

UFSC-CTC-INE

Curso de Sistemas de Informação

INE 5600 – Bancos de Dados III

# Gerência de Dados XML em Bancos de Dados

# Gerência de Dados XML em BDs

- SGBDs relacionais estendidos
  - tipo de dado XML
  - linguagem *SQL/XML* (extensão da SQL)
  - falta de padronização na atualização de dados
- SGBDs XML nativos
  - modelo de armazenamento proprietário
  - indexação de dados XML
  - suporte à tecnologia XML da W3C
    - *DTD, XSD, XPath, XQuery, DOM*
  - atualização de dados
    - linguagem *XUpdate* ou *XQuery* estendida
  - limitações na gerência de dados
    - controle de RIs, visões, transações, ...

# SGBDs Relacionais Estendidos

- Tipo de dado XML

- associado a uma coluna de uma tabela
- mantém documento XML ou conteúdo XML bem-formado
- armazenamento do tipo CLOB
- pode sofrer operações SQL/XML

- Exemplo

```
CREATE TABLE Livros (  
  ID int,  
  conteudo xml,  
  PRIMARY KEY (ID)  
);
```

# Inserção de Dados do Tipo XML

```
INSERT INTO Livros VALUES (1,  
'<livro ISBN="1"><nome>Java</nome></livro>');
```

```
INSERT INTO Livros VALUES (2,  
XMLPARSE(CONTENT  
'<livro ISBN="2"><nome>XML</nome></livro>'));
```

```
INSERT INTO Livros VALUES (3,  
XMLPARSE(DOCUMENT '<?xml version="1.0"?>  
<livro ISBN="3"><nome>SQL</nome></livro>'));
```

# Linguagem SQL/XML

- Funções para manipulação de dados XML
  - navegação e filtro sobre dados do tipo XML
    - suporte à expressões *XPath* e *XQuery*
  - geração de conteúdo XML a partir de dados relacionais ou dados do tipo XML
- Principais funções
  - `xmlelement` (gera um elemento XML)
  - `xmlattribute` (gera um atributo XML)
  - `xmlforest` (gera uma sequência de elementos XML)
  - `xmlagg` (concatena conteúdo de uma coluna do tipo XML)
  - `xmlconcat` (concatena 2 conteúdos XML)
  - `xmlexists` (predicado para expressões XPath)
  - `xpath` (executa consultas XPath) (SGBD Postgres)

# SQL/XML - Exemplos

Pessoas

<u>ID</u>	CPF	nome	sexo
1	111111111111	Joao da Silva	M
2	222222222222	Ana Souza	F
3	333333333333	Pedro Santos	M
4	444444444444	Carlos Pereira	M
5	555555555555	Maria Oliveira	F

```
SELECT xmlelement(name livro, xmlattributes('1' as ISBN));
```

Resultado: "<livro isbn="1"/>"

```
SELECT xmlelement(name pessoa, xmlattributes(CPF as cpf),  
xmlelement(name nome, nome), xmlelement(name sexo, sexo))  
FROM Pessoas  
WHERE ID = 1;
```

Resultado: "<pessoa cpf="11111111111">  
<nome>Joao da Silva</nome> <sexo>M </sexo></pessoa>"

# SQL/XML - Exemplos

```
SELECT xmlconcat('<resultado/>', xmlforest(CPF AS CPF, nome  
AS nome, sexo AS sexo))  
FROM Pessoas  
WHERE ID = 2;
```

Resultado: "<resultado/><cpf>2222222222</cpf><nome>Ana Souza  
</nome><sexo>F</sexo>"

```
SELECT xmlagg(conteudo) FROM Livros;
```

Resultado: "<livro ISBN="1"><nome>Java</nome></livro>  
<livro ISBN="2"><nome>XML</nome></livro>  
<livro ISBN="3"><nome>SQL</nome></livro>"

# SQL/XML - Exemplos

```
SELECT xpath(' /livro/nome', conteudo)  
FROM Livros;
```

Resultado: "<nome>Java</nome>  
<nome>XML</nome>  
<nome>SQL</nome>"

```
SELECT ID  
FROM Livros  
WHERE xmlexists(' /livro[@ISBN="1"] ' PASSING conteudo)
```

