    </teste>
  </quadro>
</aulaml>
```

XML

Introdução

Dentre estas tecnologias duas se destacaram e se tornaram referência:

- SAX - Simple API for XML
- DOM - Document Object Model

SAX

- API baseada em eventos.
- Se tornou a mais estável API XML largamente utilizada [DOD01].
- Iniciou como uma solução para acesso a documentos XML por programas Java.
- Hoje tem sido portada para outras linguagens de programação, tal como: C++, Pascal, Perl, Phyton, etc.

Eventos de conteúdo

```
public class SAXBasico extends  
        org.xml.sax.helpers.DefaultHandler  
{  
  
    public void startDocument() ...  
  
    public void startElement(...) ...  
  
    public void characters (...) ...  
  
    public void endElement(...) ...  
  
    public void endDocument() ...  
}
```

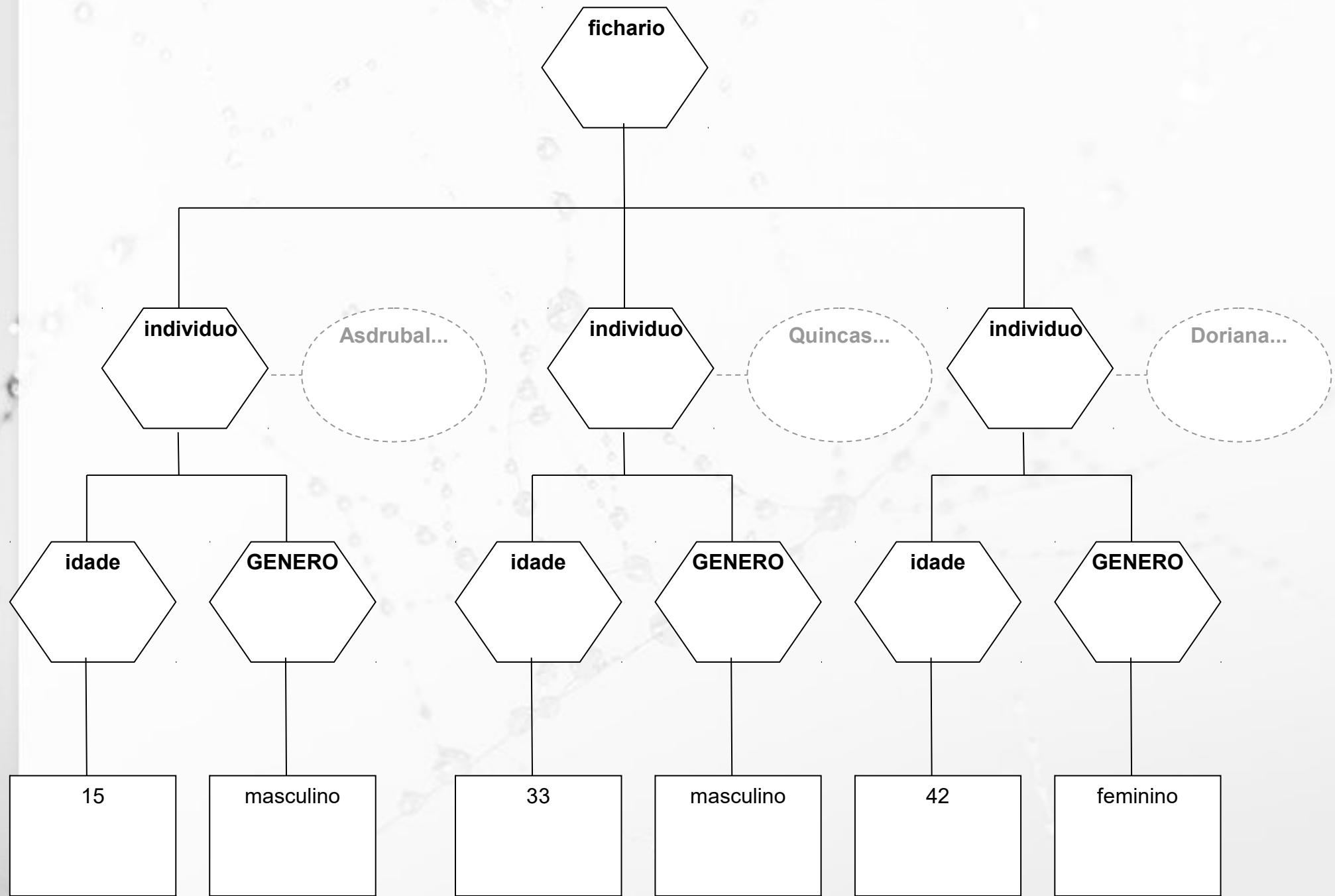
Eventos de conteúdo

Método	Acionado quando o <i>parser</i> encontra
startDocument	início do documento
startElement	início de um elemento
characters	conteúdo texto
endElement	final de um elemento
endDocument	final do documento

DOM

- DOM define uma API para documentos XML e HTML.
- Ele acrescenta ao padrão destas linguagens toda a funcionalidade e flexibilidade que um programa precisa para acessar e manipular documentos.
- Definido em IDL, ECMAScript e Java.

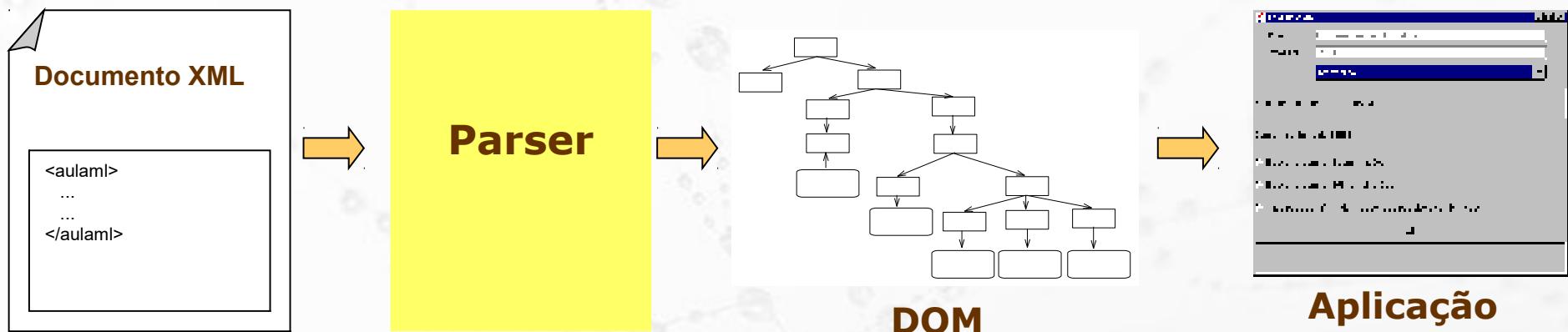
Document Object Model



DOM - Estudo de Caso

```
<FICHARIO>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</FICHARIO>
```

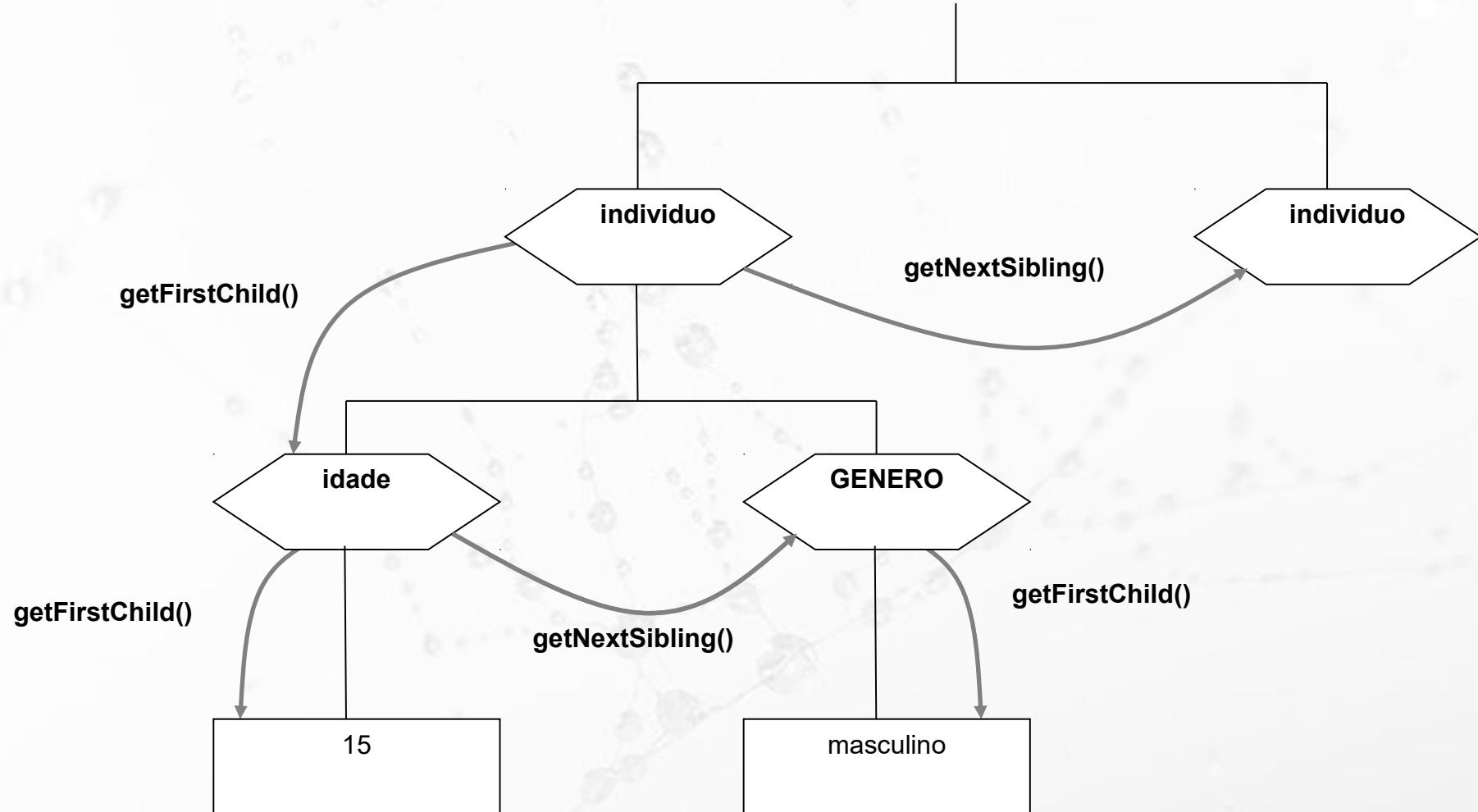
Processo



Interfaces

- **Node** - esta interface representa genericamente qualquer nó da árvore.
- **Element** - acrescenta propriedades e métodos específicos de um nó do tipo elemento.
- **Document** - interface do nó raiz da árvore que representa o documento completo.
- **NodeList** - representa uma lista de nós. Pode representar, por exemplo, a lista de filhos de um nó.

Navegar pelo Documento



JSON

- Modelo de representação de dados semiestruturados
- Baseado em JavaScript com foco em aplicações Web
- Mais simples e mais leve que XML (mas também mais limitado)
- Não define linguagens (em definição)
- XML Nutella :)

JSON x XML

```
{"employees": [  
    { "firstName": "John", "lastName": "Doe" },  
    { "firstName": "Anna", "lastName": "Smith" },  
    { "firstName": "Peter", "lastName": "Jones" }  
]}
```

