
Databasesystemer, forår 2006
IT Universitetet i København

Forelæsning 12, del 1: XML i databaser

27. april 2006

Forelæser: Rasmus Pagh

Baggrundsmateriale (ikke pensum):

- <http://www.w3.org/XML/1999/XML-in-10-points.html>
 - <http://www.zvon.org/xxl/XPathTutorial/> (inkl. Xlab XPath fortolker)
 - http://www.stylusstudio.com/sqlxml_tutorial.html
-

— Today's lecture, part I —

XML for data exchange

- What is XML?
- Defining XML formats using XML schemas.
- (Web) services and service oriented architectures (SOA).

XML in a relational databases

- Ad-hoc information.
- XML data types.
- XML query languages (XPath, XQuery).

— Integration of databases —

- In many businesses, data from a large number of heterogeneous databases need to be integrated
 - in connection with data warehousing, or
 - in connection with system integration in general
- Communication among databases is no easy task, due to differences in formats, conventions, systems, etc.

Also, modern service oriented architectures (SOA) consist of many components (services) that interact through communication protocols that transmit data.

XML

XML is a *standardized textual notation for so-called “semistructured” data*.

It is (primarily) aimed at semistructured data which is a **tree**. More specifically, an XML tree is a structure consisting of:

- *Leaves* that contain data (in the form of strings)
- *Internal nodes*, each having a number of leaves and/or other internal nodes as *children* (no cycles are allowed).

In XML notation, an internal node with label `1` is represented as `<1>...</1>`. The dots ... abbreviate the XML for the children of the node.

XML example

```
<?xml version="1.0">
<root>
  <star>
    <name>Carrie Fisher</name>
    <address><street>Maple</street><city>H'wood</city></address>
    <address><street>Locust</street><city>Malibu</city></address>
  </star>
  <star>
    <name>Mark Hamill</name>
    <address><street>Oak</street><city>B'wood</city></address>
  </star>
  <movie>
    <title>Star Wars</title>
    <year>1977</year>
  </movie>
</root>
```

— XML and WWW —

Besides data interchange, another use of XML in connection with databases is for sharing information via the World Wide Web.

- Web browsers have special facilities for viewing XML documents (e.g., containing the result of a database query).
- There are specialized languages such as XSLT that can be used to specify how XML data is to be presented in a browser (converting it to HTML).

— Schemas for XML —

When doing data interchange it is necessary to have a *common description* of the data format, i.e., we need a specification of what data is allowable.

Several languages for writing such **schemas** for XML are used. The most widespread are **DTD** (old and well-established) and **XML Schema** (new, more powerful standard).

These schema languages work by specifying a **grammar** for the XML documents allowed, i.e., a set of rules that can be used to form any allowable XML document.

Essentially, for each `<1>...</1>` (called an XML **element**) it is specified what can occur between `<1>` and `</1>`.

— XML and service architectures —

Service oriented architectures (SOA) is a current trend in IT systems.

XML is the basis in an on-going standardization of:

- Definition language for (web) *services* (WSDL).
- "Yellow pages" for (web) services (UDDI).

These, or similar, industry standards are likely to play a major role in future systems (web based and within organizations).

— Why XML? —

The reason for the success of XML and XML Schema is, in fact, *not* that it does something that could not be done before (cf. EDI, SGML).

The good new thing is **standardization**:

- There is *widespread agreement* that the XML standards will form the basis of information interchange in the future. (E.g., XML is the format chosen for information interchange in the Danish public sector.)
- Consequently, the major players in the software industry, many country administrations, etc., support XML.
- There are many tools available for XML and related technologies.

Next: XML in relational databases.

— XML data types —

The major DBMSs now offer XML as a native data type, similar to strings, integers, etc.

XML data can be processed using e.g. (a subset of) the XQuery/XPath languages. XQuery generalizes in many ways SQL. However, in general one should not expect as good performance as delivered by SQL.

The interface between SQL and XML is not well-established – there are several proprietary proposals (e.g. SQLXML), as well as a proposal in the SQL 2003 standard (SQL/XML).

— Ad-hoc information —

In many databases, the need to store new kinds of information arises continually.

Adding new attributes to a relation may be a bad idea if the new kind of information is only relevant for few rows (many NULLS).

A possible solution is to store such ad-hoc information in an XML attribute. This is particularly attractive if the new data is not used much in searches (i.e., does not need to be indexed).

— XPath and XQuery

(Lecture switches to html slides.)

— Most important points in this lecture —

As a minimum, you should after this week:

- Be able to recognize an XML document.
- Know some application areas of XML.
- Have a good idea about how XML query languages work.