Resultado: 1



# Atualização de Dados XML

- Falta de padronização em termos de capacidades de atualização
  - exemplos
    - Postgres: não possui operações de atualização
    - Oracle e DB2: funções proprietárias
- Exemplo de atualização no Oracle

```
UPDATE Livros
SET OBJECT_VALUE =
updateXML(OBJECT_VALUE,
            '/livro[@ISBN = '1']/@ISBN', '101');
```

tabela do tipo XML (XMLType)

constante que indica conteúdo XML

# SGBDs XML Nativos

- Armazenamento nativo de dados XML
- Principais estratégias de armazenamento

Árvore DOM	BDOO
preserva ordem de elementos	não preserva ordem de elementos
qualquer esquema tem a mesma estrutura ( <i>document</i> , <i>element</i> , ...)	esquema de classes gerado de acordo com os tipos de elementos complexos
clusterização por profundidade <ul style="list-style-type: none"><li>• bom p/ buscas na ordem da hierarquia</li></ul> clusterização por largura <ul style="list-style-type: none"><li>• bom p/ buscas por propriedades de um elemento</li></ul>	clusterização geralmente por instâncias da mesma classe <ul style="list-style-type: none"><li>• bom para buscas por dados de determinados tipos de elementos</li></ul>

# Tipos de Indexação

- Indexação **por valor**
  - indexa valores de elementos simples e atributos
    - exemplo: buscar elementos com valor “XML”
- Indexação **por estrutura**
  - indexa nomes de elementos ou de atributos
    - exemplo: buscar elementos cujo nome é “XML”
- Indexação ***fulltext***
  - indexa *tokens* presentes em nomes ou em valores de elementos ou de atributos
    - exemplo: buscar elementos que contenham o termo “XML” em seus valores

# Consulta a Dados XML

- Uso das linguagens *XPath* e/ou *XQuery*
- Alguns SGBDs XML com capacidades adicionais

- exemplo: SGBD Tamino (busca por padrão)

```
/livro[título ~= '*XML*']/título
```

- exemplo: SGBD eXists (busca por palavra-chave)

```
/livro[título &= 'banco XML']/título
```

# Atualização de Dados XML

- Principais abordagens
  - linguagem *XQuery* estendida
  - linguagem *XUpdate*

- *XQuery* estendida

- exemplo: SGBD Tamino

**update**

```
for $liv in input()/livro
where $liv/@ISBN = "112"
do (replace $liv/preco with
    (<preco>135.00</preco>));
```

- exemplo: SGBD BaseX

```
insert node (attribute { 'edicao' } {1})
into //livro[@ISBN='112']
```

# Atualização de Dados XML

- *XUpdate*

- Sintaxe XML

- inclusão/elementos de elementos, atributos e texto
- alteração do conteúdo de elementos e atributos

- Exemplos

```
<xupdate:append
```

```
select="//autor[nome='Joao da Silva']/eMail" child="last() ">
```

```
<xupdate:element name="eMail">Joao@teste.com</xupdate:element>
```

```
</xupdate:append>
```

```
<xupdate:remove select="//livro[@ISBN = "112"]"/>
```

# Projeto de BDs XML

- Não há uma metodologia consolidada
- Etapas do projeto tradicional de um BD
  - (i) especificação de requisitos; (ii) modelagem conceitual; (iii) modelagem lógica e (iv) modelagem física ou implementação
  - podem ser aplicadas a um BD XML
    - no caso de dados XML **fortemente semiestruturados**
      - revisão da modelagem física: considerar a existência de informação textual não-estruturada dentro do conteúdo de elementos

# Guia para Projeto de BDs XML

## 1. Especificação de requisitos

- levantamento das necessidades de dados

## 2. Modelagem conceitual

- uso de algum modelo clássico (exemplo: ER)