```
<employees>  
  <employee>  
    <firstName>John</firstName>  
    <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName>  
    <lastName>Smith</lastName>  
  </employee>  
  <employee>
```

Exercícios

- Dê exemplos de dados para cada tipo de modelo (relacional, hierárquico, grafos)
- Quais as vantagens de se usar um modelo padronizado como o XML?

Referências e Agradecimentos

- Diversos slides baseados no curso de BD do Prof. André Santanchè (UNICAMP)
 - Site: <http://www.ic.unicamp.br/~santanche>
 - Canal Youtube:
<https://www.youtube.com/santanche>

Modelagem Respostas dos Exercícios

Bancos de Dados

Luiz Celso Gomes-Jr
gomesjr@dainf.ct.utfpr.edu.br

Exercício

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:

- Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.
- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.

Exercício

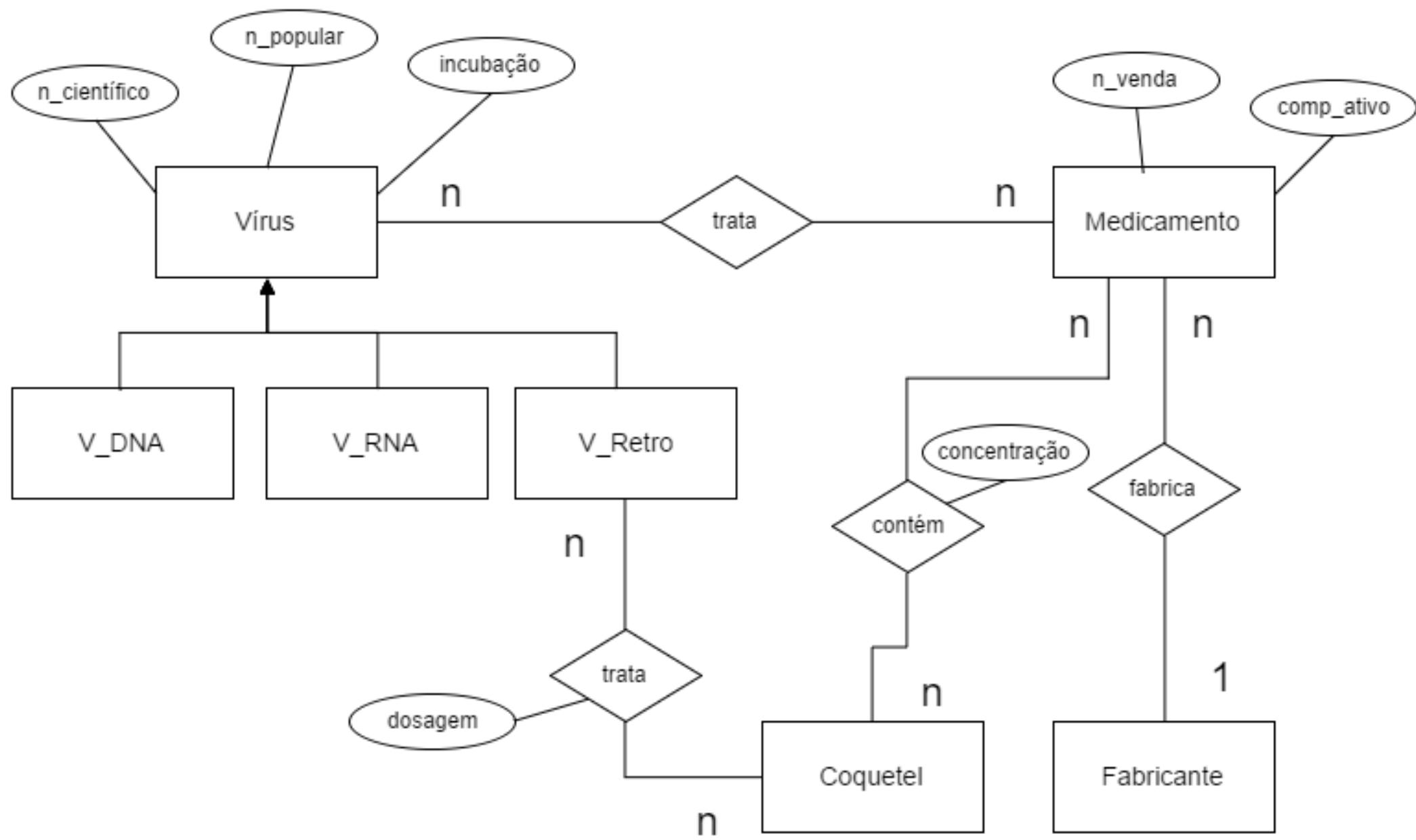
parte 2

- a) Considere que um dado medicamento pode tratar vários vírus e um vírus pode ser tratado por vários medicamentos.
- b) Medicamentos são fabricados por empresas que possuem nome e CNPJ. Uma empresa pode fabricar vários medicamentos, mas um medicamento é fabricado por uma única empresa.

Exercício

parte 3

- Vírus podem ser classificados em diversas categorias, mas considere apenas Vírus com DNA, Vírus com RNA e Retrovírus. Retrovírus são tratados com coquetéis de medicamentos. Um coquetel é composto por vários medicamentos, cada um em uma concentração específica. Um coquetel tem uma dosagem específica para o tratamento de um dado Retrovírus.

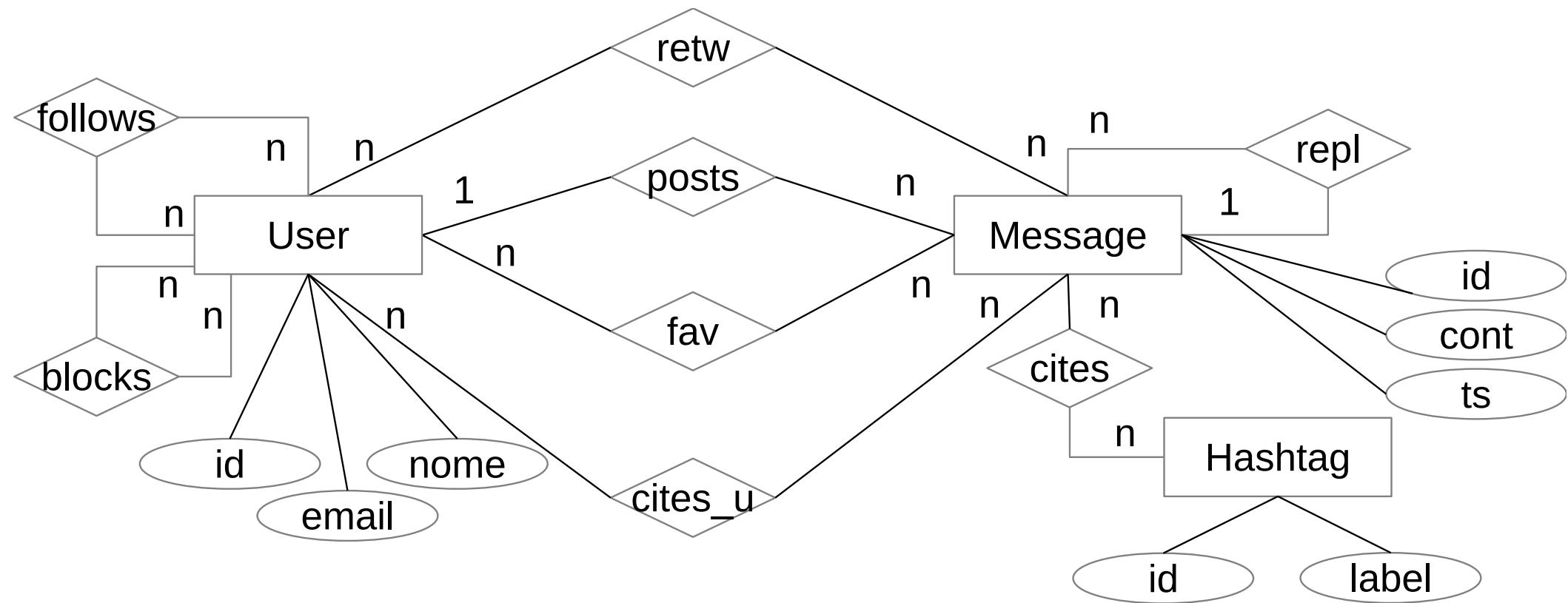


Exercício

Modele em um diagrama ER o banco de dados para o serviço de microblogs e rede social Twitter. O serviço permite o cadastro de usuários que devem informar nome, e-mail, e um identificador único para a conta. Usuários podem seguir (follow) outros usuários. Usuários também podem bloquear outros usuários.

Usuários postam mensagens (tweets). Mensagens possuem identificador, conteúdo (texto) e timestamp do momento de postagem (estilo Unix time, ou seja, representa o número total de segundos decorridos desde 0:00 de 1/1/1970). Mensagens podem mencionar outros usuários (conhecidos como at ou @) e mencionar tópicos (conhecidos como hashtags ou #). Tópicos possuem id e rótulo (e.g. "banco_de_dados"). Usuários e tópicos são automaticamente extraídos do texto do post no momento de envio com base nas strings que começam por @ ou #. Por exemplo, a mensagem "Hoje é dia de prova de #banco_de_dados! @UTFPR @Computacao" referencia o tópico "banco_de_dados" e os usuários "UTFPR" e "Computacao". Mensagens podem ser repostadas (re-tweets), marcadas como favoritas, ou respondidas por qualquer usuário. Mensagens de resposta são novas mensagens que fazem referência à mensagem original. Mensagens repostadas são idênticas às originais, mas fazem referência aos usuários que as repostaram.

Descreva no seu diagrama todas as entidades e relacionamentos relevantes, atributos, chaves primárias e cardinalidades. Minimize a redundância de informação.



Modelagem

Bancos de Dados

Luiz Celso Gomes-Jr
gomesjr@dainf.ct.utfpr.edu.br

Banco de Dados

Modelagem

André Santanchè e Patrícia Cavoto
Instituto de Computação - UNICAMP
Agosto 2015

Universo de Discurso ou Mini-mundo

- “Um banco de dados representa algum aspecto do mundo real, às vezes chamado de **mini-mundo** ou de **universo de discurso** (UoD – Universe of Discourse).”
(Elmasri & Navathe, 2011)

Universo de Discurso ou Mini-mundo

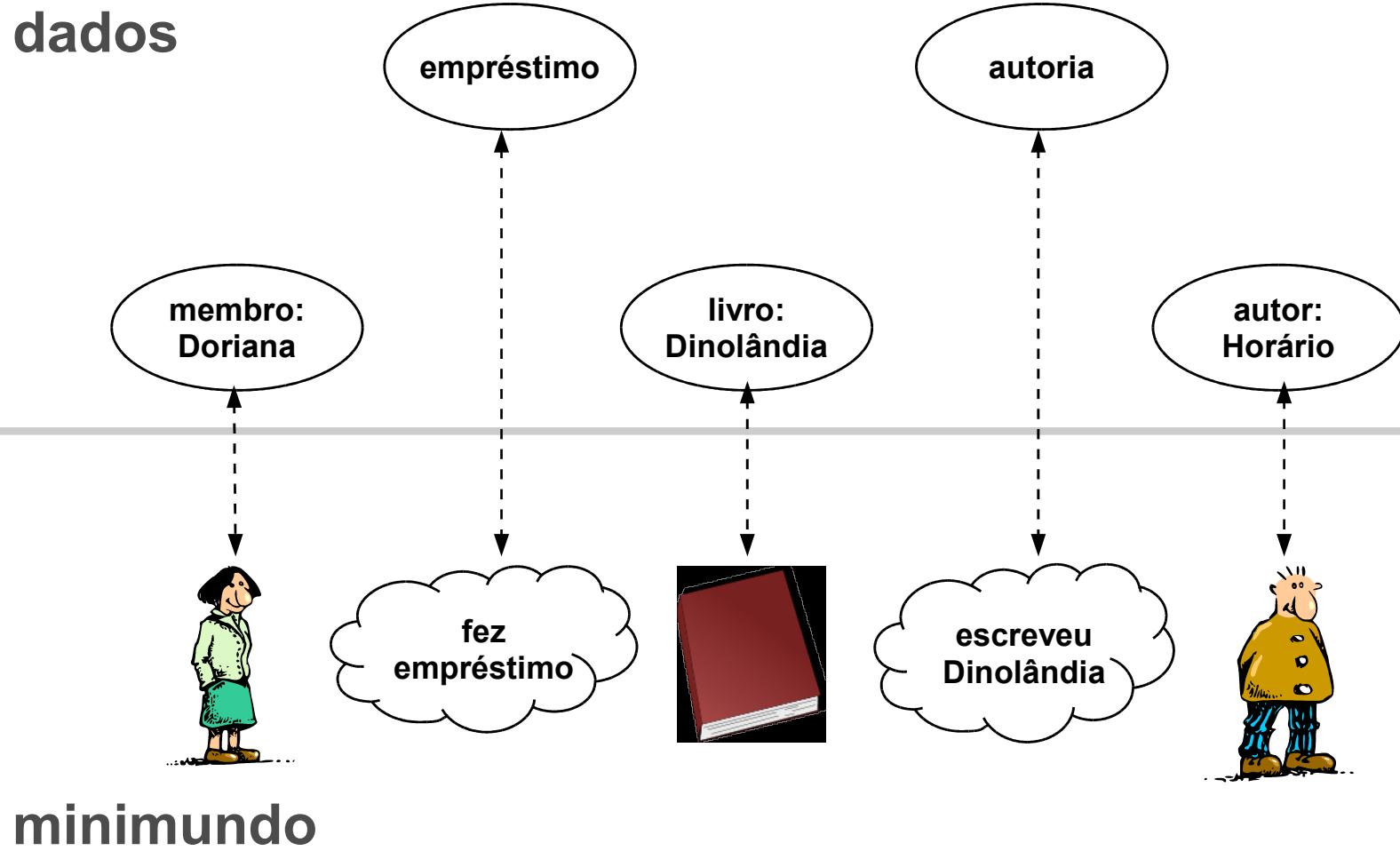
- Recorte do mundo real a ser representado
-



minimundo

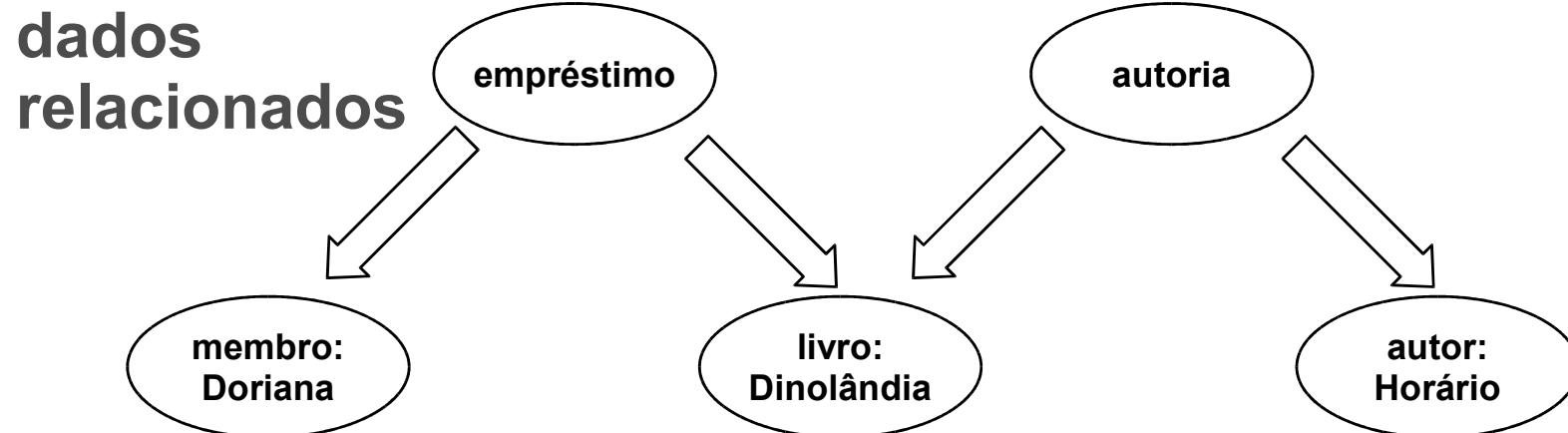
Dados

- Fatos registrados – significado implícito



Banco de Dados

- Coleção de dados relacionados



Abstração

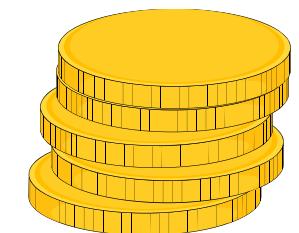
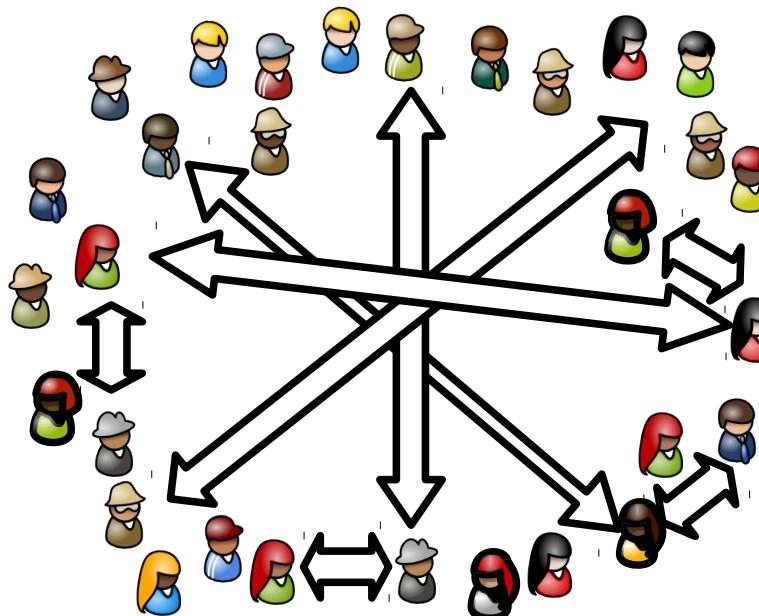
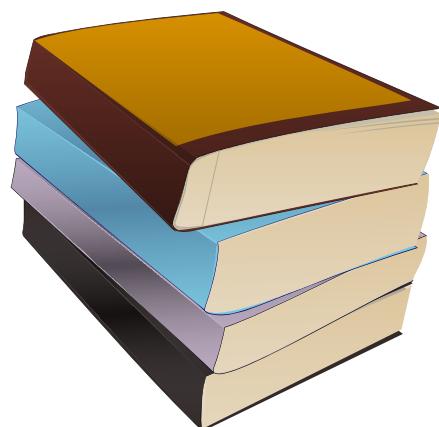
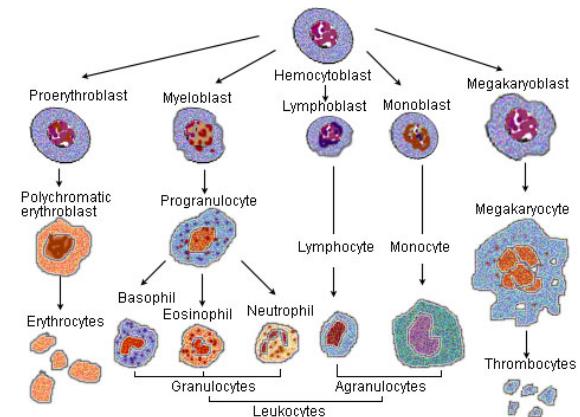
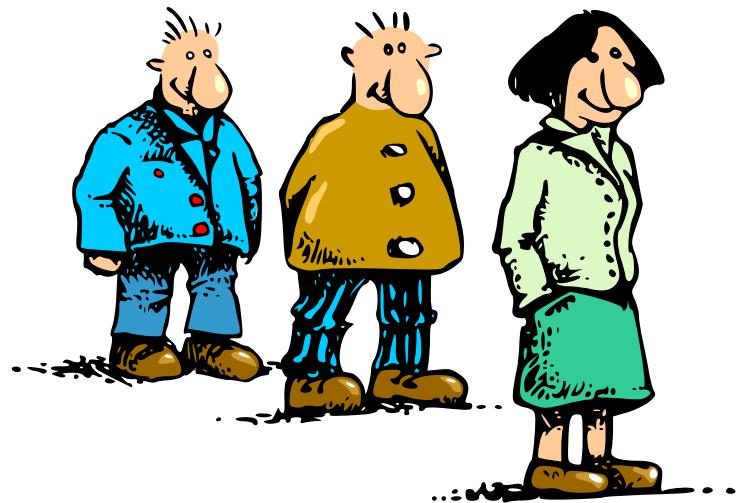
Problema x Abstração

- “Para resolver um problema é necessário escolher uma abstração da realidade”
(Almeida, 2010)

Abstração

- “**processo mental** que consiste em **escolher ou isolar um aspecto** determinado de um estado de coisas relativamente complexo, a fim de simplificar a sua avaliação, classificação ou para permitir a comunicação do mesmo” (Houaiss, 2006)
- Abstrações ajudam a gerenciar a complexidade do software (Shaw, 1984)

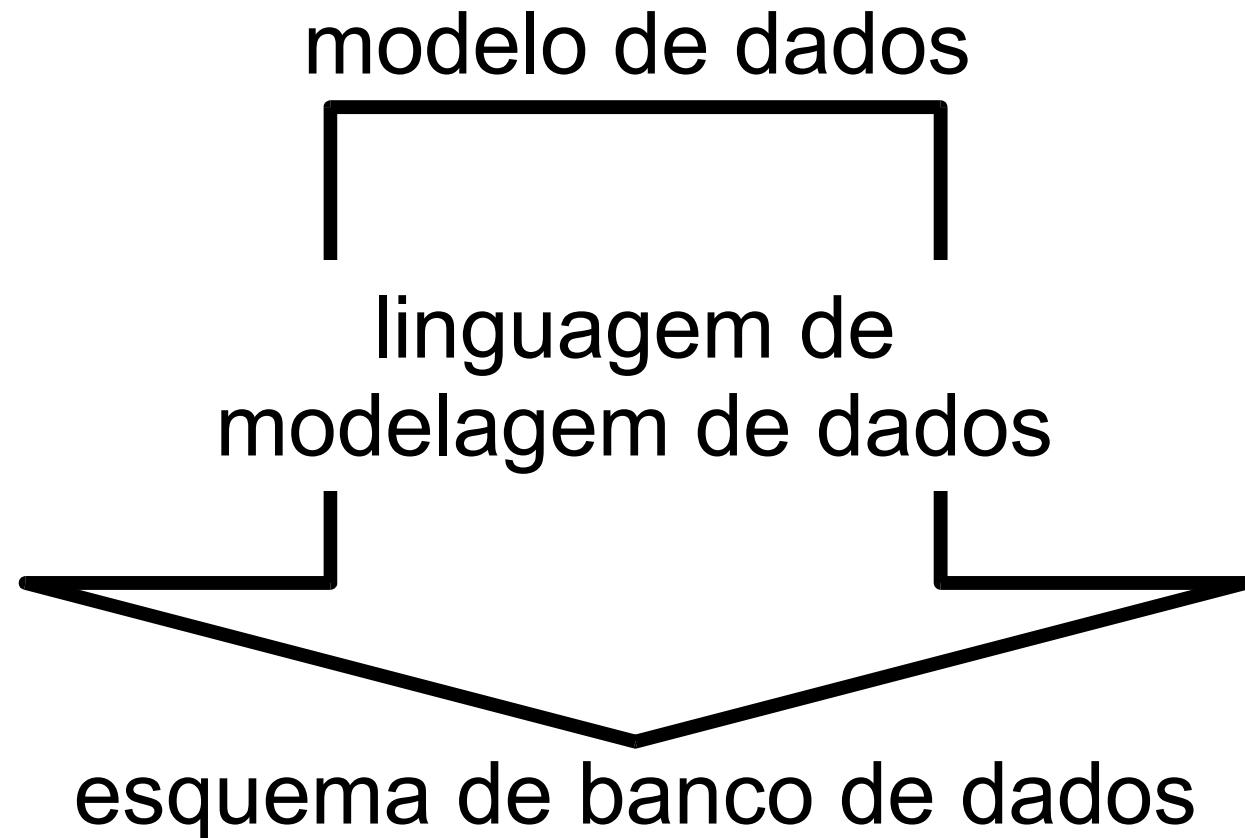
Abstrações do Dia a Dia



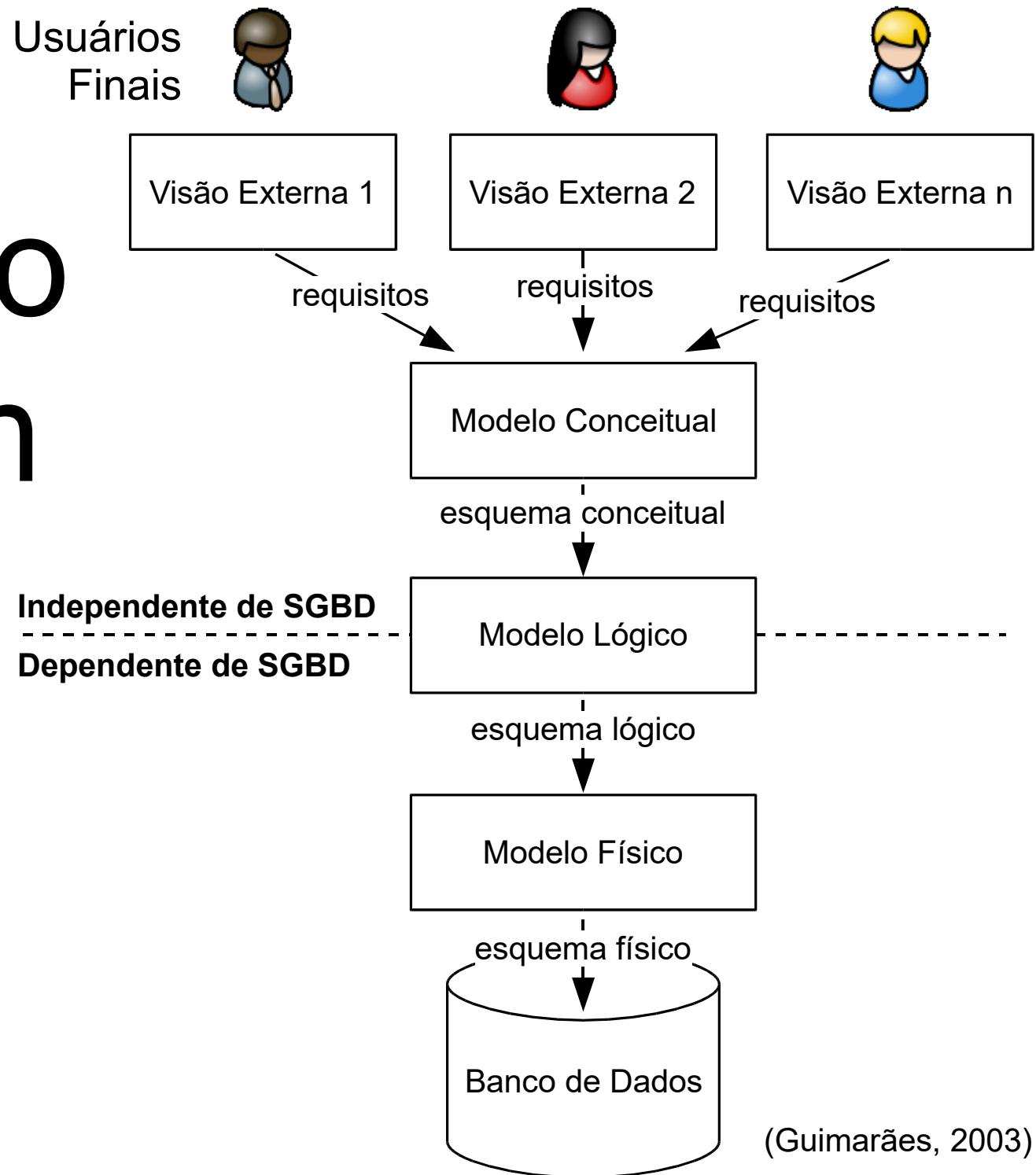
Modelo de Dados

- Modelo de dados em Banco de dados:
 - “descrição formal da estrutura de um banco de dados”
- (Heuser, 2004)

Esquema de Banco de Dados

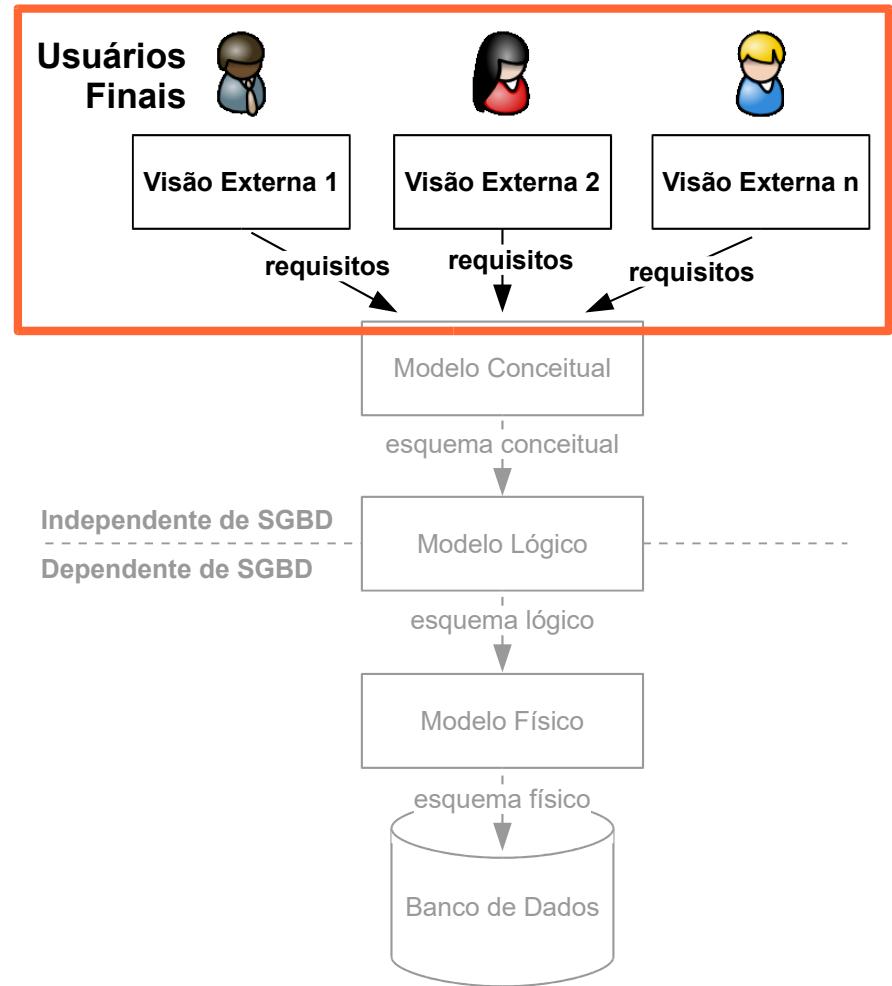


Projeto de um BD



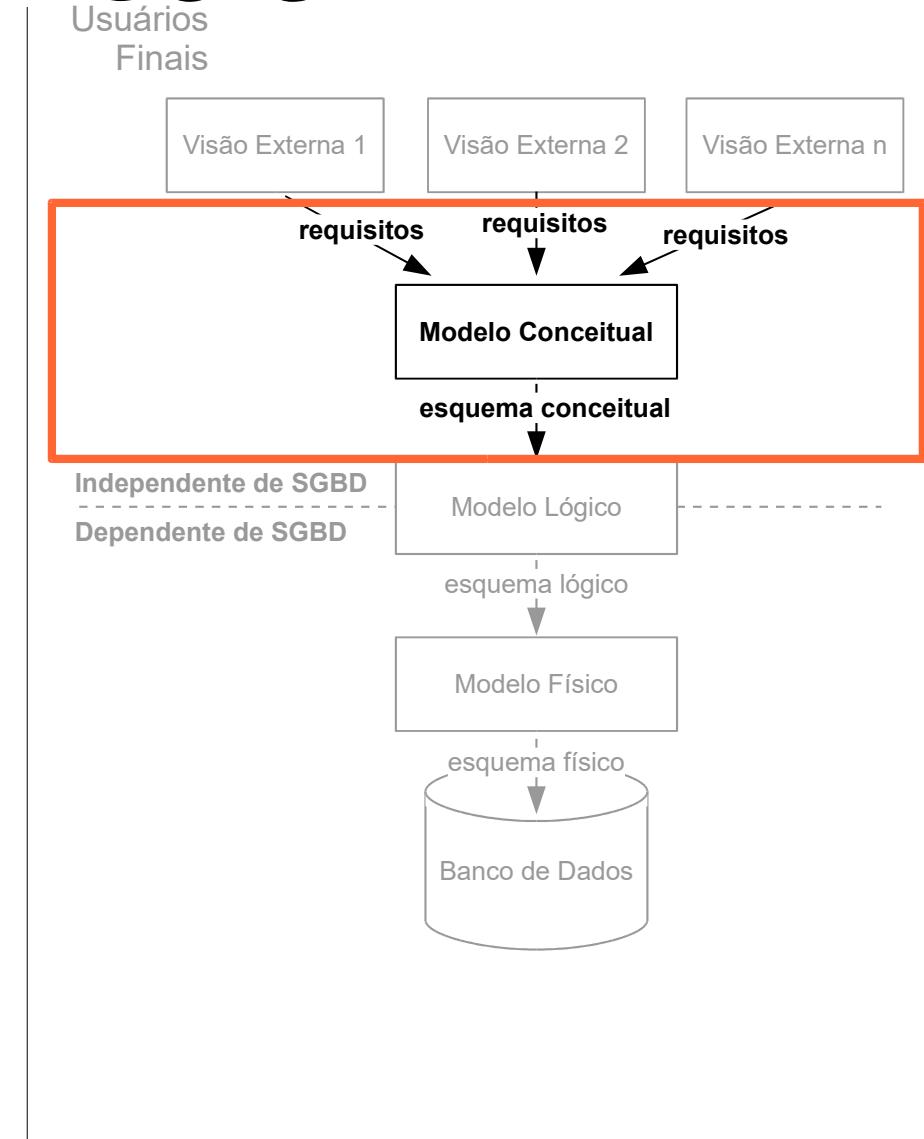
Visão Externa

- Guiada pelos requisitos dos usuários
- Usualmente representada em documentos textuais
- Visão – recorte do esquema



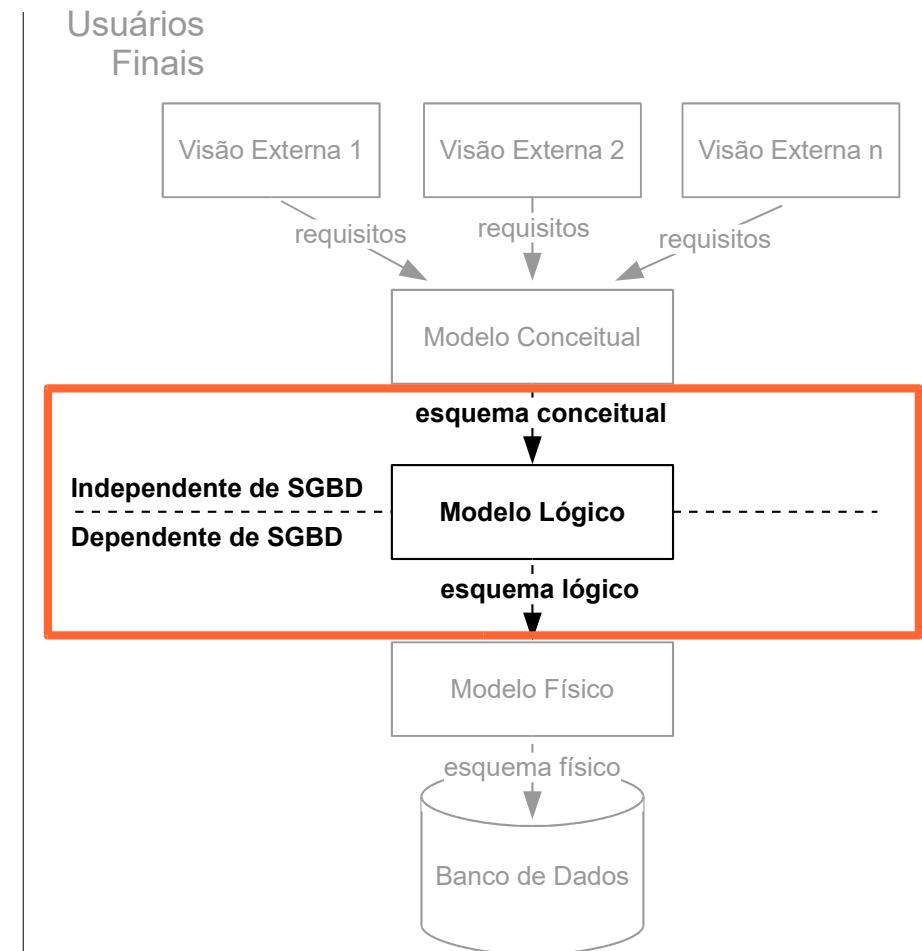
Modelo/Esquema Conceitual

- Descreve estrutura do Banco de Dados
 - entidades, tipos de dados, relações, restrições etc.
- Independente de implementação em SGBD
 - oculta detalhes de armazenamento físico



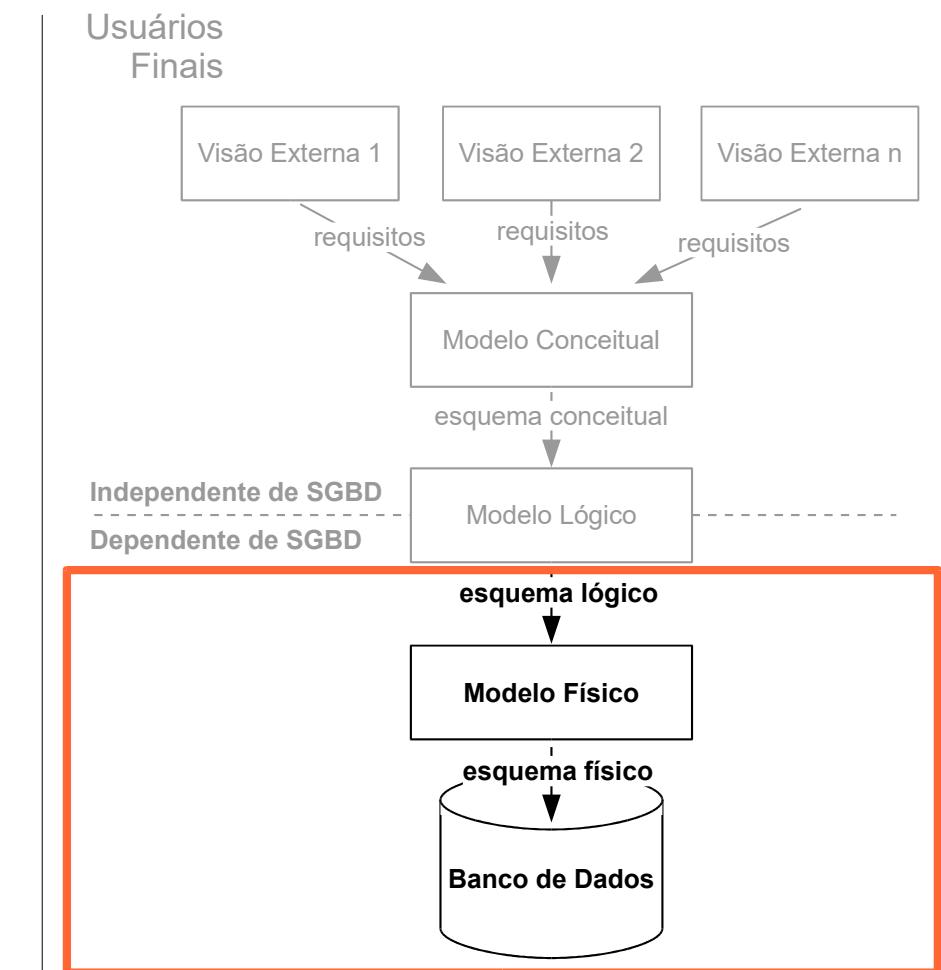
Modelo/Esquema Lógico

- Dependente de um SGBD particular
- Associado a um “modelo de dados de implementação” (Elmasri, 2005)

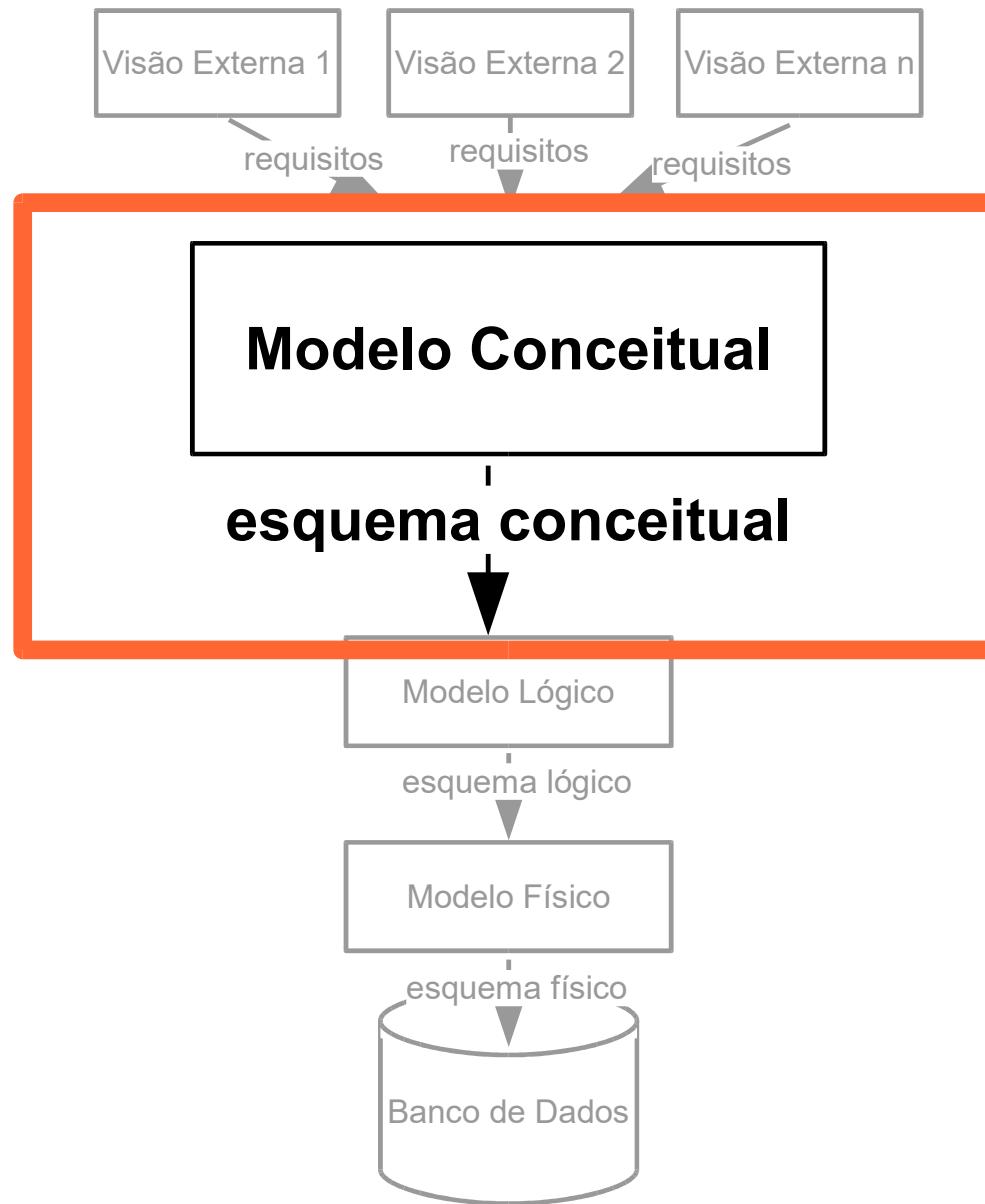


Modelo/Esquema Físico

- Descreve a estrutura de armazenamento físico



Hoje

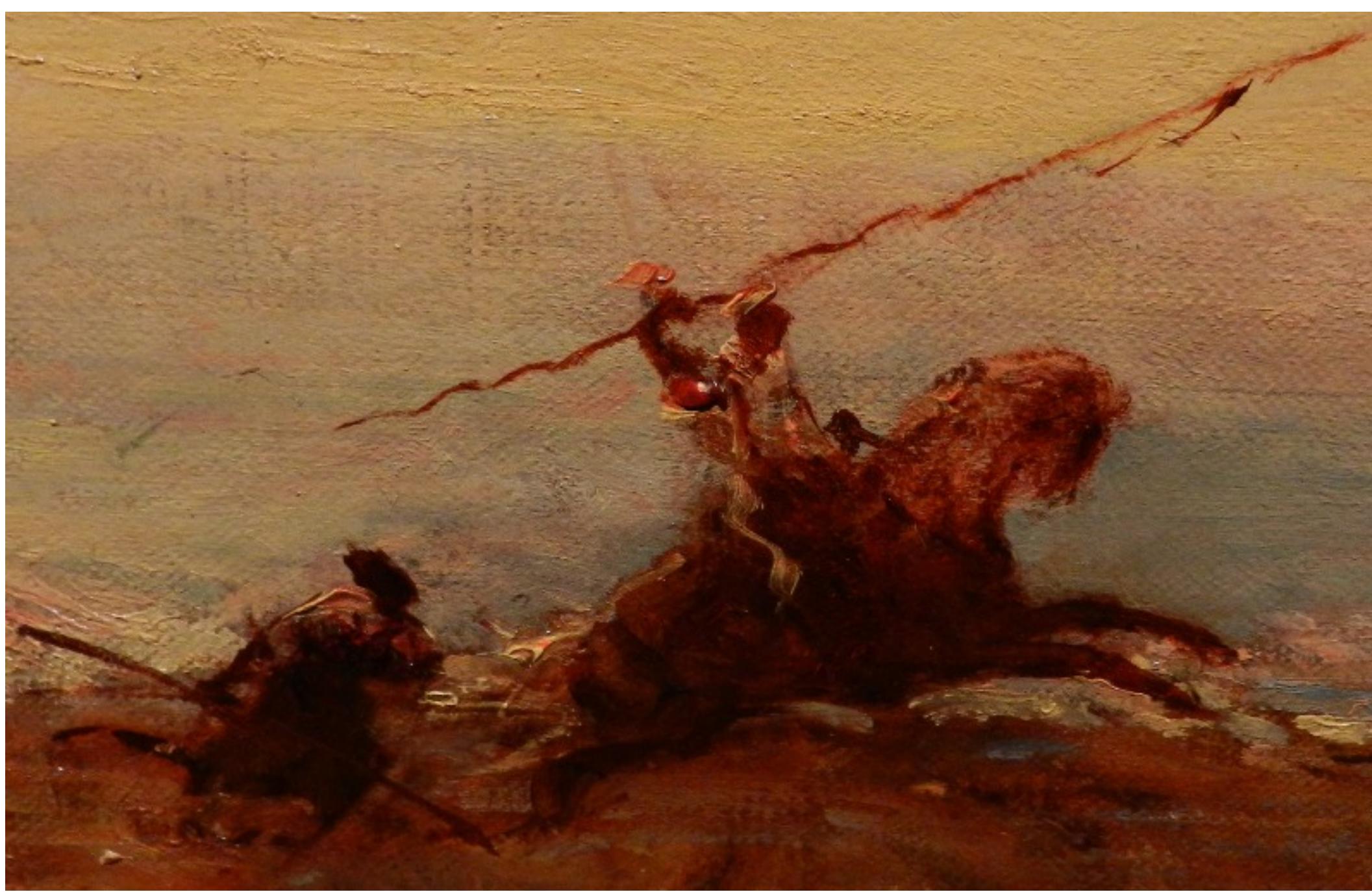


**Como modelamos o
mundo**

**Intuitivo
Entidades /
Objetos**

Fantasia à Constantinople por Felix Ziem

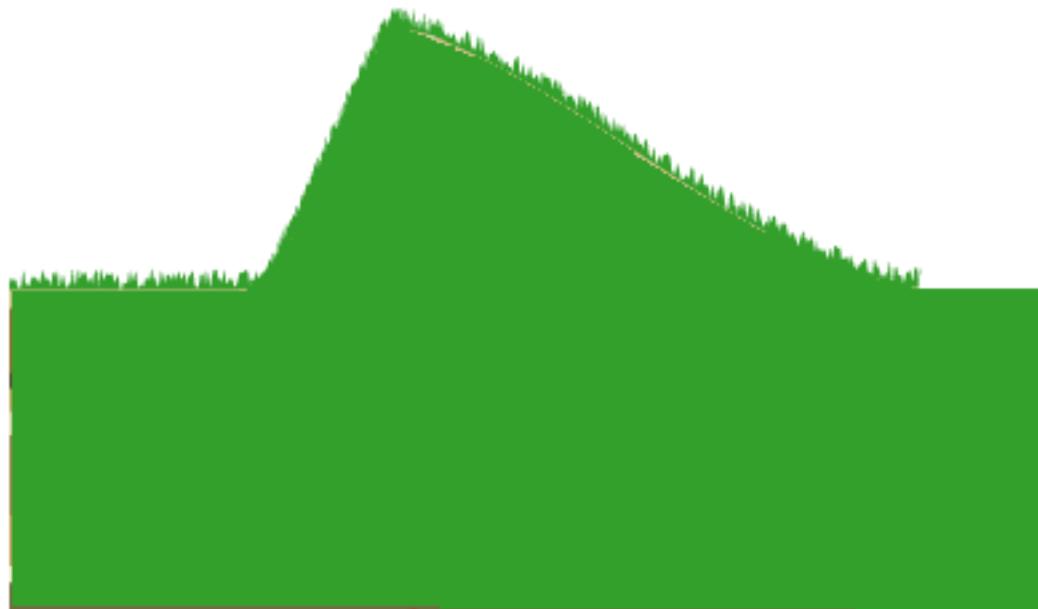






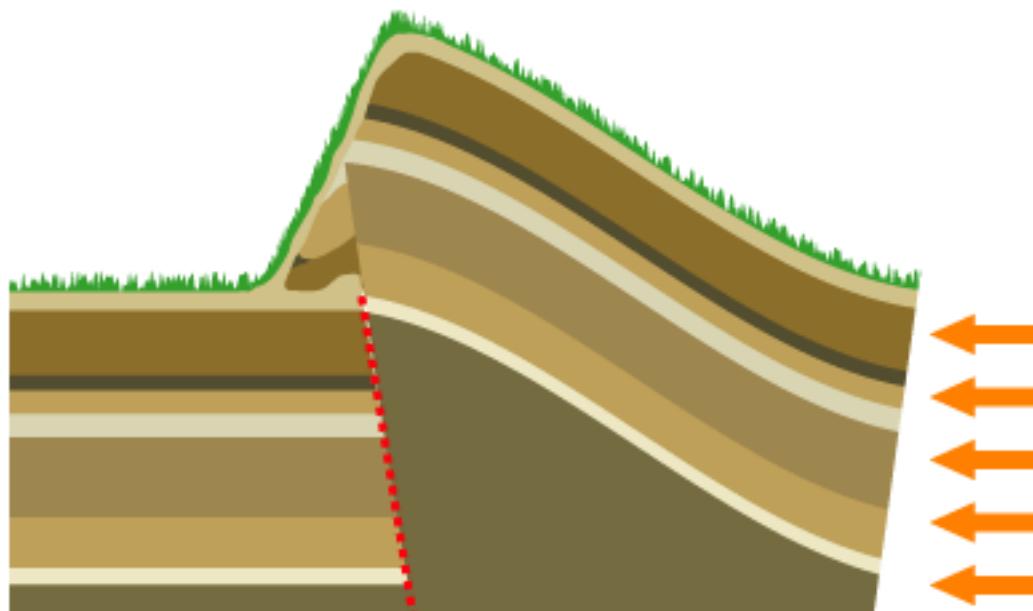
Objetos

- Montanha



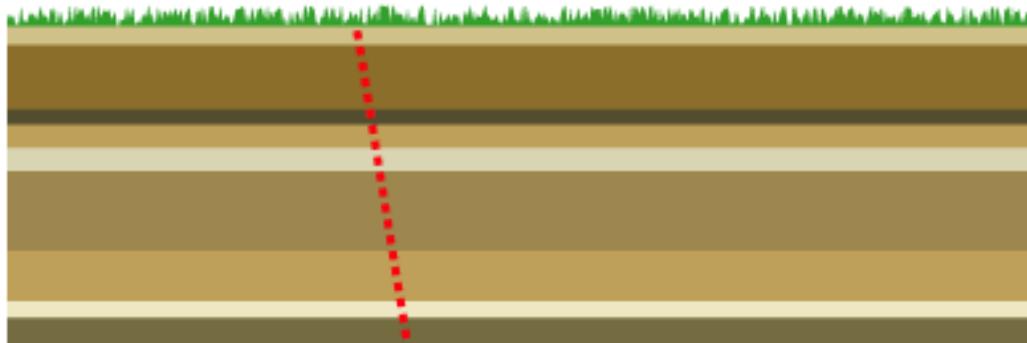
Objetos

- Montanha



Objetos

- Montanha



Objetos - Ferroviária



Noção de Objeto

- Psicologia do desenvolvimento:
 - Quando crianças representam objetos como entidades permanentes?
 - Que persistem:
 - Através do tempo e espaço
 - À oclusão

(Santos & Hood, 2009)

Noção de Objetos

- Objetos permanecem?
 - “Of course, the concept of object permanence itself is really a misnomer, as all objects comprise energy in continuous states of change.” (Santos & Hood, 2009)

**Formal
Entidades /
Objetos**

Modelo Entidade-Relacionamento (ER)

Modelo Entidade-Relacionamento

- Padrão para modelagem conceitual
- Criada por Peter Chen em 1976

Entidade

Entidade

- Objeto do universo de discurso
- Identificável distintamente
- Existência independente



Pessoa



Livro

Departamento
de uma
Organização

Categoria
de um
Livo

Modelo Orientado a Objetos (OO)

Modelo Orientado a Objetos

- SIMULA 67
 - Primeira Linguagem Orientada a Objetos
- Smalltalk
 - Projeto Dynabook
 - “Este ‘Dynabook’ foi baseado na visão de computadores pessoais baratos do tamanho de um caderno, tanto para adultos quanto crianças, com a capacidade de lidar com todas as suas respectivas necessidades de informação”. [KRE98]

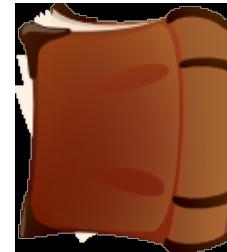
Objeto

● Objeto

- Objeto do universo de discurso
- Identificável distintamente
- Existência independente



Pessoa



Livro

Departamento
de uma
Organização

Categoria
de um
Livo

● Objeto



Pessoa



Livro

Departamento
de uma
Organização

Categoria
de um
Livo

Objetos são caracterizados por:

- identidade;
- atributos;
- comportamento.

Exemplo de Objeto

Esfera Vermelha

Objeto Esfera	
Atributos (nome, valor)	
	(peso, 200 g)
	(raio, 60 cm)
	(elasticidade, alta)
	(cor, vermelha)
Comportamento	
aumentar, diminuir, se mover	

Estereótipos / Classes

Modelo

Generalização

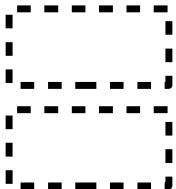
Instancias

Universo de
Discurso

Rhacophytales



color
height



Rhacophytales (1)



color



height

13

Rhacophytales (2)



color



height

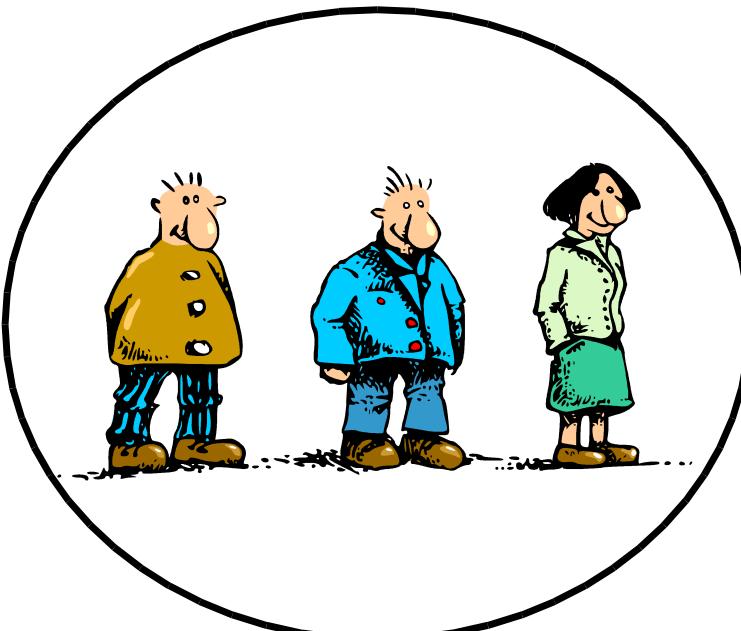
10



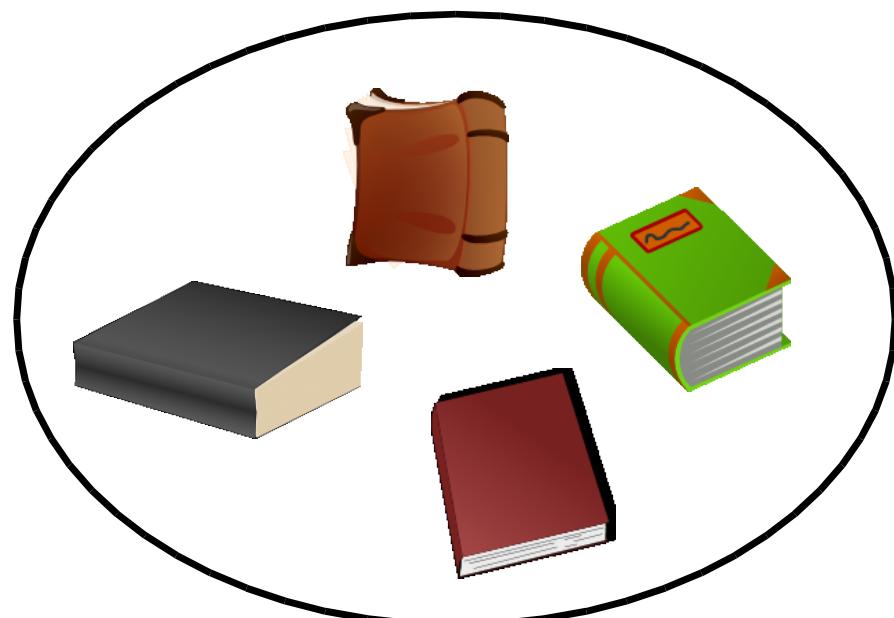
ER: Tipo Entidade

Tipo Entidade

- Tipo Entidade ou Conjunto de Entidades
 - conjunto não disjunto
 - entidades similares – mesmos atributos



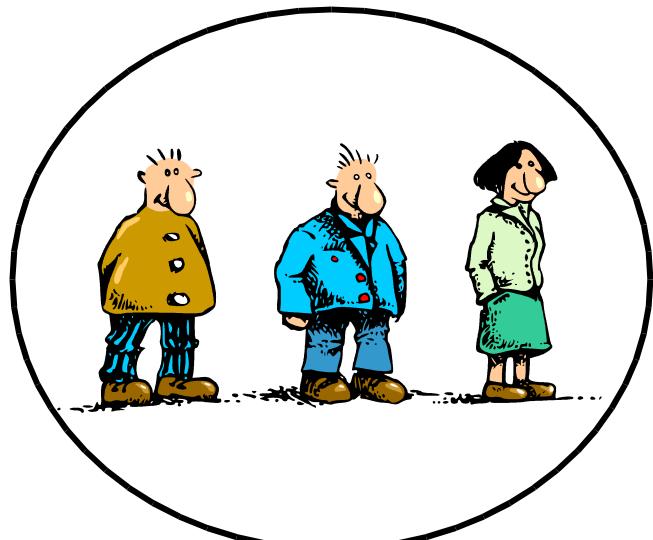
Conjunto
de Pessoas



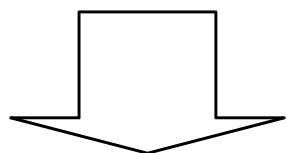
Conjunto
de Livros

Tipo Entidade

- Representação:



