

Bancos de dados XML

Conceitos e linguagens de consulta

Sidney Roberto de Sousa

MC536 - Bancos de Dados: Teoria e prática
Material base: W3Schools XPath and XQuery Tutorial
<http://www.w3schools.com/Xpath/>
<http://www.w3schools.com/xquery/default.asp>

22/05/2009

Persistência XML

- XML: linguagem de marcação de dados, utilizada para troca, compartilhamento e armazenamento de dados
- XML não persistente: existe em algum momento em memória -> troca de dados dinâmica entre aplicações
- XML persistente: pode existir de duas maneiras:
 - Documento XML: arquivo em disco
 - Banco de dados XML: coleção de dados em XML

Bancos de dados XML

- Software de persistência de dados que permite o armazenamento de dados no formato XML
- Dois tipos de banco de dados XML:
 - Com suporte a XML
 - XML nativo

Bancos de dados com suporte a XML

- Mapeia todo XML para um banco de dados tradicional -> e.g., BD relacional
- Aceita XML como entrada
- Renderiza XML como saída
- O próprio banco de dados faz esta conversão
- Ex: Postgre possui suporte a XML (biblioteca interna de manipulação XML)

Bancos de dados XML nativos

- Modelo interno depende de XML
- Possui documentos XML como unidade de armazenamento
- Apesar disso, tais documentos podem não estar armazenados necessariamente como texto!
- Podem utilizar a mesma estrutura física de armazenamento de BD's relacionais ou orientados a objeto
- Por quê? Tempo de resposta na consulta
- Hoje vamos focar em banco de dados XML nativos

Mas enfim, por que usar um banco de dados XML?

- Razão mais convincente:
 - Aumento de fluxo de dados entre bancos de dados tradicionais e documentos XML nos sistemas de informação atuais!
- Maior eficiência (e facilidade) para converter e armazenar dados em XML

Analogia: SGBD tradicional

- Modelo de dados: MER
- Unidade de armazenamento lógica: tabela
- Estruturas de dados ou formatos proprietários (e.g., arquivos comprimidos e indexados)
- Coleção de dados: banco de dados

Analogia: SGBD XML

- Modelo de dados para o documento XML -> e.g., XPath (lembra da árvore e seus nós?)
- Unidade de armazenamento lógica: documento XML
- Estruturas de dados, formatos proprietários (e.g., arquivos comprimidos e indexados) ou arquivos de texto
- Coleção de dados: a maioria oferece grupos de documentos XML, chamados *coleções*

Manipulação de dados

- Inserção: cada SGBD possui seu conjunto de instruções (ponto negativo...)
- Consulta: grande maioria suporta XPath
- Exportação: formato XML, porém a maioria suporta Extensible Stylesheet Language Transformations - XSLT
(permite *output* em XML, *plain text*, HTML, PDF, etc)

XPath

- Linguagem básica para consulta em XML
- Permite projeção e seleção sob nós XML
- Limitação: necessita que o fragmento XML (persistente ou não) seja previamente carregado
- Só possibilita exportação em XML

XQuery

- Basicamente XPath estendido
- É para o XML o que o SQL é para as tabelas de bancos de dados
- Desenvolvido para consultar dados em XML
- Não apenas documentos em XML, mas qualquer coisa que se pareça com XML
- Mais poderoso do que XPath
- Má notícia: nem todos os SGBDs XML suportam XQuery
- Boa notícia: os melhores suportam!

XQuery: um breve leque de possibilidades

- Extrair informação para usar em um Web Service
- Gerar relatórios rápidos
- Transformar XML em XHTML
- Procurar informação relevante em documentos da Web

XQuery: estrutura de consulta básica

- Estrutura de uma consulta SQL:
SELECT FROM WHERE
- Estrutura de uma consulta básica XQuery:
FROM SELECT WHERE

XQuery: consultas FLWOR

- **f**or
- **l**et
- **w**here
- **o**rders
- **r**eturn

Vamos praticar?

