

Databázové systémy a SQL

Lekce 10 - Úvod do XML

Daniel Klimeš

- Formát pro přenos strukturovaných dat
- Text se značkami (tagy)
- Připomíná HTML
- Samotný standard specifikuje jen minimum značek
- Položky (uzly - nodes)
 - Elementy
 - Atributy
 - ...
- Zanořená struktura
 - 1 kořenový element (root)
 - N zanořených elementů

- Hlavička (Processing Instruction)
 - `<?xml version="1.0" encoding="UTF-8"?>`
- Uzly
 - Elementy
 - `<tag>data</tag>`
 - `<question>Datum narození</question>`
 - Atributy
 - `<element atribut=„text“>`
 - `<name lang="cz">`
 - Komentář
 - `<!-- no need to escape <code> & in comments -->`
- Hierarchie elementů
 - Root element
 - Zanořené elementy
 - `<el_root>`
 `<el_nest>text</el_nest>`
 - `</el_root>`
 - `<cluster>`
 `<question>Datum vyšetření</question>`
 - `</cluster>`

Speciální znaky

- < místo "<"
- > místo ">"
- & místo "&"
- ' místo "'"
- " místo ""

- Textový editor - Notepad
- XML editor - XML Spy
- Moderní databáze - ORACLE
 - Nadstavba SQL
- Programovací jazyky
 - Parsery

- XPATH - formát vyhledávacích dotazů do XML dokumentů
- XML schéma - šablona XML dokumentu
- XSLT - transformace XML do jiného formátu (jiné XML, prostý text , HTML)
- Xquery - pokročilé vyhledávání v XML

- Internet
 - *www.w3.org*
 - *www.w3schools.com*

<bookstore>

<book>

<title>Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

< /book>

</bookstore>

- Uzly (nodes) – elementy, atributy, text, komentář,..
- Parent, children, siblings, ancestors, descendants

Výraz	Popis
/	Kořenový uzel
//	Uzel kdekoliv
.	Aktuální uzel
..	Rodičovský uzel
@	Atribut
text()	Obsah elementu

<code>/bookstore/book[1]</code>	První element kniha
<code>/bookstore/book[last()]</code>	Poslední element kniha
<code>/bookstore/book[last()-1]</code>	Předposlední kniha
<code>/bookstore/book[position()<3]</code>	První 2 knihy
<code>//title[@lang]</code>	Jakýkoliv element title s atributem lang
<code>//title[@lang='eng']</code>	Jakýkoliv element title s atributem lang s hodnotou 'eng'
<code>/bookstore/book[price>35.00]</code>	Knihy s cenou větší než 35
<code>/bookstore/book[price>35.00]/title</code>	Názvy knih s cenou větší než 35

<code>*</code>	Libovolný element
<code>@*</code>	Libovolný atribut
<code>node()</code>	Libovolný uzel

- Podpora ukládání XML dat
 - Datový typ XMLTYPE
- Podpora importu dat v XML formátu
 - Přímé vložení pomocí INSERT příkazu
 - XML repository
 - Propojení úložiště s tabulkou
- Podpora exportu dat v XML formátu
 - Funkce DBMS_XMLGEN.getXML
 - Výstup SQL dotazu ve formátu XML
 - Funkce sys_XMLGen
 - Export jednoho sloupce tabulky jako jednoduché XML
 - Speciální SQL funkce - skládání XML stromu
 - XMLELEMENT
 - XMLATTRIBUTES
 - XMLAGG
 - ...

- Vytvoření tabulky
 - Datový typ CLOB pro velké textové řetězce
 - Datový typ XMLTYPE pro XML formát
 - CREATE TABLE jmeno_tabulky
(velky_text CLOB,
xml_sloupec XMLTYPE);
 - SELECT DBMS_XMLGEN.getXML('SELECT study_id,
study_name FROM studies') FROM DUAL

- Generování XML
 - SELECT
 DBMS_XMLGEN.getXML('SELECT study_id, study_name FROM studies')
 FROM DUAL
 - INSERT INTO jmeno_tabulky (velky_text)
 SELECT
 DBMS_XMLGEN.getXML('SELECT study_id, study_name FROM studies')
 FROM DUAL
 - UPDATE jmeno_tabulky SET xml_sloupec = XMLTYPE(velky_text)
 - INSERT INTO jmeno_tab (xml_sloupec)
 SELECT SYS_XMLGEN(study_id) FROM studies

- XMLELEMENT
- XMLATTRIBUTES
- XMLAGG

```
SELECT XMLELEMENT (studie, XMLELEMENT (id, study_id),
XMLELEMENT (jmeno, study_name), XMLELEMENT (nazev, study_title))
FROM studies
```

```
SELECT XMLELEMENT (studie, XMLATTRIBUTES(study_id as "id",
study_name as "jmeno"), XMLELEMENT (nazev, study_title)) FROM
studies
```

```
SELECT XMLELEMENT (studie, XMLAGG( XMLELEMENT (nazev,
XMLATTRIBUTES (study_id as "id"), study_title))) FROM studies
```

- SELECT COUNT(*) FROM jmeno_tabulky WHERE **existsNode**(xml_sloupec, '//STUDY_NAME') = 1
- SELECT COUNT(*) FROM jmeno_tabulky WHERE **existsNode**(xml_sloupec, '//ROW[STUDY_ID=3]') = 1
- SELECT **extract**(xml_sloupec, '//ROW[STUDY_ID=3]/STUDY_NAME').**GetClobVal()**
FROM jmeno_tabulky WHERE existsNode(xml_sloupec, '//ROW[STUDY_ID=3]') = 1
- SELECT **extractvalue**(xml_sloupec, '//ROW[STUDY_ID=3]/STUDY_NAME')
FROM jmeno_tabulky WHERE existsNode(xml_sloupec, '//ROW[STUDY_ID=3]') = 1

- SELECT **extractvalue**(value(t), '//STUDY_NAME')
 FROM jmeno_tab,
table(xmlsequence(extract(jmeno_tab.xml_sloupec,
 '//ROW'))) t
 WHERE **extractvalue**(value(t), '//STUDY_ID') < 100

- Který formulář má nejvíce otázek? (Cluster_description)
- Která studie má nejvíce pacientů? (study_name)
- Do kterých studií se zařazovali pacienti v roce 2008 (study_name)
- Vypište názvy všech otázek ve skupině XY
- Vypište pracoviště, která nezařadila žádného pacienta
- Vypište pracoviště, která se účastní více studií
- Vytvořte pohled study_name, počet zařazených pacientů
- Zjistěte průměrnou hodnotu, minimum , maximum otázky XY
- Vypište věk pacientů při zařazení do studie XY
- Vypište hodnotu otázky XY pro všechny pacienty (včetně nevyplněných)
- Jaký je průměrný počet vyplněných formulářů na pacienta u studie XY