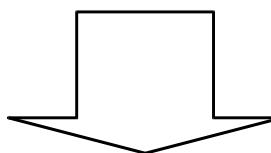
Conjunto
de Pessoas



Pessoa

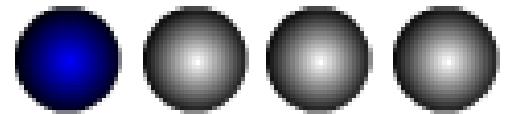


Conjunto
de Livros



Livro

OO: Classe



Classe

"Numa série ou num conjunto, grupo ou divisão que apresenta características ou atributos semelhantes." (Ferreira, 1989)

- Classificação de Carl Linné



Amphibia



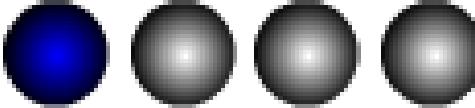
Reptilia



Aves

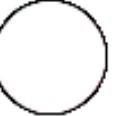
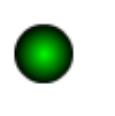


Mammalia



Classe

- Quando realizamos uma classificação de objetos, identificamos o seu comportamento e as características que eles possuem em comum.
- Classes definem:
 - Atributos que irão descrever o objeto;
 - Métodos que definem o comportamento dos mesmos.

Classe	Objeto	Objeto	Objeto
 peso raio cor	 peso: 200 g raio: 60 cm cor: vermelha	 peso: 200 g raio: 60 cm cor: azul	 peso: 50 g raio: 30 cm cor: verde

Objetos e Classes

- Os objetos são organizados/divididos em grupos chamados classes.
- Objetos da mesma classe têm:
 - o mesmo conjunto de atributos (os valores dos atributos podem ser diferentes);
 - o mesmo conjunto de métodos.

UML

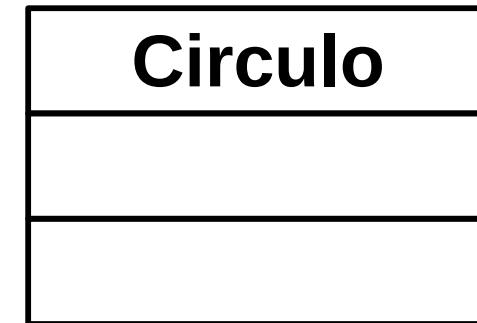
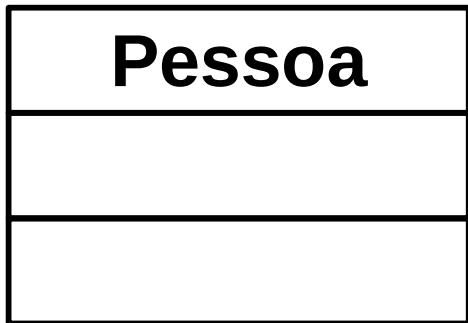
Unified Modeling Language

- <http://www.uml.org/>
- Desenvolvida entre 1994-96
- Criadores
 - Grady Booch, Ivar Jacobson and James Rumbaugh na Rational Software
- Padrão OMG em 1997
 - OMG - Object Management Group
 - <http://omg.org/>

(Wikipedia, 2015)

UML

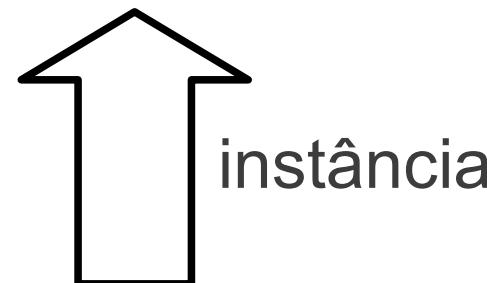
Classe



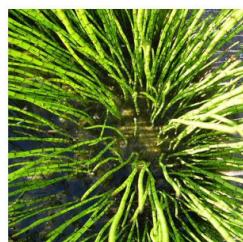
Rhacophytales



color
height



Rhacophytales (1)



color
height

Rhacophytales

color: ColorType
height: int

R1

color = green
height = 13

Exemplo de Classe

Esfera

Classe Esfera	
Atributos (nome, tipo)	
	(peso, real)
	(raio, real)
	(elasticidade, string)
	(cor, color)
Comportamento	
aumentar, diminuir, se mover	

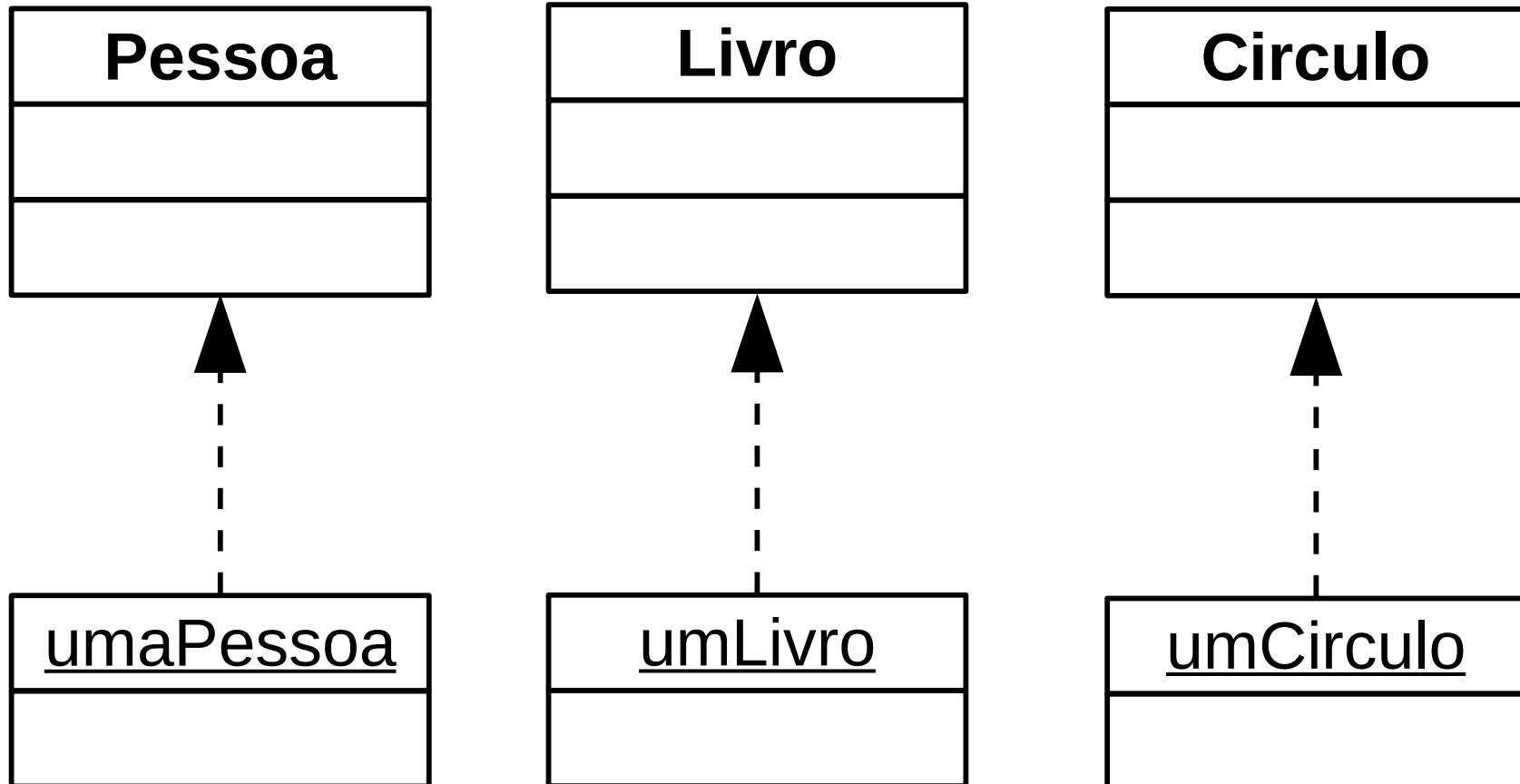
Exemplo de Objeto

Esfera Vermelha

Objeto Esfera	
Atributos (nome, valor)	
	(peso, 200 g)
	(raio, 60 cm)
	(elasticidade, alta)
	(cor, vermelha)
Comportamento	
aumentar, diminuir, se mover	

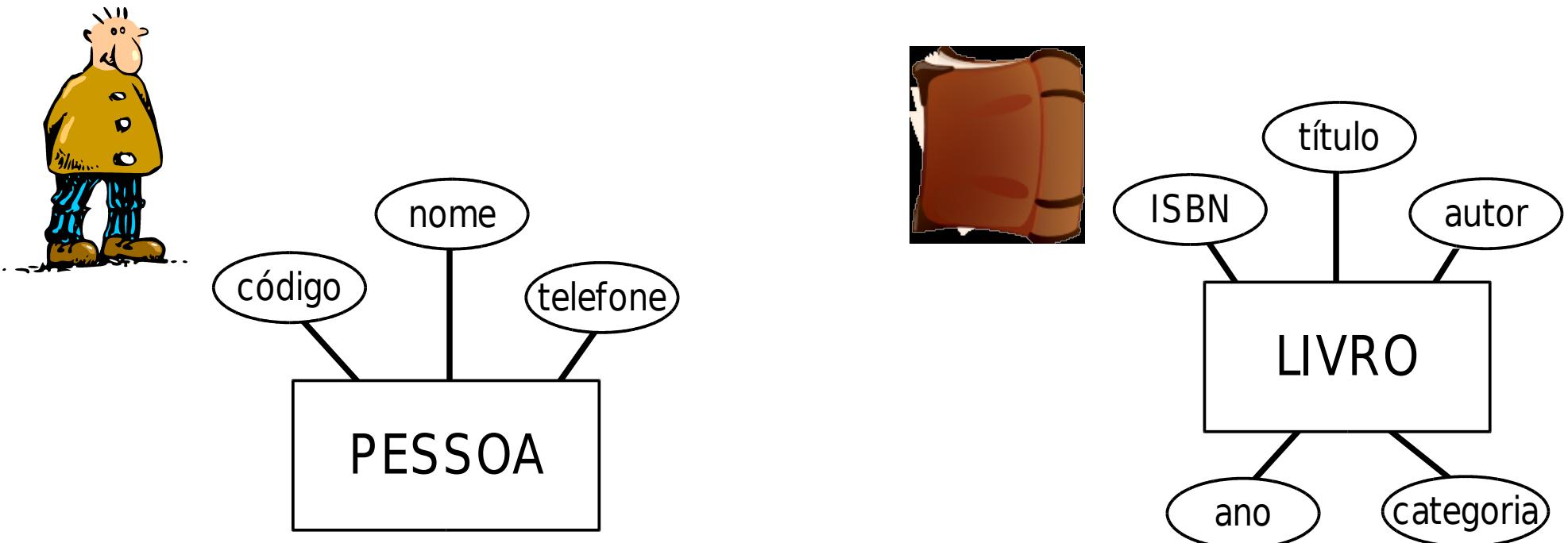
UML

Instância de Classe



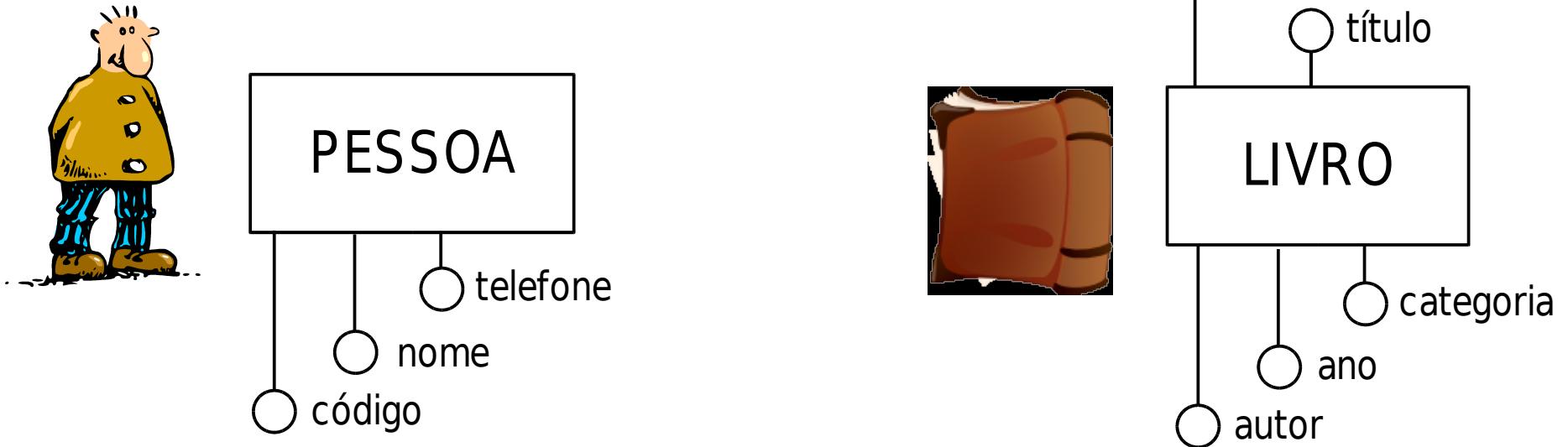
Atributos

- Cada instância de entidade ou relacionamento tem atributos que a descrevem



Atributos

Representação Alternativa



Tipos de Atributo

- Simples (atômico)

Tamanho

- Multivalorado

Autores

- Composto

Rua

Número

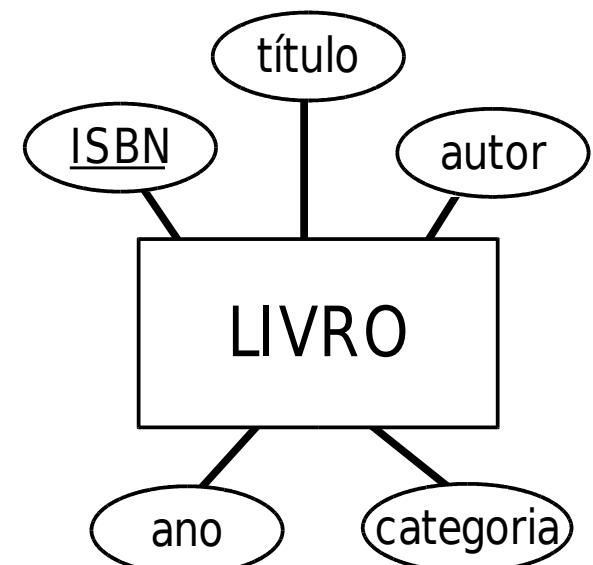
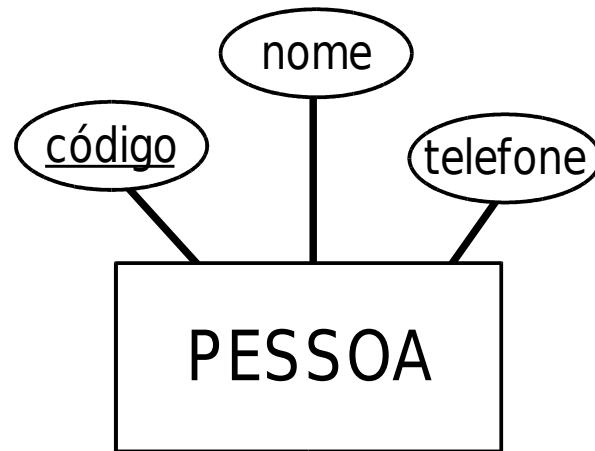
Cidade

Estado

Endereço

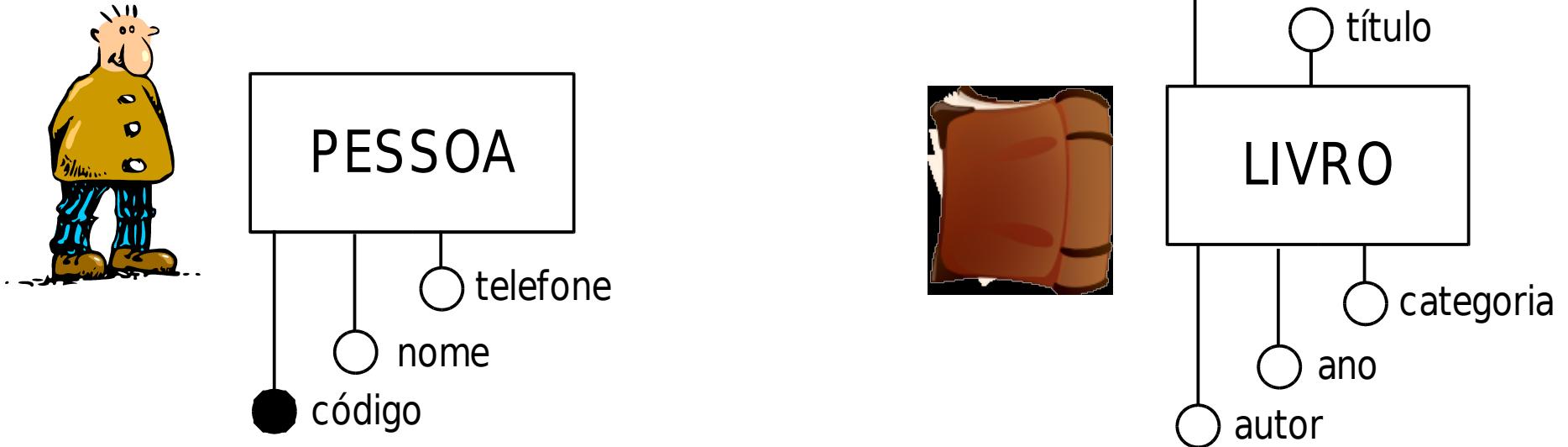
Atributos-Chave

- Servem para distinguir ocorrências da entidade
- São únicos na relação

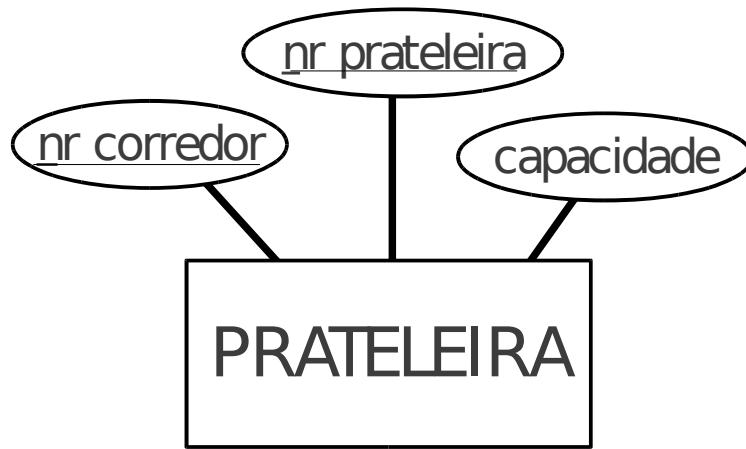


Atributos-Chave

Representação Alternativa



Atributo-chave Composto



(Heuser, 2004)

Atributos (propriedades)

Pessoa

codigo: String
nome: String
telefone: int

Livro

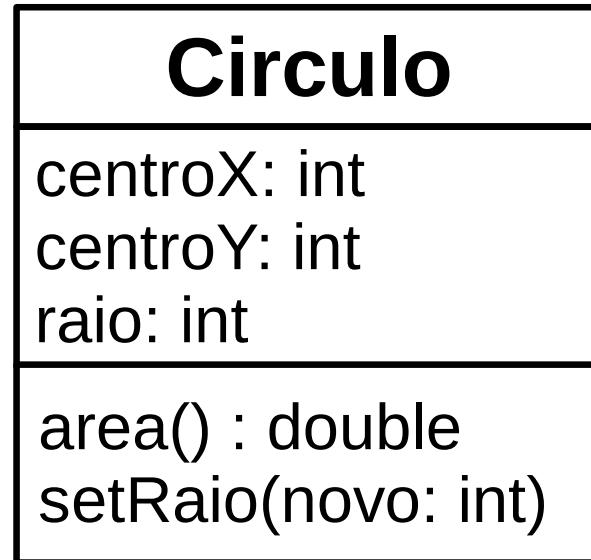
isbn: String
titulo: String
autor: String
ano: int
categoria: String

Circulo

centroX: int
centroY: int
raio: int

UML

Métodos (operações)



Exercício (parte 1)

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:

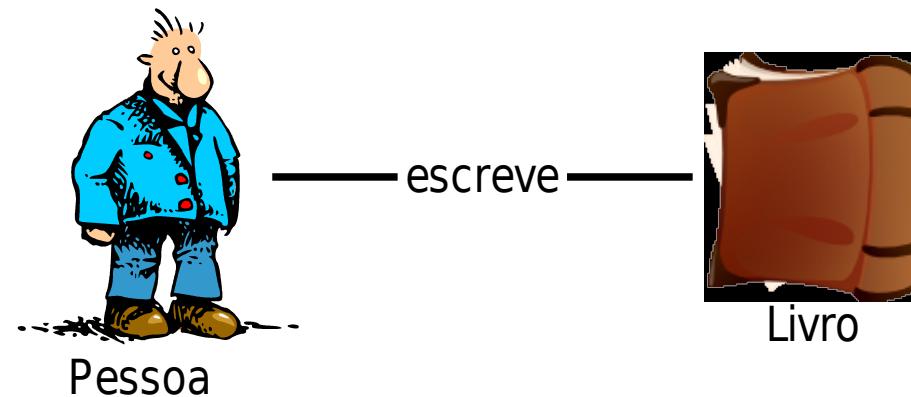
- Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.
- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.
- Em princípio considere que não há relação entre vírus e medicamentos.

Relacionamento

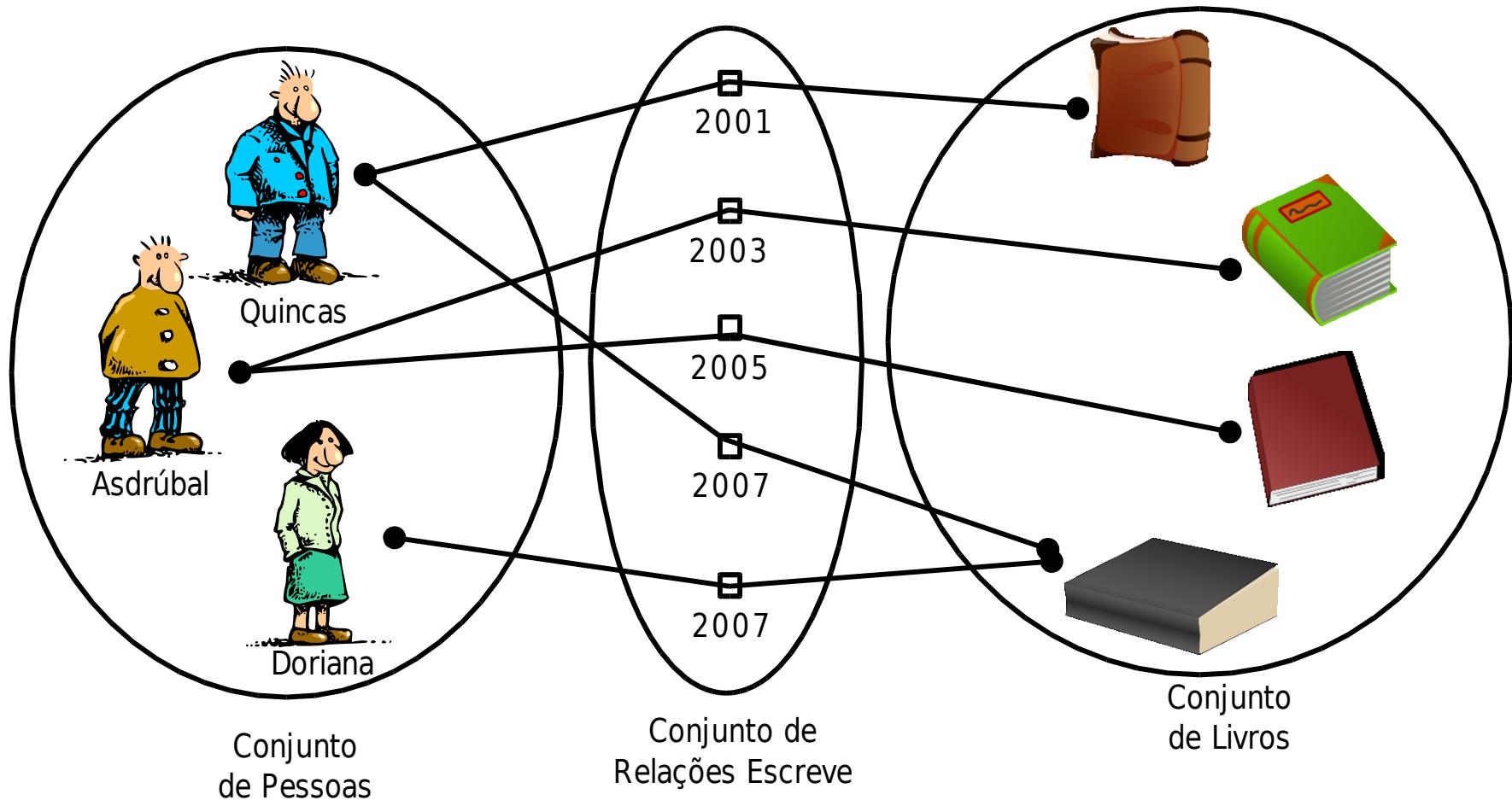
ER: Relacionamento

ER: Relacionamento

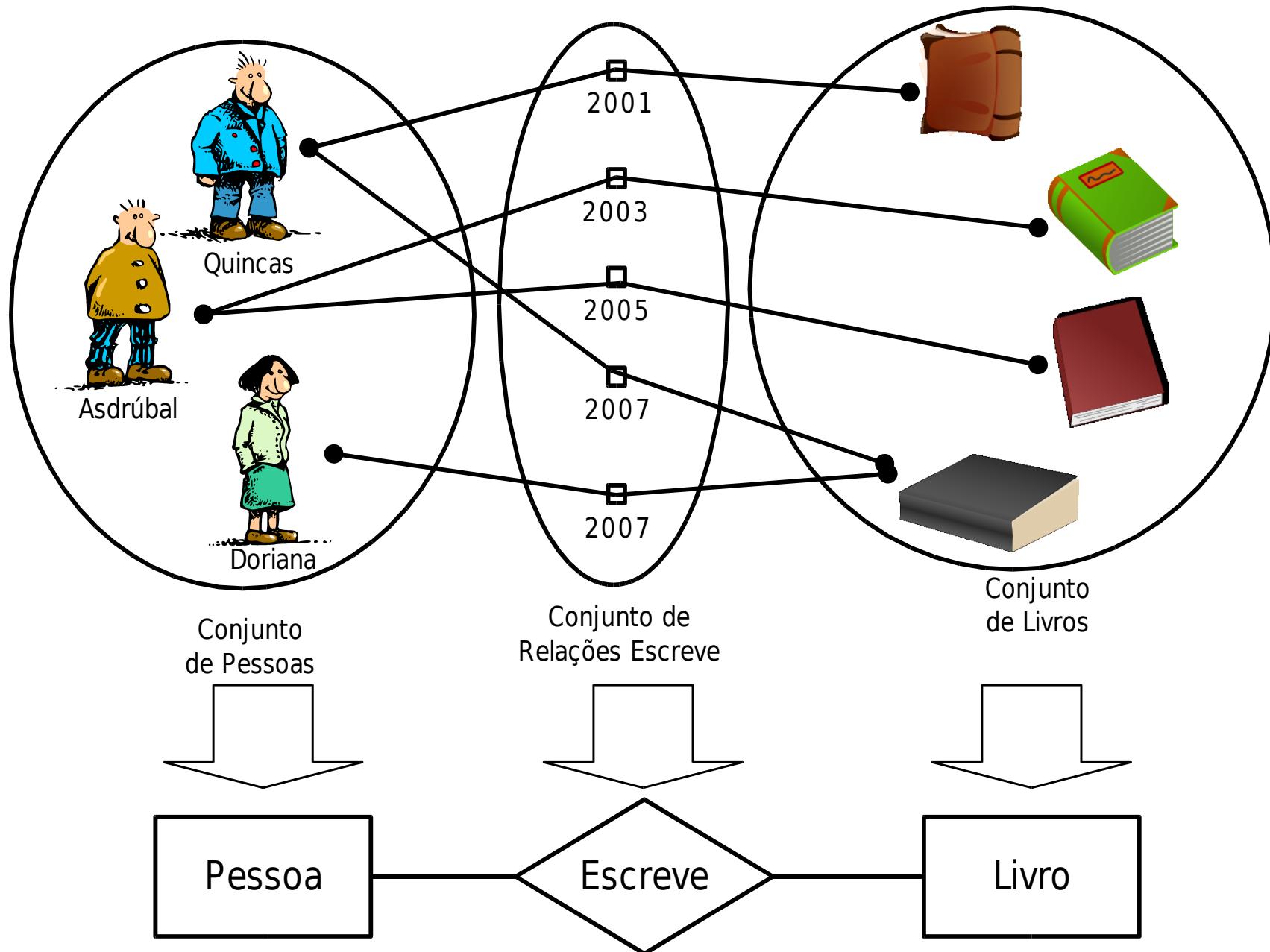
- Associação entre entidades
- Atributo de uma entidade que se refere a outra



Conjunto de Relacionamentos

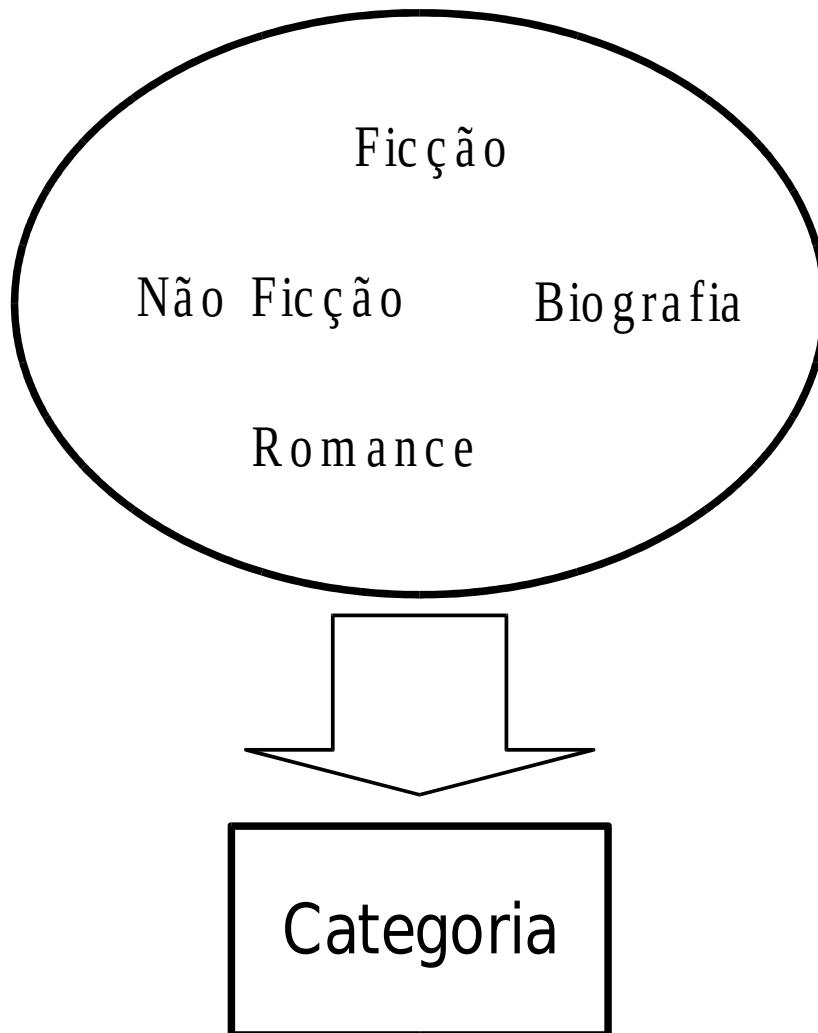


Conjunto de Relacionamentos



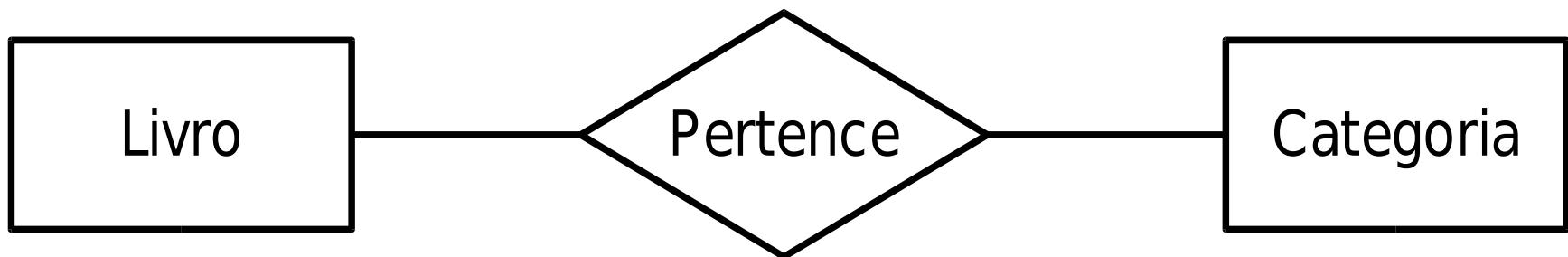
ER: Entidade

Exemplo Categoria



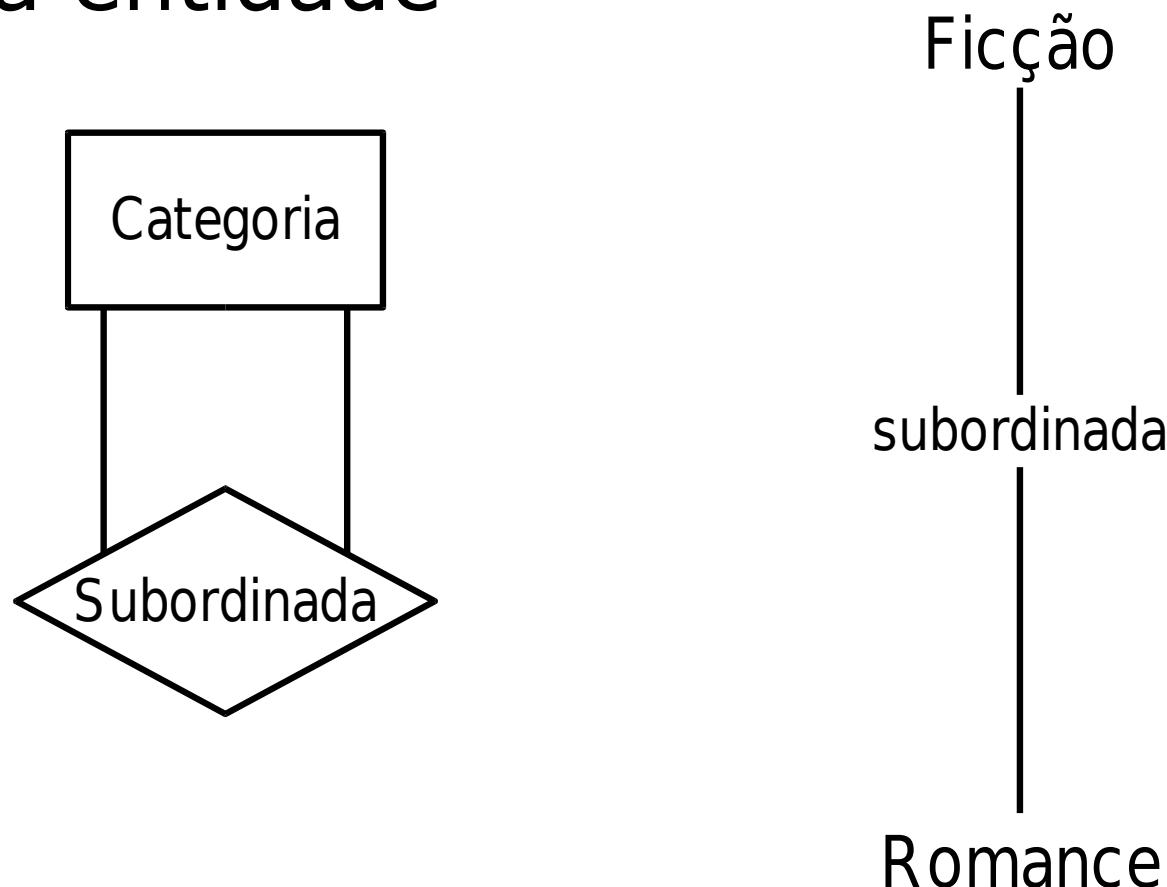
ER: Relacionamento

Exemplo Pertence



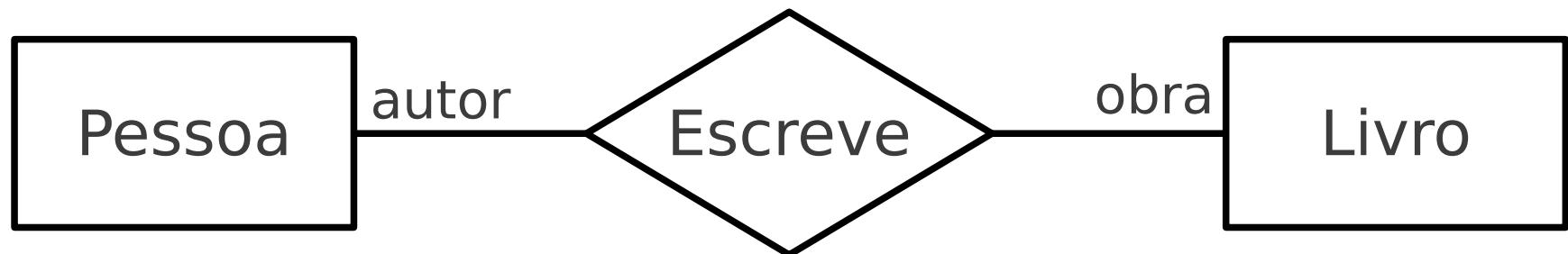
ER: Auto- Relacionamento

- Relacionamento entre ocorrências da mesma entidade

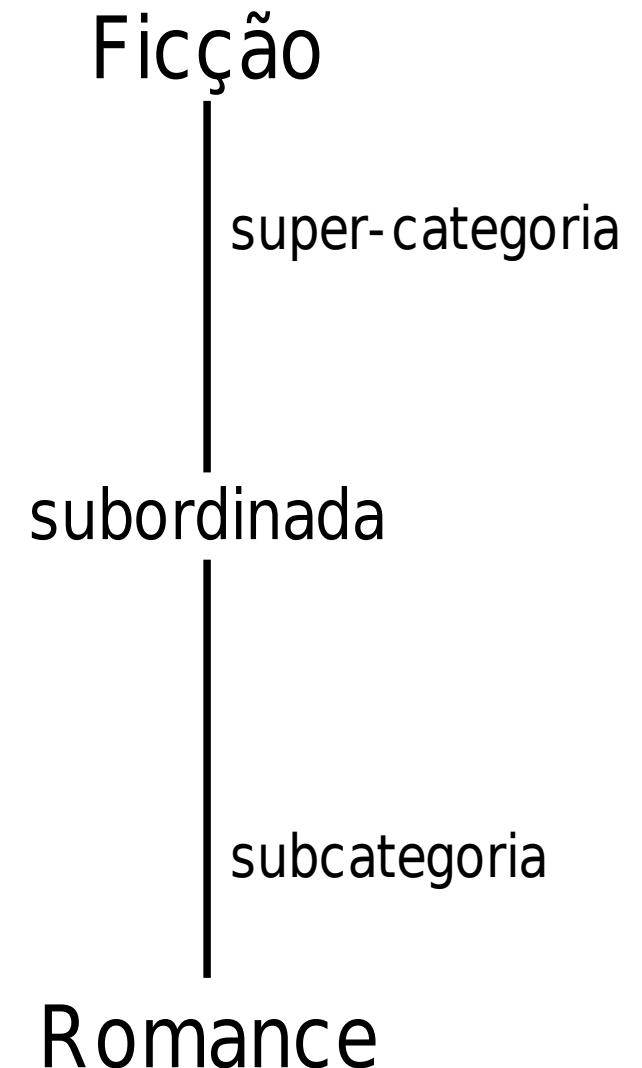
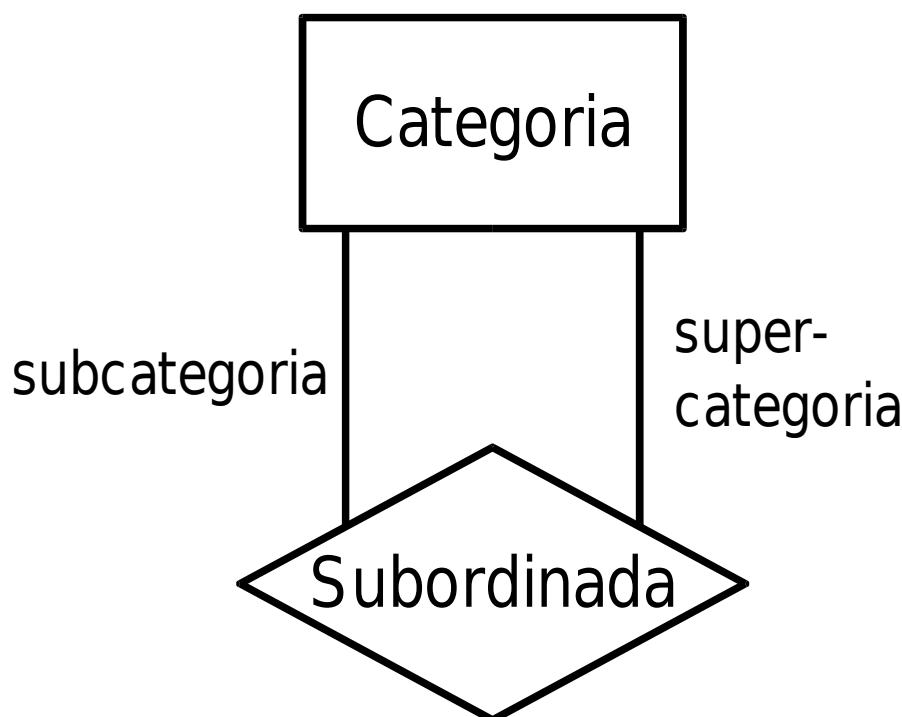


ER: Papéis

- Função que instância de entidade cumpre dentro de instância de relacionamento



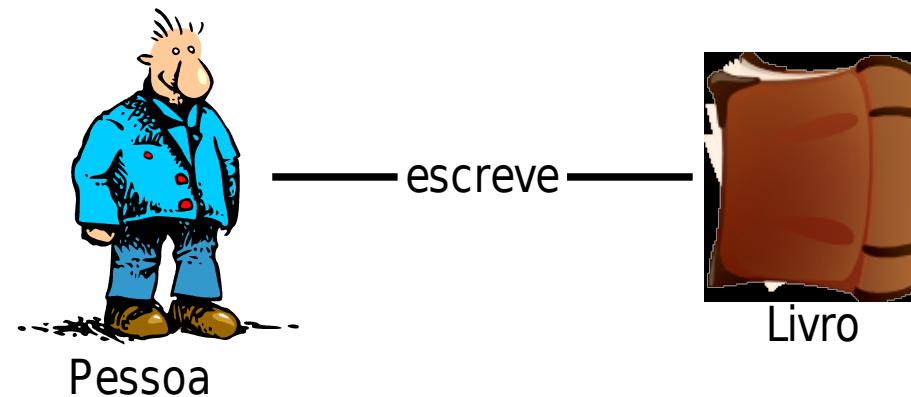
ER: Papéis



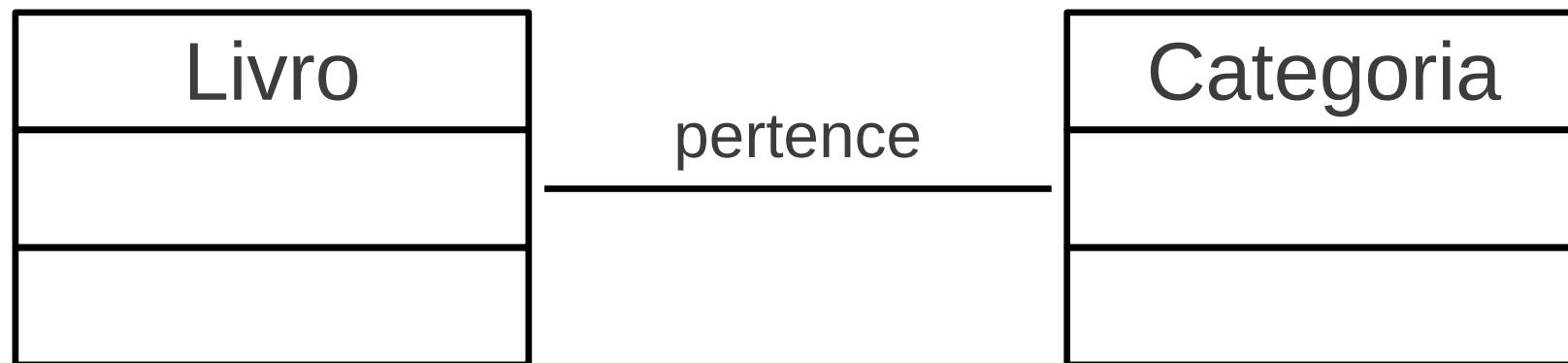
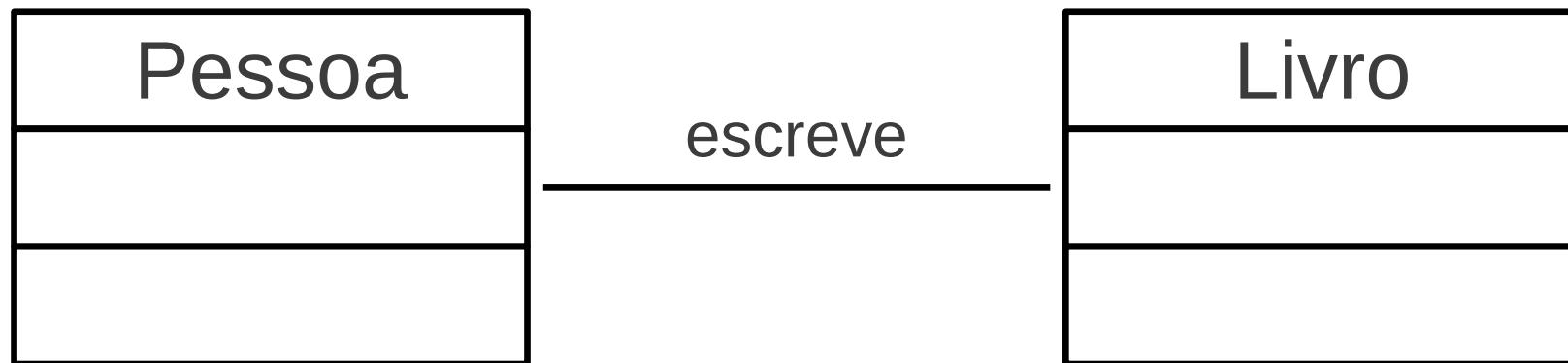
00: Relacionamento

OO: Relacionamento

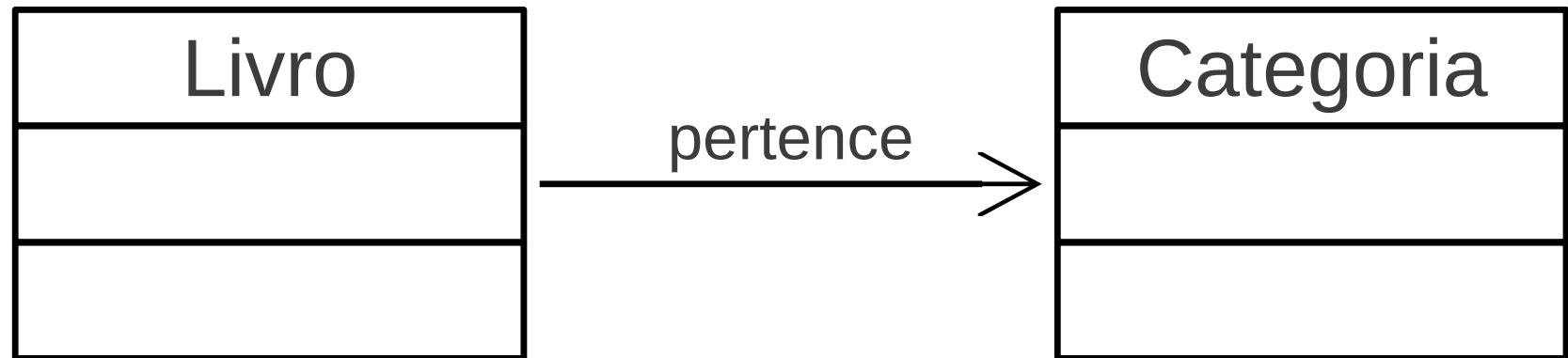
- Associação entre objetos
- Atributo de um objeto que se refere a outro
 - Atributo definido na classe



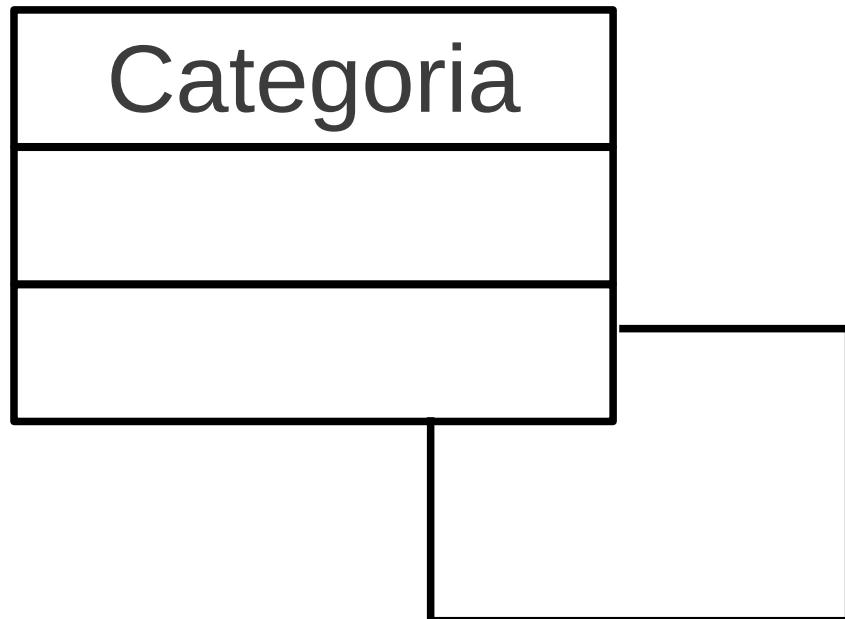
UML: Relacionamento



UML: Relacionamento Direcionado

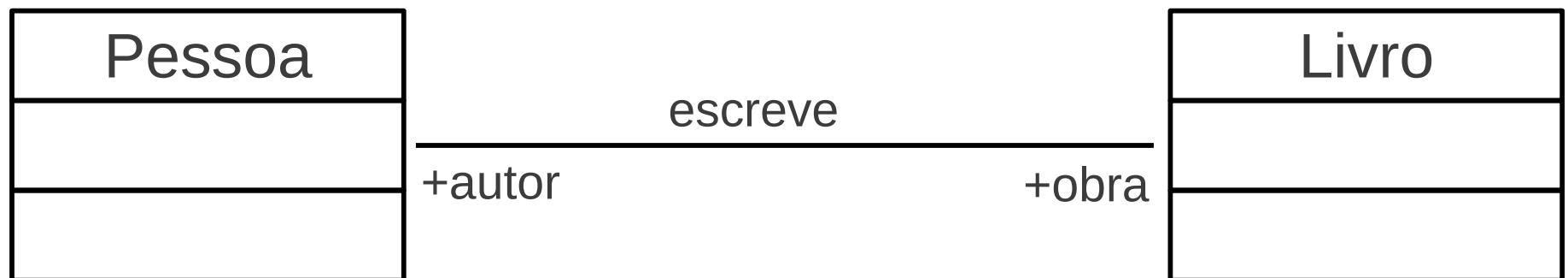
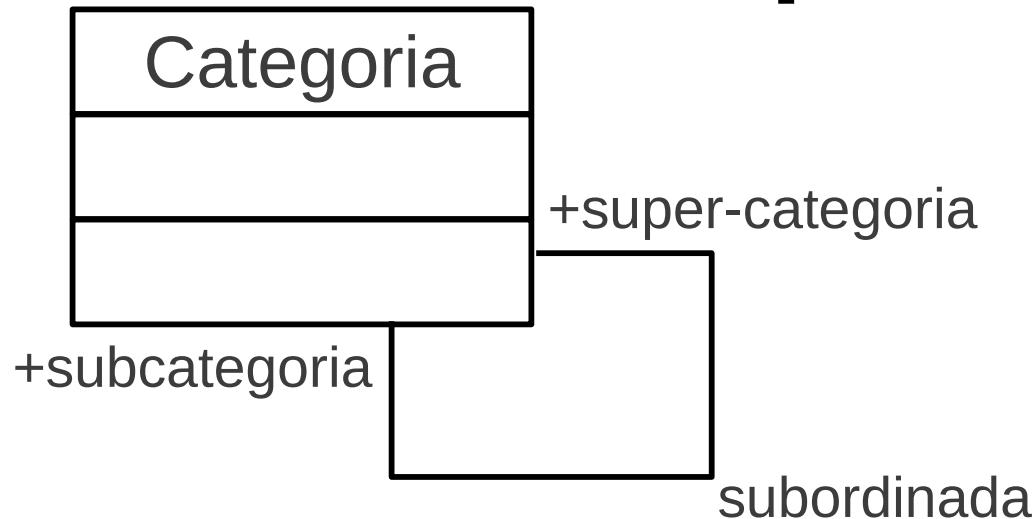


UML: Auto-relacionamiento



subordinada

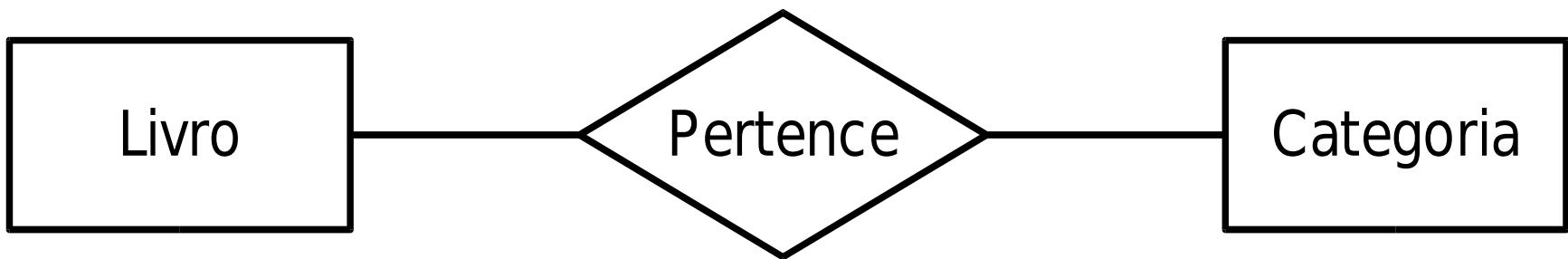
UML: Papéis



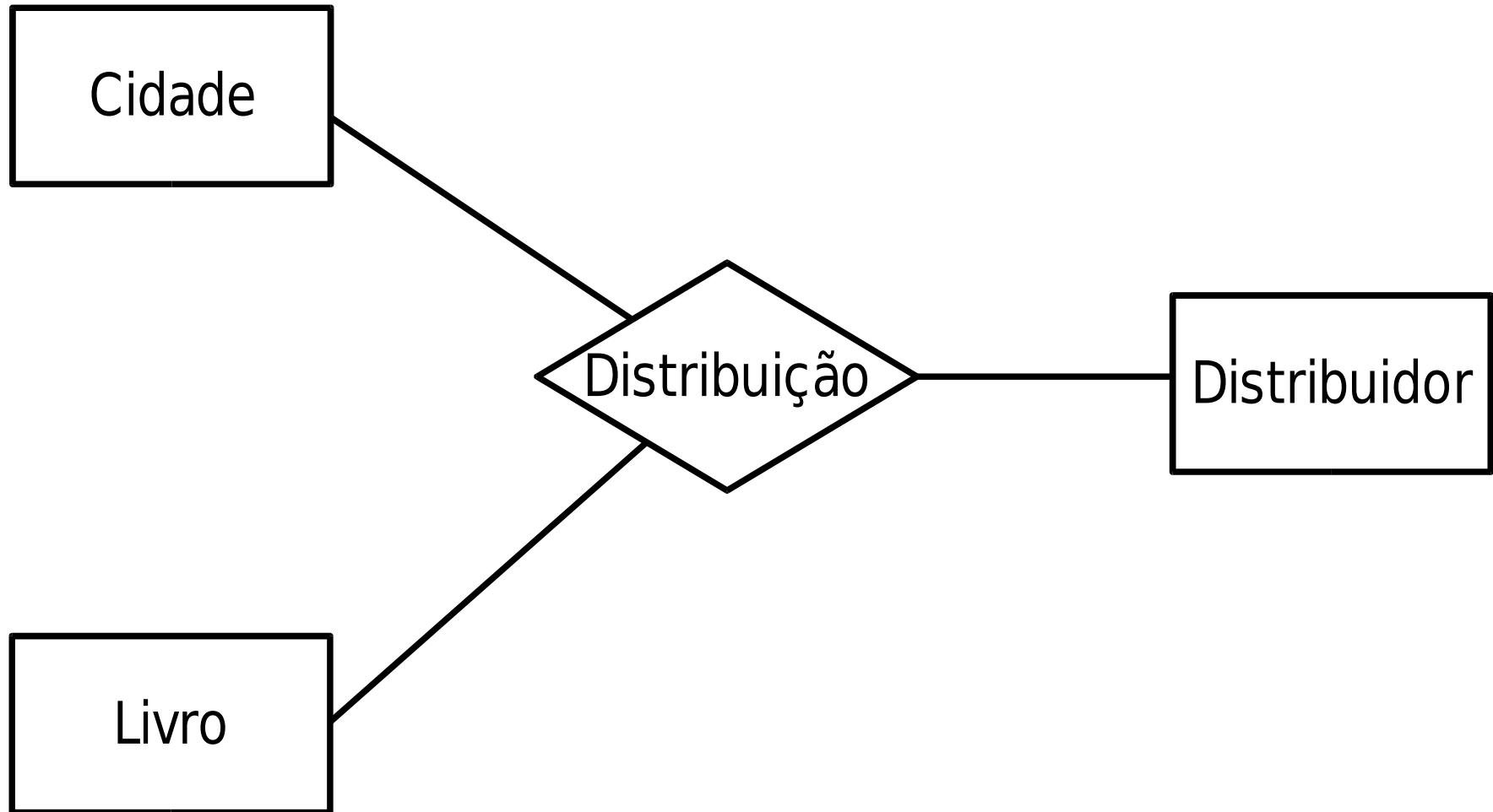
ER: Grau de Relacionamento

- Número de entidades que participam do relacionamento

Grau de Relacionamento Binário



Grau de Relacionamento Ternário



ER: Cardinalidade no Relacionamento

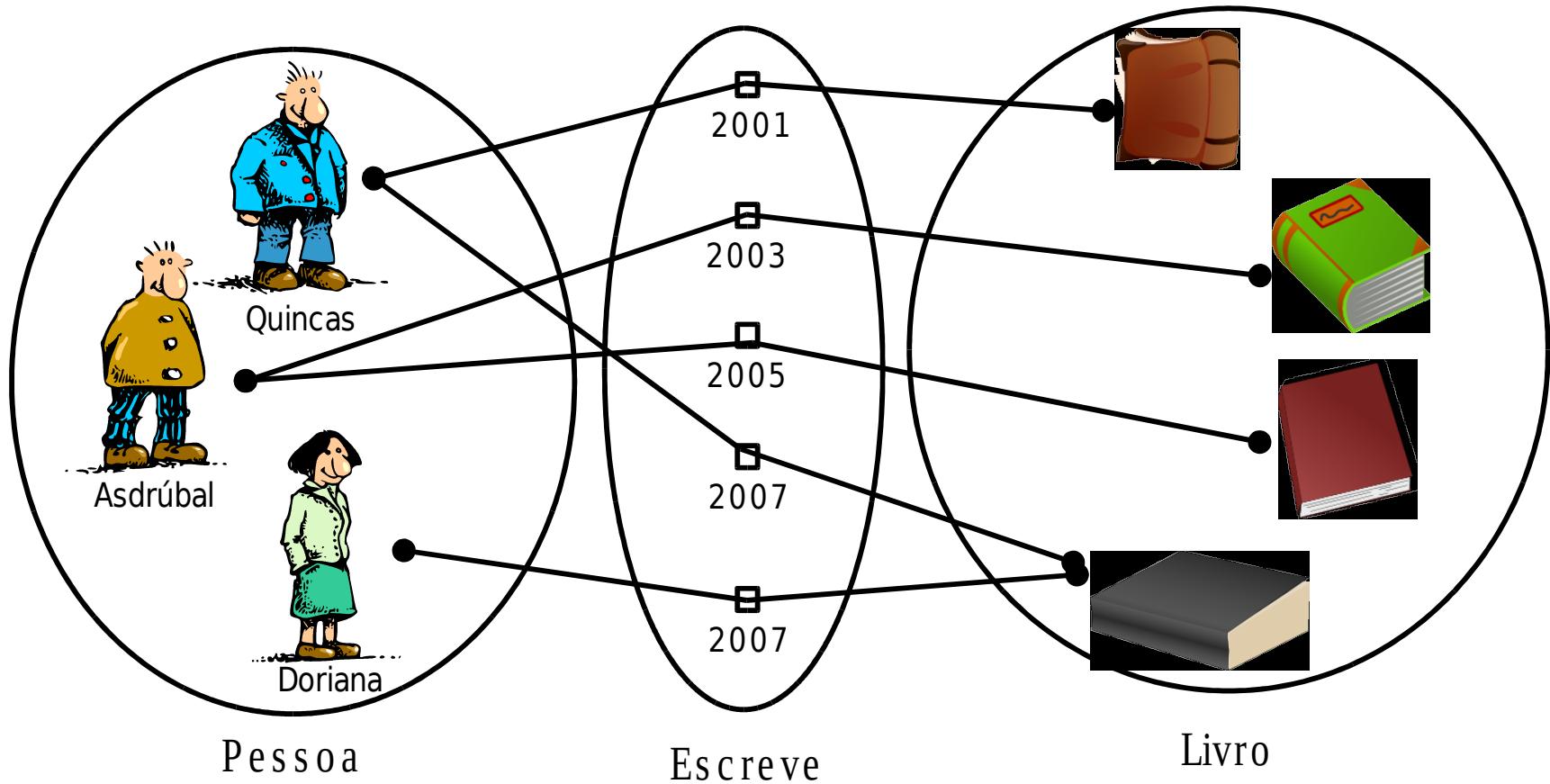
- Restrições que limitam a possibilidade de combinações de entidades em relacionamentos
- Cardinalidade:
 - Máxima
 - Mínima

Razão de Cardinalidade

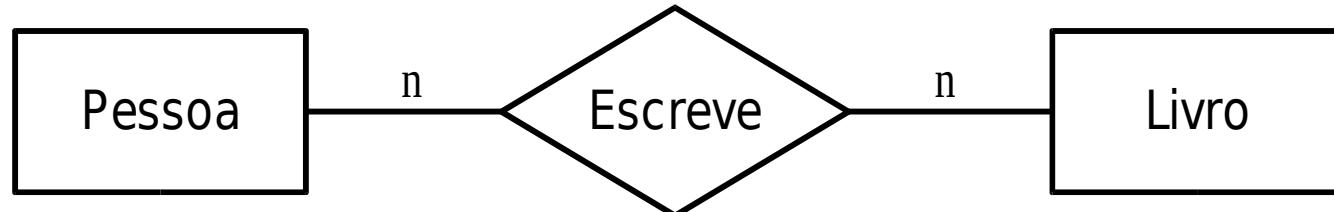
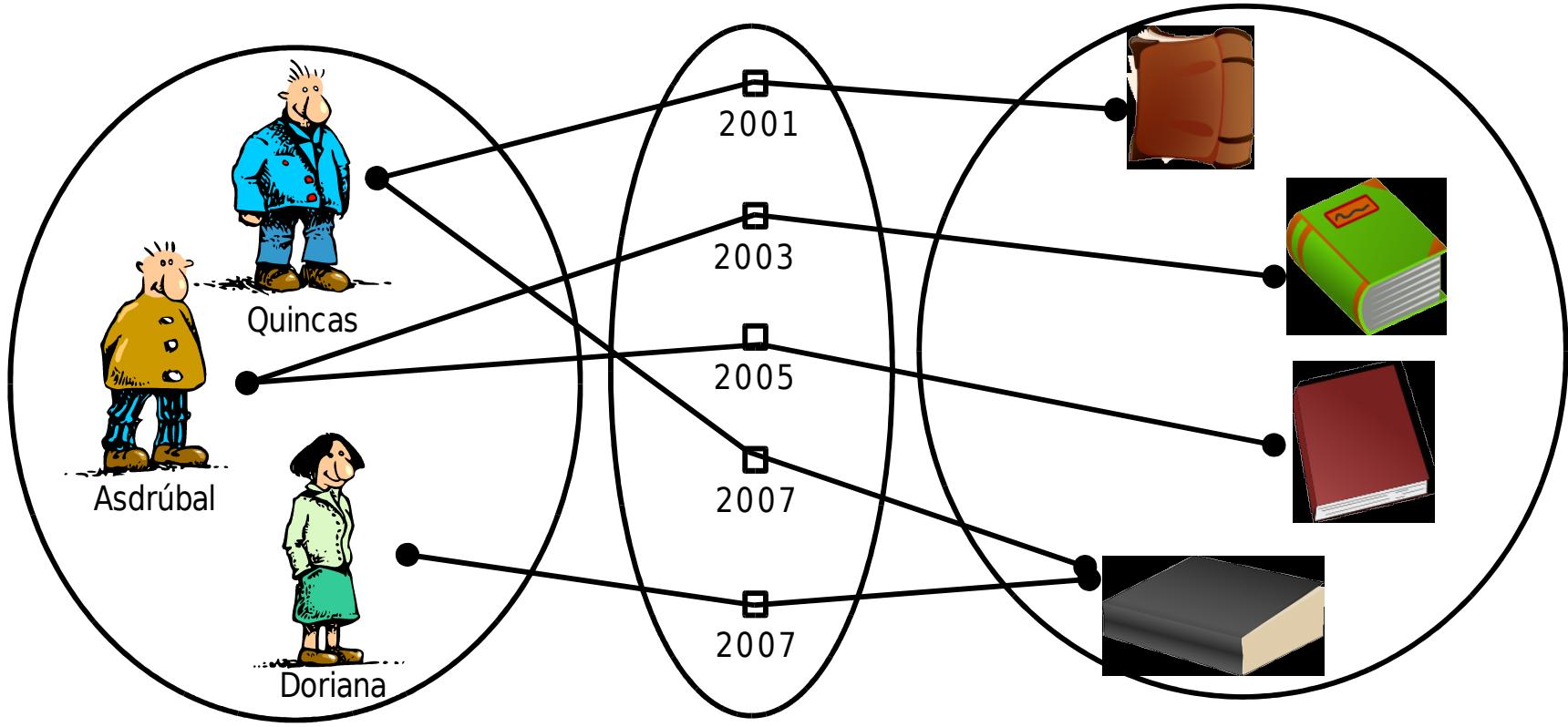
Razão de Cardinalidade

- É expressa a razão (ou proporção) de participação em um relacionamento.
- Transcrição gráfica das proporções: 1:1, 1:N, N:1 e N:N

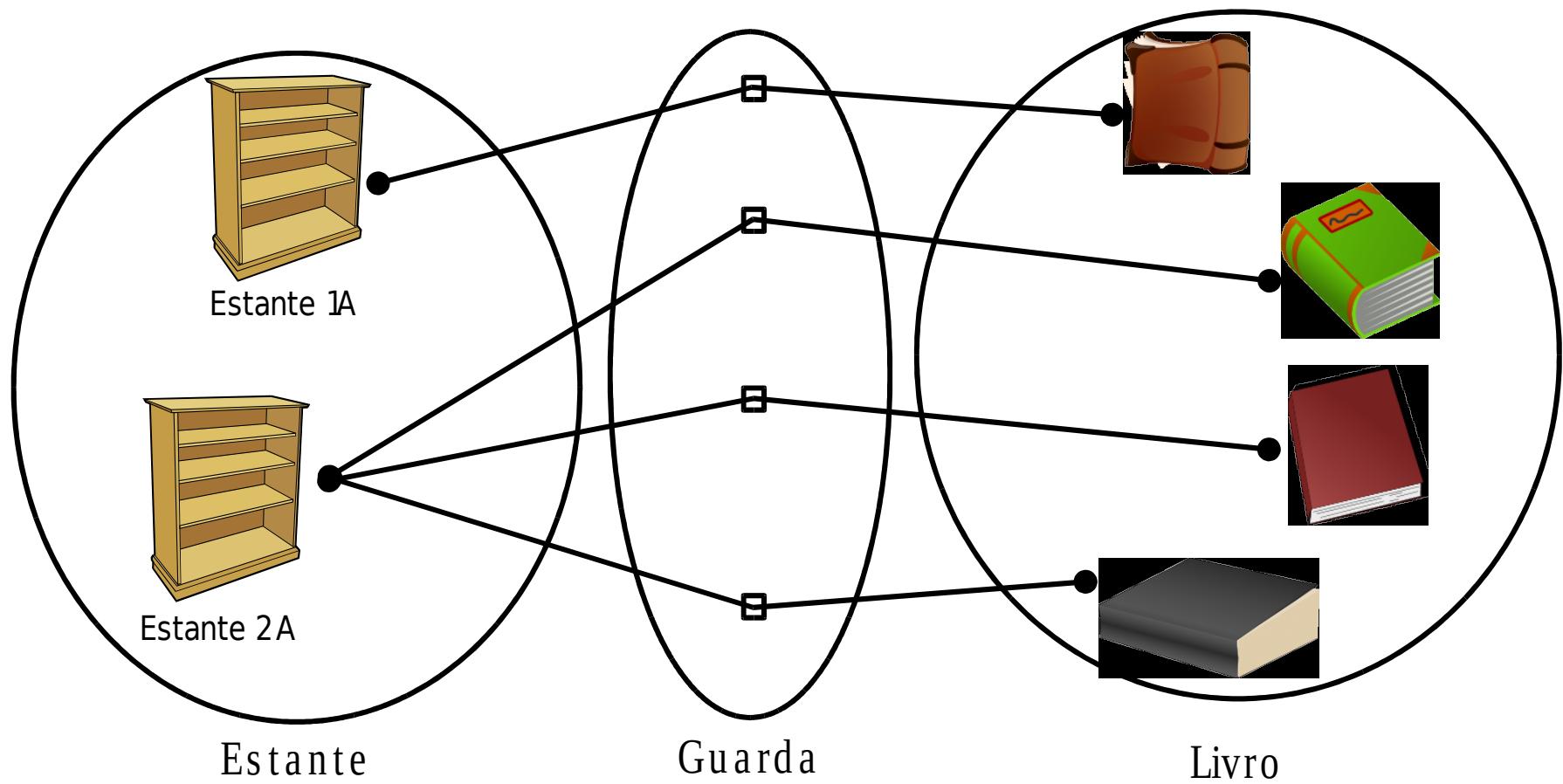
Relacionamento n:n



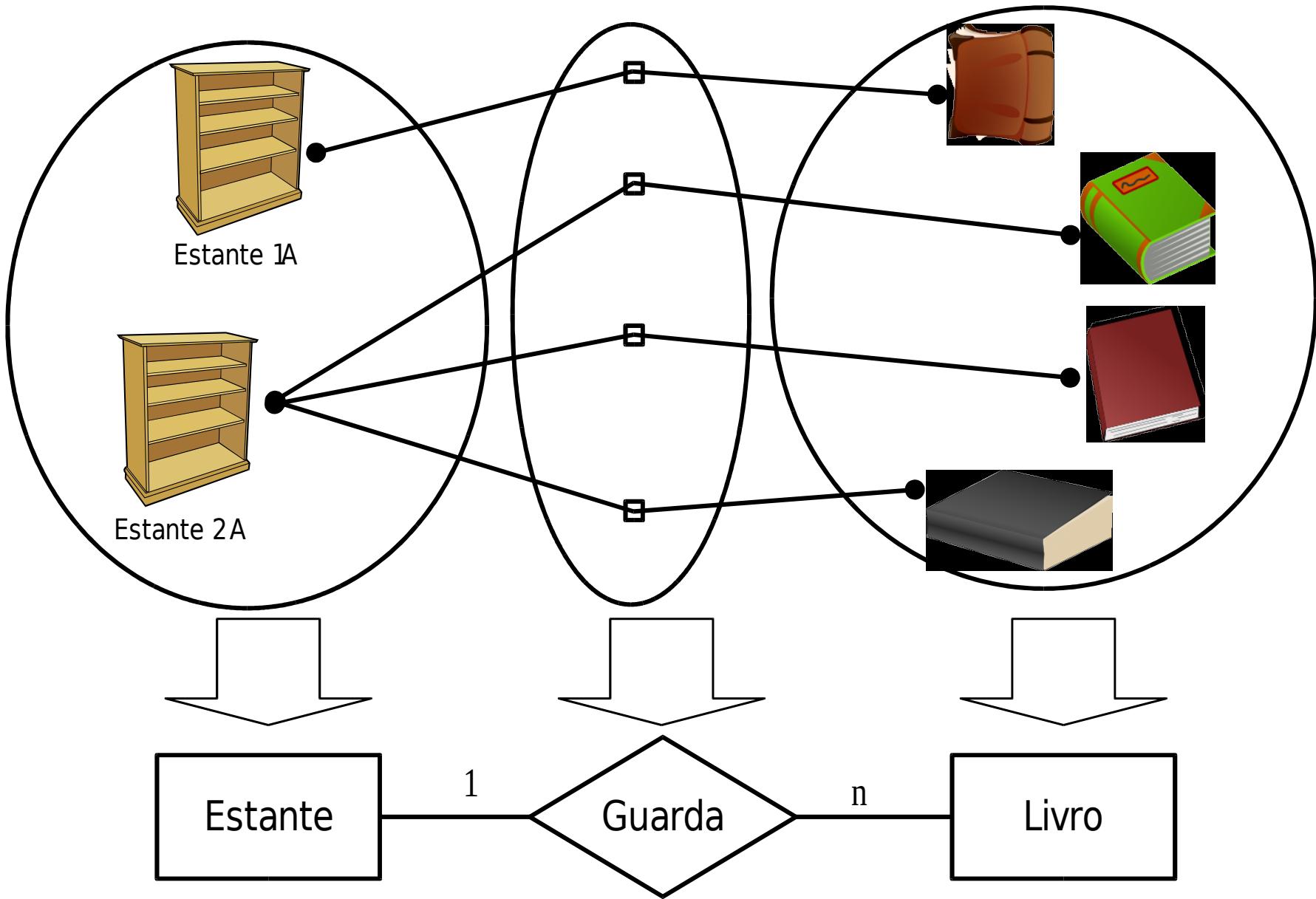
Relacionamento n:n



Relacionamento 1:n

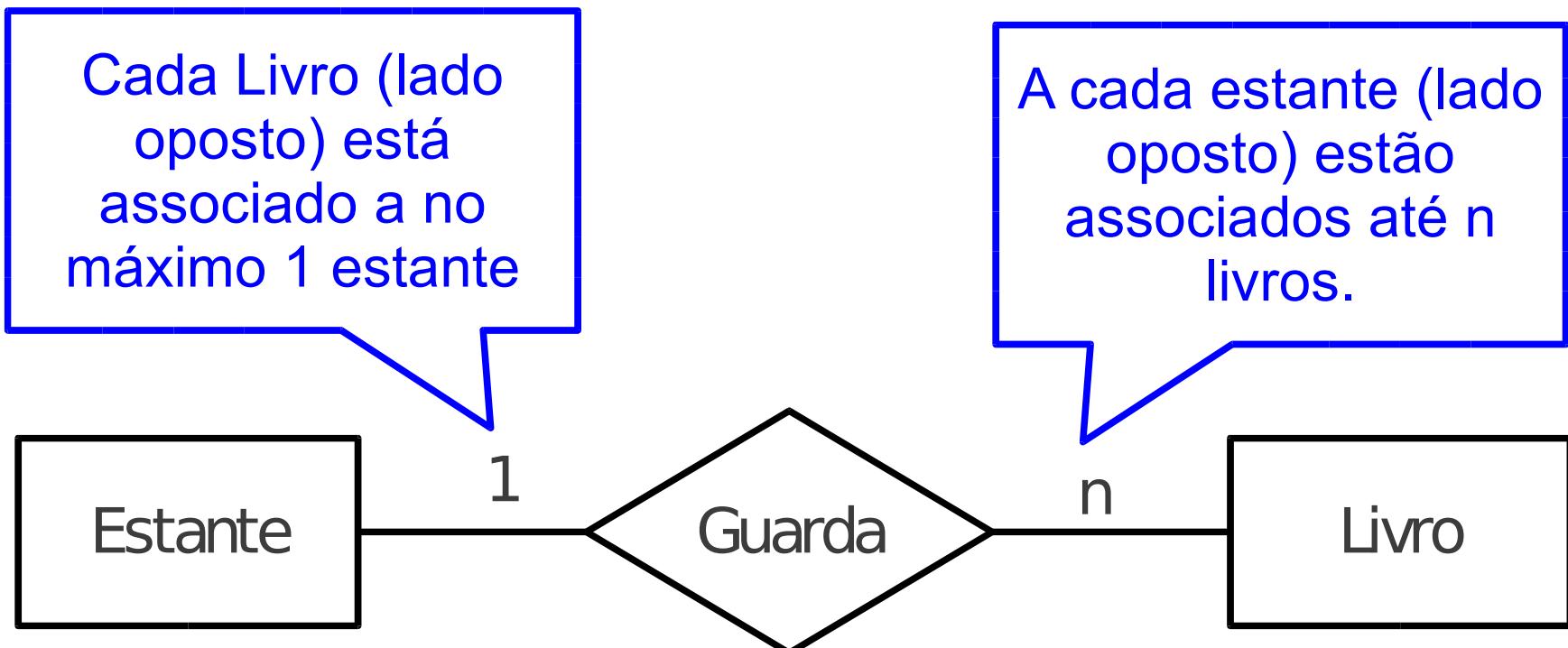


Relacionamento 1:n



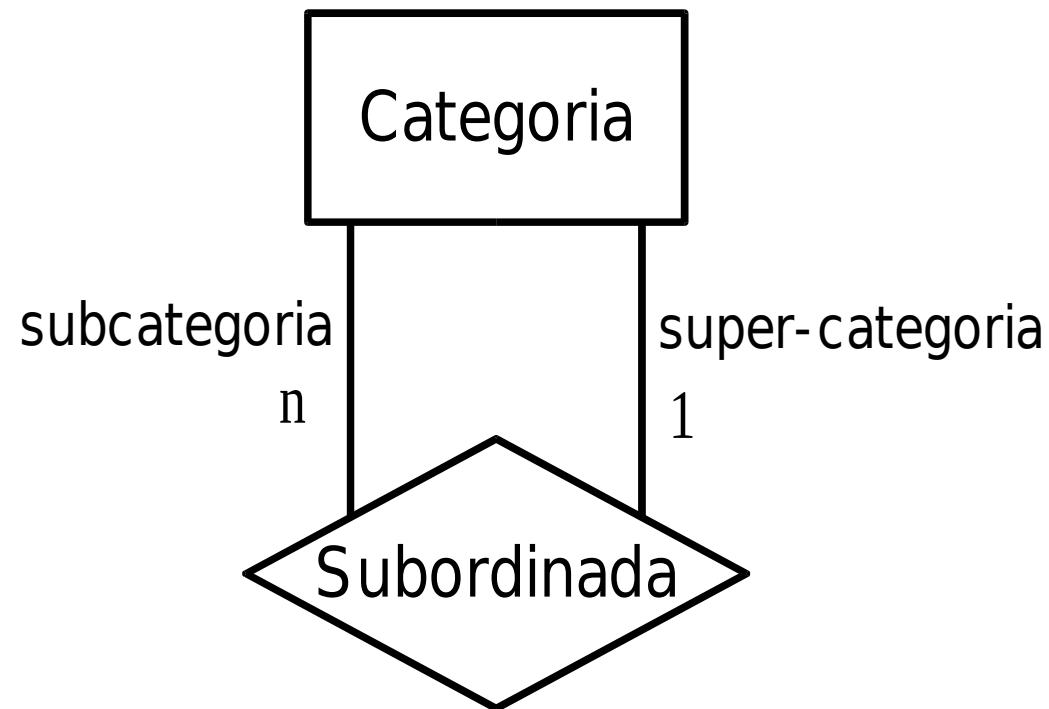
Notação de Cardinalidade

- A notação com apenas um valor de cada lado representa a razão (ou proporção) na participação. Abaixo, proporção 1:N.

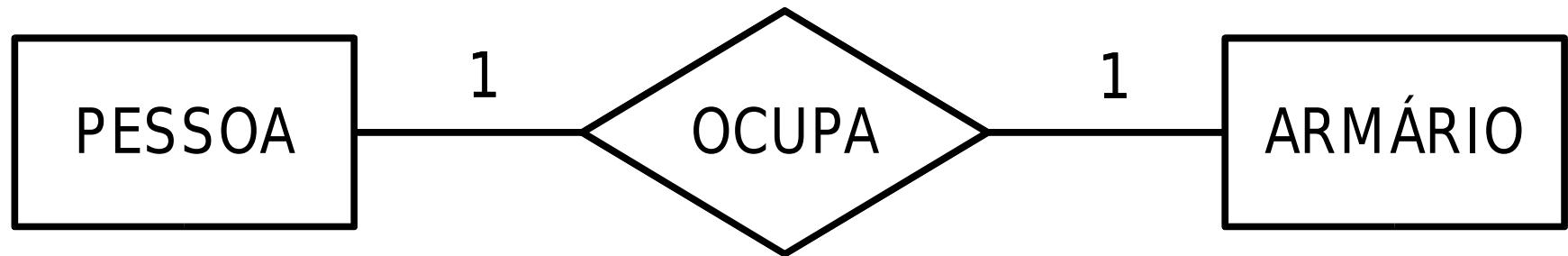


Inspirado em (Heuser, 2004)

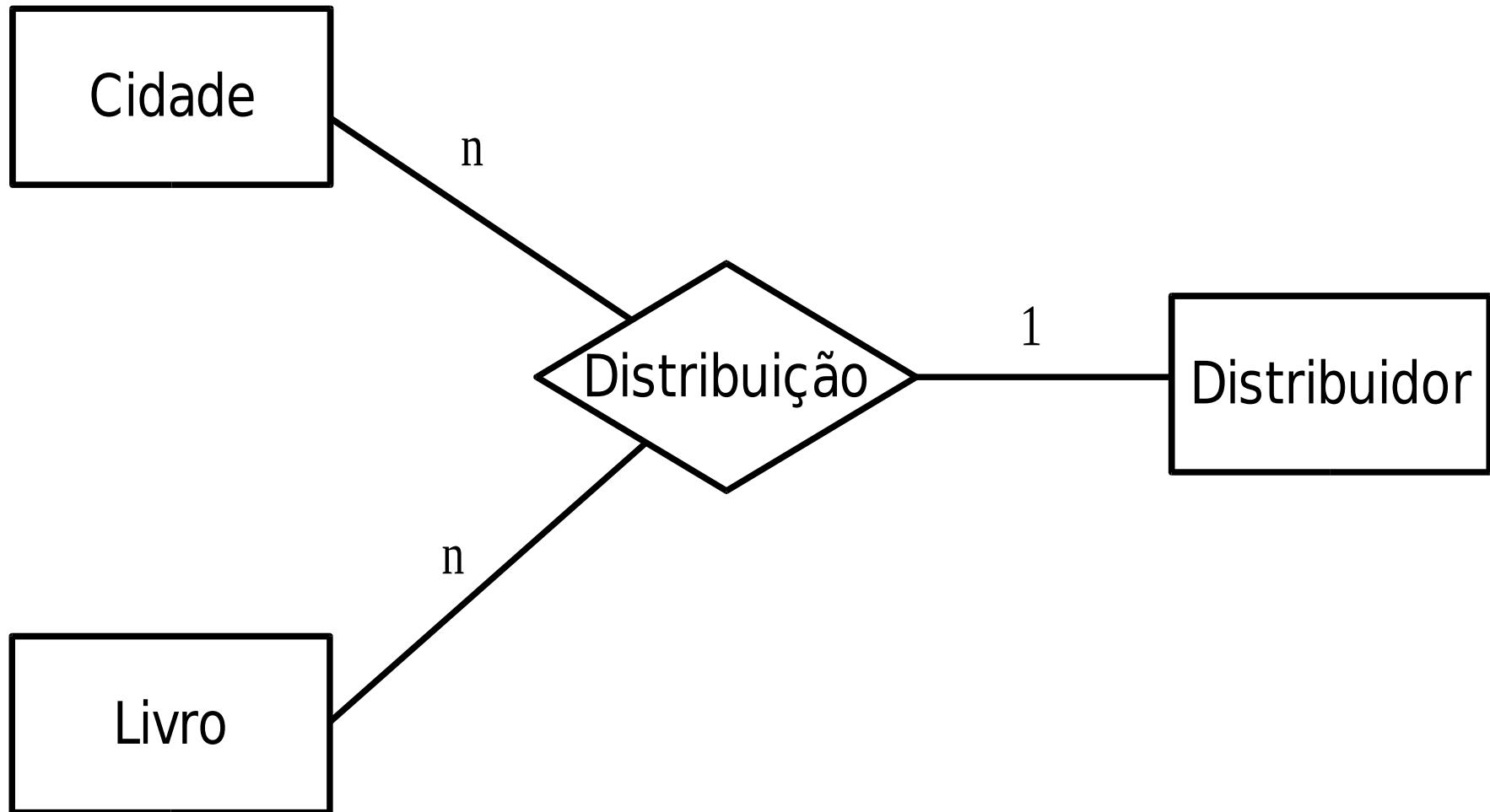
Relacionamento 1:n



Relacionamento 1:1



Cardinalidade em Relacionamento Ternário



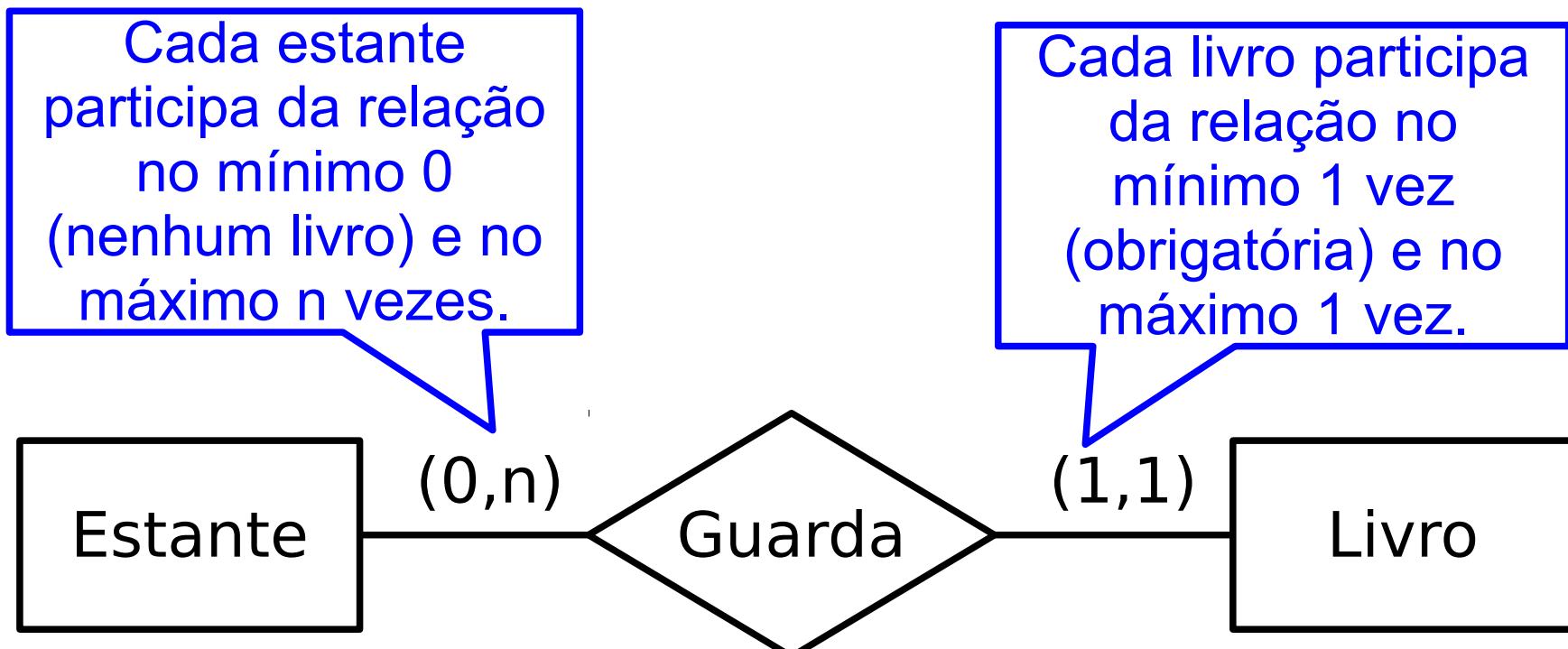
**Restrição de
Participação na
Relação
(Cardinalidade)**

Restrição de Participação na Relação

- Notação alternativa à razão de cardinalidade.
- Indica restrição mínima e máxima (min, max) de participação de cada entidade na relação.
- É indicado no lado correspondente à entidade (oposto do anterior).

Restrição de Participação na Relação

- É indicado no lado correspondente à entidade (oposto do anterior).



Restrição de Participação na Relação



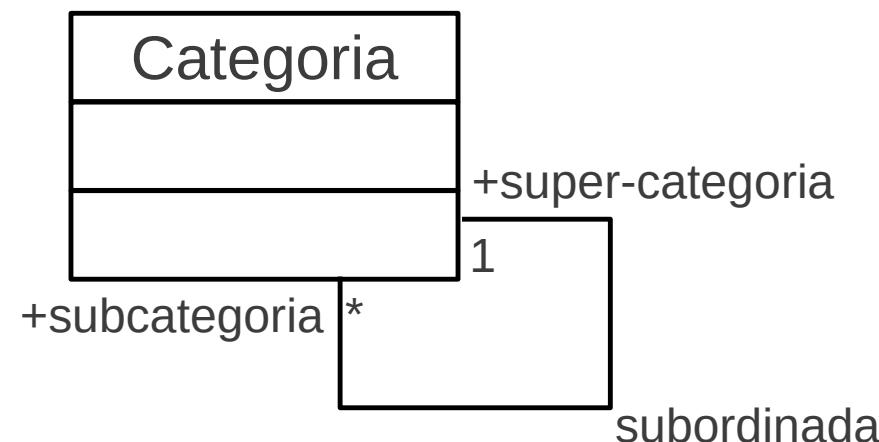
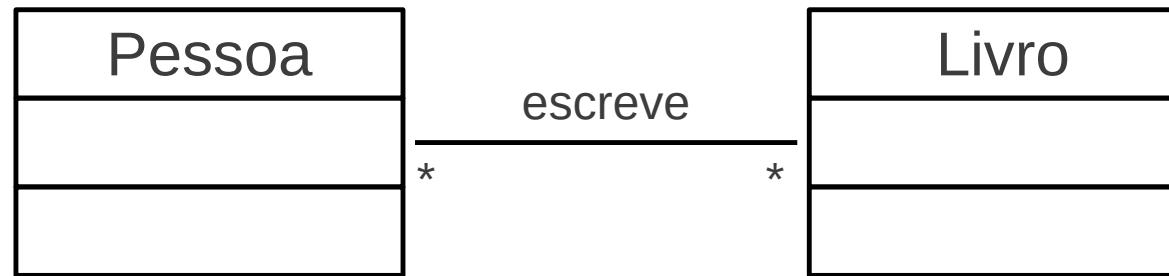
Restrição de Participação na Relação



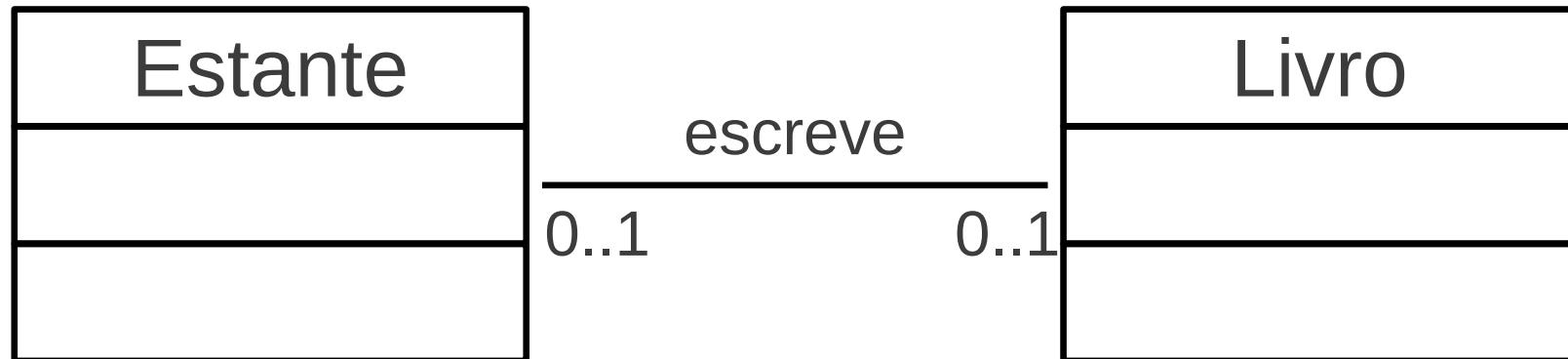
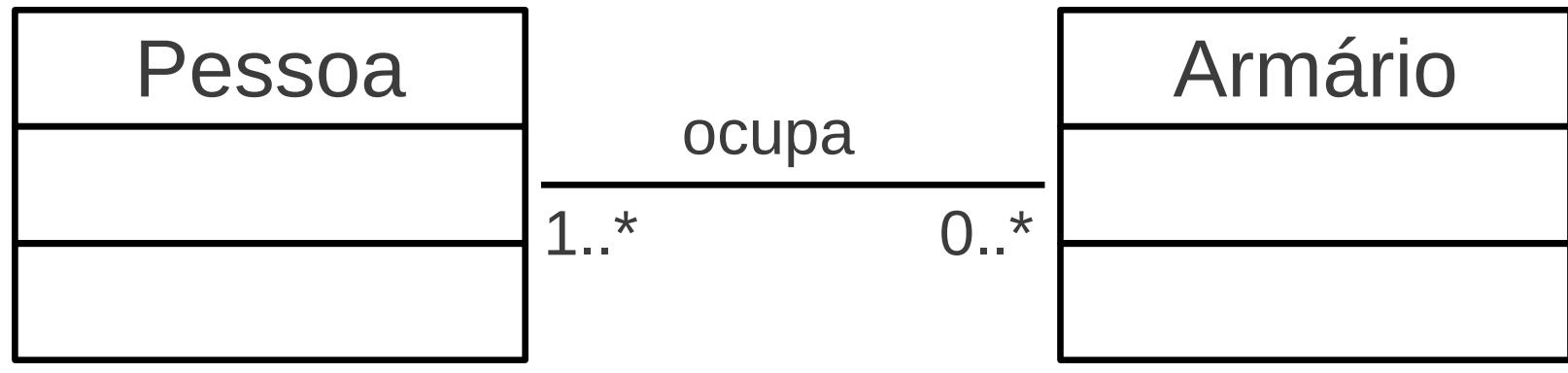
Restrições de Participação

- Relacionadas à cardinalidade mínima:
 - Participação Total (obrigatória) \Rightarrow mínima 1
 - Participação Parcial (opcional) \Rightarrow mínima 0

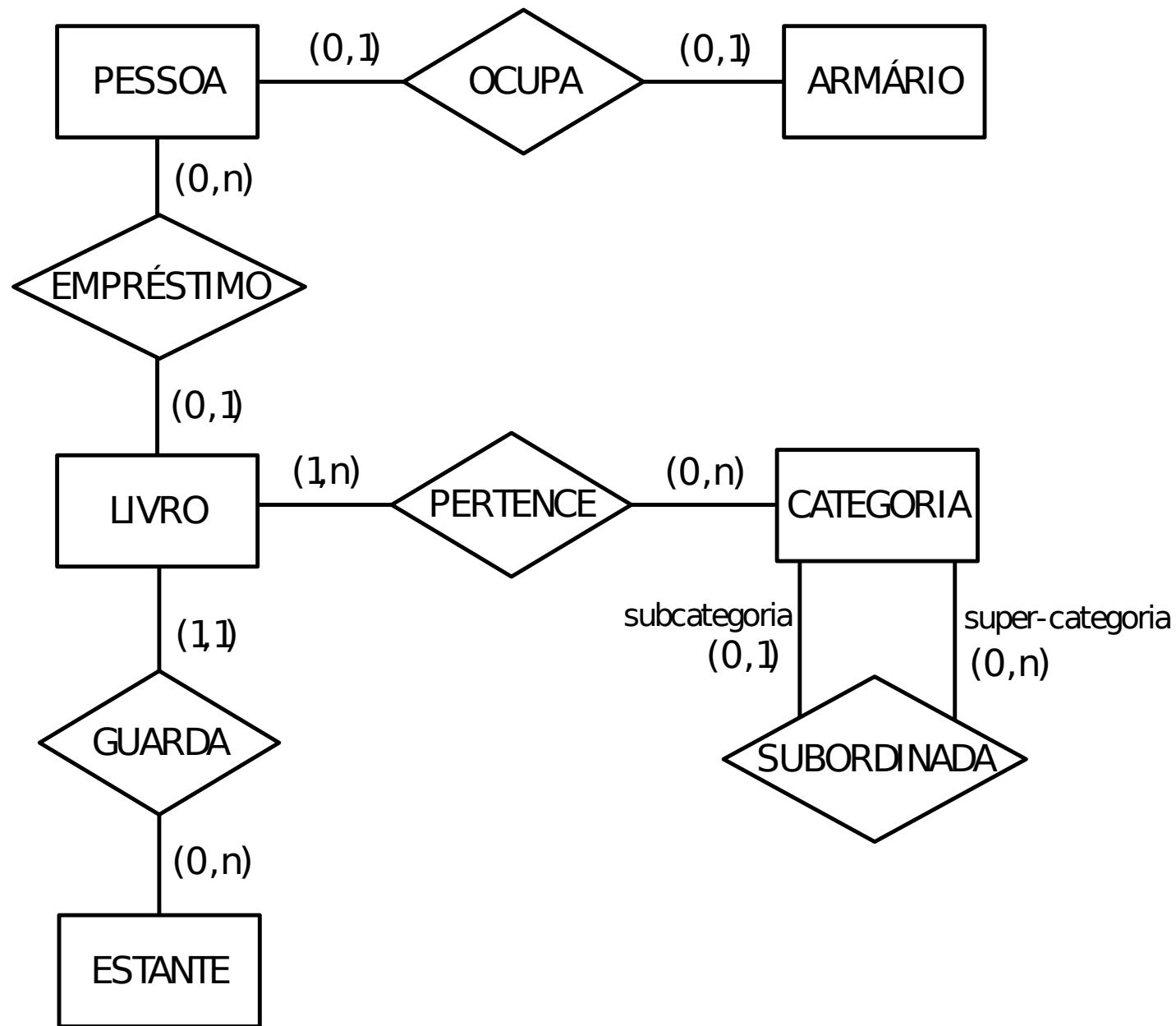
UML: Cardinalidade Máxima



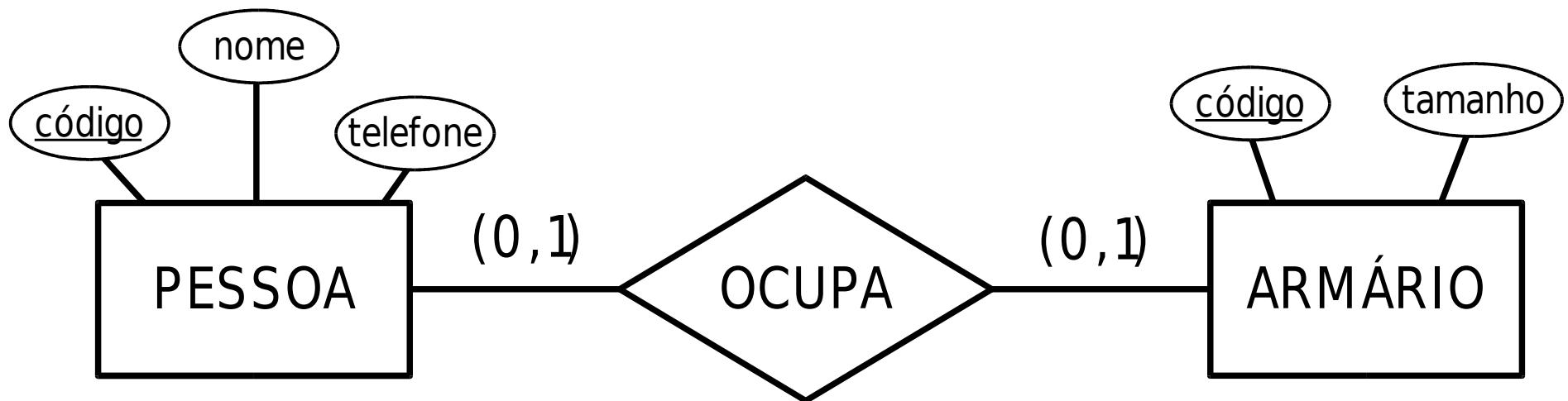
UML: Cardinalidade Mínima



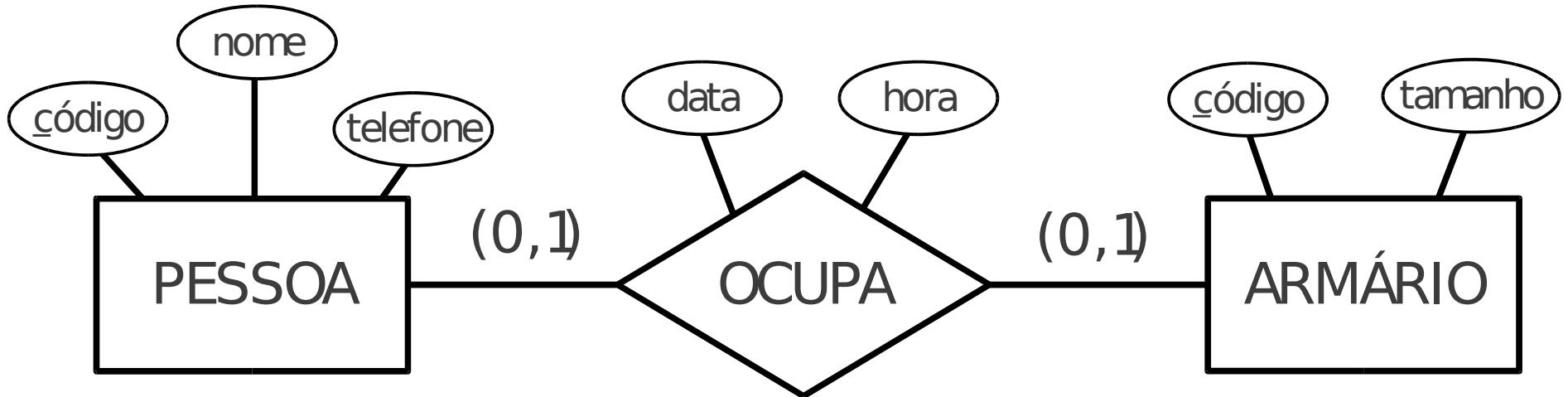
Exemplo Diagrama ER



Atributos no Relacionamento



Atributos no Relacionamento



Exercício

Uma indústria farmacêutica quer desenvolver um banco de dados para registrar os medicamentos que ela produz, bem como os vírus tratados por estes medicamentos.

Elabore uma modelo conceitual para este banco de dados conforme o detalhamento a seguir:

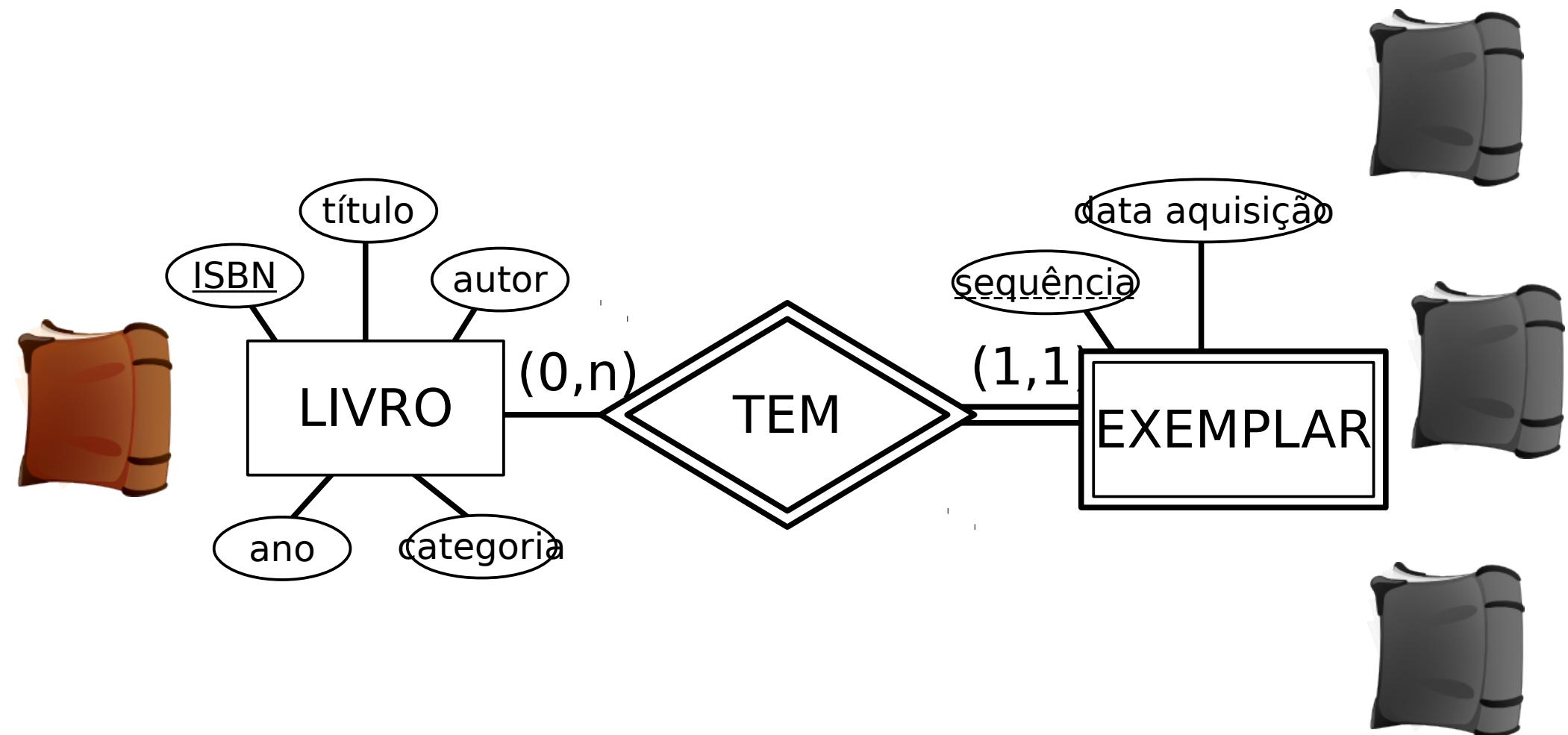
- Devem ser armazenados os nomes científicos e populares dos vírus bem como os períodos de incubação.
- Para medicamentos, o devem ser armazenados o nome de venda e o composto ativo.

Exercício

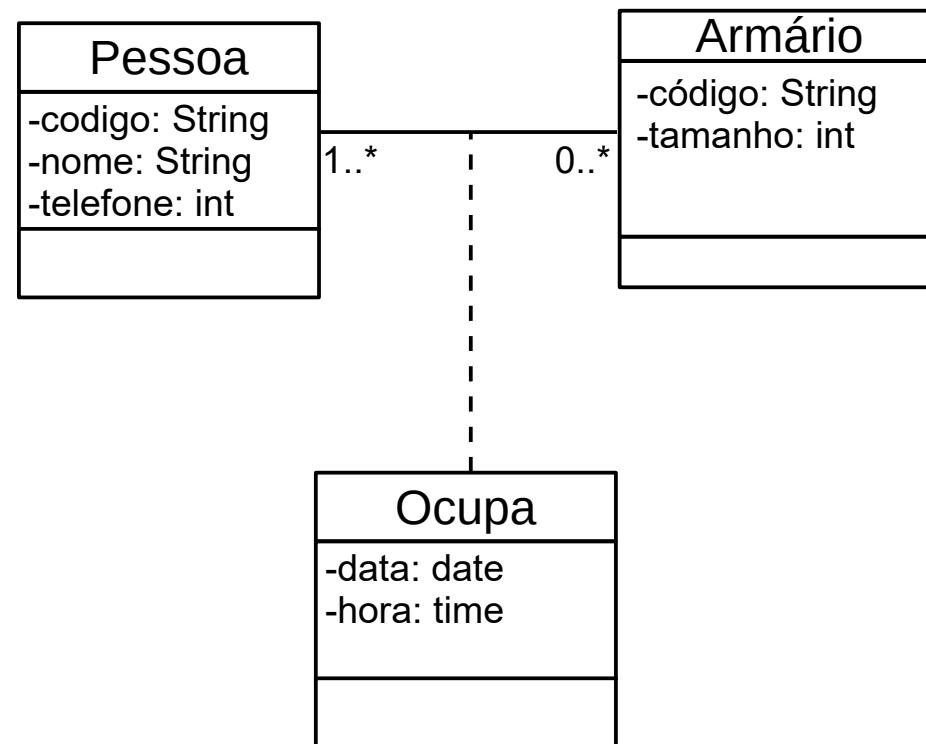
parte 2

- a) Considere que um dado medicamento pode tratar vários vírus e um vírus pode ser tratado por vários medicamentos.
- b) Medicamentos são fabricados por empresas que possuem nome e CNPJ. Uma empresa pode fabricar vários medicamentos, mas um medicamento é fabricado por uma única empresa.

Entidade Fraca



Classe de Associação

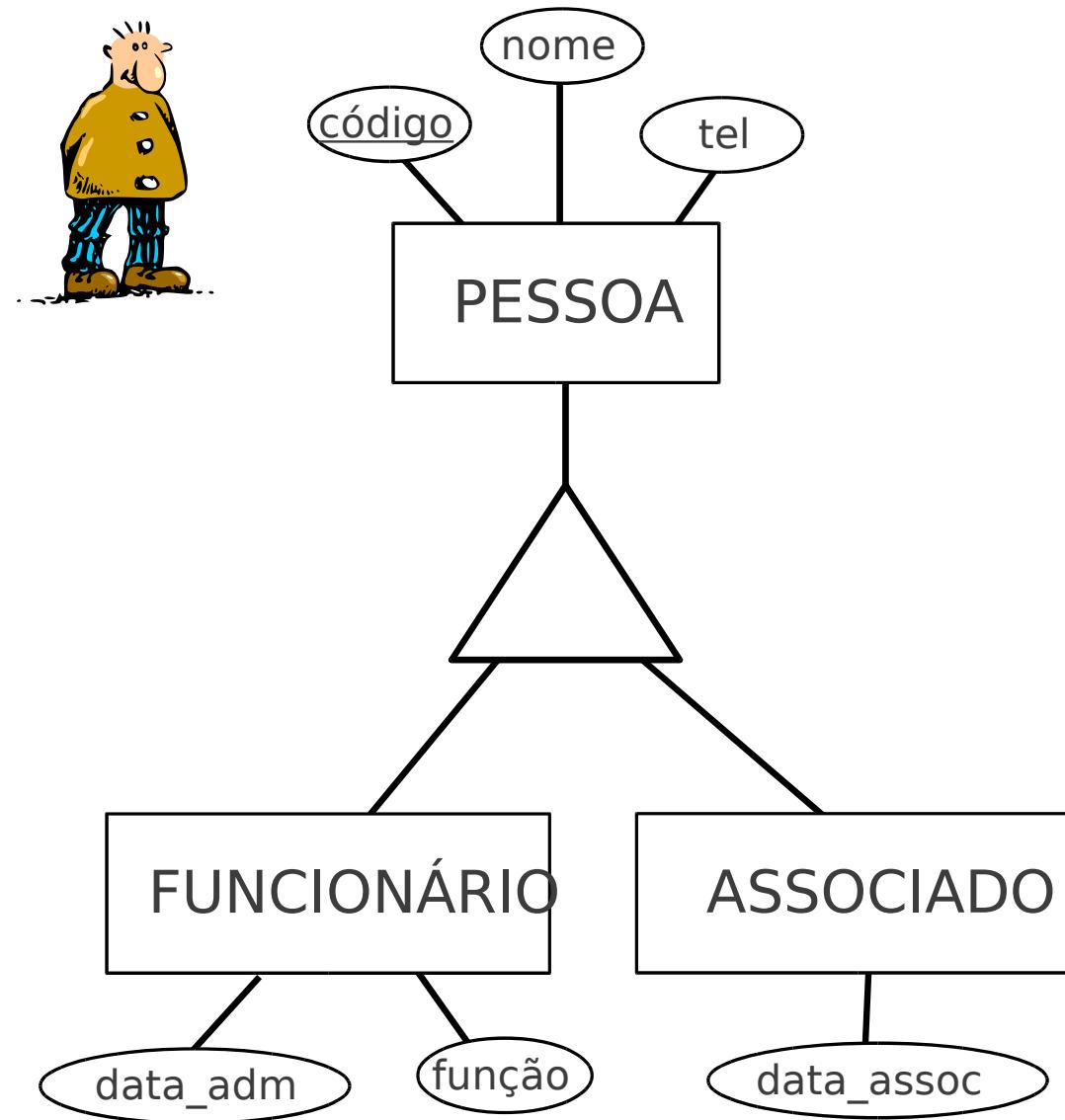


EER – ER Estendido

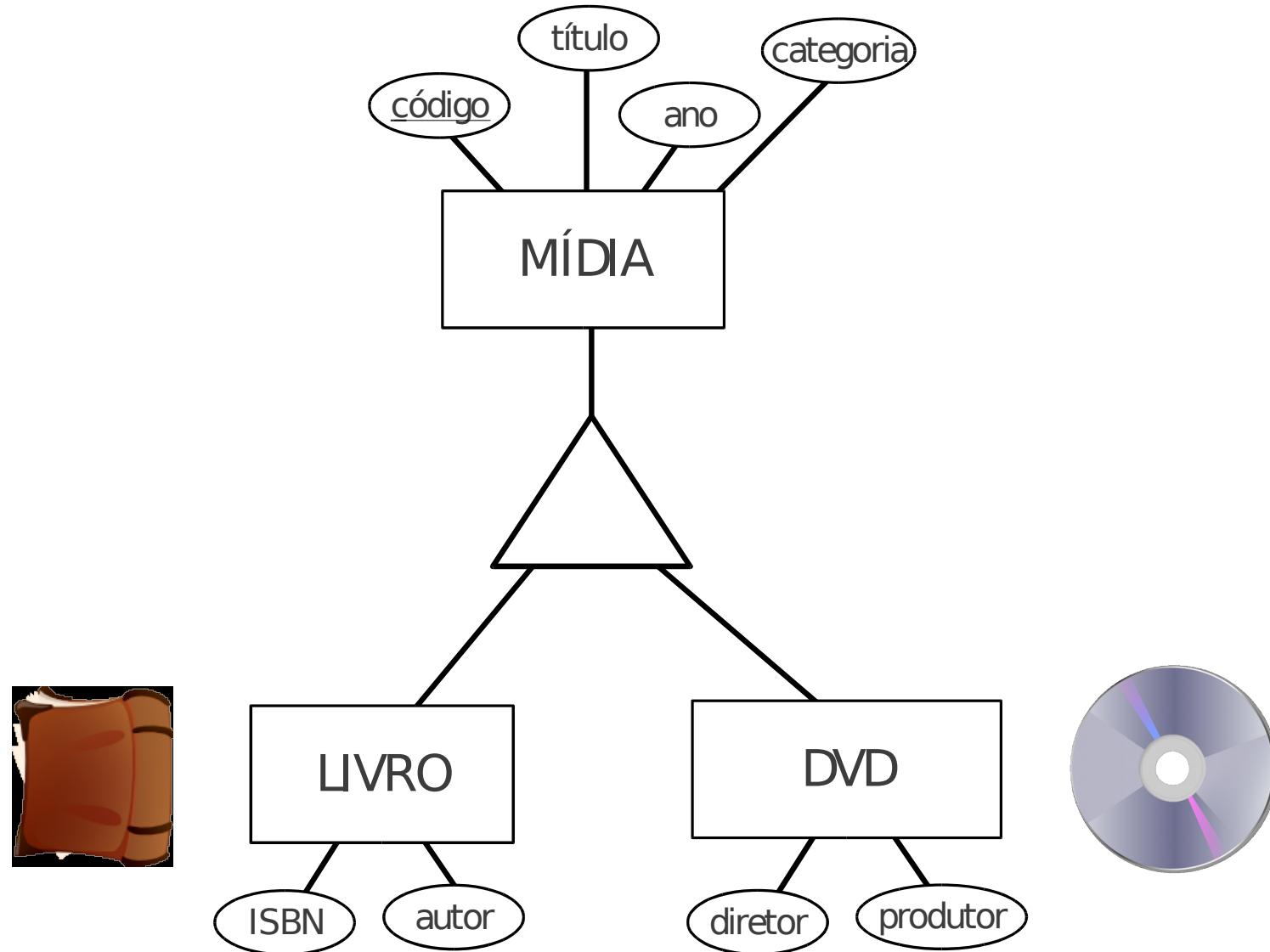
