

# Databázové systémy a SQL

---

## Lekce 11 - 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

- &lt; místo "<"
- &gt; místo ">"
- &amp; místo "&"
- &apos; místo "'"
- &quot; 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](http://www.w3.org)*
  - *[www.w3schools.com](http://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()&lt;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&gt;35.00]</code>	Knihy s cenou větší než 35
<code>/bookstore/book[price&gt;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

- 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, má nejvíce pacientů
- 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)
- Kumulativní počty náběru pacientů do studie
- Najděte pacienty zařazené do více než jedné studie

- Vytvořte zprávu/report o obsahu tabulky zapocet. Zpráva by měla obsahovat minimálně:
  - Počet záznamů
  - Přehled vyplněnosti jednotlivých sloupců (počet vyplněných, počet unikátních hodnot)
  - Pokud jde o kategoriální parametr, přidejte procentické zastoupení jednotlivých kategorií
  - Pokud jde o číslo nebo datum, zjistěte minimum, maximum, případně medián a průměr
  - Pokud zjistíte nějakou závislost mezi parametry (v hodnotách nebo vyplněnosti), slovně uveďte.