



UNIPAC - CENTRO UNIVERSITÁRIO PRESIDENTE ANTÔNIO CARLOS
CAMPUS BARBACENA

Bacharelado em Ciência da Computação



Banco de Dados

Material de Apoio

Parte XIII – XML – Conceitos Básicos

Prof. José Osvano da Silva, PMP, PSM I
joseosvano@unipac.br

1º sem / 2022

Sumário

- XML: Extensible Markup Language
- Dados estruturados, semiestruturados e não estruturados
- Modelo de dados hierárquico (em árvore) da XML
- Documentos XML, DTD e esquema XML
- Esquema XML
- Armazenando e extraíndo documentos XML de bancos de dados
- Linguagens XML
- XPath: especificando expressões de caminho em XML
- XQuery: especificando consultas em XML
- Outras linguagens e protocolos relacionados a XML
- Exercício

XML: Extensible Markup Language

- Padrão W3C que complementa HTML para intercâmbio de dados na Web
 - www.w3c.org/TR/REC-xml
- Origem: SGML
- Motivação:
 - HTML descreve apresentação XML descreve conteúdo
 - Aplicações atuais necessitam de componentes de software especiais (wrappers) para extração de dados da Web.

Dados estruturados, semiestruturados e não estruturados

- **Dados estruturados**
 - Representada em um formato estrito
 - Exemplo: informação armazenada em bancos de dados
- **Dados semiestruturados**
 - Tem determinada estrutura
 - Nem todas as informações coletadas irão ter estruturas idênticas
 - Não existe esquema pré-definido
 - Atributos podem ser adicionados a itens de dados novos

Dados estruturados, semiestruturados e não estruturados

- **Dados semiestruturados**
- Informação do esquema é misturada com os valores dos dados
 - **Dados autodescritivos (coleta de conjunto de artigos)**
 - Podem ser exibidos como um grafo direcionado
 - **Rótulos ou marcas** nas arestas direcionadas representam nomes de esquema:
 - Nomes de atributos
 - Tipos de objeto (ou tipos de entidade ou classes)
 - Relacionamentos
 - Nós internos: objetos ou atributos compostos
 - Nós externos: valores de dados reais

