

Ukládání a vyhledávání XML dat

XML – teorie a praxe značkovacích jazyků (4IZ238)

Jirka Kosek

Poslední modifikace: \$Date: 2019/11/28 21:30:44 \$

Copyright © 2001-2019 Jiří Kosek

Obsah

Ukládání XML dokumentů	3
Ukládání XML do souborů	4
Nativní XML databáze	5
Ukládání XML do relačních databází	6
Dotazování	7
Dotazování	8
XPath	9
Využití XSLT jako dotazovacího jazyka	10
XQuery	11
Rozšíření SQL	12
XQuery	13
XQuery	14
Další zdroje informací	15
Další zdroje informací	16

Ukládání XML dokumentů

Ukládání XML do souborů	4
Nativní XML databáze	5
Ukládání XML do relačních databází	6

Ukládání XML do souborů

- výhody:
 - není potřeba žádné speciální vybavení
- nevýhody:
 - s XML dokumentem lze manipulovat jen jako s celkem
 - dokumenty nelze rozumně indexovat a prohledávat

Nativní XML databáze

- databáze specializované na ukládání XML dokumentů
- místo tabulek se pracuje s kolekcemi XML dokumentů
- dokumenty v kolekci mohou, ale nemusí odpovídat určitému schématu
- lze manipulovat (výběr, změna, smazání, přidání) s libovolnou částí XML dokumentů
- lze snadno indexovat a prohledávat
- přístup k databázi – řádkové rozhraní, API, HTTP
- příklady: eXist, BaseX, MarkLogic, ...

Ukládání XML do relačních databází

- XML dokument se uloží jako CLOB/BLOB
 - snadná implementace
 - většina databází neumožňuje efektivní indexování a prohledávání BLOB položek
 - s XML dokumentem se musí pracovat jako s jedním celkem
- XML dokument se rozloží do několika relačních tabulek
 - mnoho způsobů dekompozice XML na relační model
 - relační model špatně zachycuje hierarchickou strukturu XML
 - rozložení a složení XML dokumentu je pomalé
 - mapování na relační model je většinou popsáno pomocí speciálních anotací v XML schématu
 - některé dotazy mohou být provedeny velmi rychle převedením na obyčejný SQL dotaz
- mapování XML na objektově-relační model
 - datový model OODB se po drobných úpravách (zachování pořadí dětí) dá použít pro ukládání XML dat
 - s XML dokumentem lze pracovat po částech
 - indexování a vyhledávání může využít funkce OODB
- nativní XML typ
 - zcela transparentní práce s XML
 - relační databáze zahrnuje funkčnosti nativních XML databází

Dotazování

Dotazování	8
XPath	9
Využití XSLT jako dotazovacího jazyka	10
XQuery	11
Rozšíření SQL	12

Dotazování

- navigace podle struktury dokumentu (struktury elementů)
- přístupy známé z SQL aplikované na XML strukturovaná data:
 - manipulace s fragmenty XML
 - řazení výsledků
 - agregační operace, seskupování
 - spojování dat z několika XML dokumentů v jednom dotazu
- dotazovací jazyky:
 - XPath – pouze pro jednoduché dotazy
 - XSLT – lze použít, ale příliš upovídaný
 - XQuery – speciální dotazovací jazyk navržený pro XML
 - rozšíření SQL o podporu XML dat

XPath

- výsledkem je posloupnost složená z uzlů nebo primitivních typů
- silné prostředky pro navigaci ve struktuře dokumentu
- pomocí funkce `doc` lze v jednom dotazu použít více zdrojů XML dat, lze provádět i jednoduchá spojení
- základní agregační funkce (`count`, `avg`, `sum`)
- nelze vytvořit dokument se složitější strukturou
- nelze vytvořit dokument s jinou strukturou než je v dotazovém dokumentu

Využití XSLT jako dotazovacího jazyka

- transformaci z XML → XML můžeme chápat jako provedení dotazu
- oproti XQuery „ukecanější“ syntaxe
- nelze optimalizovat tak dobře jako XQuery

XQuery

- dotazovací jazyk speciálně navržený pro XML
- spojuje možnosti XPathi s jednoduchými transformacemi a s operacemi známými z SQL
- XQuery = XPath 2.0 + FLWOR výrazy + výrazy konstruující nové elementy + uživatelsky definované funkce + několik dalších direktiv
- struktura FLWOR výrazu
 - FOR – výběr posloupnosti uzlů k dalšímu zpracování
 - LET – přiřazení proměnných pro každý prvek posloupnosti
 - WHERE – filtrování uzlů v posloupnosti
 - ORDER BY – seřazení vybraných a odfiltrovaných uzlů
 - RETURN – specifikace výstupu pro každý vybraný a odfiltrovaný uzel
- existuje velké množství implementací

Rozšíření SQL

- většina high-end databází obsahuje nějakou podporu XML
- proprietární rozšíření SQL

```
SELECT firstname, lastname, employeeid FROM Employess
WHERE employeeid = 1 FOR XML RAW
```

```
<row firstname="Nancy" lastname="Davolio" employeeid="1"/>
```

- SQL/XML – rozšíření standardu SQL o mapování mezi SQL a XML typy a názvy identifikátorů, funkce pro generování XML z výsledku SQL dotazu, načítání/ukládání XML, ...

```
select xmlelement("emp",
                  'Employee ' ,
                  xmlelement( "name", e.ename),
                  ' was hired on ',
                  xmlelement("hiredate", e.hiredate)) as result
from employess e;
```

```
<emp>
  Employee <name>John</name> was hired
  on <hiredate>2002-12-07</hiredate>
</emp>
```

- XPath/XQuery lze kombinovat s SQL

```
SELECT
  XMLQUERY('for $p in /PurchaseOrder
            where $p/Requestor = "Allan D. McEwen"
            return $p/Reference'
           PASSING object_value
           RETURNING CONTENT)
FROM PURCHASEORDER;
```

XQuery

XQuery 14

XQuery

viz <http://www.kosek.cz/xml/2005devcon/>

Další zdroje informací

Další zdroje informací	16
------------------------------	----

Další zdroje informací

- XQuery tutorial¹
- Specifikace XQuery a XPath 2²
- Saxon³ – implementace poslední verze návrhu XQuery
- Podpora XML v DB Oracle⁴
- Podpora XML v MS SQL Serveru⁵

1

<https://www.progres.com/products/db-integration-suite/db-integration-suite-developer-center/db-integration-suite-tutorials/learning-xquery/xquery-a-guided-tour>

² <http://www.w3.org/XML/Query#specs>

³ <http://saxon.sourceforge.net/>

⁴ <https://www.oracle.com/database/technologies/appdev/xmlldb.html>

⁵ <http://msdn.microsoft.com/en-us/library/bb522446.aspx>