## 3. Modelagem lógica

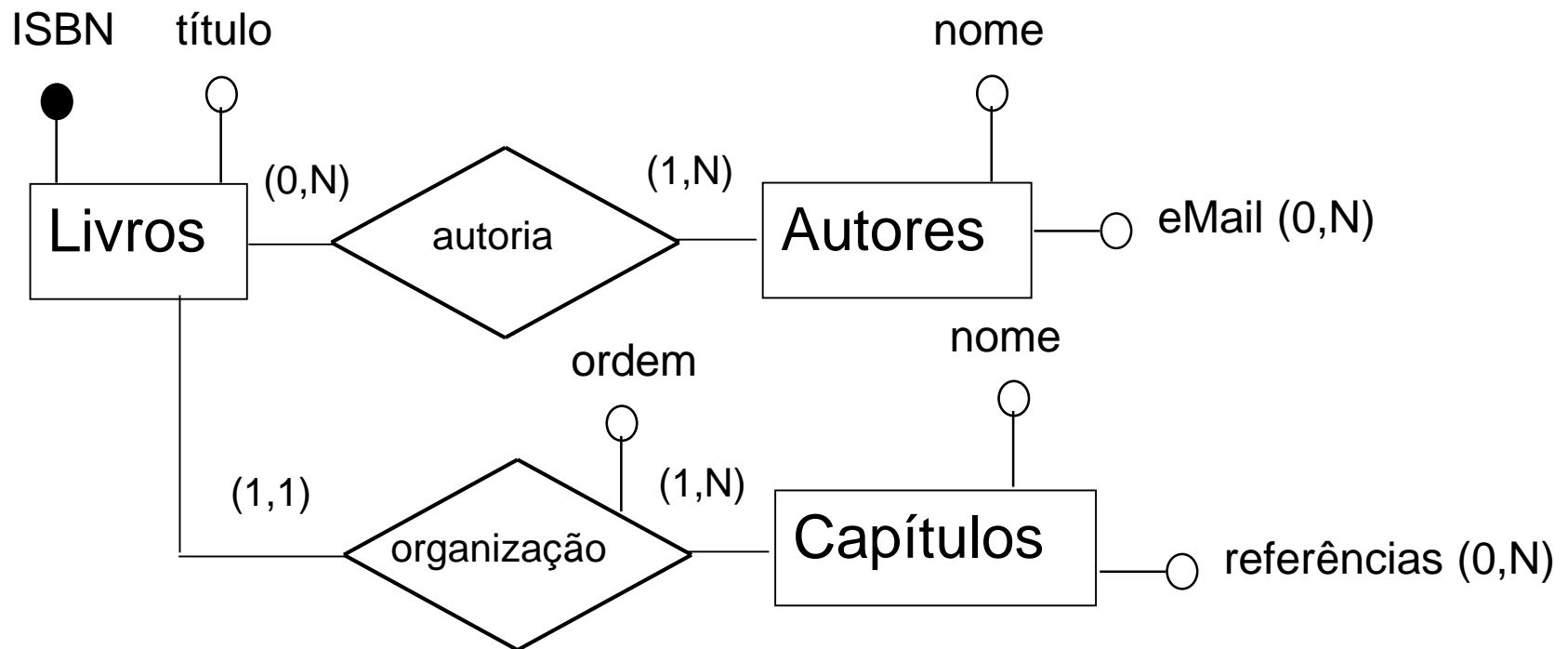
- uso de algum modelo baseado em grafo direcionado
  - adequado à representação de uma hierarquia XML

## 4. Modelagem física

- especificação do esquema XML (DTD ou XSD)