Dados estruturados, semiestruturados e não estruturados

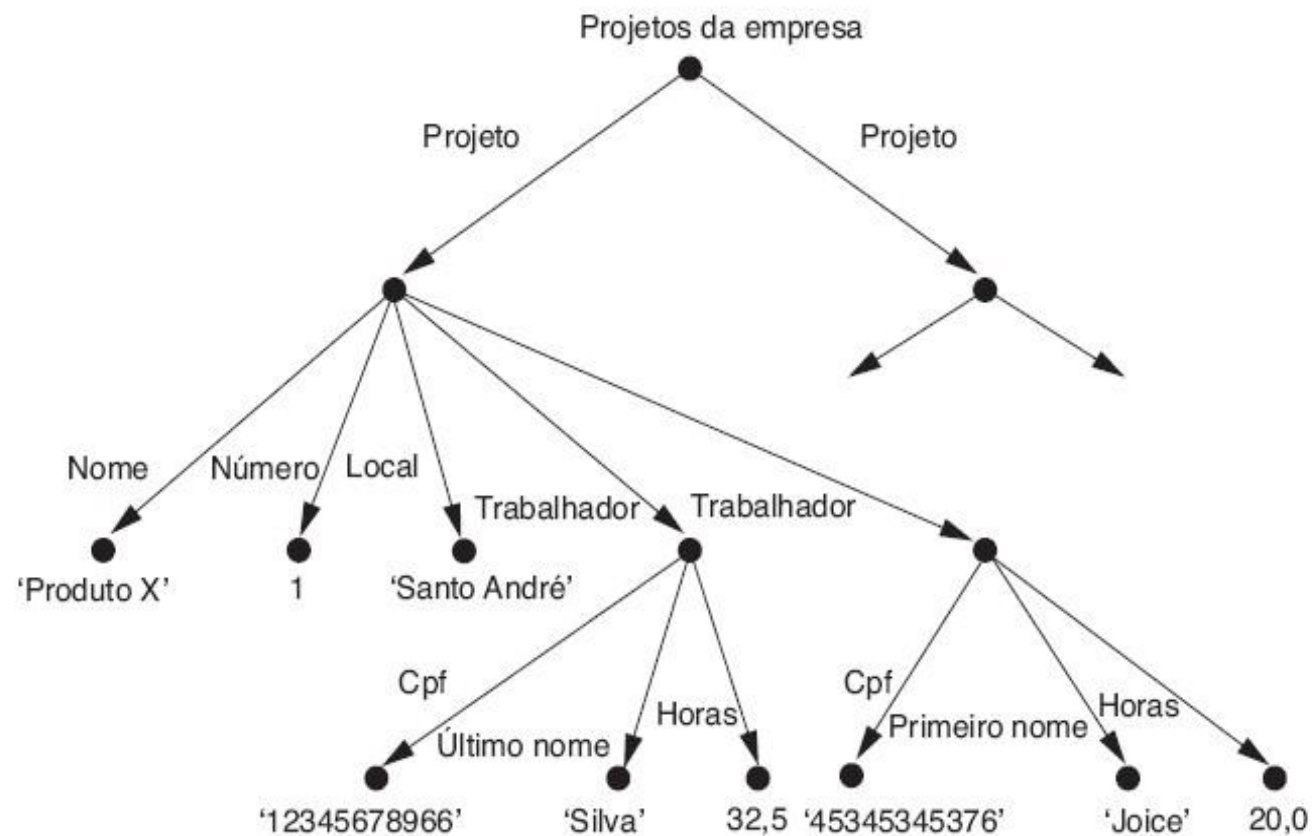


Figura 12.1

Representando dados semiestruturados como um grafo.

Dados estruturados, semiestruturados e não estruturados

- **Dados não estruturados**

- Indicação limitada de um documento de dados que contém informações incorporadas a ele

- **Tag HTML**

- Texto que aparece entre os sinais: `< ... >`

- **Tag de fim**

- Tag com uma barra: `</...>`

```
<HTML>
  <HEAD>
  ...
  </HEAD>
  <BODY>
    <H1>Lista de projetos da empresa e os funcionários em cada projeto</H1>
    <H2>O projeto ProdutoX:</H2>
    <TABLE width="100%" border=0 cellpadding=0 cellspacing=0>
      <TR>
        <TD width="50%"><FONT size="2" face="Arial">João Silva:</FONT></TD>
        <TD>32,5 horas por semana</TD>
      </TR>
      <TR>
        <TD width="50%"><FONT size="2" face="Arial">Joice Leite:</FONT></TD>
        <TD>20,0 horas por semana</TD>
      </TR>
    </TABLE>
    <H2>O projeto ProdutoY:</H2>
    <TABLE width="100%" border=0 cellpadding=0 cellspacing=0>
      <TR>
        <TD width="50%"><FONT size="2" face="Arial">João Silva:</FONT></TD>
        <TD>7,5 horas por semana</TD>
      </TR>
      <TR>
        <TD width="50%"><FONT size="2" face="Arial">Joice Leite:</FONT></TD>
        <TD>20,0 horas por semana</TD>
      </TR>
      <TR>
        <TD width="50%"><FONT size="2" face="Arial">Fernando Wong:</FONT></TD>
        <TD>10,0 horas por semana</TD>
      </TR>
    </TABLE>
    ...
  </BODY>
</HTML>
```

Figura 12.2
Parte de um documento HTML representando dados não estruturados.