- ER original não suporta generalização/especialização
- ER Estendido (EER) – acrescenta estes recursos

EER:
Especialização/Gener
alização

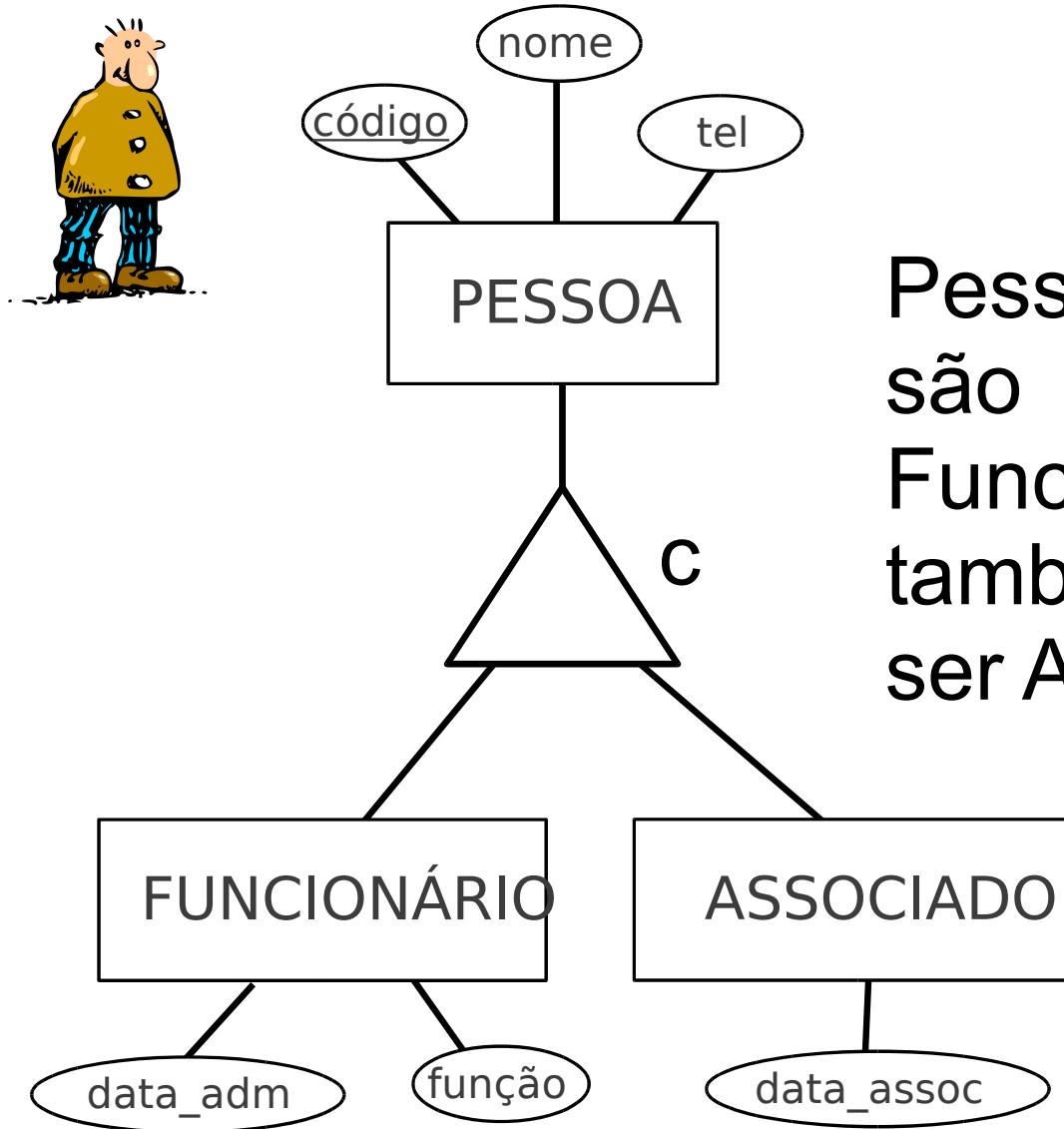
Generalização / Especialização



Generalização / Especialização

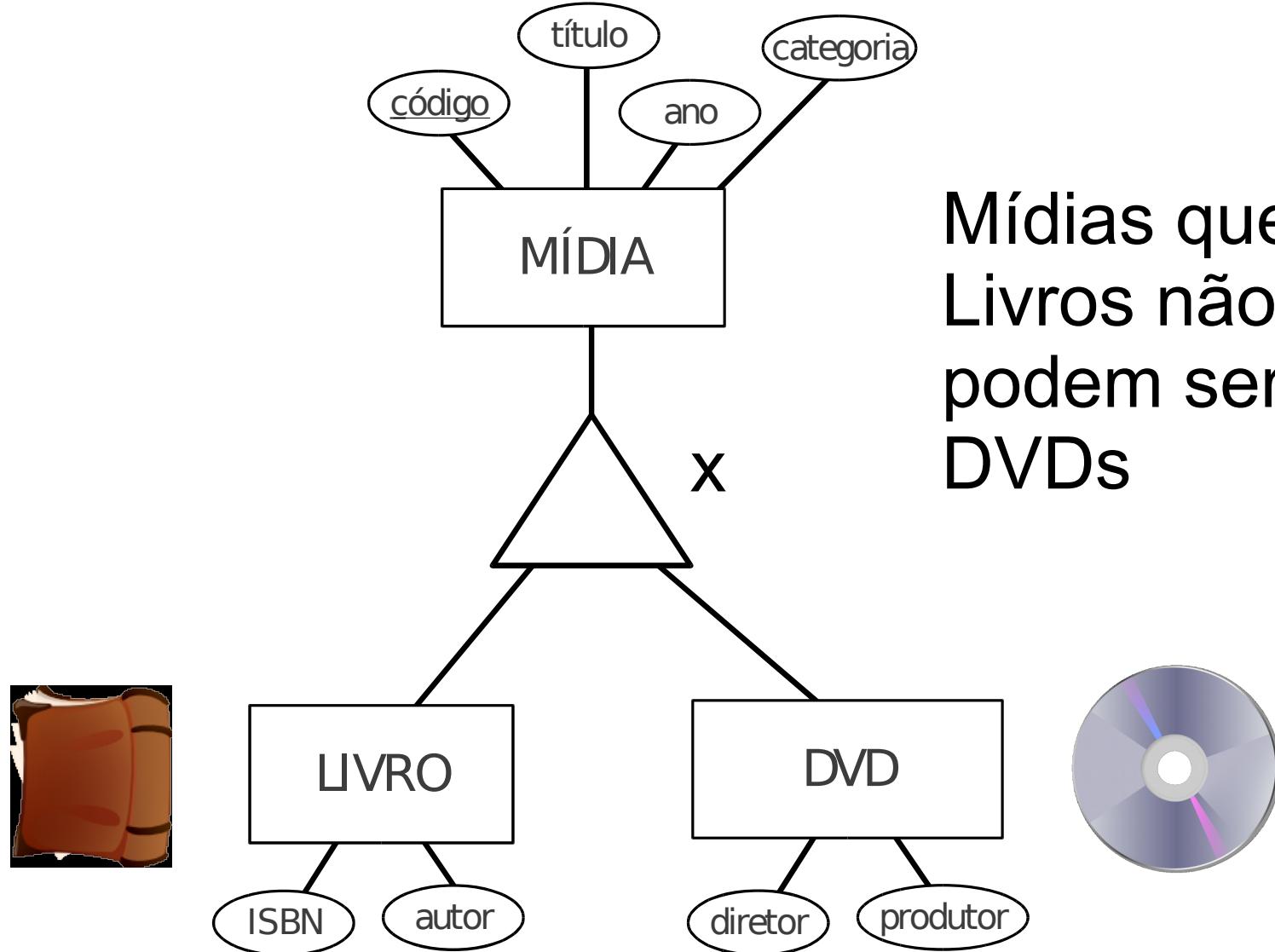


Generalização / Especialização Compartilhada ou Superposta



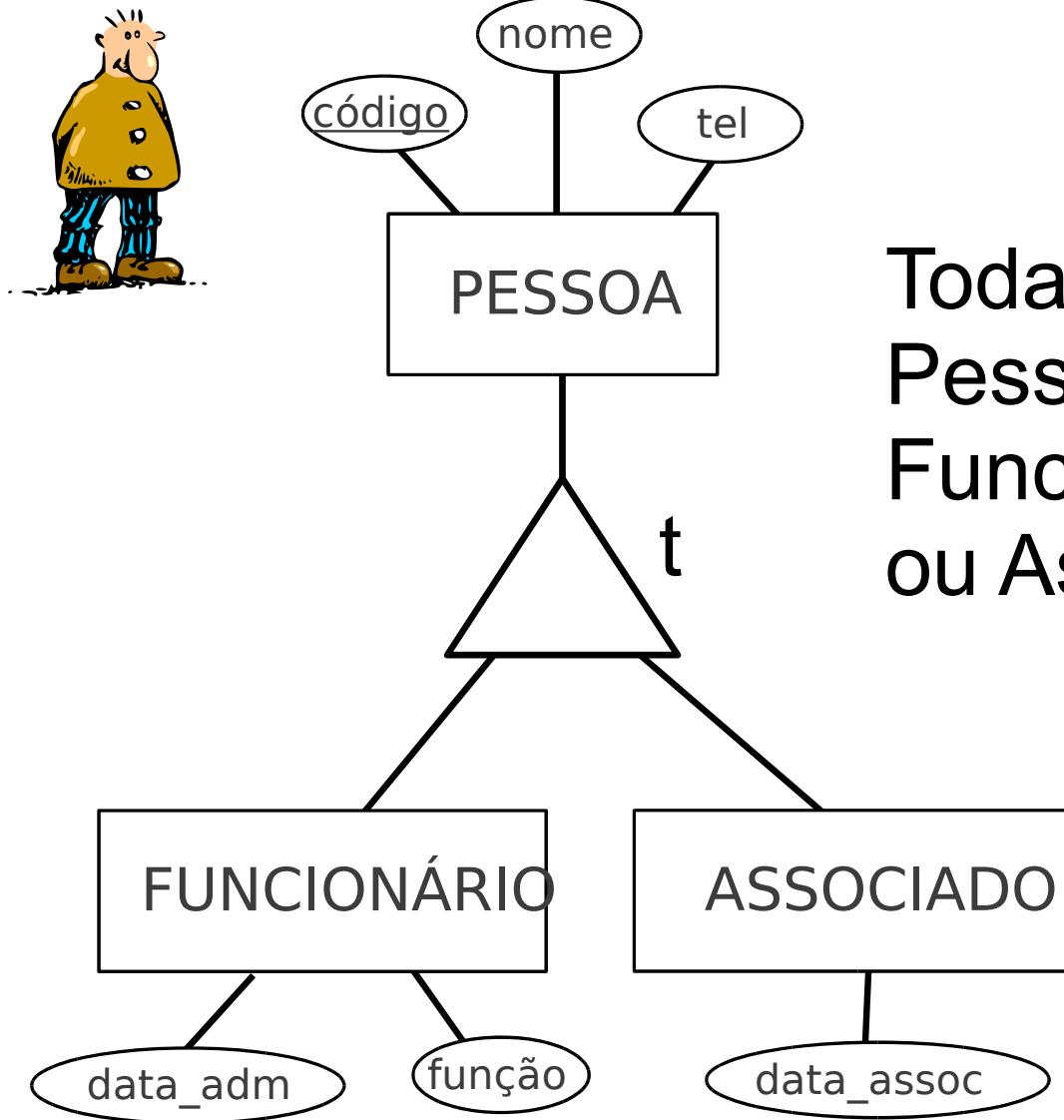
Pessoas que
são
Funcionários
também podem
ser Associados

Generalização / Especialização Exclusiva ou Disjunta



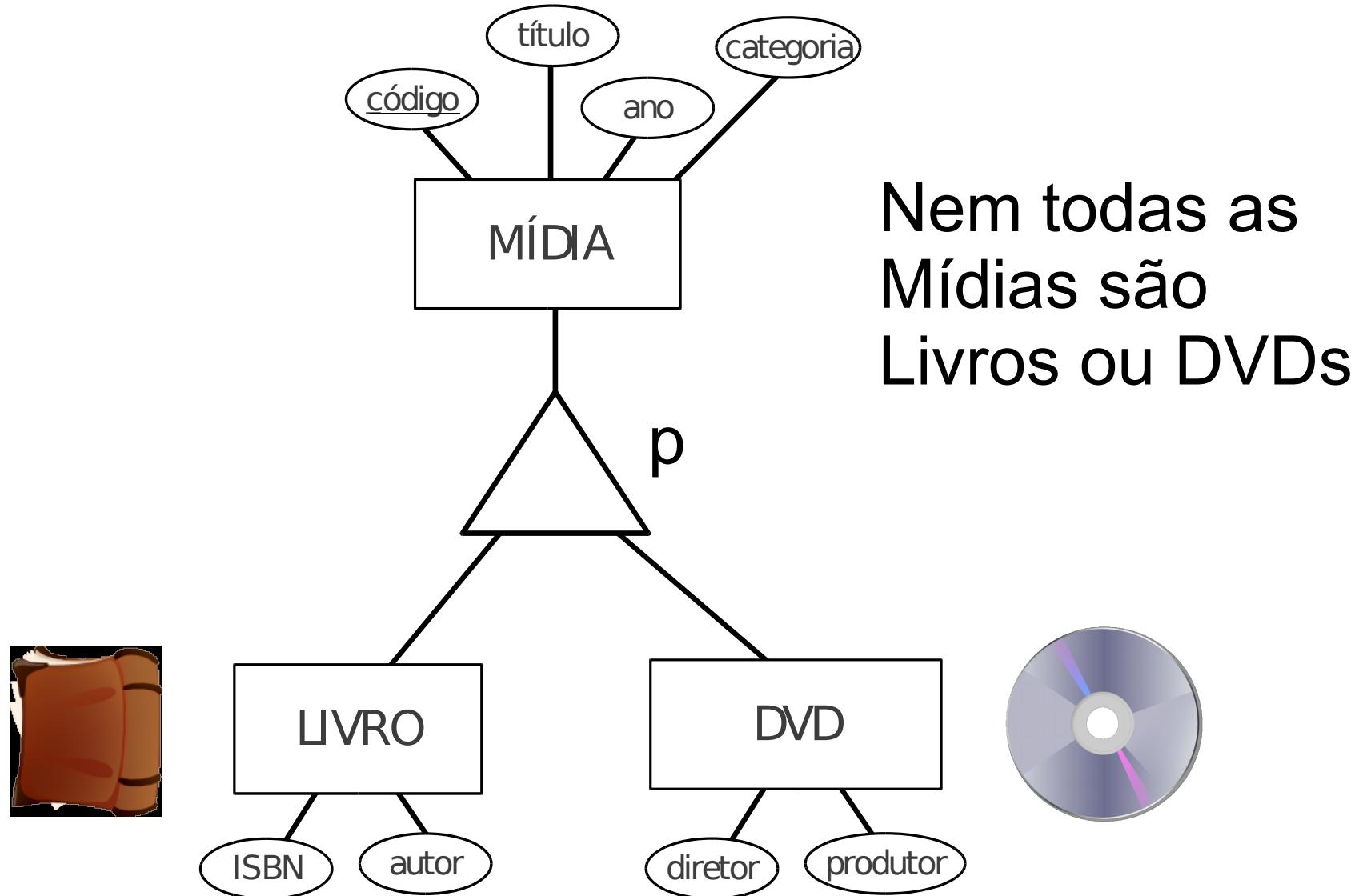
Generalização / Especialização

Total

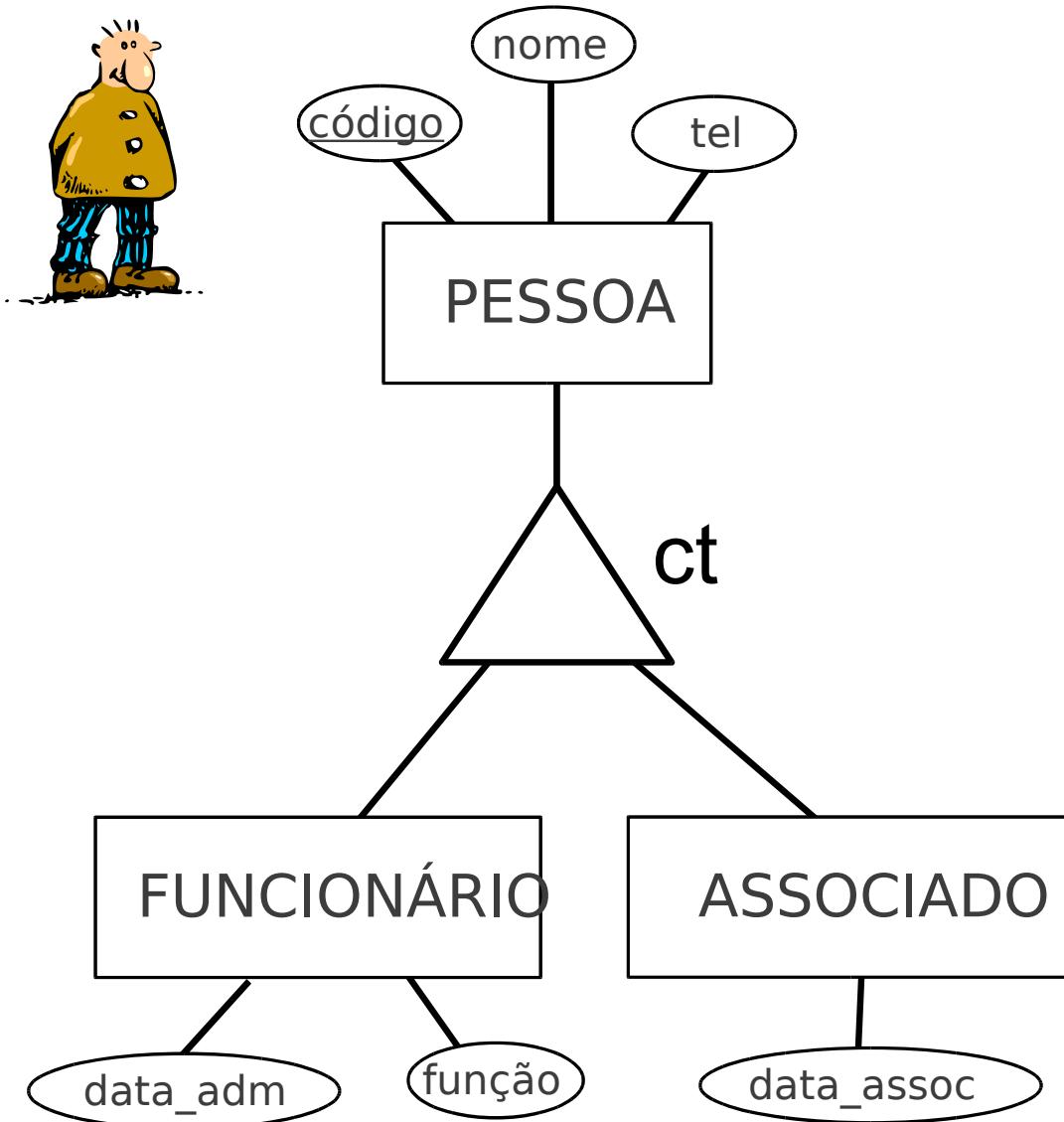


Todas as
Pessoas são
Funcionários
ou Associados

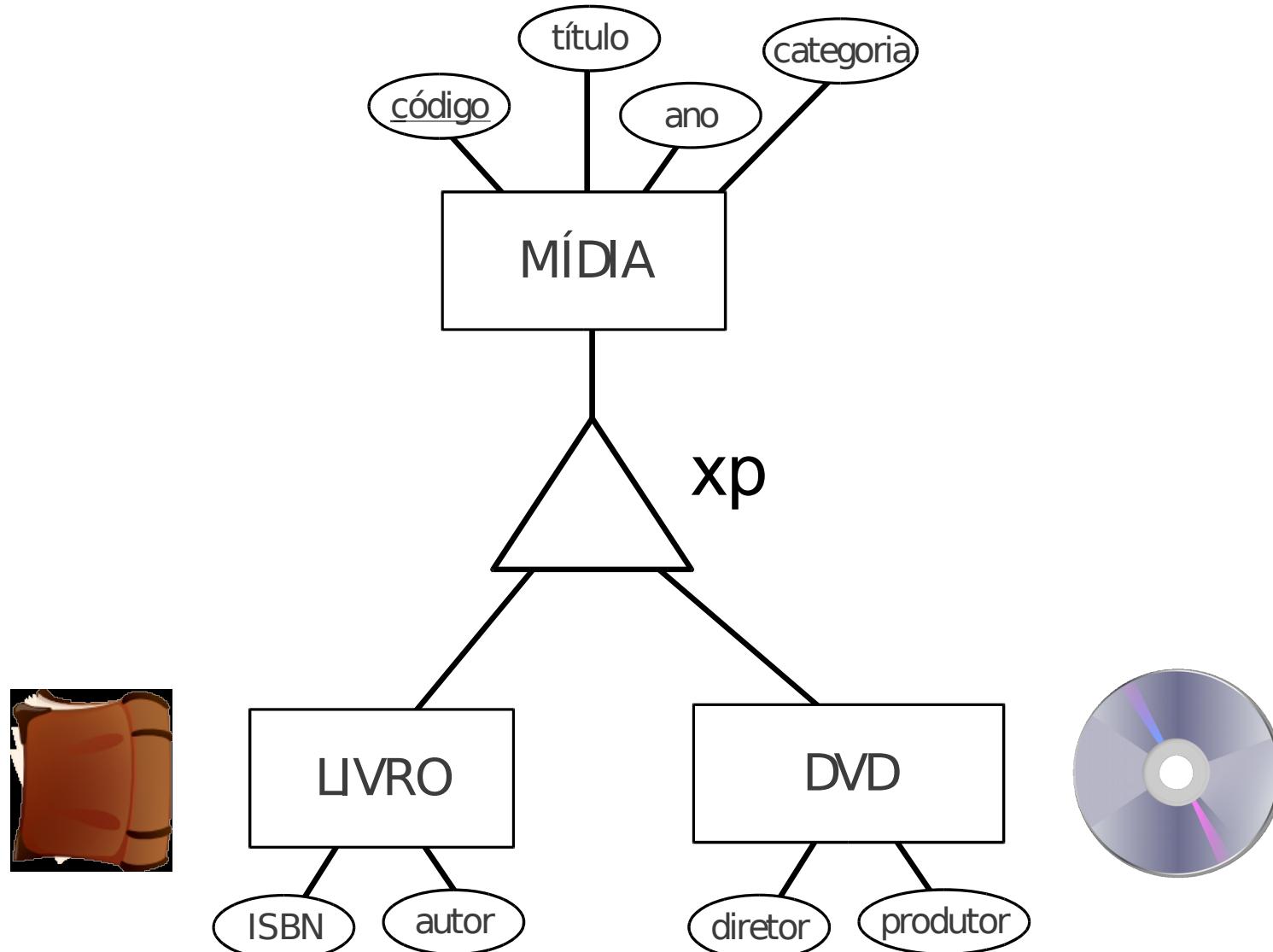
Generalização / Especialização Parcial



Generalização / Especialização Compartilhada e Total



Generalização / Especialização Exclusiva e Parcial



Generalização / Especialização

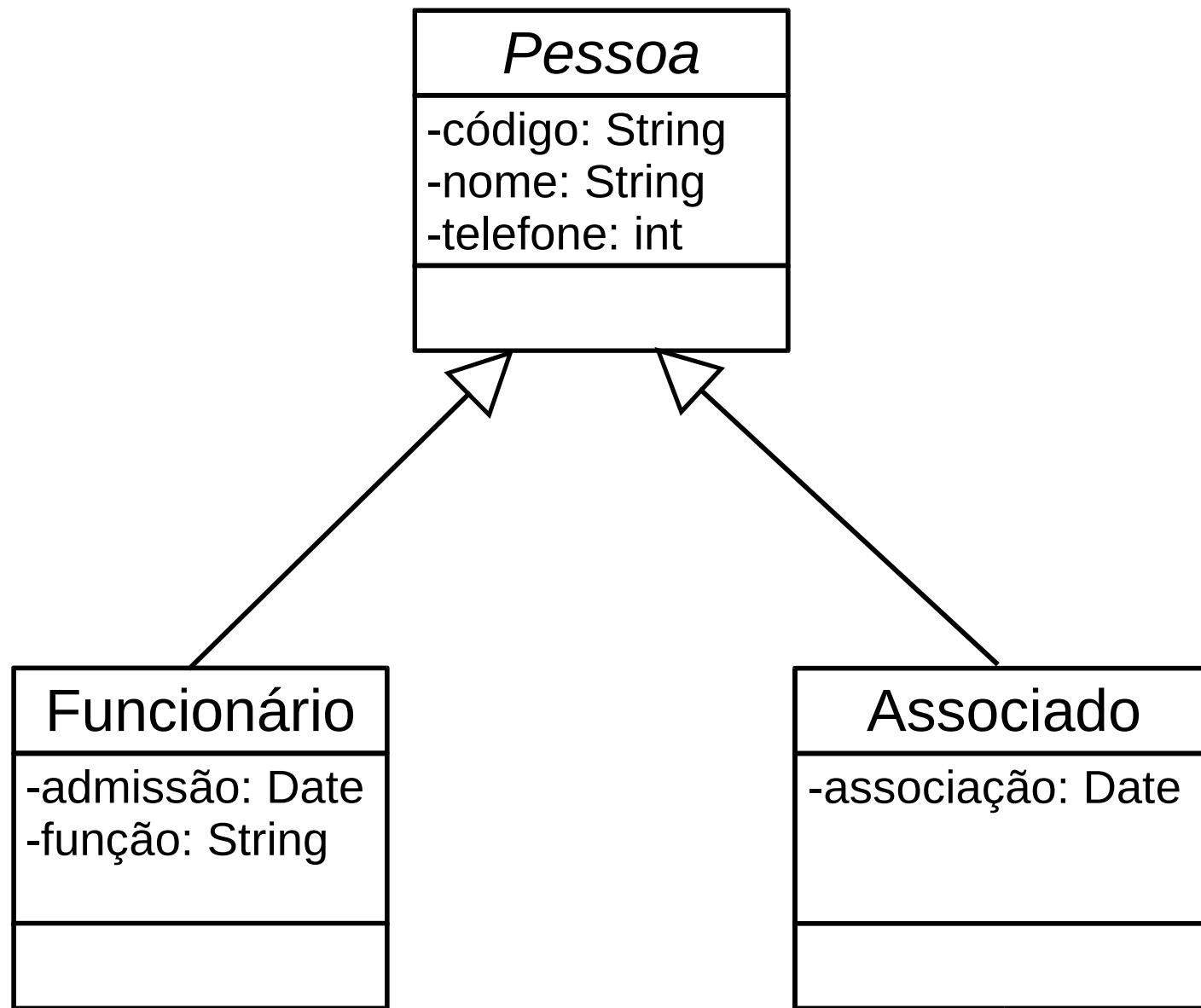
Tipos

	Total (t)	Parcial (p)
Exclusiva (x)	xt	xp
Compartilhada (c)	ct	cp

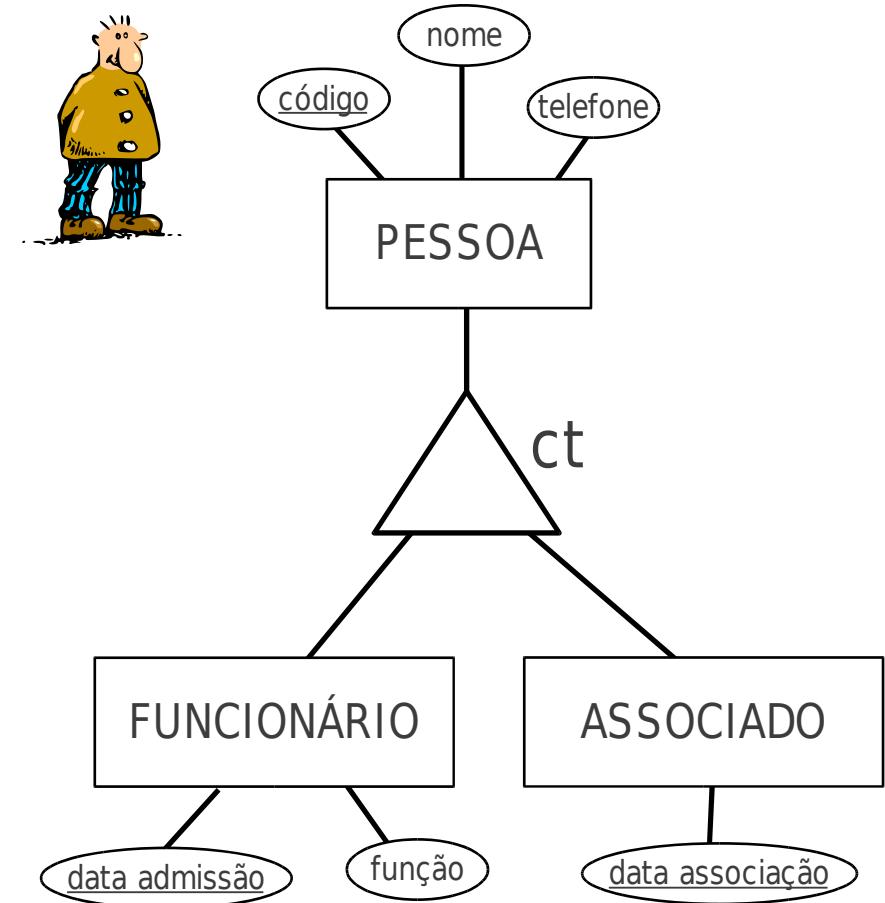
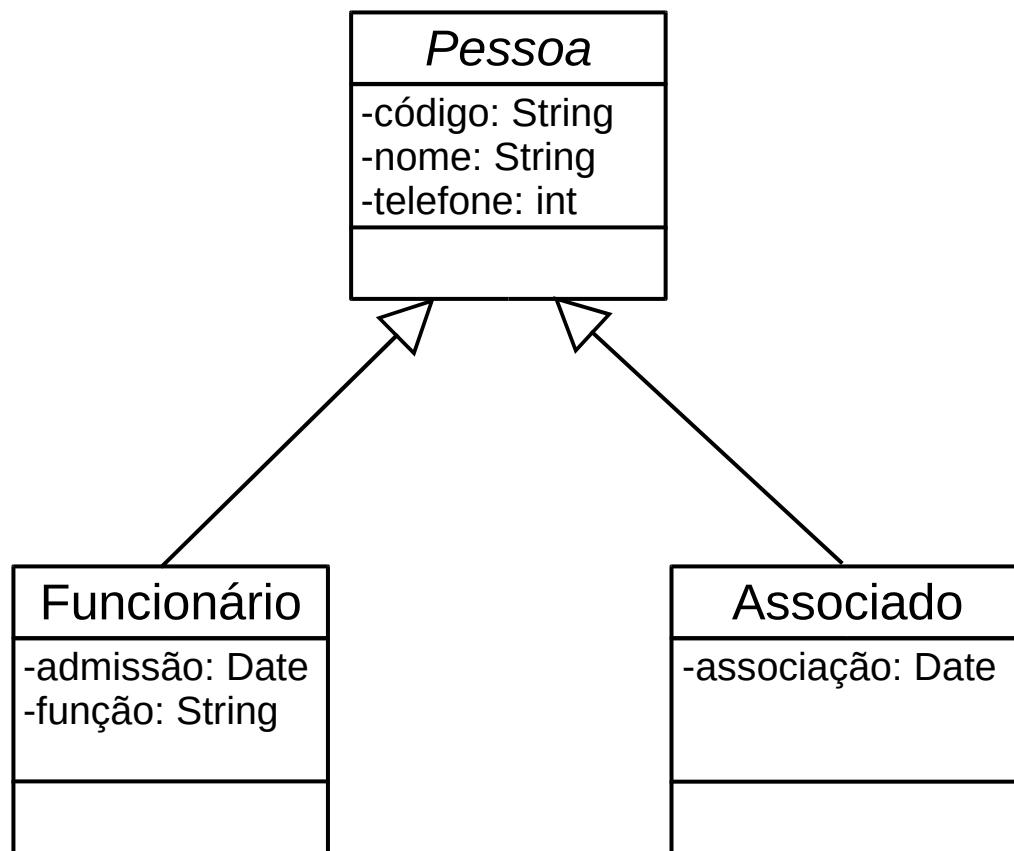
(Heuser, 2004)

OO: Herança

UML: Herança

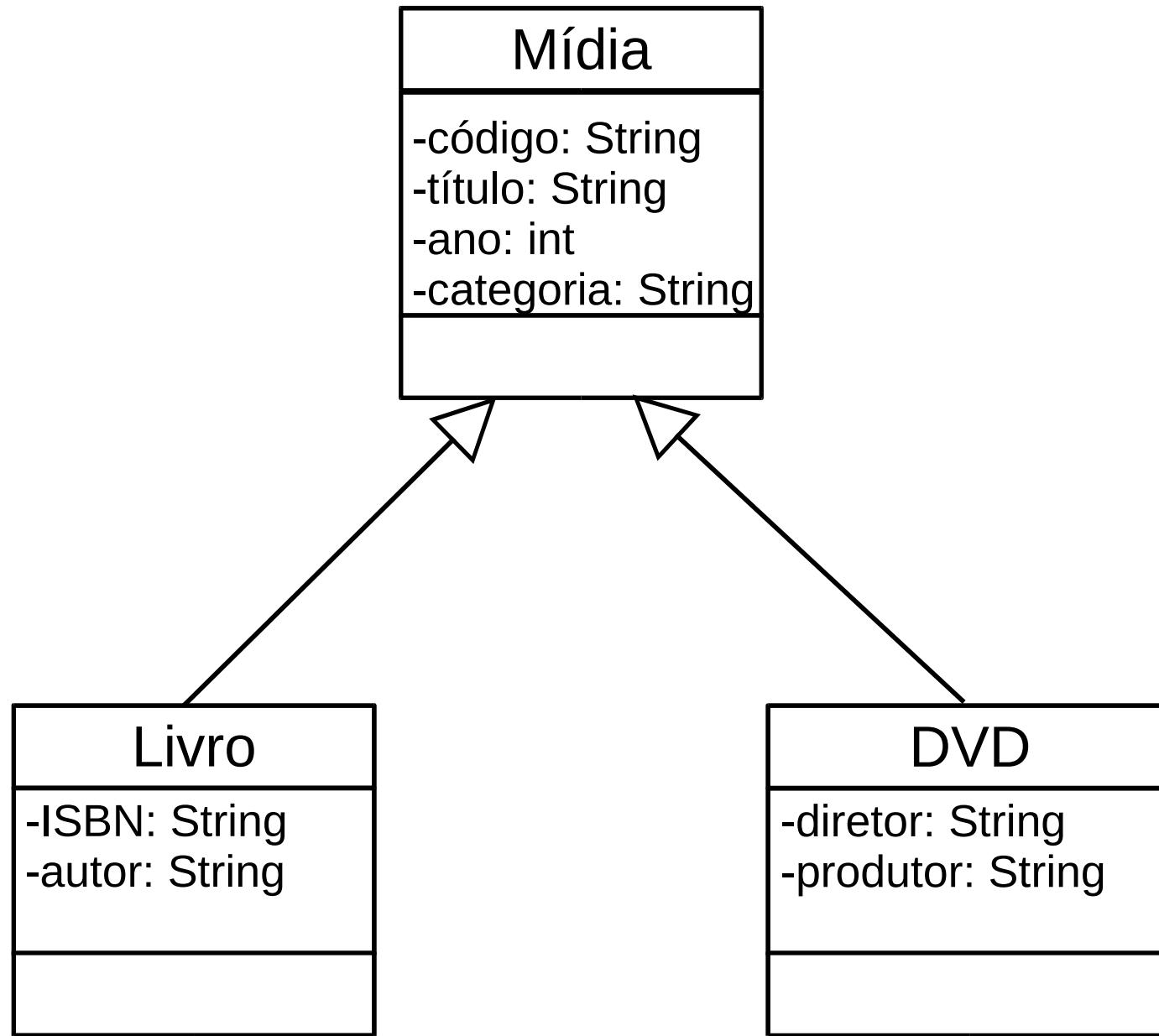


Especialização total x Classe abstrata

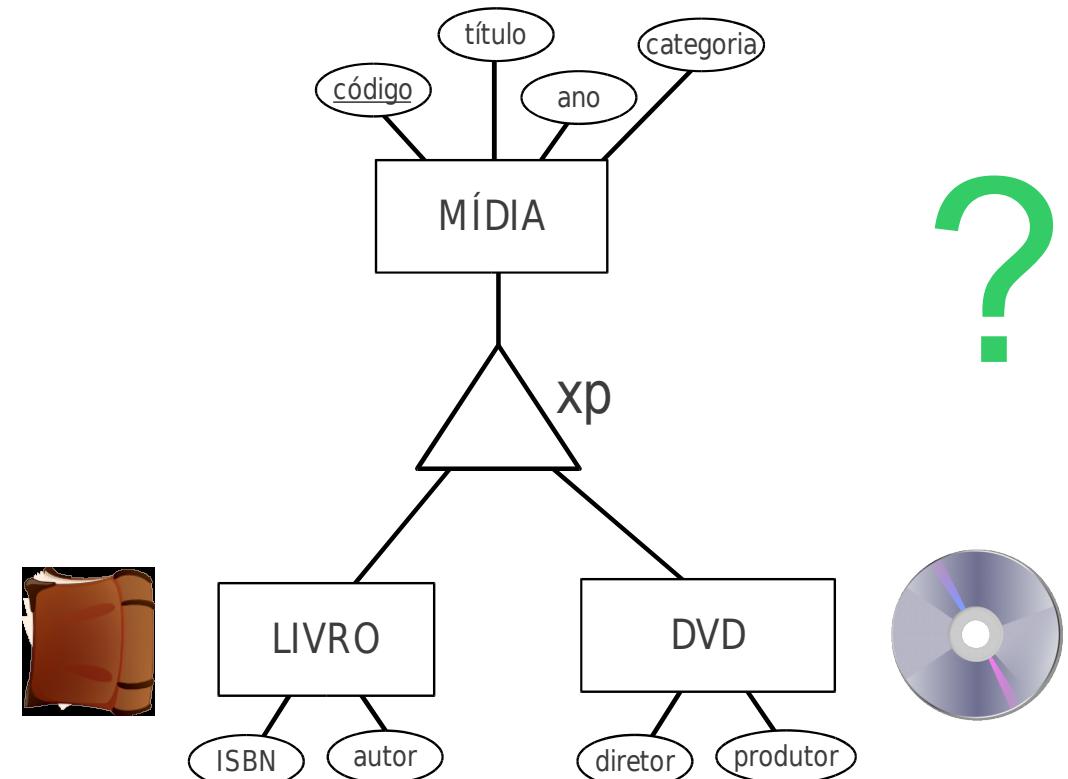
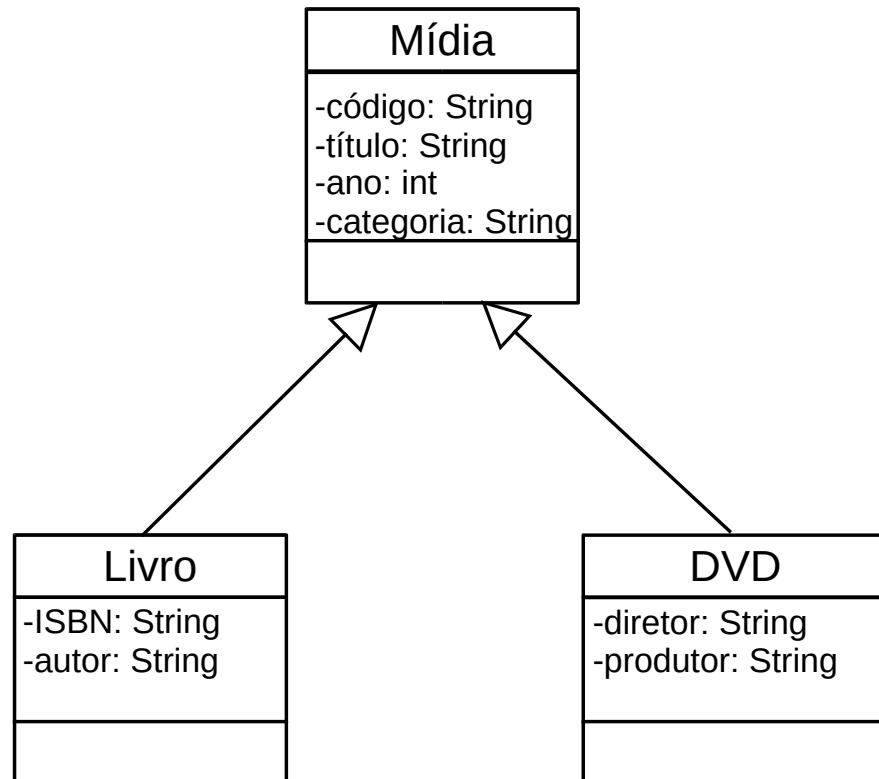


especialização total x classe abstrata

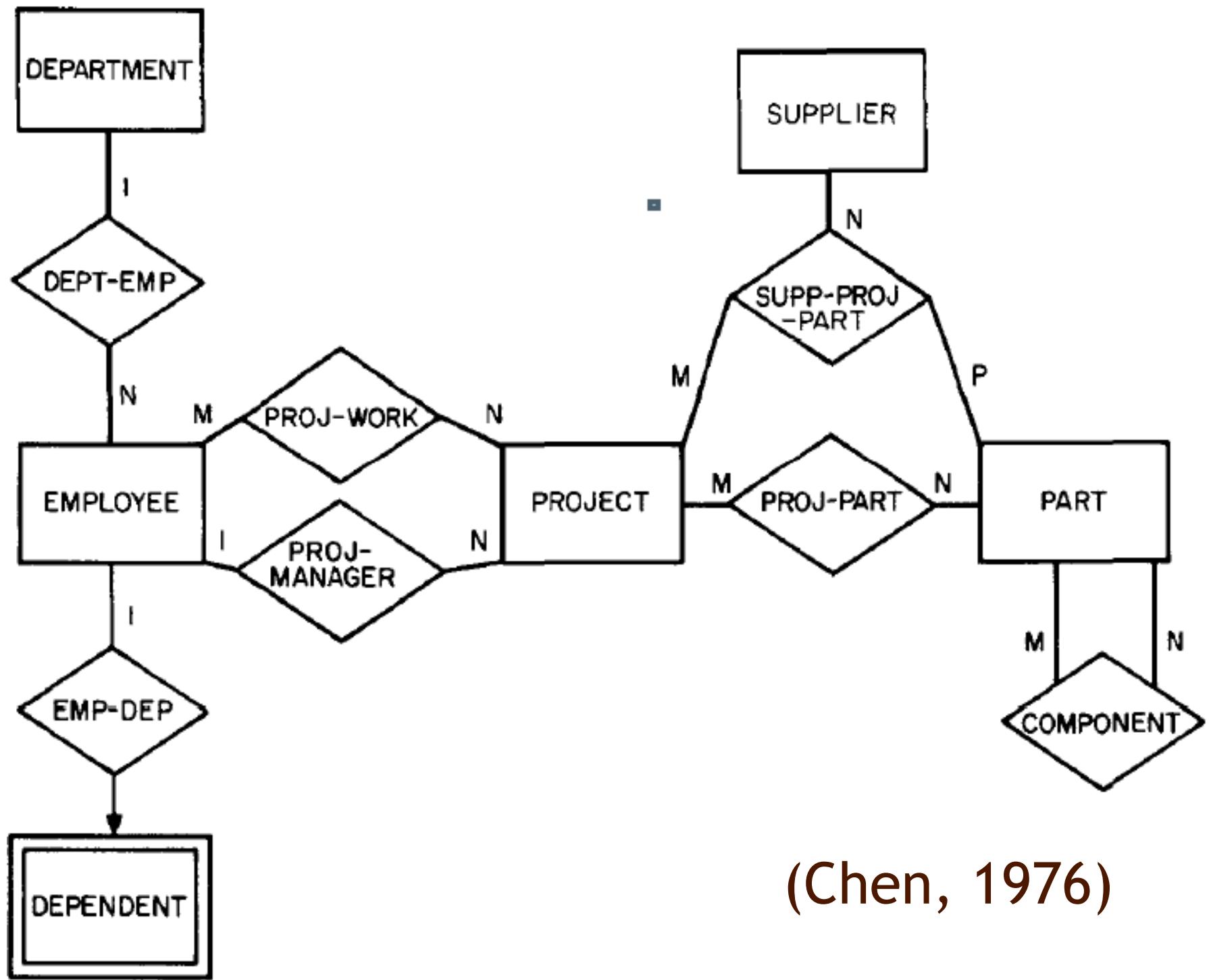
UML: Herança



Especialização parcial x Classe



?



(Chen, 1976)

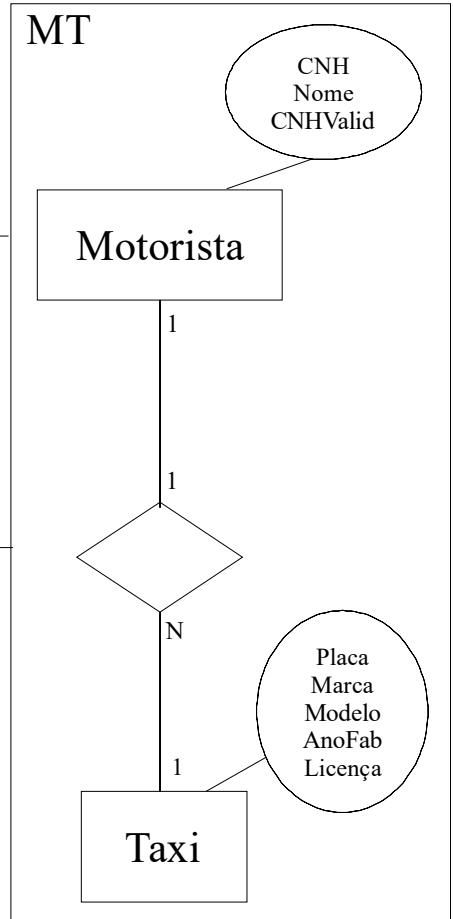
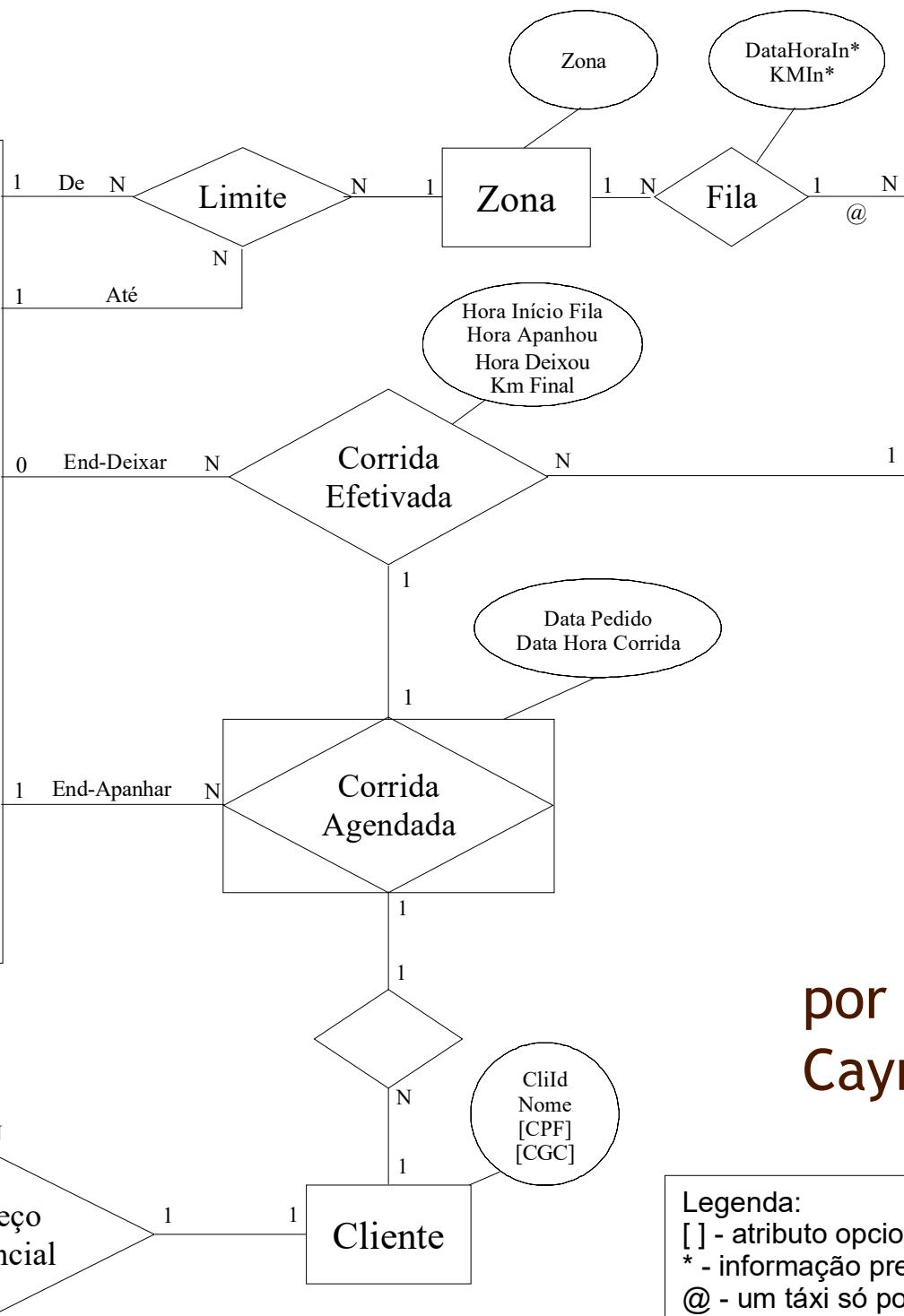
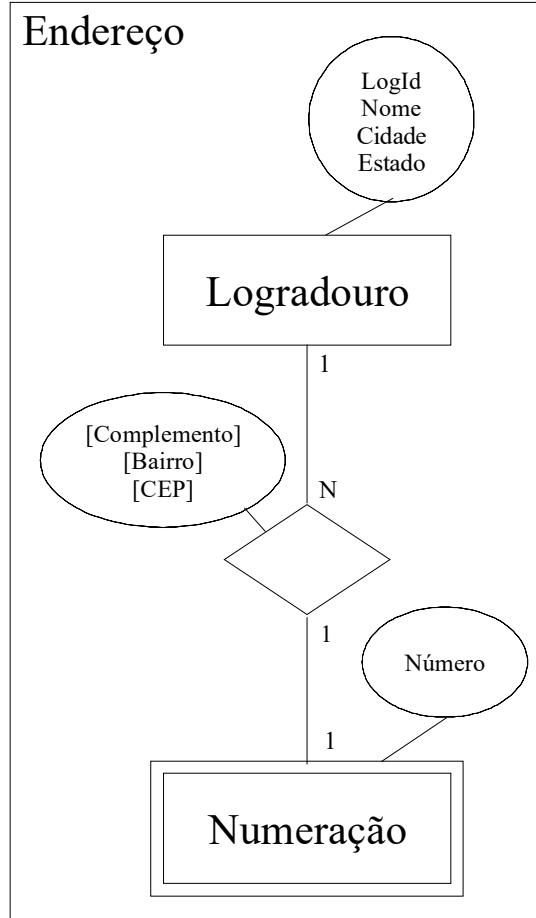
Exercício

parte 3

- Vírus podem ser classificados em diversas categorias, mas considere apenas Vírus com DNA, Vírus com RNA e Retrovírus. Retrovírus são tratados com coquetéis de medicamentos. Um coquetel é composto por vários medicamentos, cada um em uma concentração específica. Um coquetel tem uma dosagem específica para o tratamento de um dado Retrovírus.

Caso dos Taxis

- Exemplo criado por prof. Geovane Cayres Magalhães
 - <http://www.ic.unicamp.br/~geovane/mo410-091/caso.html>



por prof. Geovane
Cayres Magalhães

Legenda:
 [] - atributo opcional
 * - informação preenchida após inclusão inicial
 @ - um táxi só pode aparecer uma vez na fila

Referências

- Chen, Peter Pin-Shan (1976) **The entity-relationship model – toward a unified view of data**. ACM Trans. Database Systems, ACM, 1, 9-36.
- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4^a edição em português.
- Guimarães, Célio (2003) **Fundamentos de Bancos de Dados: Modelagem, Projeto e Linguagem SQL**. Editora UNICAMP, 1^a edição.
- Heuser, Carlos Alberto (2004) **Projeto de Banco de Dados**. Editora Sagra Luzzato, 5^a edição.

Referências

- Ramakrishnan, Raghu; Gehrke, Johannes (2003)
Database Management Systems. McGraw-Hill,
3rd edition.

Referências Bibliográficas

- Almeida, Charles Ornelas , Guerra, Israel; Ziviani, Nivio (2010) **Projeto de Algoritmos** (transparências aula).
- Bloom, Paul (2007) **Introduction to Psychology** – transcrição das aulas (aula 17). Yale University.
- Ferreira, Aurélio B. H. (1989) **Minidicionário da Língua Portuguesa**. Rio de Janeiro, Editora Nova Fronteira.
- Houaiss, Instituto Antônio. **Dicionário Houaiss da Língua portuguesa** (2006) Editora Objetiva, Março.
- IBM - International Business Machines Corporation. **IBM Smalltalk Tutorial** [Online] <http://www.wi2.uni-erlangen.de/sw/smalltalk/>
- Liskov, Barbara; Zilles, Stephen. **Programming with abstract data types** (1974) ACM SIGPLAN Notices, 9 (4) p. 50.

Referências Bibliográficas

- Meyer, Bertrand (1997) **Object-Oriented Software Construction – Second Edition.** USA, Prentice-Hall, Inc.
- Miller, Robert (2004) **6.831 User Interface Design and Implementation (lecture notes).** MIT OpenCourseware.
- Rocha, Heloisa Vieira da, Baranauskas, Maria Cecilia Calani (2003) **Design e Avaliação de Interfaces Humano-Computador.** NIED/UNICAMP.
- Santos, L. R., & Hood, B. M. (2009). **Object representation as a central issue in cognitive science.** The Origins of Object Knowledge: The Yale Symposium on the Origins of Object & Number Representation. Oxford: Oxford University Press.
- Shaw, M. **Abstraction Techniques in Modern Programming Languages** (1984) IEEE Software, 1, 4, 10-26.

Referências

- Bloom, Paul (2007) **Introduction to Psychology** – transcrição das aulas (aula 17). Yale University.
- Chen, Peter Pin-Shan (1976) **The entity-relationship model – toward a unified view of data**. ACM Trans. Database Systems, ACM, 1, 9-36.
- Dijkstra, E. W. (1986) **On a cultural gap**. The Mathematical Intelligencer. vol. 8, no. 1, pp. 48-52.
- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4a. edição em português.
- Elmasri, Ramez; Navathe, Shamkant B. (2011) **Sistemas de Bancos de Dados**. Pearson, 6a. edição em português.
- Guimarães, Célio (2003) **Fundamentos de Bancos de Dados: Modelagem, Projeto e Linguagem SQL**. Editora UNICAMP, 1a. edição.

Referências e Agradecimentos

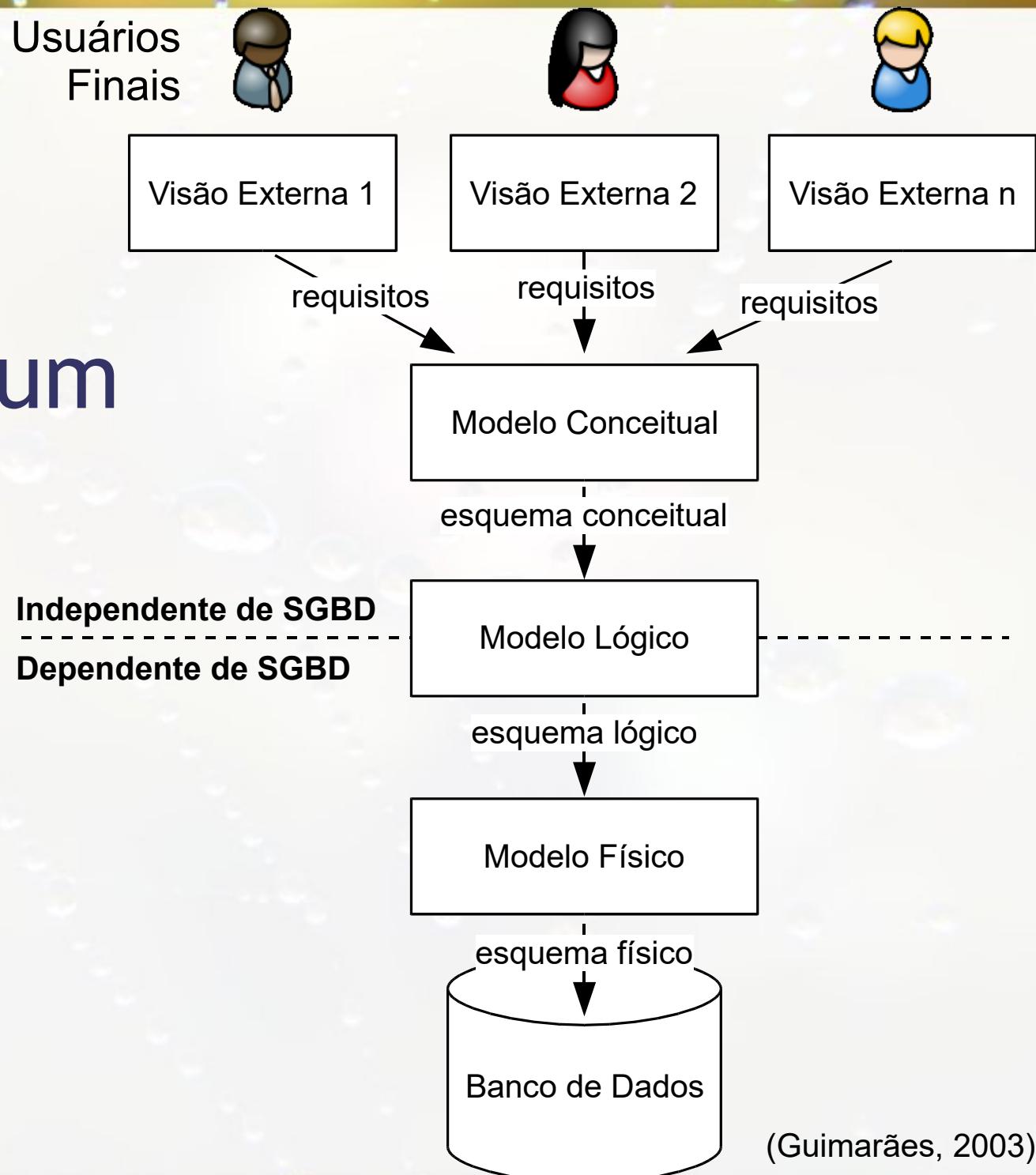
- Diversos slides baseados no curso de BD do Prof. André Santanchè (UNICAMP)
 - Site: <http://www.ic.unicamp.br/~santanche>
 - Canal Youtube:
<https://www.youtube.com/santanche>

Modelo Lógico

Banco de Dados: Teoria e Prática

André Santanchè e Patrícia Cavoto
Instituto de Computação - UNICAMP
Agosto 2015

Projeto de um BD



Modelo/Esquema Lógico

- Dependente de um SGBD particular
- Associado a um “modelo de dados de implementação” (Elmasri, 2005)



Flat File Model

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	
Record 3	SR-301	33	

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Object-Oriented Model

Object 1: Maintenance Report

Date	
Activity Code	
Route No.	
Daily Production	
Equipment Hours	
Labor Hours	

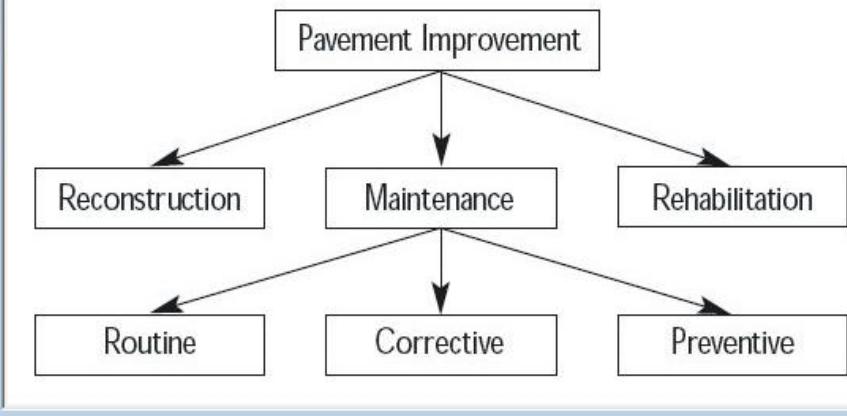
Object 1 Instance

01-12-01
24
I-95
2.5
6.0
6.0

Object 2: Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

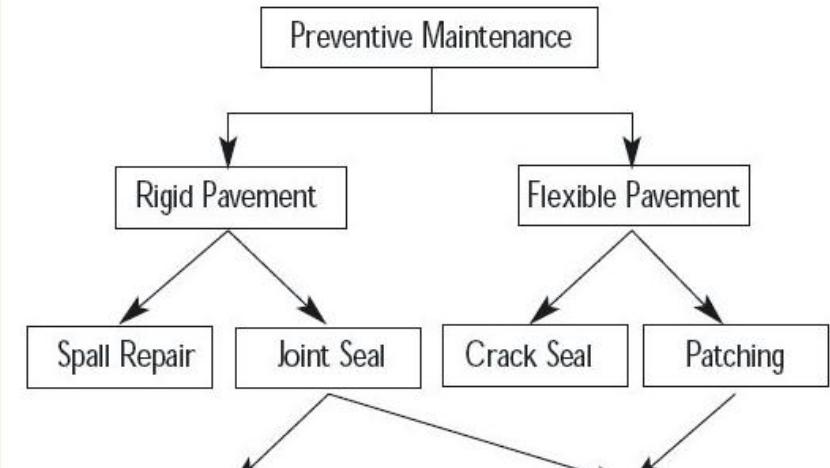
Hierarchical Model



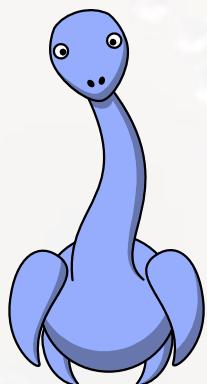
By Marcel Douwe Dekker

https://en.wikipedia.org/wiki/File:Database_models.jpg

Network Model



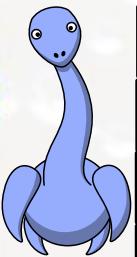
Describing Prehistoric Animals



MNHN A. C. 8592

Is a	Plesiosaurus dolichodeirus	
Origin	Lyme Regis	England
Recognized	1824	
Size	5	

Describing Prehistoric Animals



SIPB R 90

Is a	Plesiosaurus dolichodeirus	
Origin	Lyme Regis	England
Recognized	1830	
Size	5	



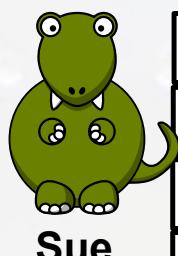
STC223

Is a	Plesiosaurus gurgitis	
Origin	St. Croix	Switzerland
Recognized	1964	
Size	3.5	



MNHN 1912.20

Is a	Triceratops horridus	
Origin	Lance Creek	EUA
Recognized	1889	
Size	9	



Sue

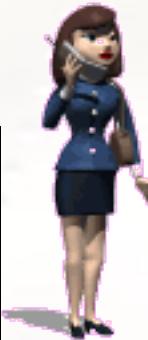
FMNH PR2081

Is a	Tyrannosaurus rex	
Origin	Hell Creek	EUA
Recognized	1990	
Size	12.3	

Modelo Relacional

Cliente (C)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

Corrida (R1)

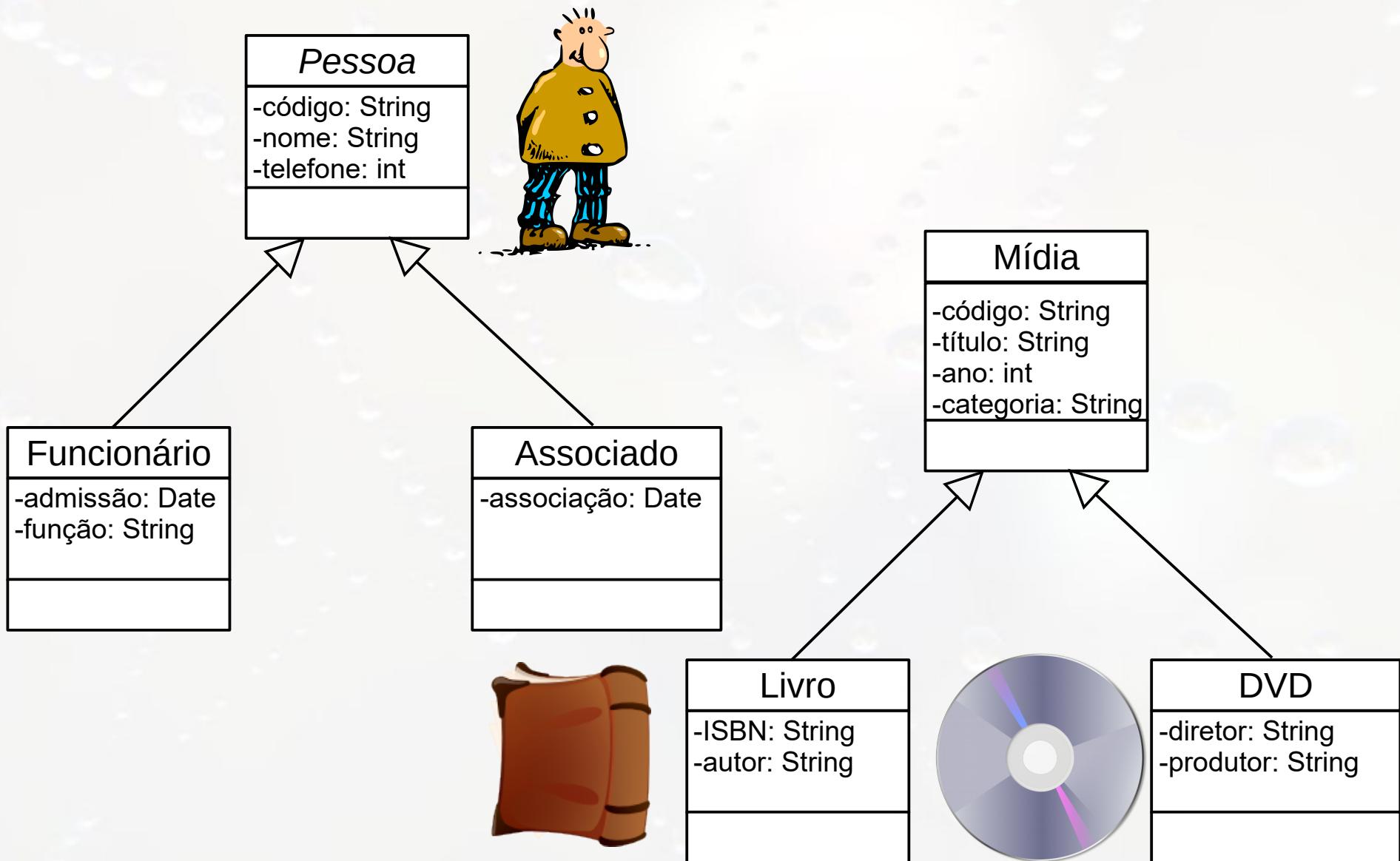
<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo original
por prof. Geovane
Cayres Magalhães

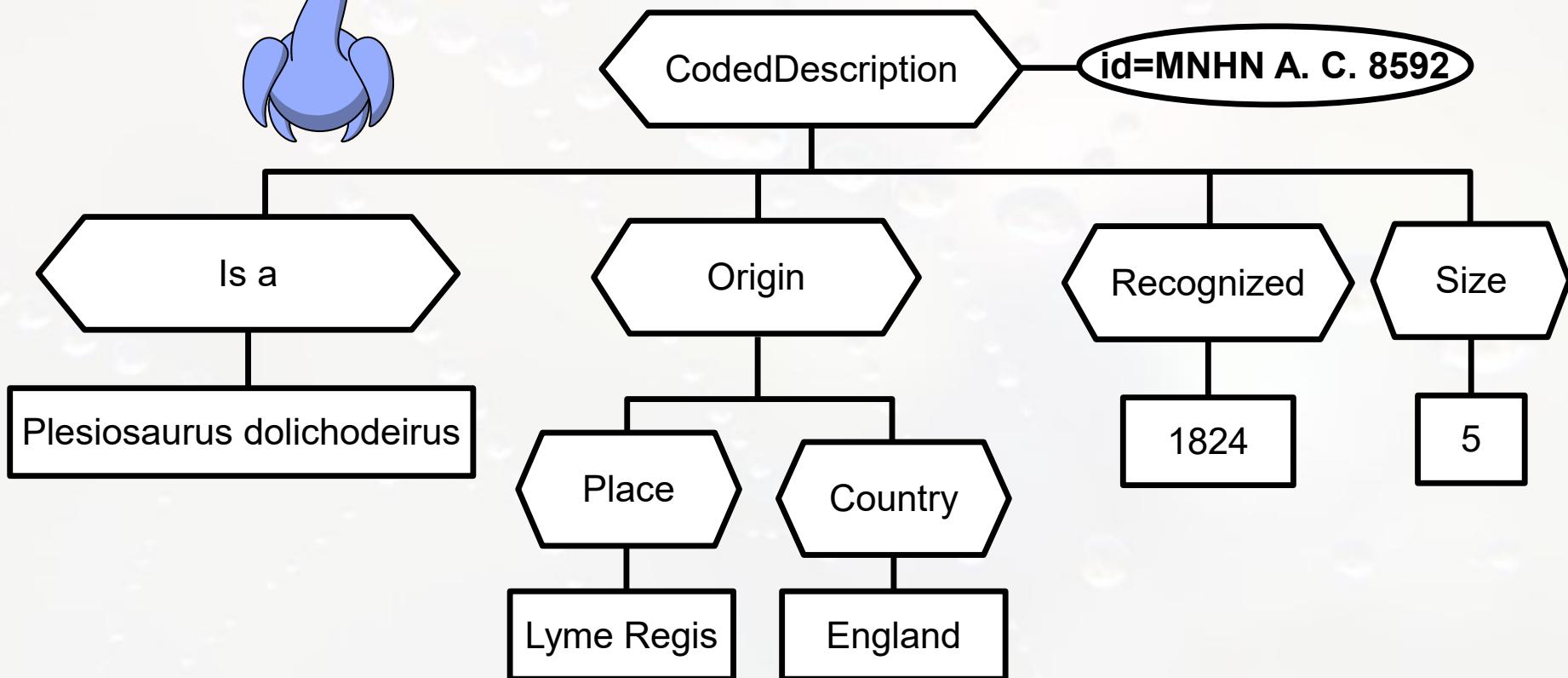
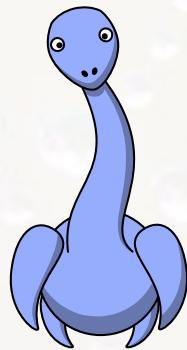
Modelo Relacional

Id	Is a	Origin Place	Origin Country	Recognized	Size
MHN A. C. 8592	Plesiosaurus dolichodeirus	Lyme Regis	England	1824	5
SIPB R 90	Plesiosaurus dolichodeirus	Lyme Regis	England	1830	5
STC223	Plesiosaurus gurgitis	St. Croix	Switzerland	1964	3.5
MHN 1912.20	Triceratops horridus	Lance Creek	EUA	1889	9
FMNH PR2081	Tyrannosaurus rex	Hell Creek	EUA	1990	12.3

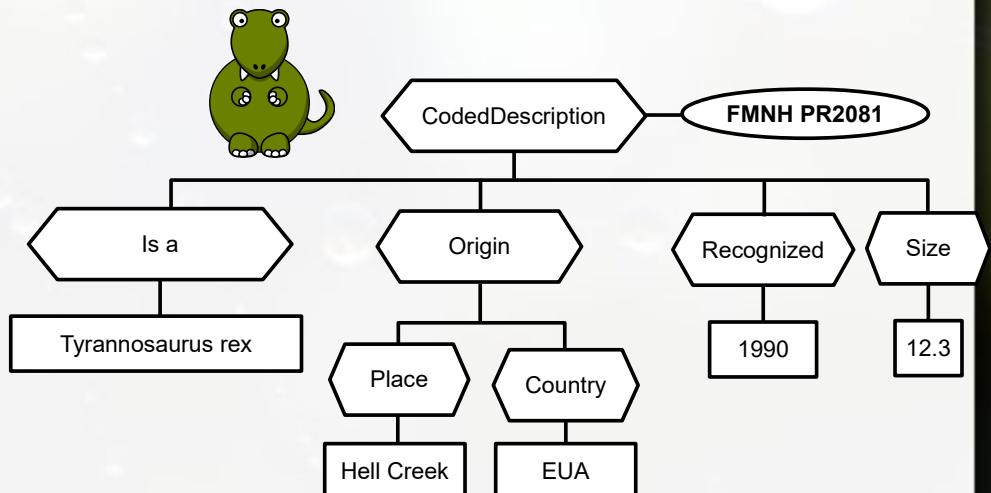
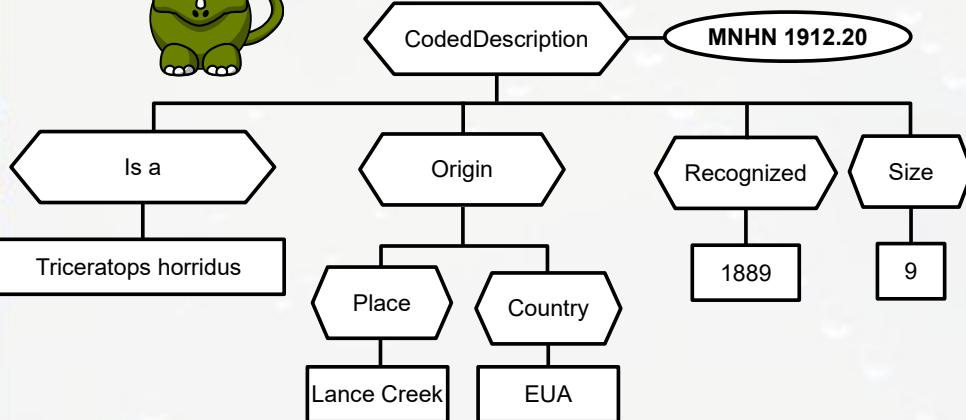
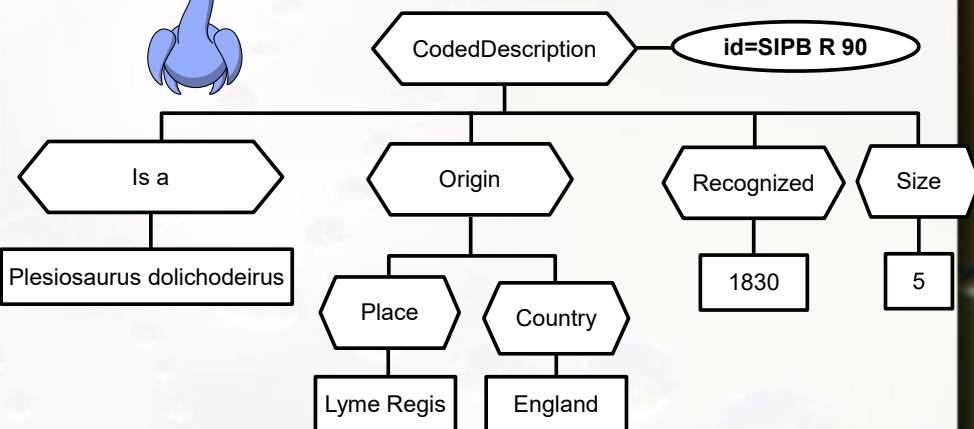
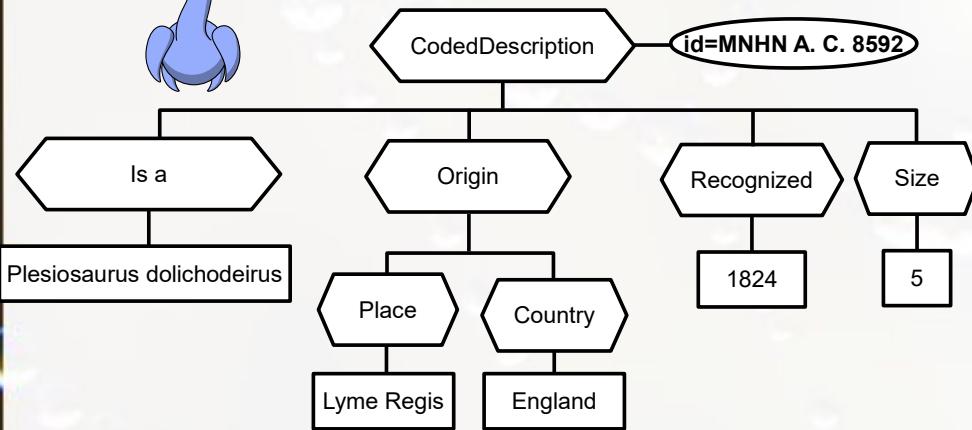
Modelo Orientado a Objetos



Modelo Hierárquico (Documentos/XML)

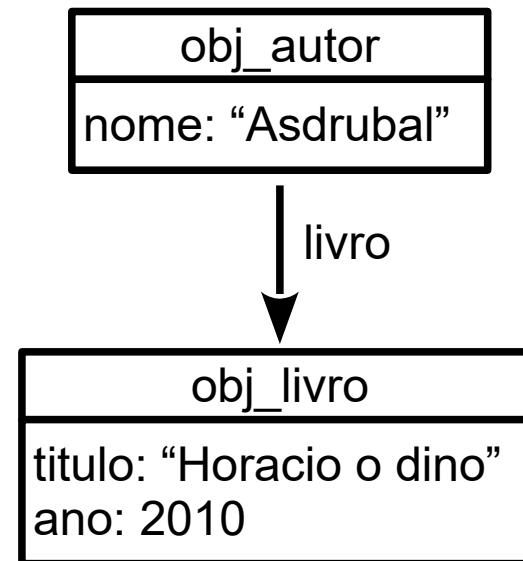


Modelo Hierárquico (Documentos/XML)

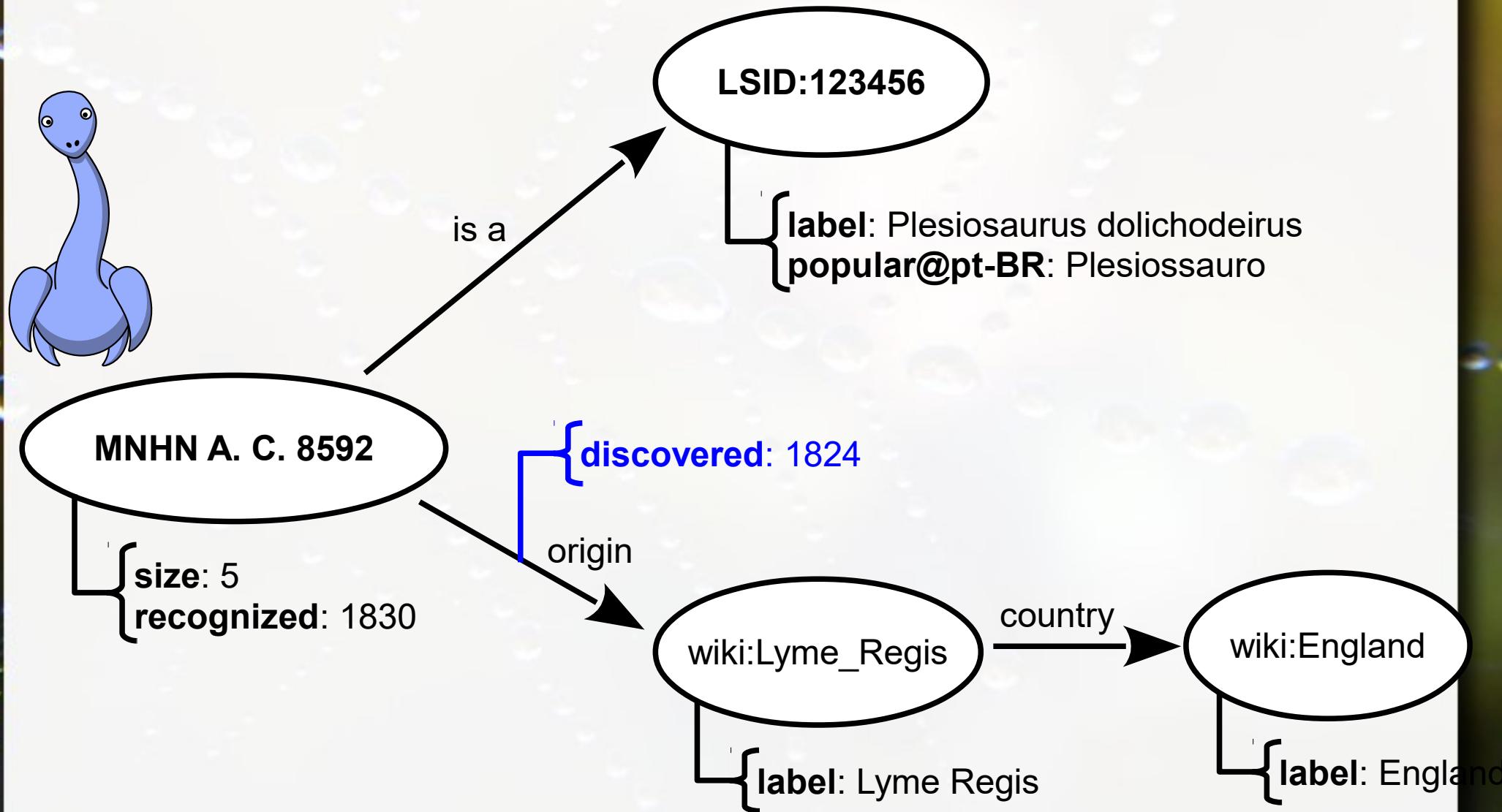


Modelo Hierárquico (JSON - Documentos + OO)

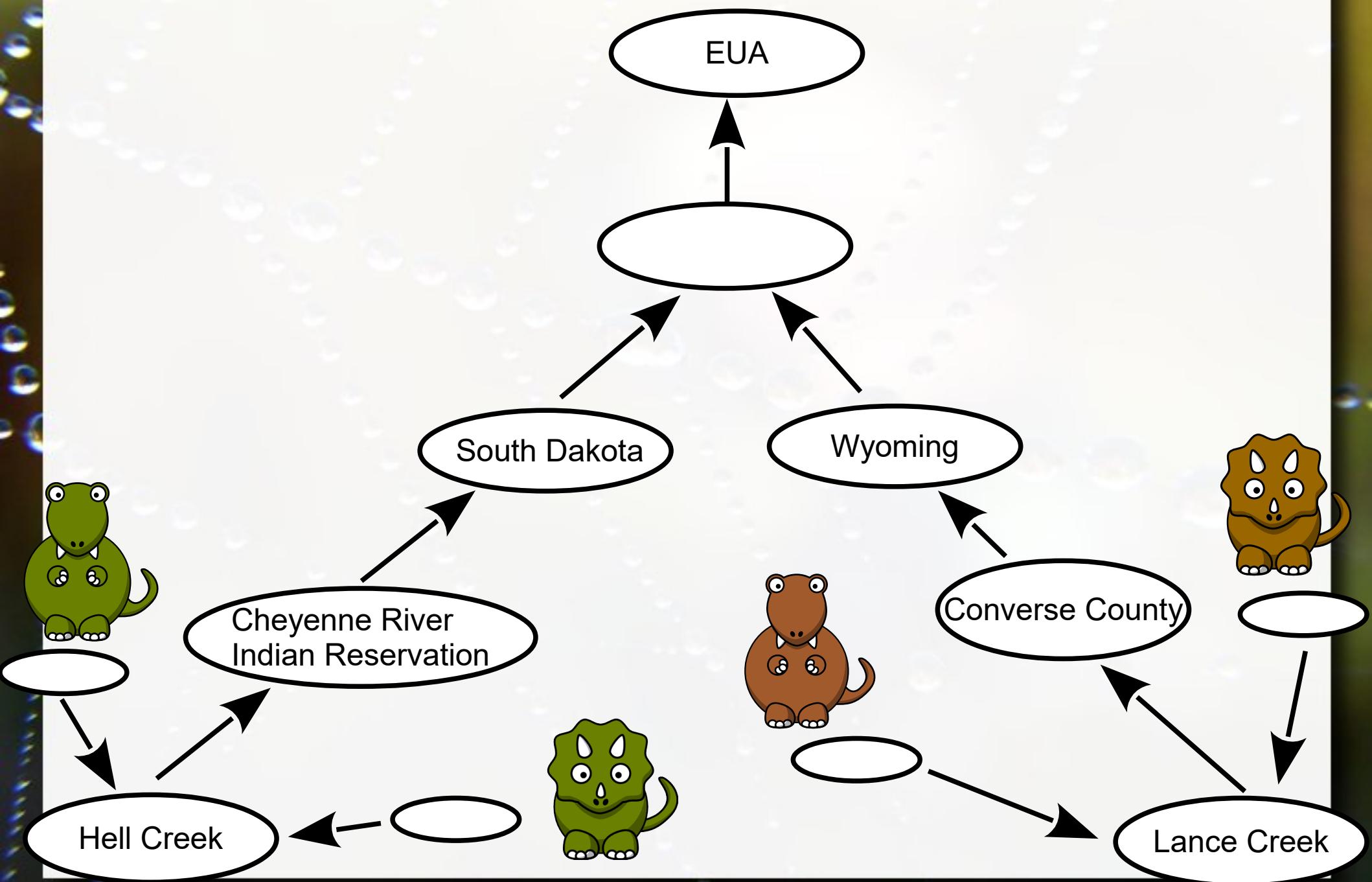
```
{  
  "nome": "Asdrubal",  
  "livro": {  
    "titulo": "Horacio o dino",  
    "ano": 2010  
  }  
}
```



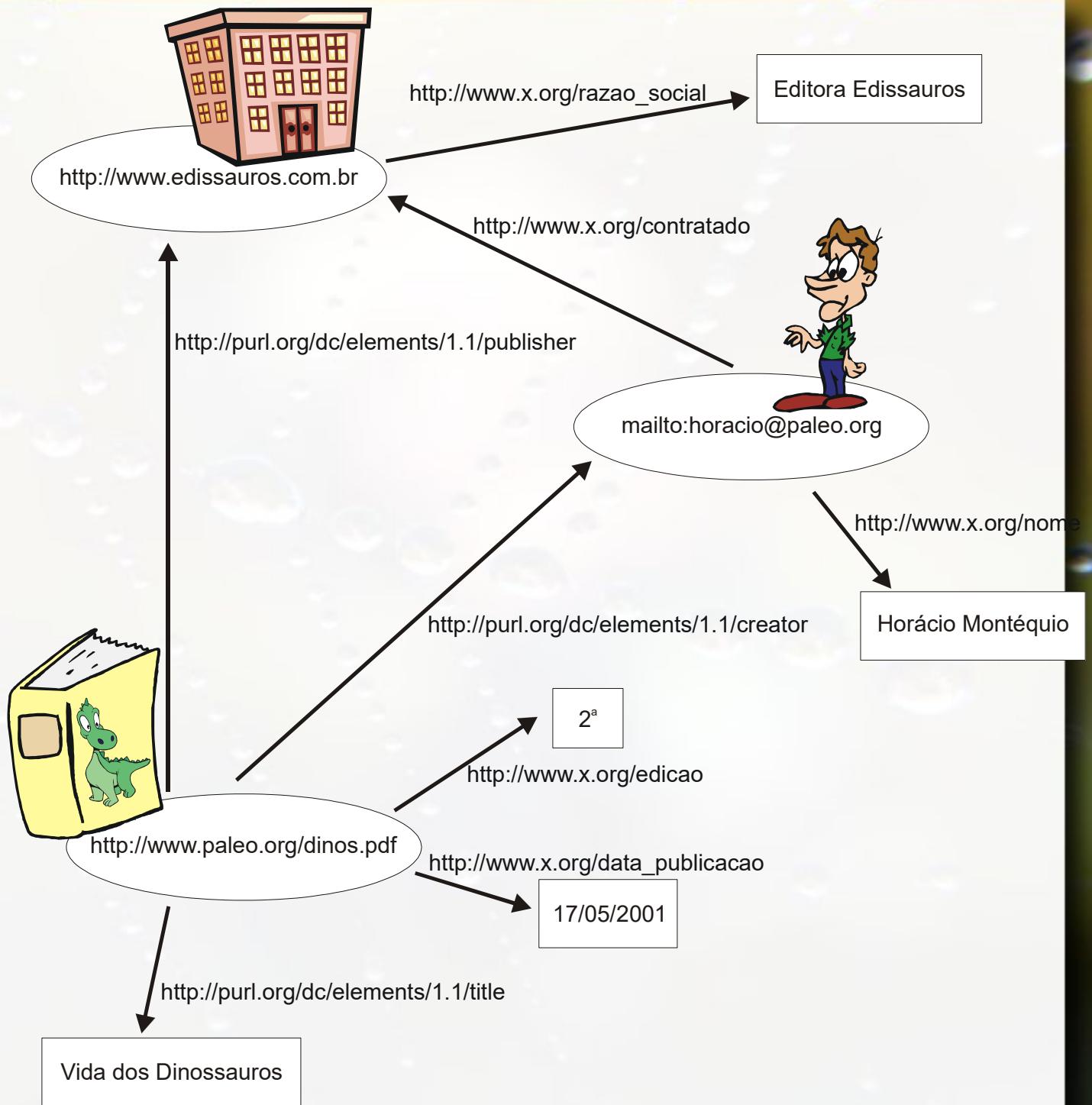
Modelo de Grafos



Modelo de Grafos



Modelo de Grafos



Modelo Chave/Valor