# Modelagem Conceitual - Exemplo



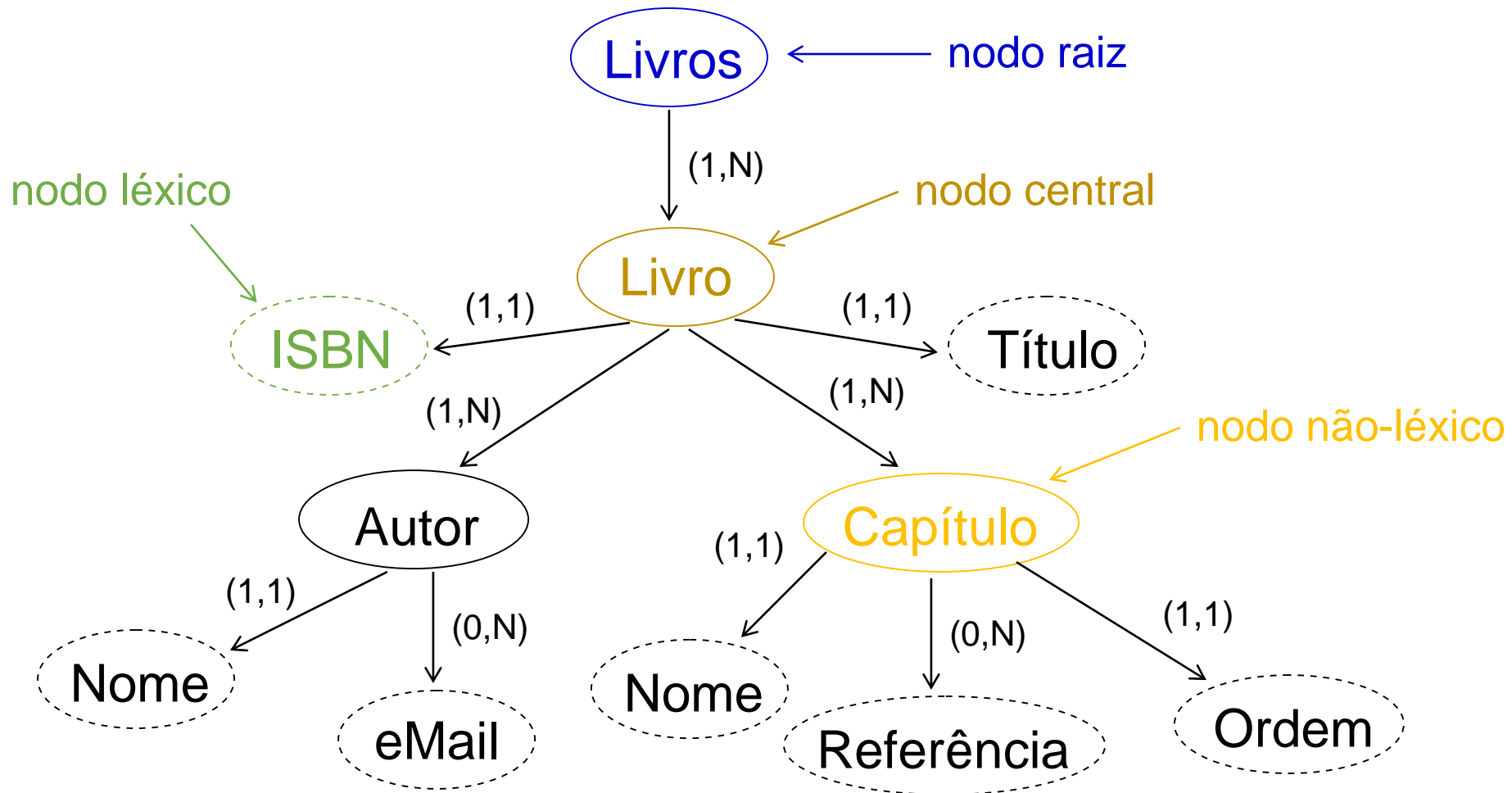
# Modelagem Lógica

- Grafo direcionado com 2 tipos de nodos
- **Nodos não-terminais (ou não-léxicos)**
  - mapeamento de entidades do ER
  - modelam elementos compostos do XML
- **Nodos terminais (ou léxicos)**
  - mapeamento de atributos do ER
  - modelam conteúdo de elemento ou de atributo no XML
- **Arestas rotuladas com restrições de cardinalidade**
  - mapeamento de relacionamentos ou ligações entidade-atributo do ER
  - modelam relacionamentos hierárquicos ou ligações elemento-atributo no XML
  - 2 ou mais arestas podem ter uma **restrição de disjunção** (útil para modelar especializações exclusivas)