Dados estruturados, semiestruturados e não estruturados

- HTML usa um grande número de tags predefinidas
- Documentos HTML
 - Não incluem informações de esquema sobre o tipo de dado
- Página HTML **estática**
 - Toda a informação a ser exibida está escrita explicitamente como um texto fixo no arquivo HTML

Modelo de dados hierárquico (em árvore) da XML

- **Elementos e atributos**
 - Conceitos de estruturação principais são usados para construir um documento XML
- **Elementos complexos**
 - Construídos com base em outros elementos hierarquicamente
- **Elementos simples**
 - Contêm valores de dados
- **Nomes de tag XML**
 - Definidos para descrever o significado dos elementos

Modelo

```
<?xml version="1.0" standalone="yes"?>
<Projetos>
  <Projeto>
    <Nome>ProdutoX</Nome>
    <Numero>1</Numero>
    <Localizacao>Santo_Andre</Localizacao>
    <Dept_nr>5</Dept_nr>
    <Trabalhador>
      <Cpf>12345678986</Cpf>
      <Ultimo_nome>Silva</Ultimo_nome>
      <Horas>32,5</Horas>
    </Trabalhador>
    <Trabalhador>
      <Cpf>45345345376</Cpf>
      <Primeiro_nome>Joice</Primeiro_nome>
      <Horas>20,0</Horas>
    </Trabalhador>
  </Projeto>
  <Projeto>
    <Nome>ProdutoY</Nome>
    <Numero>2</Numero>
    <Localizacao>Itu</Localizacao>
    <Dept_nr>5</Dept_nr>
    <Trabalhador>
      <Cpf>12345678986</Cpf>
      <Horas>7,5</Horas>
    </Trabalhador>
    <Trabalhador>
      <Cpf>45345345376</Cpf>
      <Horas>20,0</Horas>
    </Trabalhador>
    <Trabalhador>
      <Cpf>33344555587</Cpf>
      <Horas>10,0</Horas>
    </Trabalhador>
  </Projeto>
  ...
</Projetos>
```

Figura 12.3
Um elemento XML complexo, chamado <Projeto>.

re) da XML

Modelo de dados hierárquico (em árvore) da XML

- **Modelo de árvore ou modelo hierárquico**
- Tipos principais de documentos XML
 - **Documentos XML centrados em dados**
 - **Documentos XML centrados no documento**
 - **Documentos XML híbridos**
- **Documentos XML sem esquema**
 - Não seguem um esquema predefinido de nomes de elemento e estrutura de árvore correspondente.

Documentos XML, DTD e esquema XML

- **Bem formado**
 - **Declaração XML**
 - Indica a versão da XML que está sendo usada, bem como quaisquer outros atributos relevantes
 - Cada elemento precisa incluir um par correspondente de tags de início e de fim
 - dentro das tags de início e de fim do elemento pai
- **DOM** (Document Object Model)
 - Manipula a representação de árvore resultante correspondente a um documento XML bem formado

Documentos XML, DTD e esquema XML

- **SAX** (Simple API for XML)
 - Processamento de documentos XML no ato
 - Notifica o programa de processamento por meio de chamadas de eventos sempre que uma tag de início ou fim for encontrada
 - Facilita o processamento de grandes documentos
 - Permite **streaming**

Documentos XML, DTD e esquema XML

- **Válido**

- Documento deverá ser bem formado
- Documento deve seguir um esquema específico
- Pares de tag de início e de fim devem seguir a estrutura especificada em um arquivo XML **DTD** (Document Type Definition) separado, ou arquivo de esquema XML

Documentos XML, DTD e esquema XML

- XML DTD
 - Tipos de dados na DTD não são muito genéricos
 - Sintaxe especial
 - Requer processadores especializados
 - Todos os elementos DTD são sempre forçados a seguir a ordenação especificada do documento
 - Elementos não ordenados não são permitidos

Esquema XML

- **Linguagem de esquema XML**
 - Padrão para especificar a estrutura de documentos XML
 - Usa as mesmas regras de sintaxe dos documentos XML normais
 - Os mesmos processadores podem ser utilizados em ambos