XPath

Projeção em nós

- Projeção sob XML é muito simples
- Basta indicar o caminho do(s) elemento(s) desejado(s) na árvore
- Projeção sob elementos:
/bookstore/book/title
- Projeção sob atributos:
/bookstore/book/@category

Projeção em nós: Sintaxe

- `/`: Projeção a partir da raiz
- `//`: Projeção de um nó não importa onde ele esteja
`//author`
- `..`: Projeção do pai do nó atual
`/bookstore/book/author/..`
- `@`: Projeção sob atributos
`//@lang`
- `text()`: Projeção sob nós de texto
`/bookstore/book/author/text()`

Seleção de nós

- Uso de predicados para selecionar um nó específico
- Selecionar nó específico:
book[2] -> seleciona segundo livro
- Selecionar último nó:
author[last()] -> selecionar último autor
- Selecionar todos os nós a partir de um determinado nó:
book[position() > 1] -> todos os livros após o primeiro
- Selecionar todos os nós que possuem determinado atributo:
//title[@lang]

Seleção de nós

- Selecionar todos os nós que possuem determinado atributo com determinado valor:

```
//title[@lang='eng']
```

- Selecionar todos os elementos que satisfazem uma condição:

```
//book[price > 30.00]
```

```
//book[price != 30]
```

Exercícios

- Selecione a nota de Matemática de todos os alunos
- Selecione o nome de todos os alunos que tiveram notas menores ou iguais a 5
- Selecione o título de todos os livros da categoria Web

Seleção com mais de uma condição

- É possível combinar condições
- Selecionar todos os livros com preço entre 20 e 35:
`//book[price > 20 and price < 35]`
- Selecionar todos os livros com preço menor do que 20 ou maior do que 35:
`//book[price < 20 or price > 35]`

Exercícios

- Selecione o nome dos alunos que tiveram nota superior a 8 em português e menos de 2 faltas nesta disciplina

Selecionando vários caminhos

- É possível selecionar mais de um caminho na árvore
- Selecionar todos os títulos e preços dos livros:
`//book/title | //book/price`

Funções XPath

- Possui mais de 100 funções pré-definidas
- Funções para lidar com números, strings, erros, entre outros
- Para saber mais:

http://www.w3schools.com/Xpath/xpath_functions.asp

Vamos praticar?

XQuery

Consultas em XQuery

- Tudo o que fizemos até agora pode ser feito em XQuery
- Porém, com XQuery é possível carregar o documento XML desejado
- Tente isto:
`doc("CAMINHO_DA_PASTA_AULA/book.xml")//book[price < 30 or price > 35]`
- Estrutura de consulta:
 - FROM -> `doc("CAMINHO_DA_PASTA_AULA/book.xml")`
 - SELECT -> `//book`
 - WHERE -> `[price < 30 or price > 35]`

Consultas FLWOR

- Lembra um laço de repetição dentro de uma função
- Exemplo:
for \$x in doc("CAMINHO_DA_PASTA_AULA/book.xml")//book
where \$x/price > 10 return \$x/title

Estrutura de consulta

- FOR -> for \$x in
doc("CAMINHO_DA_PASTA_AULA/book.xml")//book
- (opcional) LET -> utilizar uma variável
let \$x := (1 to 5)
return <test>\$x</test>
- WHERE -> where \$x/price > 10
- (opcional) ORDER BY -> order by \$x/title
- RETURN -> return \$x/title

Exercício

- Escreva uma consulta FLWOR que retorne em ordem alfabética o nome de todos os alunos (alunos.xml) que tem nota inferior a 8 em matemática

Condições em XQuery

- É possível construir expressões IF-THEN-ELSE em XQuery
- Ex:
for \$x in doc("CAMINHO_DA_PASTA_AULA/book.xml")//book
return if(\$x/@category="CHILDREN")
then <crianca>{data(\$x/title)}</crianca>
else <adulto>{data(\$x/title)}</adulto>

Gerando HTML

- Uma das maiores utilidades do XQuery é gerar HTML dinamicamente
- Possibilita que uma página possa ser atualizada apenas alterando o conteúdo do documento XML (banco de dados)
- Basta saber um pouco de HTML e fazer consultas em XQuery

Exemplo de geração de HTML

- Após executar o exemplo, copie o resultado e salve como resultado.html

```
<html>
<body>
<h1>Bookstore</h1>
<ul>
{
for $x in doc("CAMINHO_DA_PASTA_AULA/book.xml")//book
order by $x/title
return <li>{data($x/title)}. Category: {data($x/@category)}</li>
}
</ul>
</body>
</html>
```


Exercício

- Gere uma planilha em HTML que contenha o nome de cada aluno, sua nota de matemática e o seu número de faltas nesta disciplina