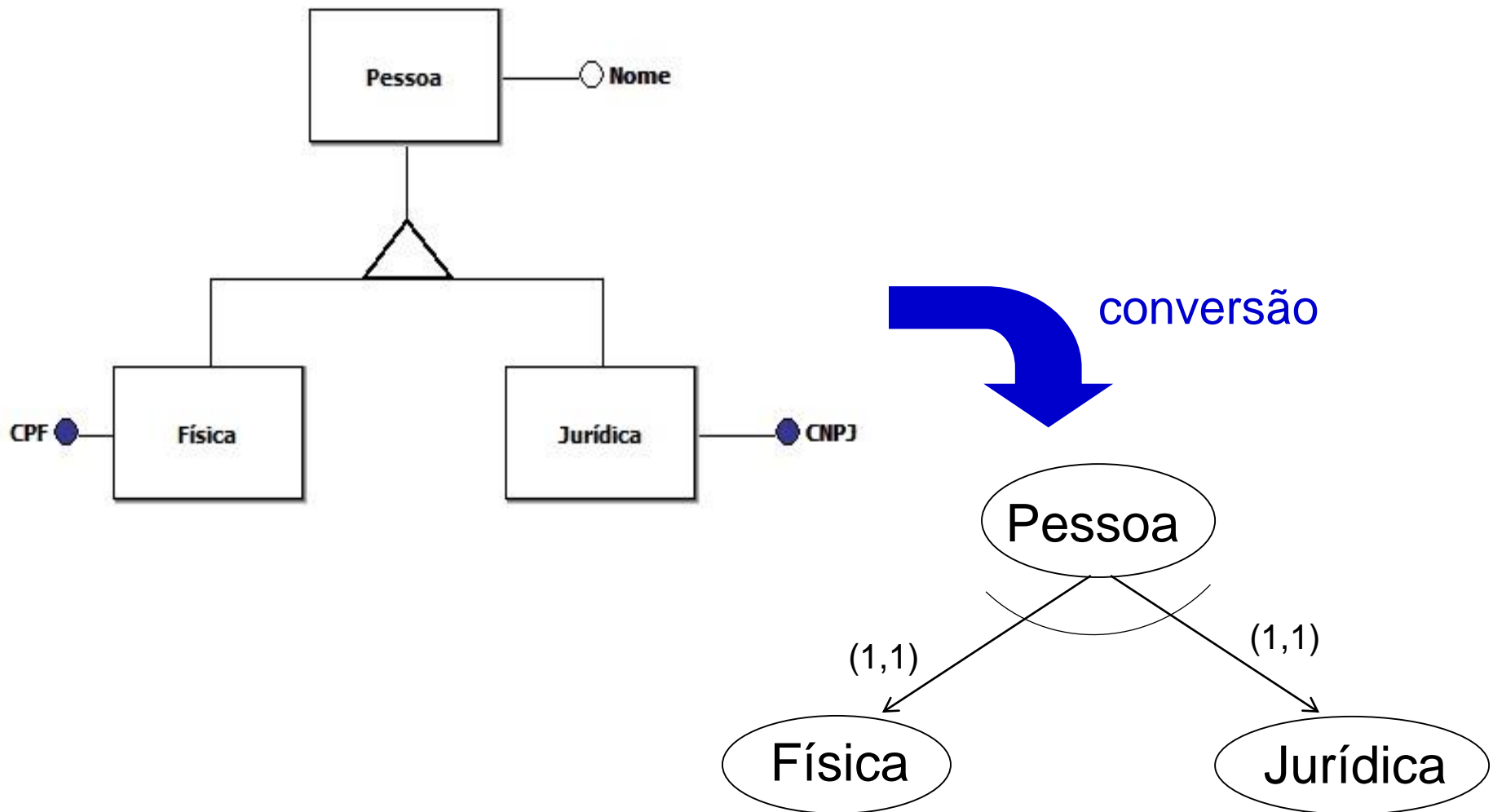
# Modelagem Lógica

- Eleição do **nodo não-léxico central**
  - **entidade central na modelagem conceitual**
    - a partir dela uma hierarquia de nodos é definida com base nos seus relacionamentos no ER (preferência por caminhos com cardinalidade 1:1 ou 1:N)
    - exemplo: **Livro**
  - **um nodo raiz deve ser definido como pai deste nodo** (com cardinalidade 1:N)
    - sugestões de nomenclatura
      - conjunto de ocorrências da entidade central (ex.: **Livros**)
      - nome do domínio (ex.: **Livraria**, **Biblioteca**, ...)
- Mais de um nodo central pode existir...
  - entidades “independentes” (ex.: livros e funcionários de uma biblioteca)
  - todos serão filhos do nodo raiz (ex.: biblioteca)