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">Esquema Empresa (Definição Elemento) - Criado por Babak Hojabri</xsd:documentation>
  </xsd:annotation>
  <xsd:element name="empresa">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="departamento" type="Departamento" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element name="funcionario" type="Funcionario" minOccurs="0" maxOccurs="unbounded">
          <xsd:unique name="UnicoNomeDependente">
            <xsd:selector xpath="dependenteFuncionario" />
            <xsd:field xpath="nomeDepartamento" />
          </xsd:unique>
        </xsd:element>
        <xsd:element name="projeto" type="Projeto" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
    <xsd:unique name="unicoNomeDepartamento">
      <xsd:selector xpath="departamento" />
      <xsd:field xpath="nomeDepartamento" />
    </xsd:unique>
    <xsd:unique name="unicoNomeProjeto">
      <xsd:selector xpath="projeto" />
      <xsd:field xpath="nomeProjeto" />
    </xsd:unique>
  </xsd:element>
</xsd:schema>

```

Esquema XML

- Identificar o conjunto específico de elementos da linguagem de esquema XML (tags) sendo usado
 - Especificar um arquivo armazenado em um local de Website
- **Namespace XML**
 - Define o conjunto de comandos (nomes) que podem ser usados

Esquema XML

- Conceitos do esquema XML:
 - Descrição e namespaces XML
 - Anotações, documentação e linguagem
 - Elementos e tipos
 - Elemento de primeiro nível
 - Tipos de elemento e ocorrências mínima e máxima
 - Chaves
 - Estruturas de elementos complexos
 - Atributos compostos

Armazenando e extraindo documentos XML de bancos de dados

- Abordagens mais comuns
 - Usar um SGBD para armazenar os documentos como texto
 - Pode ser usada se o SGBD tiver um módulo especial para processamento de documento
 - Usar um SGBD para armazenar conteúdos de documento como elementos de dados
 - Exige algoritmos de mapeamento para projetar um esquema de banco de dados compatível com a estrutura do documento XML

Armazenando e extraindo documentos XML de bancos de dados

- Projetar um sistema especializado para armazenar dados XML nativos
 - Chamados de **SGBDs XML nativos**
- Criar ou publicar documentos XML personalizados de bancos de dados relacionais pré-existent
 - Usa uma camada separada de software de middleware para tratar de conversões

Linguagens XML

- Dois padrões de linguagens de consulta
 - **Xpath**
 - Especifica expressões de caminho a fim de identificar certos nós (elementos) ou atributos em um documento XML que combina padrões específicos
 - **XQuery**
 - Usa expressões XPath, mas tem construções adicionais

XPath: especificando expressões de caminho em XML

- Expressão XPath
 - Retorna uma sequência de itens que satisfazem certo padrão, conforme especificado pela expressão
 - Podem ser valores (ou nós de folha), elementos ou atributos
 - **Condições qualificadoras**
 - Restringir ainda mais os nós que satisfazem o padrão
- **Separadores** usados ao se especificar um caminho:
 - Barra simples (/) e barra dupla (//)

XPath: especificando expressões de caminho em XML

1. `/empresa`
2. `/empresa/departamento`
3. `//funcionario [salarioFuncionario gt 70.000]/nomeFuncionario`
4. `/empresa/funcionario [salarioFuncionario gt 70.000]/nomeFuncionario`
5. `/empresa/projeto/trabalhadorProjeto [horas ge 20,0]`

Figura 12.6

Alguns exemplos de expressões XPath em documentos XML que seguem o arquivo de esquema XML *empresa* na Figura 12.5.

XPath: especificando expressões de caminho em XML

- Nome do atributo é iniciado pelo símbolo @
- Símbolo **curinga** *
 - Representa qualquer elemento
 - Exemplo: /empresa/*
- **Eixos**
 - Mover em várias direções a partir do nó atual na expressão de caminho
 - Incluem nó atual, filho, descendente, atributo, pai, ancestral, irmão anterior e irmão seguinte

XPath: especificando expressões de caminho em XML

- Principal restrição de expressões de XPath
 - Caminho que especifica o padrão também especifica os itens a serem recuperados
 - Difícil especificar certas condições sobre o padrão enquanto se especifica separadamente quais itens do resultado devem ser recuperados.

XQuery: especificando consultas em XML

- XQuery expressão FLWR
 - Quatro cláusulas principais da XQuery
 - Forma:
 - FOR <vínculos de variável para nós (elementos individuais)>
 - LET <vínculos de variável para coleções de nós (elementos)>
 - WHERE <condições qualificadoras>
 - RETURN <especificação de resultado da consulta>
 - Zero ou mais instâncias da cláusula FOR, bem como da cláusula LET
 - WHERE opcional
 - RETURN deve aparecer uma única vez

XQuery: especificando consultas em XML