setItem(chave, valor)	adiciona/atualiza par chave-valor
getItem(chave)	recupera o valor associado à chave
key(n)	recupera a enésima chave
removeItem(chave)	remove o par que possui a chave
length	indica o número de pares chave-valor
clear()	remove todos os dados do repositório

Referências

- Heuser, Carlos Alberto (2004) **Projeto de Banco de Dados**. Editora Sagra Luzzato, 5^a edição.
- Ramakrishnan, Raghu; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3rd edition.

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.

André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at:
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

O Modelo Relacional

Banco de Dados

Luiz Celso Gomes-Jr

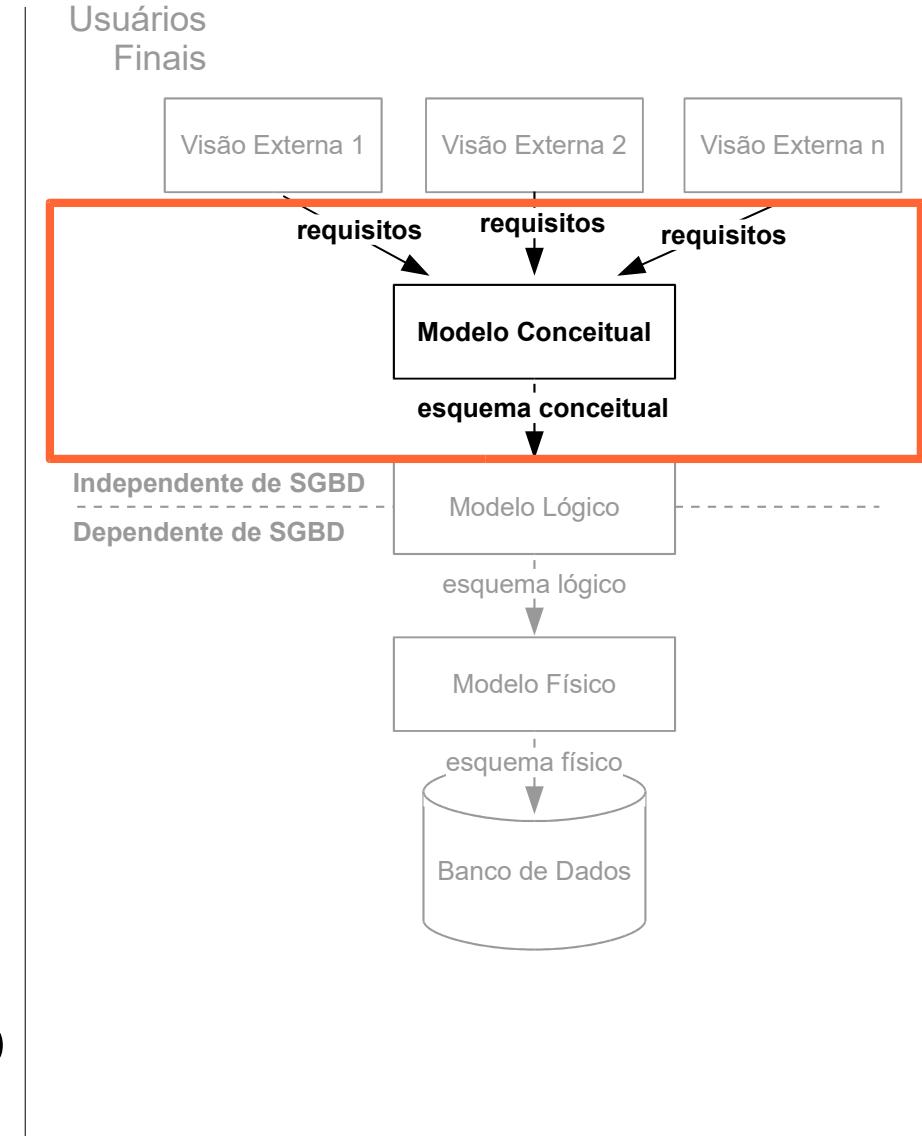
gomesjr@dainf.ct.utfpr.edu.br

Agenda

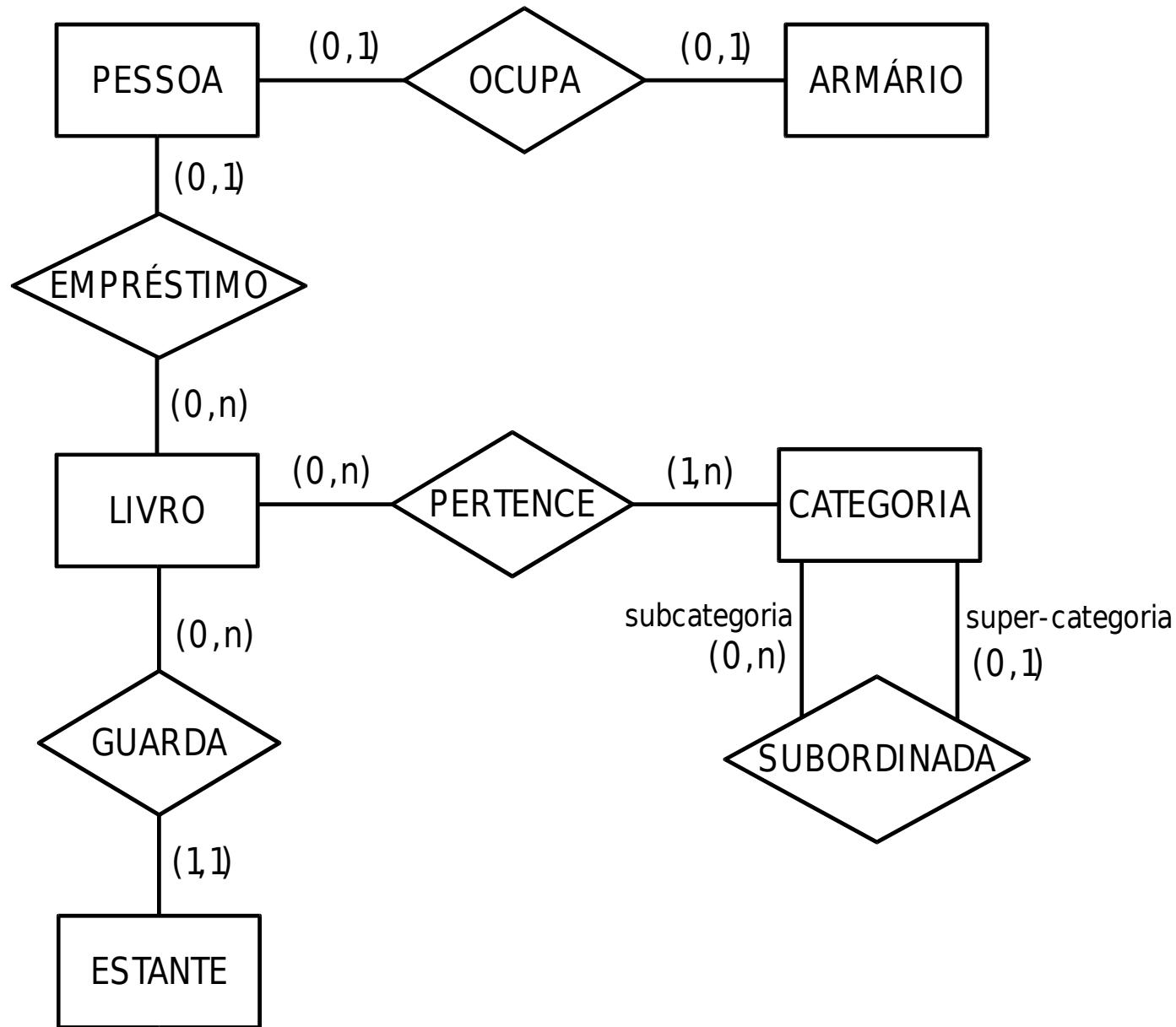
- Recapitação
- Modelo Relacional
- Restrições de Integridade
- Exercícios
- Além do modelo relacional

Modelo/Esquema Conceitual

- Descreve estrutura do Banco de Dados
 - entidades, tipos de dados, relações, restrições etc.
- Independente de implementação em SGBD
 - oculta detalhes de armazenamento físico

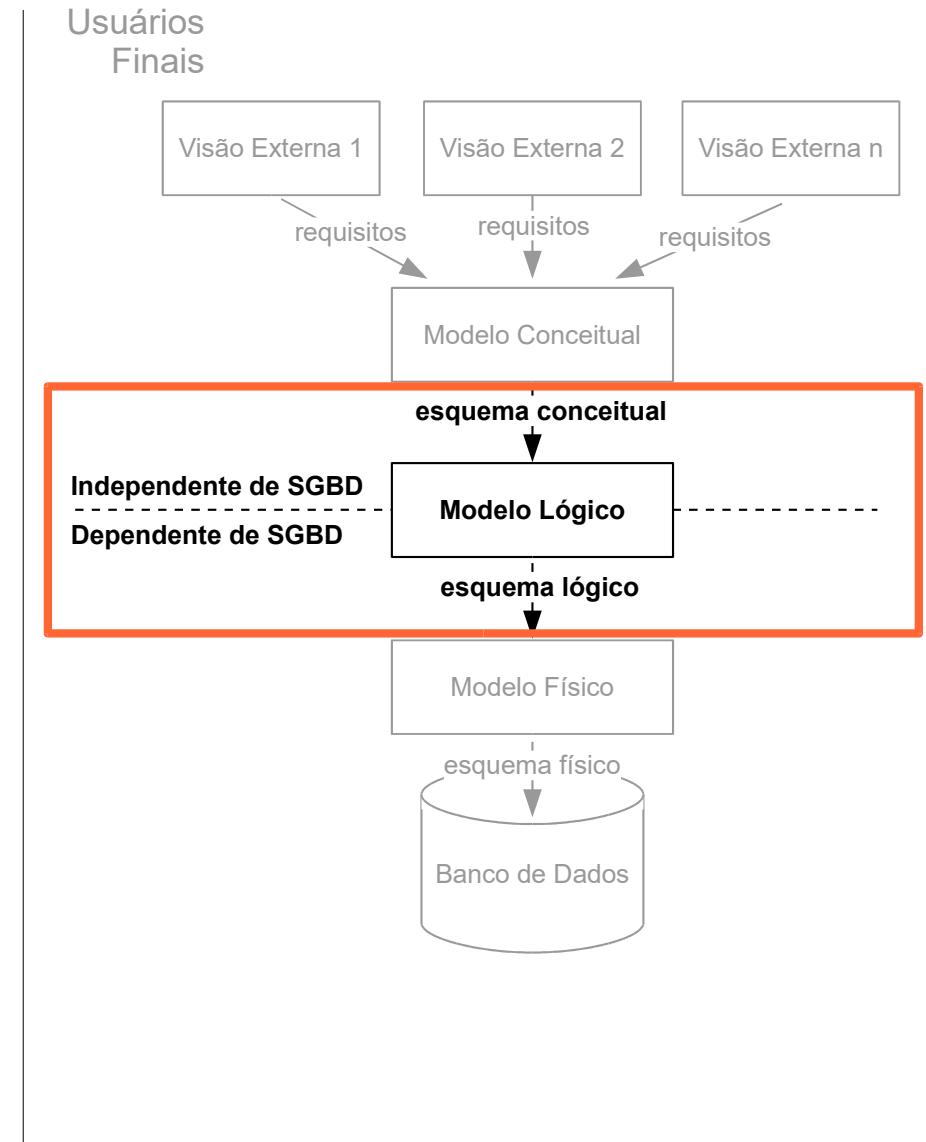


Modelo ER



Modelo/Esquema Lógico

- Dependente de um SGBD particular
- Associado a um “modelo de dados de implementação” (Elmasri, 2005)



Modelo Relacional

- Proposto por E. F. Codd em 1970 no artigo:
 - “A Relational Model for Large Shared Data Banks”
- Resposta a um cenário complexo de incompatibilidade de modelos e sistemas
- Independência da estrutura interna
 - “Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed [...]”
(Codd, 1970)

Modelo Relacional

- Modelo mais amplamente utilizado por SGBDs
- Maiores empresas de informática oferecem soluções: IBM, Microsoft, Oracle, SAP
- Grandes projetos Open Source: MySQL, PostgreSQL, SQLite
- De celulares à data centers
- Mercado de U\$24bi (2012)

Larry Ellison, ORACLE



Relação

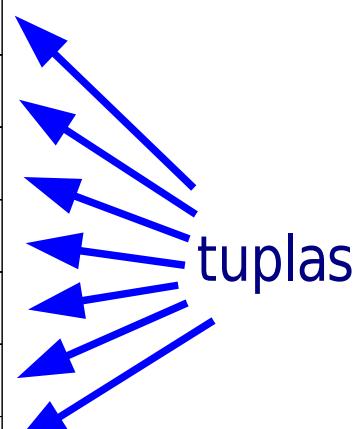
- Banco de Dados Relacional contém um conjunto de **Relações**
- **Relação** – informalmente pode ser visualizada como uma tabela

Relação (Tabela)

- Relação = conjunto não ordenado de tuplas (linhas)
- Não existem tuplas duplicadas

Livro  relação

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565



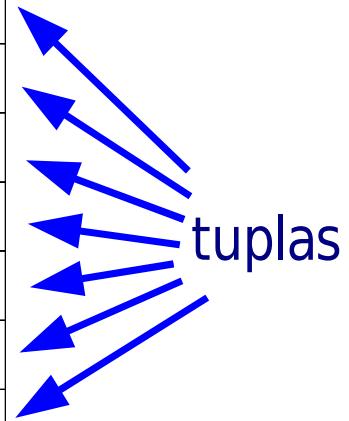
tuplas

Relação (Tabela)

- Tupla = conjunto ordenado de atributos
- Valores de atributos são atômicos e monovalorados

Livro  relação

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565



Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
 - a) nº de tabelas de uma aplicação típica (média)
 - b) nº de tabelas de uma aplicação grande e complexa
 - c) nº total de tabelas distintas, considerando todas as aplicações
 - d) média de nº de atributos por tabela
 - e) número de atributos das "maiores" tabelas
 - f) nº de DBAs envolvidos
 - g) nº de administradores de dados envolvidos

Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
 - a) nº de tabelas de uma aplicação típica (média): **30 tabelas**
 - b) nº de tabelas de uma aplicação grande e complexa: **500 tabelas (quinhentas!)**
 - c) nº total de tabelas distintas, considerando todas as aplicações: **10.000 tabelas**
 - d)média de nº de atributos por tabela: **10 atributos**
 - e)número de atributos das "maiores" tabelas: **80 atributos**

Exercício de aquecimento

- Estime os dados abaixo pensando em uma grande empresa como a Petrobras:
 - n° de DBAs envolvidos: cerca de 15 (parte física, replicação, backup, controle de acesso, performance, servidores - não contando as pessoas de infraestrutura que mantém os servidores em si)
 - n° de administradores de dados envolvidos: cerca de 40 (envolvidos na elaboração e manutenção dos modelos e esquemas, manutenção de metadados, elaboração de scripts, procedures, views; dominam os assuntos / negócio)

Relação (Tabela)

- Nome dos atributos – título das colunas

Livro

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565

nome dos
atributos

Esquema

Livro

ISBN	Título	Categoria	Autor	Ano
9580471444	Vidas Secas	Romance	Graciliano Ramos	1938
958047950X	Agosto	Romance	Rubem Fonseca	1990
0554253216	Micrographia	Ciências	Robert Hooke	1665
0195087445	Divina Comédia	Poesia	Dante Alighieri	1308
0559274289	Le Opere	Ciências	Galileu Galilei	1811
0451526929	Hamlet	Drama	William Shakespeare	1599
1603033785	Othello	Drama	William Shakespeare	1565

esquema

tuplas

Esquema

- Esquema de uma Relação:
 - Denotado por $R(A_1, A_2, \dots, A_n)$
 - $R \Rightarrow$ nome da relação
 - $A_1, A_2, \dots, A_n \Rightarrow$ atributos da relação
- Exemplo:
 - LIVRO (ISBN, Título, Categoria, Autor, Ano)
 - Nome da relação: LIVRO
 - Atributos: ISBN, Título, Categoria, Autor, Ano

Nome do Atributo

- Indica o significado dos valores do atributo
- Designa o papel realizado por um domínio na relação

Domínio do Atributo

- Determina os valores possíveis para um atributo
- Conjunto de valores atômicos
- Representação: $D(A_x)$ – domínio do atributo A_x
- Um domínio tem uma definição lógica
 - $D(\text{ISBN})$: identificador de até 13 dígitos
- Um domínio está associado a um tipo de dados
 - $D(\text{Título})$: string de até 100 caracteres
 - $D(\text{Ano})$: inteiro de 4 dígitos

Esquema + Domínios

- Notação Usual:
 - nome da relação e atributos + tipos
 - Exemplo:
 - LIVRO (ISBN: string, Título: string, Categoria: string, Autor: string, Ano: integer)

Relação ou Estado da Relação

- Uma relação r de um esquema $R(A_1, A_2, \dots, A_n)$
 - conjunto de tuplas $r=\{t_1, t_2, \dots, t_n\}$ em que
 - t_i lista ordenada de valores $t = \langle v_1, v_2, \dots, v_n \rangle$ em que
 - v_i é um elemento de $\text{dom}(A_i)$ ou um valor nulo
 - $t[A_i]$ ou $t.A_i \rightarrow$ valor do atributo i
 - $t[i] \rightarrow$ notação alternativa (posicional)
(Elmasri, 2010)

Tupla

- Esquema relação
 - LIVRO (ISBN, Título, Categoria, Autor, Ano)
- Tupla
 - $t = <9580471444, \text{Vidas Secas}, \text{Romance}, \text{Graciliano Ramos}, 1938>$
 - $t[\text{Categoria}] = <\text{Romance}>$
 - $t[\text{Ano}] = <1938>$
 - $t[2] = <\text{Vidas Secas}>$

Valor Nulo

- Tuplas podem conter o valor nulo (NULL) em atributos
- Nulo indica valor:
 - desconhecido
 - não disponível
 - indefinido (não se aplica à tupla)

(Elmasri, 2010)

Relação (matemática)

- Construído a partir do conceito de **conjuntos matemáticos**
- Fundamentação matemática é importante para definição de linguagens de consulta e para otimização de processamento

Relação (matemática)

- Considerando os conjuntos S_1, S_2, \dots, S_n (não necessariamente distintos)
- R é uma relação destes n conjuntos se:
 - for uma relação de n tuplas em que:
 - primeiro elemento for de S_1
 - segundo elemento for de S_2
 - ...
- R é um subconjunto do produto cartesiano

$S_1 \times S_2 \times \dots \times S_n$

(Codd, 1970)

Relação (matemática)

- Uma relação é um conjunto, portanto não admite valores repetidos de tuplas

Sumário dos Conceitos

Termo Informal	Termo Formal
Tabela	Relação
Cabeçalho da Coluna	Atributo
Todos os valores possíveis de uma Coluna	Domínio
Linha	Tupla
Definição da Tabela	Esquema da Relação
Tabela populada	Estado da Relação

(Elmasri, 2007)

Restrições de Integridade

- Devem ser verdadeiras para cada instância do banco de dados
- Restrições:
 - de domínio
 - de chave
 - de integridade de entidades
 - de integridade referencial

Restrições de Domínio

- Valores dos atributos devem ser atômicos
- Valor do atributo:
 - tem que ser do domínio do atributo
 - pode ser nulo (se permitido pelo atributo)

Restrições de Chave

- Chave: identifica tuplas e é usada para estabelecer relacionamentos entre tabelas
- Superchave
 - conjunto de atributos de uma relação
 - não existem duas tuplas em uma relação com a mesma superchave
- Chave
 - Superchave que atende à “condição mínima”:
 - Se qualquer atributo for removido deixa de ser superchave

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave 

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave ✘

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave?

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Superchave 

Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave?

Exemplo

Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave ✕

Exemplo Superchave & Chave

ISBN	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências
0195087445	Divina Comédia	Dante Alighieri	1308	Poesia
0559274289	Le Opere	Galileu Galilei	1811	Ciências
0451526929	Hamlet	William Shakespeare	1599	Drama
1603033785	Othello	William Shakespeare	1565	Drama

Chave

Chave Primária

- Chave cujos valores distinguem uma tupla das demais dentro de uma relação
- Identifica a tupla de forma única
- Usada como referência a partir de outra tupla
- Atributos da chave primária recebem sublinhado:
 - LIVRO (ISBN, Título, Categoria, Autor, Ano)

Chave Primária

Livro



LIVRO (ISBN, Título, Autor, Ano, Categoria)

<u>ISBN</u>	Título	Autor	Ano	Categoria
9580471444	Vidas Secas	Graciliano Ramos	1938	Romance
958047950X	Agosto	Rubem Fonseca	1990	Romance
0554253216	Micrographia	Robert Hooke	1665	Ciências

Chave Primária Estante



ESTANTE (Código, Tamanho)

ESTANTE

<u>Código</u>	Tamanho
1A	simples
2A	duplo
1B	simples
2B	duplo

Exercício 1

parte 1

- Liste as superchaves e chaves da seguinte relação:

COL1	COL2	COL3
A	10	F
A	15	F
D	15	M
B	5	F
A	5	M
B	10	M

Agradecimentos

- Diversos slides foram baseados no curso de BD do Prof. André Santanchè (UNICAMP)
 - Site: <http://www.ic.unicamp.br/~santanche>
 - Canal Youtube:
<https://www.youtube.com/santanche>

Exercício 1

parte 2

- As informações contidas na relação em questão são suficientes para determinar a chave primária? Justifique.

Exercício 1

parte 3

- Uma relação sempre terá uma chave?
Justifique.

Integridade de Entidade

- Valor da chave primária não pode ser nulo
 - já que chave primária identifica tuplas
- Integridade de Entidade e Integridade de Chave se aplicam a relações individuais

(Elmasri, 2010)

BD Relacional

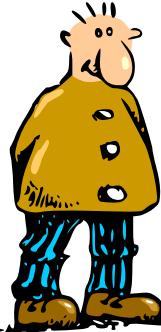
- Banco de Dados Relacional
 - conjunto de esquemas $S = \{R_1, R_2, \dots, R_n\}$
 - conjunto de restrições de integridade RI
- Estado ou instância do Banco de Dados Relacional
 - conjunto de estados da relação $DB = \{r_1, r_2, \dots, r_n\}$
 - r_i é instância de R_i
 - Estados de r_i satisfazem restrições de integridade

(Elmasri, 2010)

Chave Estrangeira

- Conjunto de campos em uma relação que é usado para fazer referência à chave primária da segunda relação
- Valor de cada chave estrangeira deve corresponder à chave primária existente da relação referenciada
- Funciona como um ‘ponteiro lógico’
(Ramakrishman, 2003)

Pessoa ocupa Armário



PESSOA (Código, Nome, Telefone)



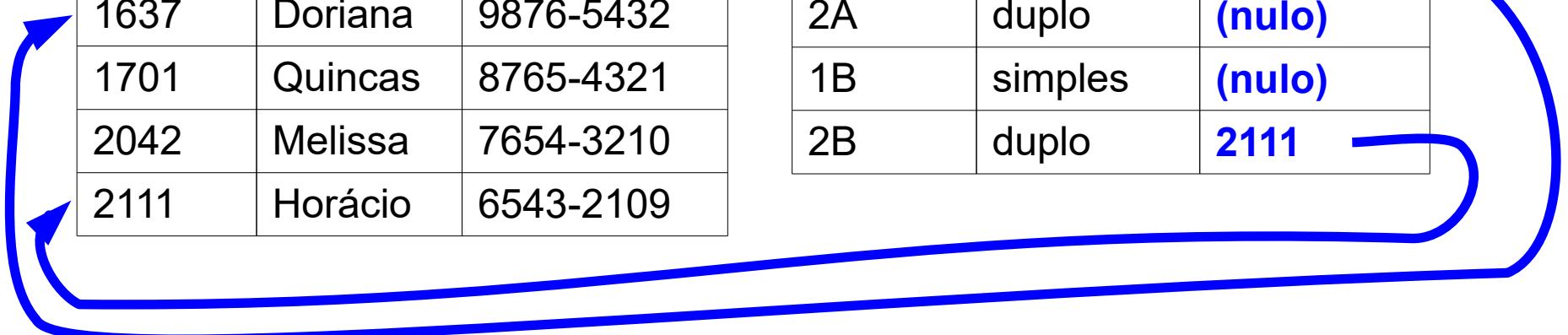
ARMÁRIO (Código, Tamanho, **Ocupante**)

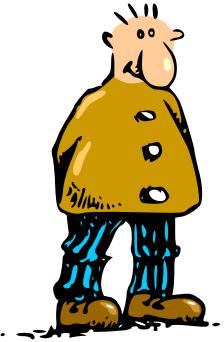
PESSOA

<u>Código</u>	Nome	Telefone
1525	Asdrúbal	5432-1098
1637	Doriana	9876-5432
1701	Quincas	8765-4321
2042	Melissa	7654-3210
2111	Horácio	6543-2109

ARMÁRIO

<u>Código</u>	Tamanho	Ocupante
1A	simples	1637
2A	duplo	(nulo)
1B	simples	(nulo)
2B	duplo	2111





Pessoa ocupa Armário

PESSOA (Código, Nome, Telefone)



ARMÁRIO (Código, Tamanho)

OCUPA (CodPessoa, CodArmário, Data, Hora)

PESSOA

<u>Código</u>	Nome	Telefone
1525	Asdrúbal	5432-1098
1637	Doriana	9876-5432
1701	Quincas	8765-4321
2042	Melissa	7654-3210
2111	Horácio	6543-2109

OCUPA

<u>CodPessoa</u>	<u>CodArmário</u>	Data	Hora
1637	1A	03/08	10:20
2111	2B	03/08	11:45

ARMÁRIO

<u>Código</u>	Tamanho
1A	simples
2A	duplo
1B	simples
2B	duplo

Exemplo: Táxis

Cliente

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Dorianा	567.387.387-44
1780	Quincas	546.373.762-02



Exemplo: Táxis

Táxi

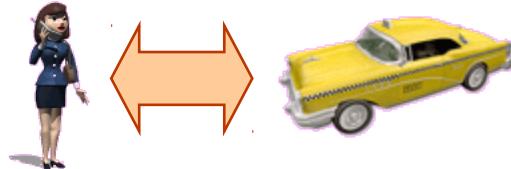
<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Exemplo: Táxis

Corrida

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003



Exemplo: Táxis

Chaves Estrangeiras

Corrida

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

} Chave Primária

- SGDB deve garantir a integridade dos dados na inserção, exclusão e alteração de dados

Integridade Referencial

- Tupla deve referenciar tupla existente, utilizando chaves válidas
- SGDB deve garantir a consistência das referências

Exemplo: Táxis

Corrida

<u>CId</u>	<u>Placa</u>	<u>Data Pedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

- Inserção de Corrida: CId e Placa devem existir
- Exclusão de Táxi ou Cliente: não é permitida se existirem corridas que fazem referência

Exercícios (2)

- Faça o modelo relacional do seguinte diagrama:



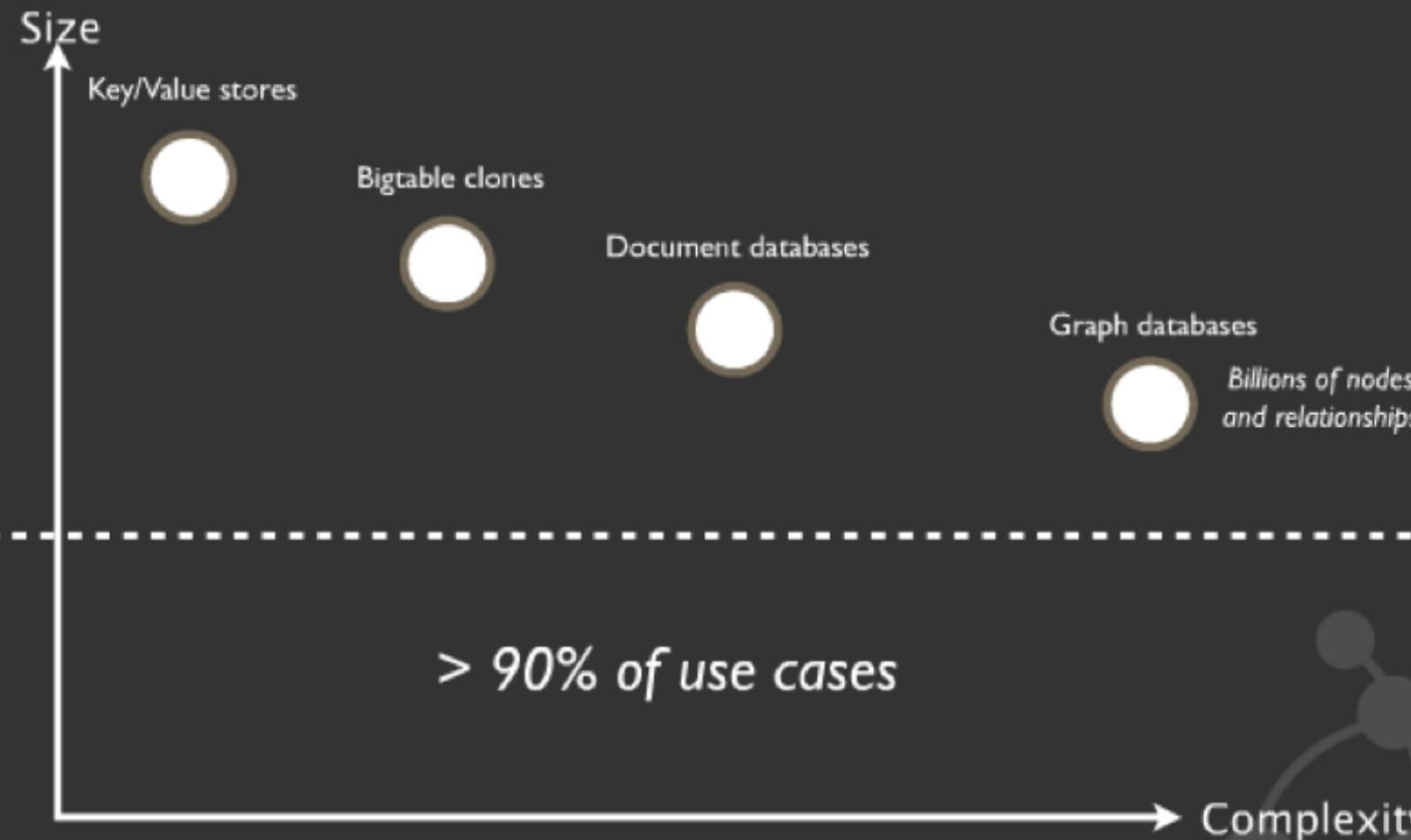
- Desenvolva versões com uma e duas tabelas
- Adicione atributos importantes e defina chaves primárias
- Mostre dados de exemplo

Próximas aulas

- Mapeamento ER -> Relacional
- Modelagem/Normalização
- Consultas
- Otimização de consultas
- Controle de transação

Próximas aulas

Scaling to size vs. Scaling to complexity



Conclusão

- Modelo Relacional dominou a área de banco de dados por décadas
- Importante em diversas aplicações
- Restrições de integridade garantem coesão dos dados
- Algumas aplicações demandam outros modelos

Perguntas?

Referências

- Codd, Edgar Frank (1970) **A relational model of data for large shared data banks.** Communications ACM 13(6), 377-387.
- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados.** Pearson, 6^a edição em português.
- Ramakrishnan, Raghu; Gehrke, Johannes (2003) **Database Management Systems.** McGraw-Hill, 3rd edition.

Aplicações - SQL

Banco de Dados: Teoria e Prática

André Santanchè

Instituto de Computação - UNICAMP

Março de 2016



Linguagens de Query

- Para manipulação e recuperação de dados
- Linguagens de Query (LQ) em BD:
 - Fundamentação formal
 - Subsidiam otimização
- LQ <> linguagens de programação
 - não se espera que sejam “Turing completas”.
 - não pensadas para cálculos complexos.
 - suportam acessos simples e eficientes a extensos conjuntos de dados

Linguagens de

- Para manipulação e recuper
- Linguagens de Query (LQ) em BD:
 - Fundamentação formal
 - Subsidiam otimização
- LQ <> linguagens de programação
 - não se espera que sejam “Turing completas”.
 - não pensadas para cálculos complexos.
 - suportam acessos simples e eficientes a extensos conjuntos de dados

Uma linguagem é dita “Turing completa” se puder ser demonstrado que ela é computacionalmente equivalente à máquina de Turing.

SQL

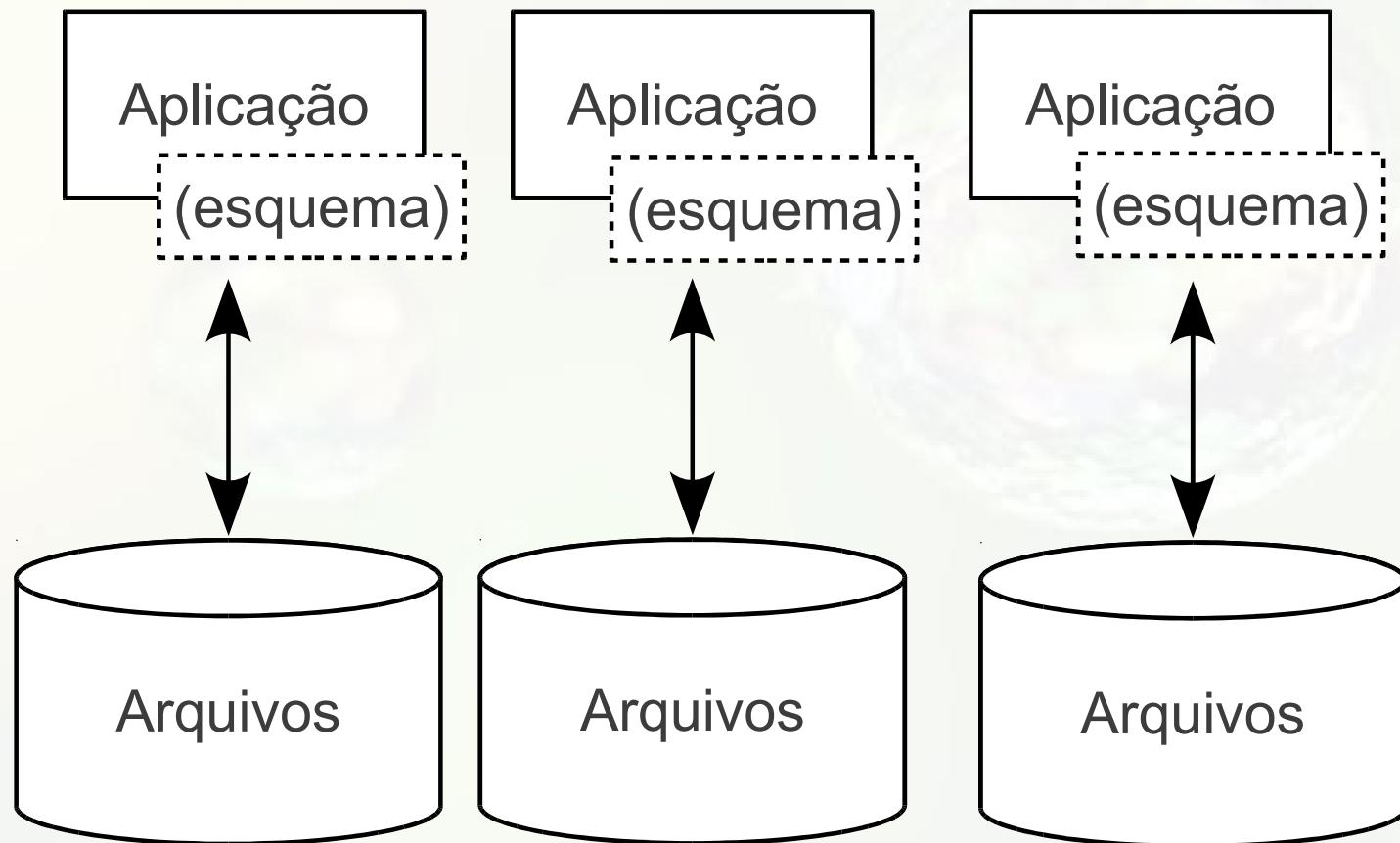
- SQL - Structured Query Language
- Originalmente: SEQUEL - Structured English QUERy Language
- Criada pela IBM Research
 - Interface BD Relacional → SYSTEM R

SQL

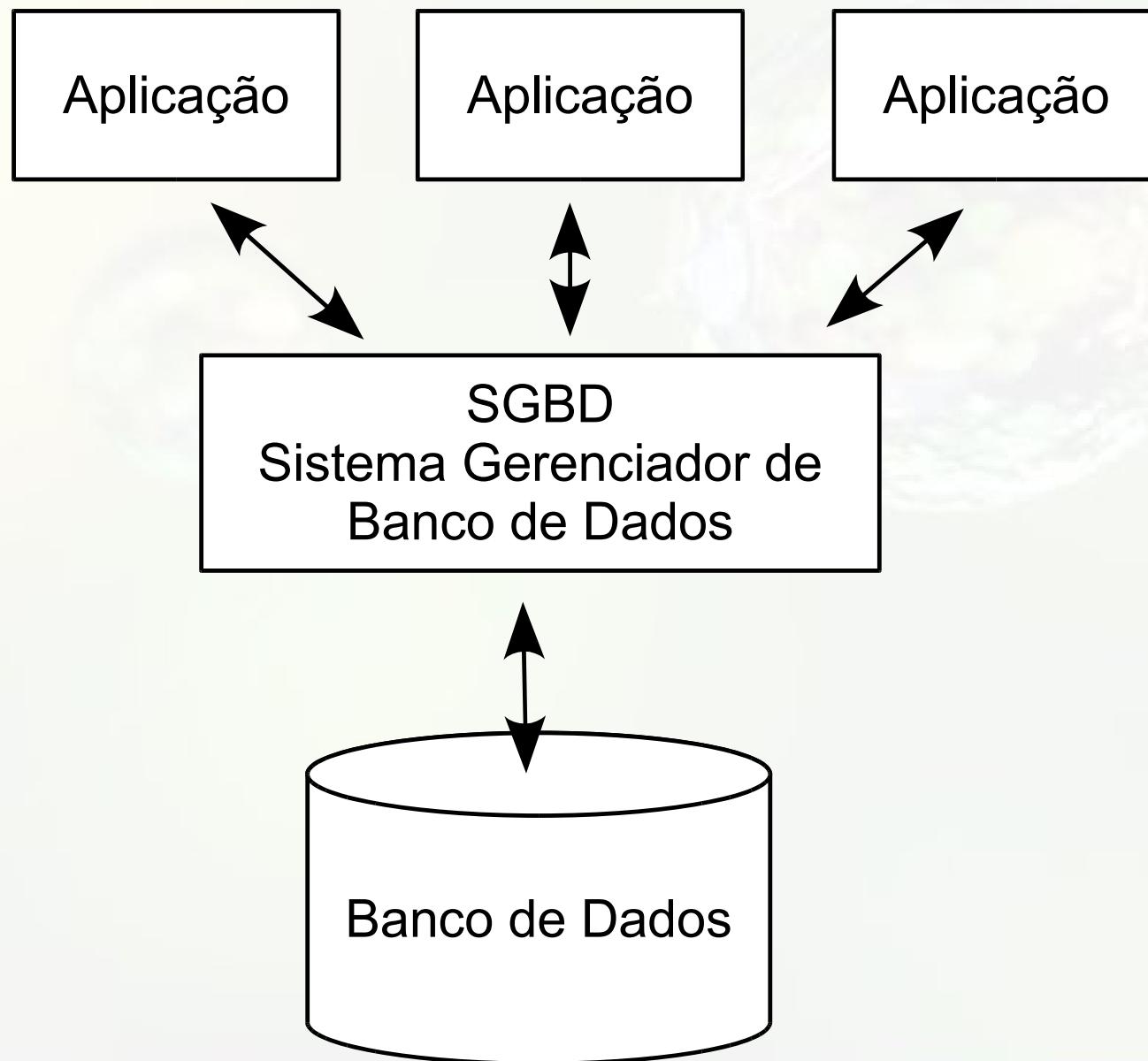
Padronização

- ANSI + ISO
- SQL-86 ou SQL1
- SQL-92 ou SQL2
- SQL:1999 ou SQL3
- SQL:2003
- SQL:2006

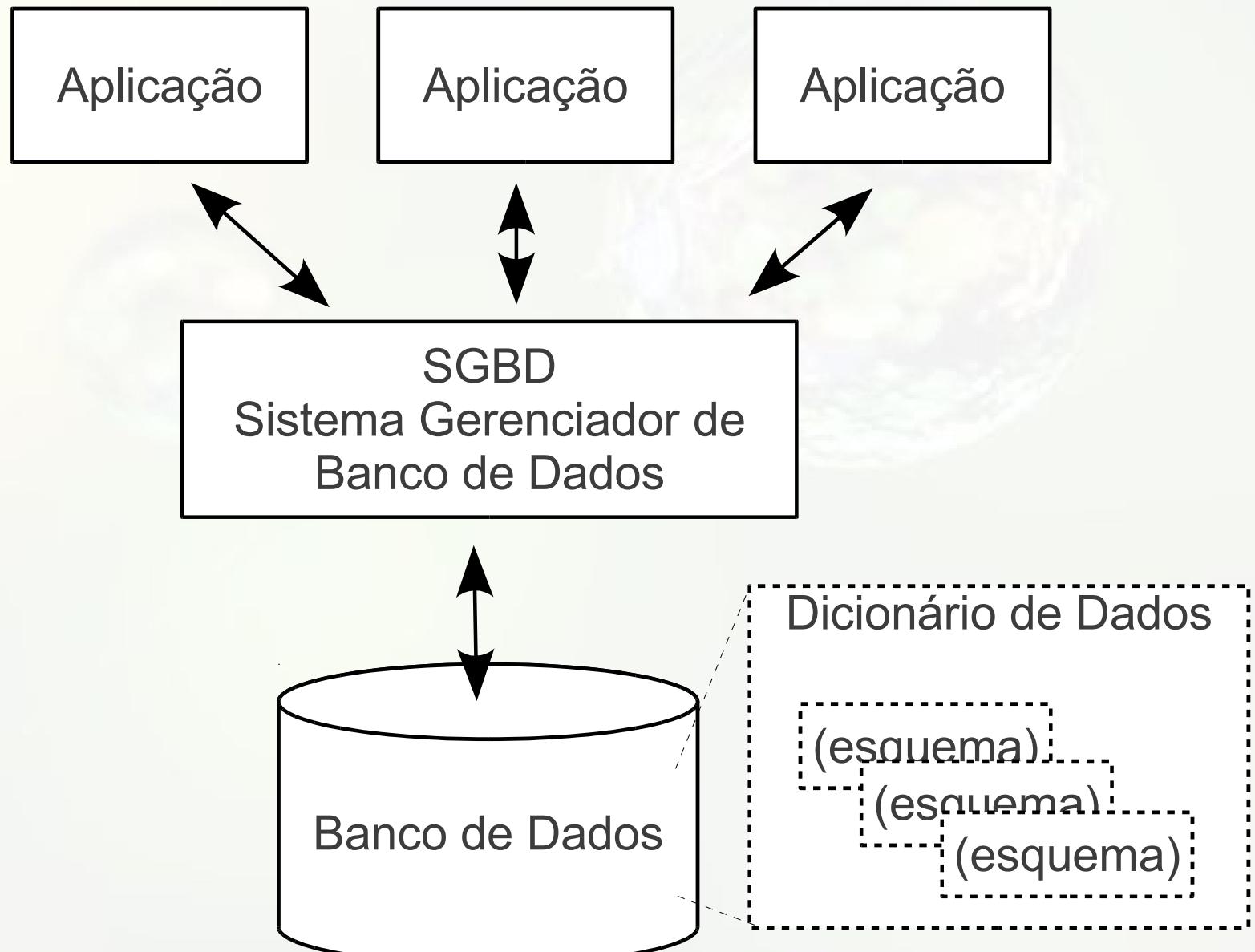
Aplicações e Armazenamento Arquivos



Aplicações e Armazenamento SGBD

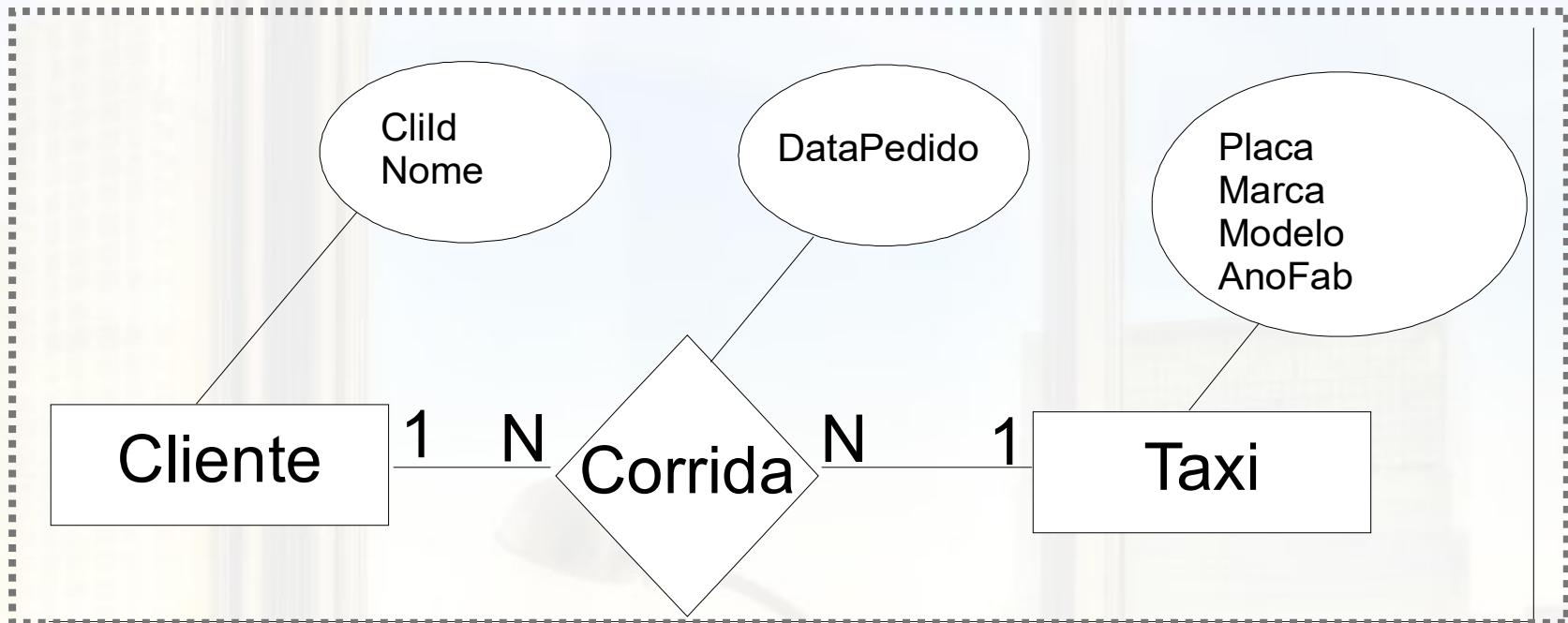


Dicionário de Dados



Caso Prático - Taxis

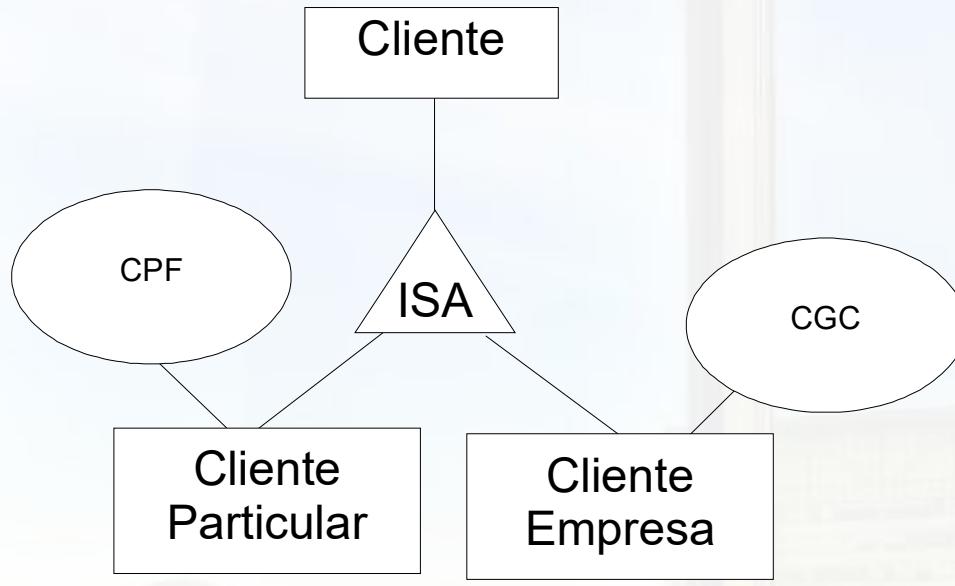
Esquema Conceitual - Exemplo Táxis



Este é um subconjunto do Estudo de Caso proposto “Despacho e controle de Táxis via terminais móveis ligados on-line com um sistema multi-usuário” por prof. Geovane Cayres Magalhães

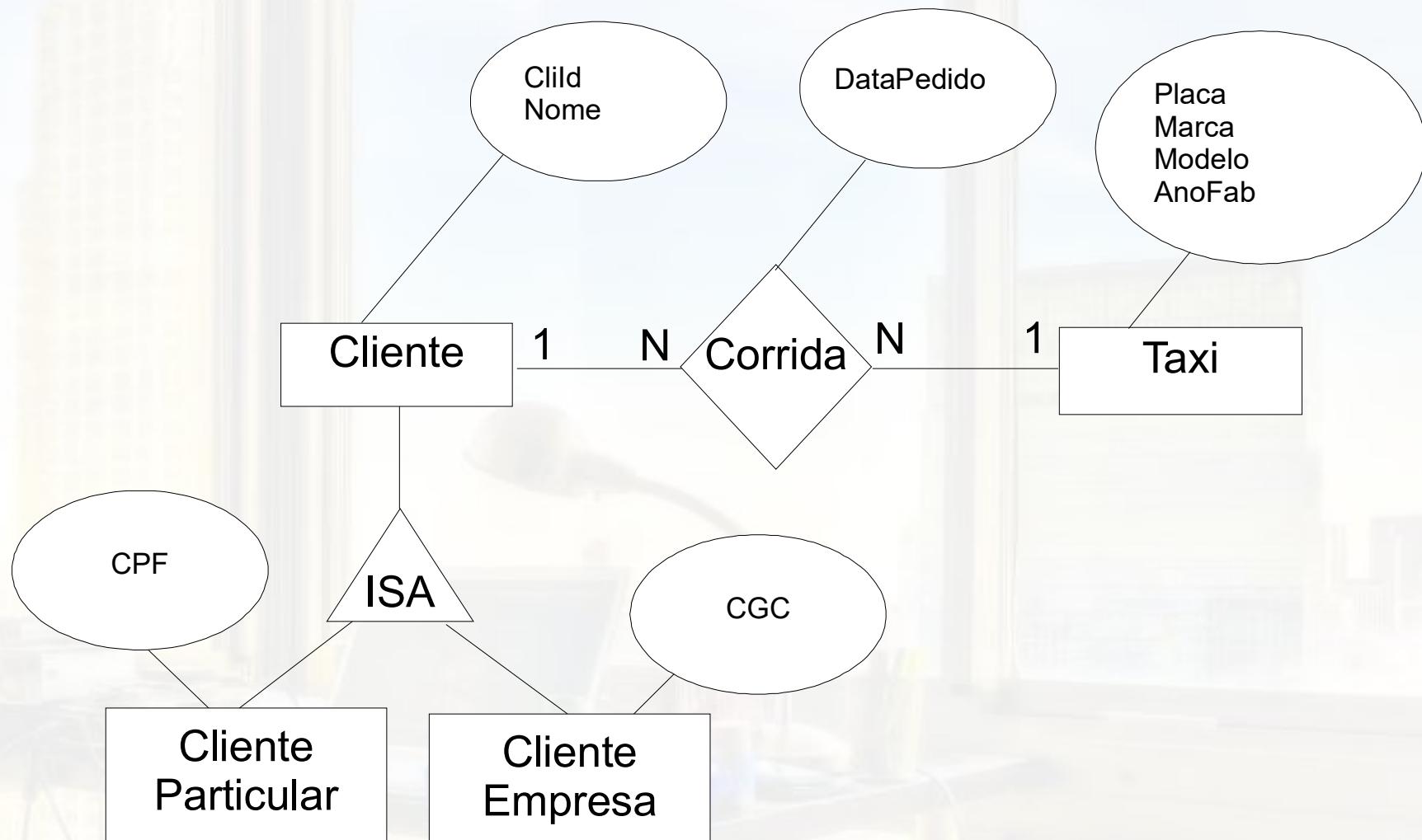
Esquema Conceitual - Exemplo

Cliente



Para ilustrar o tema apresentado, foram acrescentadas duas entidades que são especialização de Cliente. A primeira representa um indivíduo que irá pagar a conta, a segunda representa um funcionário de uma empresa conveniada, para a qual a conta será enviada. Um cliente pode pertencer a ambas especializações.

Esquema Conceitual completo Táxis



Tabelas para exemplo - Táxis

Cliente Particular (CP)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Cliente Empresa (CE)

<u>CliId</u>	Nome	CGC
1532	Asdrúbal	754.856.965/0001-54
1644	Jepeto	478.652.635/0001-75
1780	Quincas	554.663.996/0001-87
1982	Zandor	736.952.369/0001-23

Tabelas para exemplo - Táxis

Táxi (TX)

<u>Placa</u>	<u>Marca</u>	<u>Modelo</u>	<u>AnoFab</u>
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003



CREATE SCHEMA

- CREATE SCHEMA <esquema>
AUTHORIZATION <id_autorizado>
- Java: executeUpdate(...)

CREATE TABLE

- CREATE TABLE <tabela>
(<campo₁> <tipo> [NULL|NOT NULL] [restrição],
[...,<campo_n> <tipo> [NULL|NOT NULL] [restrição],
PRIMARY KEY <chave_primaria>])
- Java: executeUpdate(...)

CREATE TABLE