# Modelagem Lógica - Exemplo



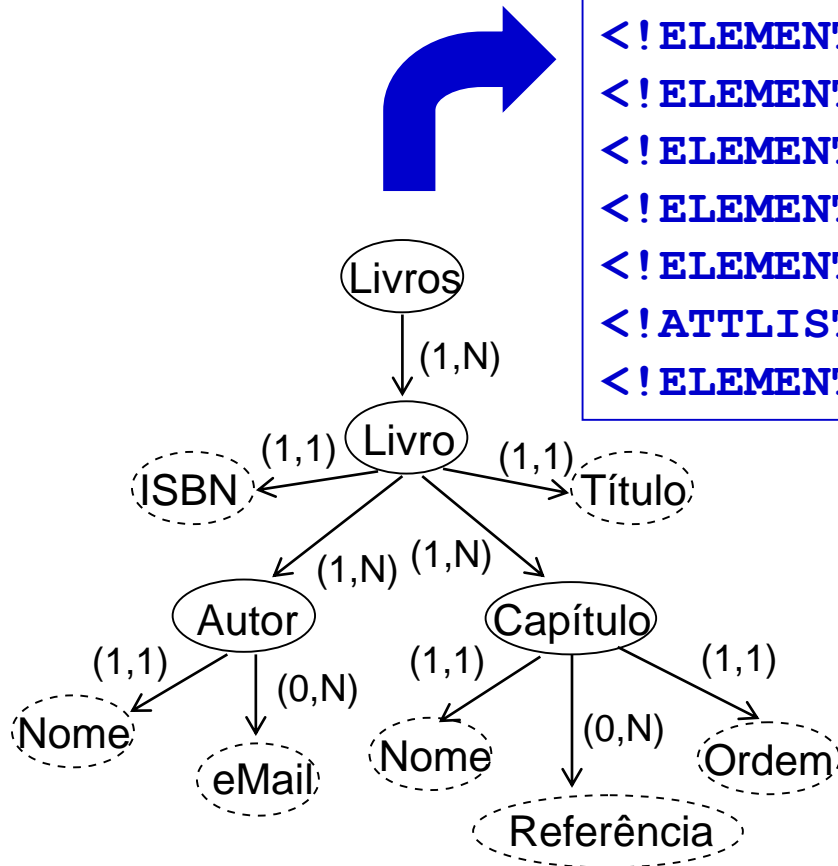
# Modelagem Lógica – Exemplo de Restrição de Disjunção



# Modelagem Física

- Definição de elementos e atributos do esquema
  - nodos não-léxicos  $\Rightarrow$  elementos compostos
  - nodos léxicos  $\Rightarrow$  elementos simples ou atributos
- Definição da ordem de subelementos
  - ordenar as arestas que partem do nodo não-léxico
- Modelagem física de um nodo léxico
  - como atributo
    - economia de espaço no documento XML
    - pode-se definir algumas RIs específicas
  - como elemento
    - conteúdos extensos (uso de *tags* para representa-lo)
    - necessário quando se deseja subelementos com cardinalidade  $> 1$
    - melhor legibilidade do documento XML

# Modelagem Física - Exemplo



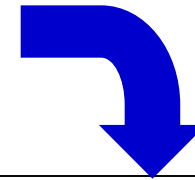
```
<!ELEMENT Livros (Livro+)>  
<!ELEMENT Livro (Título, Autor+, Capítulo+)>  
<!ATTLIST Livro ISBN CDATA>  
<!ELEMENT Título (#PCDATA)>  
<!ELEMENT Autor (Nome, eMail*)>  
<!ELEMENT Nome (#PCDATA)>  
<!ELEMENT eMail (#PCDATA)>  
<!ELEMENT Capítulo (Nome, Referência*)>  
<!ATTLIST Capítulo ordem CDATA>  
<!ELEMENT Referência (#PCDATA)>
```

DTD

# Modelagem Física - Revisão

```
<!ELEMENT Livros (Livro+)>
<!ELEMENT Livro (Título, Autor+, Capítulo+)>
<!ATTLIST Livro ISBN CDATA>
<!ELEMENT Título (#PCDATA)>
<!ELEMENT Autor (Nome, eMail*)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT eMail (#PCDATA)>
<!ELEMENT Capítulo (Nome, Referência*)>
<!ATTLIST Capítulo ordem CDATA>
<!ELEMENT Referência (#PCDATA)>
```

capítulos possuem  
conteúdo  
semiestruturado



```
<!ELEMENT Livros (Livro+)>
<!ELEMENT Livro (Título, Autor+, Capítulo+)>
<!ATTLIST Livro ISBN CDATA>
<!ELEMENT Título (#PCDATA)>
<!ELEMENT Autor (Nome, eMail*)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT eMail (#PCDATA)>
<!ELEMENT Capítulo (Nome, Texto)>
<!ELEMENT Texto (#PCDATA | Referência)*>
<!ATTLIST Capítulo ordem CDATA>
<!ELEMENT Referência (#PCDATA)>
```



# Atividade 4

Apresente uma modelagem lógica para um BD XML e uma modelagem física em DTD a partir desta modelagem ER abaixo no domínio de um Museu