```
LET $d := doc(www.empresa.com/info.xml)
FOR $x IN $d/empresa/projeto[NumeroProjeto = 5]/
  trabalhadorProjeto, $y IN $d/empresa/funcio-
    nario
WHERE $x/horas gt 20.0 AND $y.cpf = $x.cpf
RETURN <res> $y/nomeFuncionario/primeiroNome,
  $y/nomeFuncionario/ultimoNome,
  $x/horas </res>
```

1. FOR \$x IN
doc(www.empresa.com/info.xml)
//funcionario [salarioFuncionario gt 70.000]/nomeFuncionario
RETURN <res> \$x/primeiroNome, \$x/ultimoNome </res>
2. FOR \$x IN
doc(www.empresa.com/info.xml)/empresa/funcionario
WHERE \$x/salarioFuncionario gt 70.000
RETURN <res> \$x/nomeFuncionario/primeiroNome, \$x/nomeFuncionario/ultimoNome </res>
3. FOR \$x IN
doc(www.empresa.com/info.xml)/empresa/projeto[numeroProjeto = 5]/trabalhadorProjeto,
\$y IN doc(www.empresa.com/info.xml)/empresa/funcionario
WHERE \$x/horas gt 20,0 AND \$y.cpf = \$x.cpf
RETURN <res> \$y/nome_funcionario/primeiroNome, \$y/nomeFuncionario/ultimoNome, \$x/horas</res>

Figura 12.7

Alguns exemplos de consultas XQuery em documentos XML que seguem o arquivo de esquema XML *company* da Figura 12.5.

XQuery: especificando consultas em XML

- Xquery possui construções muito poderosas para especificar consultas complexas
- www.w3.org
 - Contém documentos descrevendo os padrões mais recentes relacionados a XML e Xquery.

Outras linguagens e protocolos relacionados a XML

- Extensible Stylesheet Language (XSL)
 - Definir como um documento deve ser renderizado para exibição por um navegador Web
- Extensible Stylesheet Language for Transformations (XSLT)
 - Transformar uma estrutura em outra
- Web Services Description Language (WSDL)
 - Descrição de Web Services em XML

Outras linguagens e protocolos relacionados a XML

- Simple Object Access Protocol (SOAP)
 - Protocolo independente de plataforma e de linguagem de programação para transmissão de mensagens e chamadas de procedimento remoto
- Resource Description Framework (RDF)
 - Linguagens e ferramentas para trocar e processar descrições de metadados (esquema) e especificações pela Web

Exemplo

```
select * from Cliente
```

```
select * from Cliente
```

```
FOR XML RAW('Cliente'),
```

```
ELEMENTS, ROOT('Clientes')
```

Sugestão de Leitura

- DATE, C. J. **Introdução a sistemas de bancos de dados.** Tradução da 8ª edição americana de Daniel Vieira. Rio de Janeiro: Campus, 2004. 803 p.
- Capítulo: 27 - A World Wide Web e XML

Exercício 16

- Crie uma estrutura XML para armazenar dois registros de cada uma das entidades abaixo:
 - Ambulatorio (numeroA, andar, capacidade)
 - Medico (CRM, rg, nome, idade, cidade, especialidade, #numeroA)
 - Paciente (RG, nome, idade, cidade, doenca)
 - Consulta (#CRM, #RG, data, hora)
 - Funcionario (RG, nome, idade, cidade, salario)

Dúvidas



José Osvano da Silva
joseosvano@unipac.br