```
CREATE TABLE Taxi (
    Placa VARCHAR(7) NOT NULL,
    Marca VARCHAR(30) NOT NULL,
    Modelo VARCHAR(30) NOT NULL,
    AnoFab INTEGER,
    Licenca VARCHAR(9),
    PRIMARY KEY(Placa)
);
```

```
CREATE TABLE Cliente (
    CliId VARCHAR(4) NOT NULL,
    Nome VARCHAR(80) NOT NULL,
    CPF VARCHAR(14) NOT NULL,
    PRIMARY KEY(CliId)
);
```

CREATE TABLE FOREIGN KEY

- CREATE TABLE <tabela>

...

FOREIGN KEY (<coluna_estr>₁, ..., <coluna_estr>_n)
REFERENCES <tabela_ref>(<coluna_ref>[, ..., <coluna_ref>])
[ON DELETE <ação_ref>]
[ON UPDATE <ação_ref>]

- <ação_ref>

- NO ACTION → impede a ação na tabela mestre <tabela_ref>
- CASCADE → propaga a ação da tabela mestre
- SET NULL → valores de referências alterados para nulo
- SET DEFAULT → valores de referências alterados para default

CREATE TABLE FOREIGN KEY

```
CREATE TABLE Corrida (
    CliId VARCHAR(4) NOT NULL,
    Placa VARCHAR(7) NOT NULL,
    DataPedido DATE NOT NULL,
    PRIMARY KEY(CliId, Placa, DataPedido),
    FOREIGN KEY(CliId)
        REFERENCES Cliente(CliId)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    FOREIGN KEY(Placa)
        REFERENCES Taxi(Placa)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
) ;
```

Exercício 1

- Escreva uma comando SQL para criar os esquemas:
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

INSERT

- **INSERT INTO <tabela>**
[(<campo₁>[, ..., <campo_n>])]
VALUES (<valor₁>[, ..., <valor_n>])
- **executeUpdate(...)**

Exemplos INSERT

```
INSERT INTO Cliente  
    VALUES ('1755', 'Doriana', '567.387.387-44');
```

```
INSERT INTO Taxi  
    VALUES ('DAE6534', 'Ford', 'Fiesta', 1999, 'MN572345');
```

```
INSERT INTO Corrida  
    VALUES ('1755', 'DAE6534', '2003-02-15');
```

Comandos INSERT para Taxi

```
INSERT INTO Cliente VALUES ('1532', 'Asdrúbal', '448.754.253-65');  
INSERT INTO Cliente VALUES ('1755', 'Dorian', '567.387.387-44');  
INSERT INTO Cliente VALUES ('1780', 'Quincas', '546.373.762-02');  
  
INSERT INTO Taxi VALUES ('DAE6534', 'Ford', 'Fiesta', 1999, 'MN572345');  
INSERT INTO Taxi VALUES ('DKL4598', 'Wolkswagen', 'Gol', 2001, 'AU876543');  
INSERT INTO Taxi VALUES ('DKL7878', 'Ford', 'Fiesta', 2001, 'OP102938');  
INSERT INTO Taxi VALUES ('JDM8776', 'Wolkswagen', 'Santana', 2002, 'QM365923');  
INSERT INTO Taxi VALUES ('JJM3692', 'Chevrolet', 'Corsa', 1999, 'UU335577');  
  
INSERT INTO Corrida VALUES ('1755', 'DAE6534', '2003-02-15');  
INSERT INTO Corrida VALUES ('1780', 'JDM8776', '2003-02-18');  
INSERT INTO Corrida VALUES ('1755', 'DKL7878', '2003-02-16');  
INSERT INTO Corrida VALUES ('1780', 'DKL4598', '2003-02-17');  
INSERT INTO Corrida VALUES ('1532', 'DKL4598', '2003-02-18');  
INSERT INTO Corrida VALUES ('1780', 'DAE6534', '2003-02-16');
```

Exercício 2

- Escreva um comando SQL para inserir uma tupla na tabela Pessoa com os seus dados e dados de familiares próximos (cerca de 2 linhas). Preencha a tabela Cidade com as cidades listadas na tabela Pessoa e suas respectivas siglas de estado. Use dados fictícios se preciso.

SELECT

- **SELECT * | <campo₁>[,..., <campo_n>]
FROM <tabela₁>[,..., <tabela_n>]
WHERE <condição/junção>**
- **executeQuery(...)**

SELECT Projeção

SELECT Marca, Modelo FROM Taxi

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT Projeção

SELECT Marca, Modelo FROM Taxi



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT Projeção

SELECT Marca, Modelo FROM Taxi

Marca	Modelo
Ford	Fiesta
Wolkswagen	Gol
Ford	Fiesta
Wolkswagen	Santana
Chevrolet	Corsa

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002

Exercício 3

- Para a tabelas que você montou no exercício 1, escreva um comando SQL que retorne:
 - a) nomes de todas as mães
 - b) nomes de todas as mães com filhos maiores de 12 anos

SELECT Between

**SELECT * FROM Taxi WHERE AnoFab BETWEEN
1999 AND 2001;**

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT IN

```
SELECT * FROM Taxi WHERE Modelo  
    IN ('Corsa', 'Santana');
```

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT LIKE

```
SELECT Placa, Marca FROM Taxi  
WHERE Marca LIKE 'Wolks%';
```

	placa character varying(7)	marca character varying(30)
1	DKL4598	Wolkswagen
2	JDM8776	Wolkswagen

- % → qualquer cadeia com 0 a n caracteres
- _ → exatamente um caractere (qualquer)
- = → caractere de escape
 - e.g., serve para encontrar um caractere _

AS (alias)

- SELECT <campo₁> [AS] <alias₁>
[,..., <campo_n> [AS] <alias_n>]
...
- SELECT ...
FROM <tabela₁> [AS] <alias₁>
[,..., <tabela_n> [AS] <alias_n>]
...

SELECT Alias

```
SELECT Cl.Clid, Cl.Nome  
FROM Cliente AS Cl;
```

	cliid character varying(4)	nome character varying(80)
1	1532	Asdrúbal
2	1755	Doriana
3	1780	Quincas

SELECT Funções

```
SELECT Placa, Modelo || ' - ' || upper(Marca) AS  
      MarcaModelo FROM Taxi;
```

	placa character	marcamodelo text
1	DAE6534	Fiesta - FORD
2	DKL4598	Gol - WOLKSWAGEN
3	DKL7878	Fiesta - FORD
4	JDM8776	Santana - WOLKSWAGEN
5	JJM3692	Corsa - CHEVROLET

SELECT DISTINCT e ALL

- **SELECT DISTINCT ...**

- Seleciona apenas tuplas distintas (definição do modelo relacional)

- **SELECT ALL ...**

- A cláusula ALL é implícita se não especificada (o padrão do SQL é não seguir a unicidade do modelo relacional)

SELECT ORDER BY

- SELECT ...

ORDER BY <campo₁>[, ..., <campo_n>]

[DESC]

- Ordena de acordo com a lista de campos
- Use DESC para ordem decrescente

SELECT
ORDER BY

SELECT nome FROM Cliente
ORDER BY nome DESC;

	nome character varying(80)
1	Quincas
2	Dorianा
3	Asdrúbal

Produto Cartesiano

- SELECT ...
 FROM <tabela₁>, <tabela₂>
 <não há condição que ligue tabelas>
- Não há associação de atributo da <tabela₁>
 com atributo da <tabela₂>

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
Corrida.CliId, Corrida.Placa,  
Corrida.DataPedido  
FROM Cliente, Corrida
```

<u>CliId</u>	<u>Nome</u>
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida
```

<u>CliId</u>	<u>Nome</u>
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

<u>(CliId)</u>	<u>Nome</u>	<u>(CliId)</u>	<u>Placa</u>	<u>DataPedido</u>
1532	Asdrúbal	1755	DAE6534	15/02/2003
1532	Asdrúbal	1982	JDM8776	18/02/2003

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida
```

<u>CliId</u>	<u>Nome</u>
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

<u>(CliId)</u>	<u>Nome</u>	<u>(CliId)</u>	<u>Placa</u>	<u>DataPedido</u>
1532	Asdrúbal	1755	DAE6534	15/02/2003
1532	Asdrúbal	1982	JDM8776	18/02/2003
1755	Doriana	1755	DAE6534	15/02/2003
1755	Doriana	1982	JDM8776	18/02/2003

Produto Cartesiano

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida
```

<u>CliId</u>	<u>Nome</u>
1532	Asdrúbal
1755	Doriana
1780	Quincas

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

(CliId)	Nome	(CliId)	Placa	DataPedido
1532	Asdrúbal	1755	DAE6534	15/02/2003
1532	Asdrúbal	1982	JDM8776	18/02/2003
1755	Doriana	1755	DAE6534	15/02/2003
1755	Doriana	1982	JDM8776	18/02/2003
1780	Quincas	1755	DAE6534	15/02/2003
1780	Quincas	1982	JDM8776	18/02/2003

Junção - Join (1)

- SELECT ...

FROM <tabela₁>, <tabela₂>

WHERE <tabela₁>.<atr> = <tabela₂>.<atr>

- Join implícito

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida  
 WHERE Cliente.CliId = Corrida.CliId
```

(CliId)	Nome	(CliId)	Placa	DataPedido
1532	Asdrúbal	1755	DAE6534	15/02/2003
1532	Asdrúbal	1982	JDM8776	18/02/2003
1755	Doriana	1755	DAE6534	15/02/2003
1755	Doriana	1982	JDM8776	18/02/2003
1780	Quincas	1755	DAE6534	15/02/2003
1780	Quincas	1982	JDM8776	18/02/2003

Obter, para cada corrida, informações de id e nome do cliente bem como placa e data da corrida

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida  
 WHERE Cliente.CliId = Corrida.CliId
```

(CliId)	Nome	(CliId)	Placa	DataPedido
1532	Asdrúbal	1755	DAE6534	15/02/2003
1532	Asdrúbal	1982	JDM8776	18/02/2003
1755	Doriana	1755	DAE6534	15/02/2003
1755	Doriana	1982	JDM8776	18/02/2003
1780	Quincas	1755	DAE6534	15/02/2003
1780	Quincas	1982	JDM8776	18/02/2003

Join (1)

```
SELECT Cliente.CliId, Cliente.Nome,  
       Corrida.CliId, Corrida.Placa,  
       Corrida.DataPedido  
  FROM Cliente, Corrida  
 WHERE Cliente.CliId = Corrida.CliId
```

(CliId)	Nome	(CliId)	Placa	DataPedido
1755	Doriana	1755	DAE6534	15/02/2003

Exercício 4

- Para a tabelas que você montou no exercício 1, desenhe uma tabela com mais dados (fictícios se preferir) e escreva um comando SQL que retorne:
 - nomes de parentes que nasceram no mesmo estado que você
 - retorne todas as duplas de irmãos (não se preocupe com duplicidade de irmãos)

Desafio

- Qual o modelo de Taxi para cada Corrida?

Cliente (C)

<u>ClId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelo de Taxi para cada Corrida

```
SELECT Co.DataPedido, Co.Placa, T.Modelo  
FROM Corrida Co, Taxi T  
WHERE Co.Placa = T.Placa;
```

Cliente (C)

<u>ClId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Desafio

- Quais os modelos de Taxi tomados por cada Cliente?

Cliente (C)

<u>ClId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Modelos de Taxi por Cliente

```
SELECT Cl.Nome, Co.DataPedido, Co.Placa, T.Modelo  
FROM Cliente Cl, Corrida Co, Taxi T  
WHERE Cl.CliId = Co.CliId AND Co.Placa = T.Placa;
```

Cliente (C)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Agrupamento

Funções de Agregação

- COUNT(*) \Rightarrow contagem
- SUM(<coluna>) \Rightarrow soma
- AVG(<coluna>) \Rightarrow média
- MAX(<coluna>) \Rightarrow maior valor
- MIN(<coluna>) \Rightarrow menor valor

AGREGAÇÃO

SELECT AVG(anofab) FROM Taxi;

Táxi:

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

Resultado:

avg
2000.4

Média de ano de
fabricação

GROUP BY

- **SELECT * | <campo₁>[,..., <campo_n>]
FROM <tabela₁>[,..., <tabela_n>]
WHERE <condição/junção>
GROUP BY <coluna_agrupar>
HAVING <condição_grupo>**

GROUP BY

SELECT modelo, count(*) FROM Taxi GROUP BY modelo;

Táxi:

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

Resultado:

modelo	count
Gol	1
Corsa	1
Santana	1
Fiesta	2

Contagem por modelo

GROUP BY

```
SELECT modelo, anofab, count(*) FROM Taxi  
GROUP BY modelo, anofab;
```

Resultado:

modelo	anofab	count
Escort	2001	1
Escort	2000	1
Santana	2002	1
Fiesta	1999	1
Corsa	1999	1
Gol	2000	1
Santana	1998	2
Fiesta	2001	1
Gol	2001	1

(usando dados completos)

Contagem por ano de fabricação do modelo

Exercício 6

- Escreva uma sentença SQL, baseada no esquema abaixo, que retorne o número de pessoas da família em cada estado:
 - Pessoa(nome, nome_da_mãe, ano_nascimento, nome_cidade_natal)
 - nome_cidade_natal → CHE Cidade
 - Cidade(nome_cidade, sigla_estado)

Consultas Aninhadas

SELECT Seleção

SELECT * FROM Taxi WHERE AnoFab > 2000

<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolkswagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolkswagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

SELECT IN e NOT IN

- SELECT ...
WHERE <campo> IN
(SELECT <campo> ...)
- SELECT ...
WHERE <campo> NOT IN
(SELECT <campo> ...)

**SELECT
FROM**

- **SELECT ...
FROM (SELECT <campo> ...)**

SELECT EXISTS e NOT EXISTS

- SELECT ...
WHERE EXISTS
(SELECT <campo> ...)
- SELECT ...
WHERE NOT EXISTS
(SELECT <campo> ...)

SELECT Comparação

- SELECT ...
 WHERE <campo> <comparação>
 (SELECT <campo> ...)

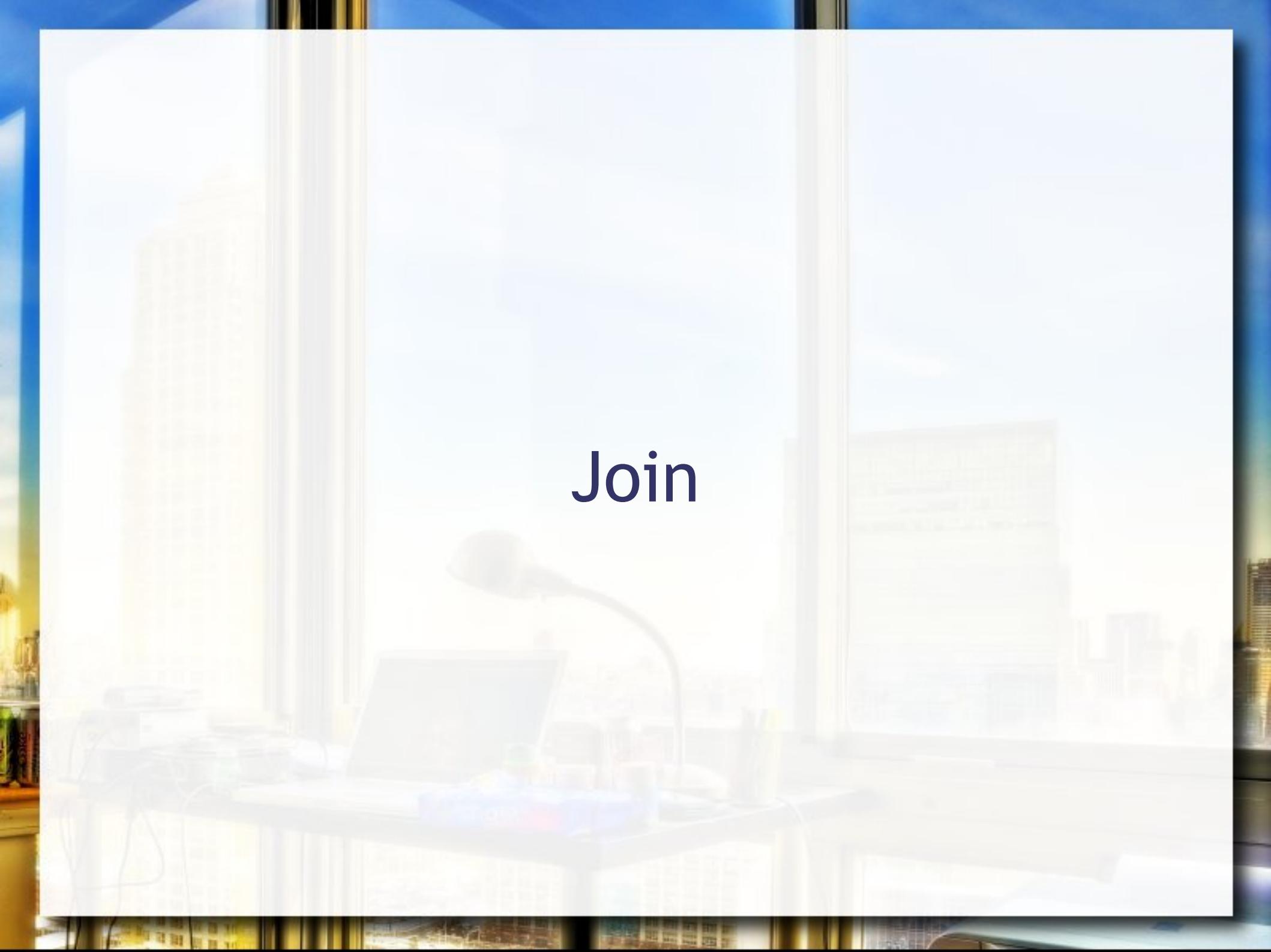
Exemplo:

SELECT Placa FROM Corrida

 WHERE Corrida.DataPedido = (SELECT
 MIN(DataPedido) FROM Corrida);

Exercício 7

- Para a tabelas que você montou no exercício 1, escreva um comando SQL que retorne todos os primos por parte de mãe, que você for capaz de inferir a partir da tabela. Considere que você tem como ponto de partida o nome da sua avó.
- Utilize duas estratégias:
 - VIEW
 - SELECT aninhado

A faint, semi-transparent background image shows a person sitting at a desk in a library, surrounded by bookshelves filled with books. The person appears to be reading or looking at something on the desk.

Join

Join

- SELECT ...
 FROM <tabela> JOIN <tabela>
 ON <condição> ...
- Tipo clássico de join explicitado
- Também conhecido como INNER JOIN

Natural Join

- SELECT ...
FROM <tabela> NATURAL JOIN <tabela>
- Condição não especificada
- EQUIJOIN: Verifica igualdade de cada par de atributos com o mesmo nome

Outer Join

- SELECT ...

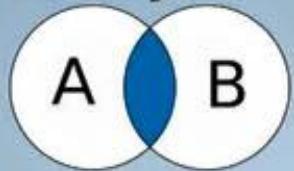
```
FROM <tabela> <join> <tabela>  
ON <condição> ...
```

- <join>

- LEFT JOIN - toda tupla à esquerda aparece
- RIGT JOIN - toda tupla à direita aparece
- FULL JOIN - toda tupla aparece

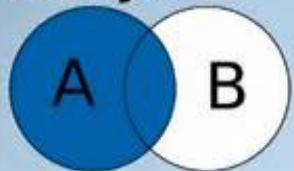
SQL JOINS

INNER JOIN



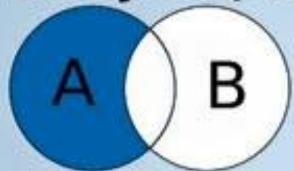
```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```

LEFT JOIN



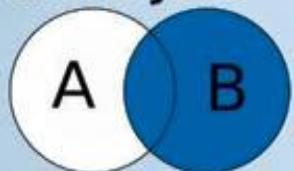
```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```

LEFT JOIN (sans l'intersection de B)



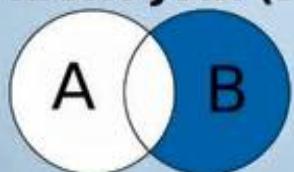
```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

RIGHT JOIN



```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```

RIGHT JOIN (sans l'intersection de A)



```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

FULL JOIN



```
SELECT *
```

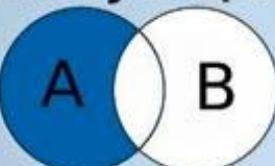


A

B

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```

LEFT JOIN (sans l'intersection de B)

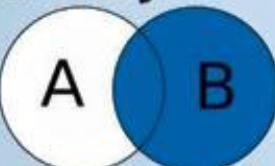


A

B

```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

RIGHT JOIN

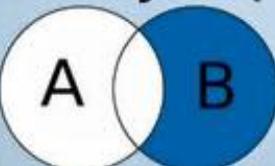


A

B

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```

RIGHT JOIN (sans l'intersection de A)

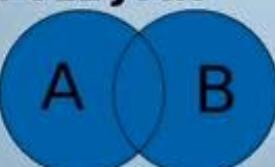


A

B

```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

FULL JOIN

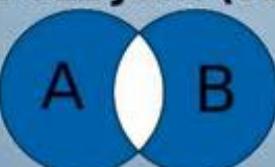


A

B

```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key
```

FULL JOIN (sans intersection)



A

B

```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```

União, Interseção e Diferença

- SELECT ...
<operador>
SELECT ...
- <operador>
 - UNION
 - INTERSECT
 - EXCEPT

Modificando Dados

DELETE (CUIDADO!!!)

- `DELETE FROM <tabela1>
WHERE <condição>`
- `executeUpdate(...))`

UPDATE

- UPDATE <tabela>

- SET <campo₁>=<valor₁>

- [,..., <campo_n>=<valor_n>]

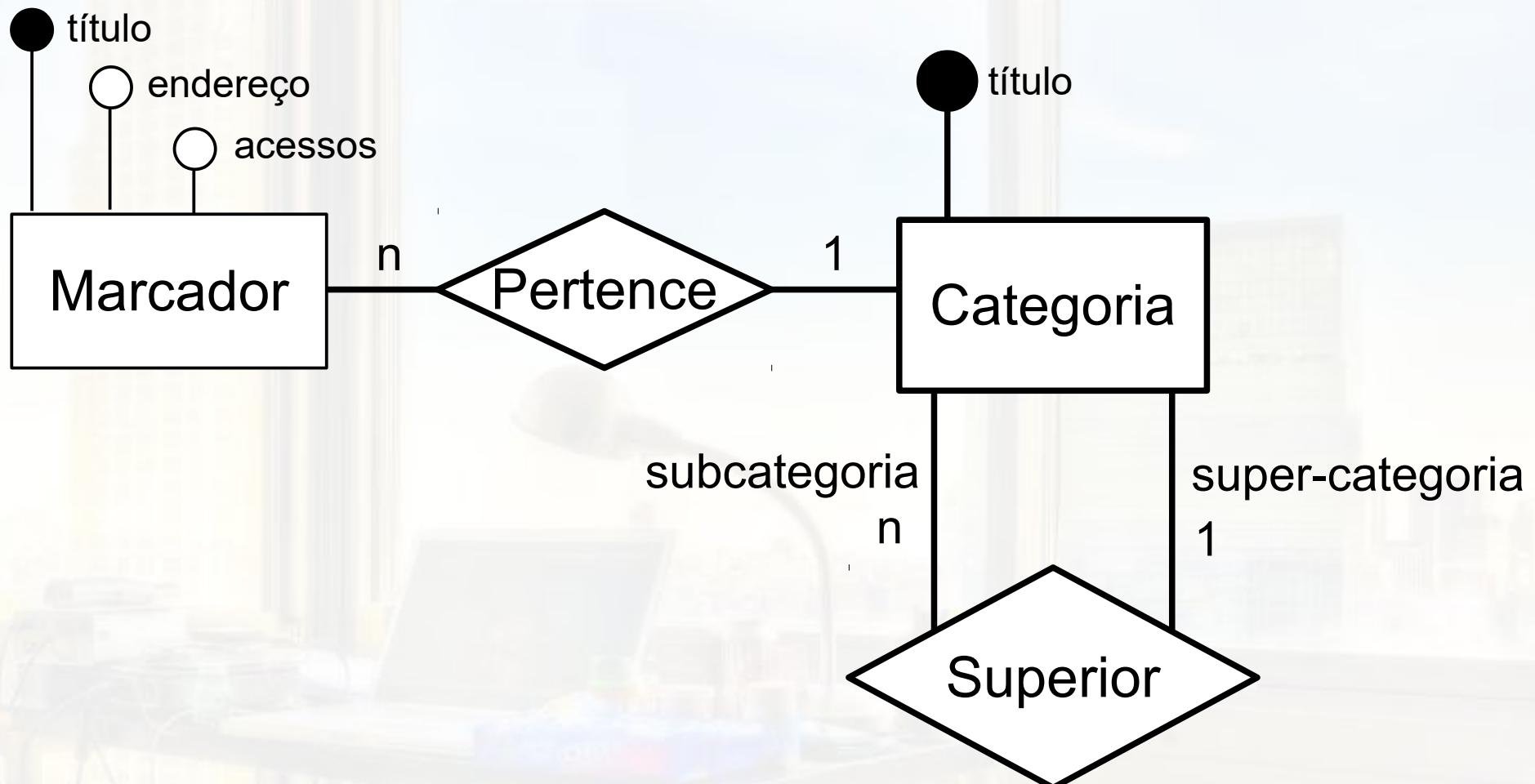
- WHERE <condição>

- executeUpdate(...)

Aplicando o UPDATE

Marcadores e Categorias

Modelo ER



Marcadores e Categorias

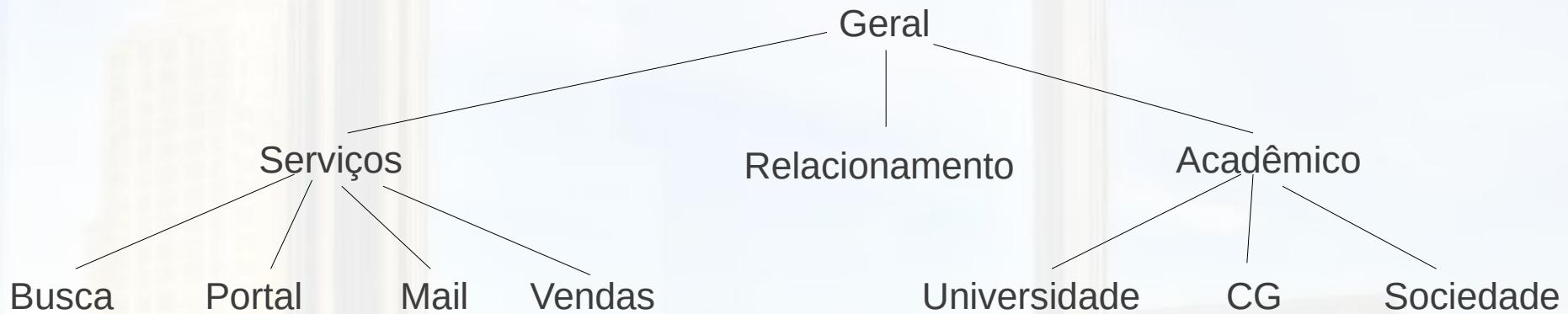
Modelo Relacional

Marcador (Titulo, Endereco, Acessos, Categoria)

Título	Endereco	Acessos	Categoria
Terra	http://www.terra.com.br	295	Portal
POVRay	http://www.povray.org	2	CG
SBC	http://www.sbc.org.br	26	Sociedade
Correios	http://www.correios.com.br	45	Serviços
GMail	http://www.gmail.com	296	Mail
Google	http://www.google.com	1590	Busca
Yahoo	http://www.yahoo.com	134	Serviços
Orkut	http://www.orkut.com	45	Serviços
iBahia	http://www.ibahia.com	3	Portal
Submarino	http://www.submarino.com.br	320	Serviços

Tabela Taxonomia

Modelo Relacional



Categoria	Superior
Geral	
Serviços	Geral
Acadêmico	Geral
Relacionamento	Geral
Busca	Serviços
Portal	Serviços
Mail	Serviços
Vendas	Serviços
Universidade	Acadêmico
CG	Acadêmico
Sociedade	Acadêmico

Marcadores e Categorias

Modelo Relacional

Marcador (Titulo, Acessos, Endereco, Categoria)

- Categoria: chave estrangeira para Taxonomia

Taxonomia (Categoria, Superior)

Estudo de Caso

SQL

- UPDATE Marcadores
 - SET Categoria = <nova>
 - WHERE Categoria = <antiga>
- UPDATE Taxonomia
 - SET Categoria = <nova>
 - WHERE Categoria = <antiga>
- UPDATE Taxonomia
 - SET Superior = <nova>
 - WHERE Superior = <antiga>

SELECT aninhado também pode ser
usado em operações de **UPDATE** e
DELETE

Prepared Statement

Utilizando o PreparedStatement

- SELECT FROM Marcadores
WHERE Titulo = ?
- <comando>.setString(<numero>, <valor>)

Utilizando o PreparedStatement

- INSERT INTO Marcadores
VALUES (? , ? , ? , ?)
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

Utilizando o PreparedStatement

- UPDATE Marcadores
 - SET Categoria = ?
 - WHERE Categoria = ?
- <comando>.setString(<numero>, <valor>)
- <comando>.setInt(<numero>, <valor>)

Visões

VIEW

- CREATE VIEW <nome> AS
SELECT ...

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.

André Santanchè

<http://www.ic.unicamp.br/~santanche>

Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença, com restrições adicionais:
 - Se você é estudante, você não está autorizado a utilizar estes slides (total ou parcialmente) em uma apresentação na qual você esteja sendo avaliado, a não ser que o professor que está lhe avaliando:
 - lhe peça explicitamente para utilizar estes slides;
 - ou seja informado explicitamente da origem destes slides e concorde com o seu uso.
- Mais detalhes sobre a referida licença Creative Commons veja no link:
<http://creativecommons.org/licenses/by-nc-sa/2.5/br/>