XML and JSON in Databases

Marek Rychlý
rychly@fit.vut.cz

Brno University of Technology
Faculty of Information Technology
Department of Information Systems

Data Storage and Preparation (UPA)
25 November 2025

Outline

Extensible Markup Language (XML)

- A markup language and text-based file format for the Internet.
  (one of the “Web standards”, i.e., by WWW Consortium/W3C and for Web)
- For platform-independent storing/transmitting arbitrary data.
  (it is standardized and well-recognized by major vendors across many products)
- An XML document consists of

declaration	of an XML document with version (1.0) and encoding (e.g., utf-8);
elements	for logical parts with name, attributes, and the content between the start-tag and end-tag, or empty;
tags	for markup constructs that begin with < and end with >;
attributes	for name-value pairs that exists within the elements;
...	encoded chars, CDATA, processor directives, etc.

- A correct XML document must be

well-formed	which is syntactically correct in its format,
valid	which is semantically correct in its content.

- The valid XML document must follow a specific XML schema.
  (the XML schema is usually referred from the XML document by XSI attributes)

An Example of XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="./menu.css"?>
<?xml-stylesheet alternate="yes" type="text/xsl" href="./menu.xsl"?>
<breakfastMenu xmlns="http://mypub.com/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://mypub.com/ http://mypub.com/menu.xsd">
  <food>
    <name>Pilsner Urquell</name>
    <price>50.0</price>
    <description>A pale lager beer brewed in Plzen.</description>
    <calories>215</calories>
  </food>
  <!-- ... another instances of the food element -->
</breakfastMenu>
```

The example puts elements into default namespace `http://mypub.com/menu` and refers the corresponding XSD schema. There are also two processor directives to provide formatting visual styles in CSS and XSLT.

XML Schema

- The content of an XML document is restricted by an XML schema.
  (utilized by an XML parser to check validity of the document)
- The XML schema is described by an external XML document.
  (e.g., by an XSD file for W3C XML Schema Definition)
- There are several types of XML schema documents.
  - **Document Type Definition (DTD)** – from SGML, deprecated;
  - **W3C XML Schema** – de-facto standard in XML validation;
  - **RELAX NG** – a strong language, the ability to validate text nodes;
  - **Schematron** – rule-based with XPath rules, also quite strong.
- To understand an XML doc. structure, the schema is necessary.
  (XML schema in XML databases vs. relational schema in relational databases)

An Example of XSD Document

(it corresponds to the previous example of XML document)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
  targetNamespace="http://mypub.com/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="breakfastMenu" type="breakfast-menu-type"/>
  <xs:complexType name="breakfast-menu-type">
    <xs:sequence>
      <xs:element type="food-type" name="food"
        maxOccurs="unbounded" minOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="food-type">
    <xs:all>
      <xs:element type="xs:string" name="name"/>
      <xs:element type="xs:string" name="price"/>
      <xs:element type="xs:decimal" name="description"/>
      <xs:element type="xs:integer" name="calories"/>
    </xs:all>
  </xs:complexType>
</xs:schema>
```

Types of XML Documents

Data-centric XML documents

- for data collections with complex structures
  (usually numerical and non-text attribute-value data)
- usually utilized for export/import of data
  (e.g., from/to non-XML databases or applications)
- good and regular structure, order unimportant
  (the ordering of attribute-value data is not important)

Document-centric XML documents

- for text documents that are marked up as XML
  (to capture document structure, e.g., paragraphs, sections, footnotes etc.)
- utilized for rendering visual outputs,
  (does not make sense to store them into databases as structured data)
- irregular structure, order important
  (one element can be used in various contexts, e.g., XHTML `<span>`)

An Example of Document-centric XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Title of document</title>
  </head>
  <body>
    A break: <br />
    A horizontal rule: <hr />
    An image: <img src="./happy.gif" alt="Happy face" />
  </body>
</html>
```

A data-centric example was provided before on page 5.

XML Path Language (XPath)

- Querying XML data/documents by a path to matching elements.
  (a context-based location path consists of a sequence of location steps)
- To find XML nodes based on the path from the context or a root element, restricted by given predicates or modified by functions.
- XPath is utilized by Schematron and in XSLT.
  (to target elements in rules for validation or transformation, respectively)

```
(: get food names with the price below 100 :)
/breakfastMenu/food[price < 100]/name

(: get food with the name staring with P letter :)
/*/food[starts-with(name,'P')]

(: get a string of all names separated by the semicolon :)
string-join(//name/text(),'; ')
```

XML Query (XQuery)

- A functional prog. language to query and transform XML docs.
  (generally applicable and extendable to any collections of structured and unstructured data, e.g., to JSONs or binary data)
- Comparable to XSLT, however, with significantly different goals.
  (XQuery is for querying, XSLT is for transformations)

```
for
  $i in fn:doc("m.xml")/breakfastMenu/food[name = 'Pilsner Urquell'],
  $j in fn:doc("m.xml")/breakfastMenu/food[price < $i/price]
  order by $j/price, $j/name
return
  <cheaper-than-pilsner>{ $j/name, $j/price }</cheaper-than-pilsner>
```

Types of XML Data Adoption/Support (1)

DBs supporting XML; Native XML DBs

Databases with Support of XML Data

- The most of the current RDBMS do support XML data.
- XML data is stored as structured/object data and may be queried.
- Oracle XML DB, PostregSQL XML type, MySQL XML functs., etc.

Native XML Databases

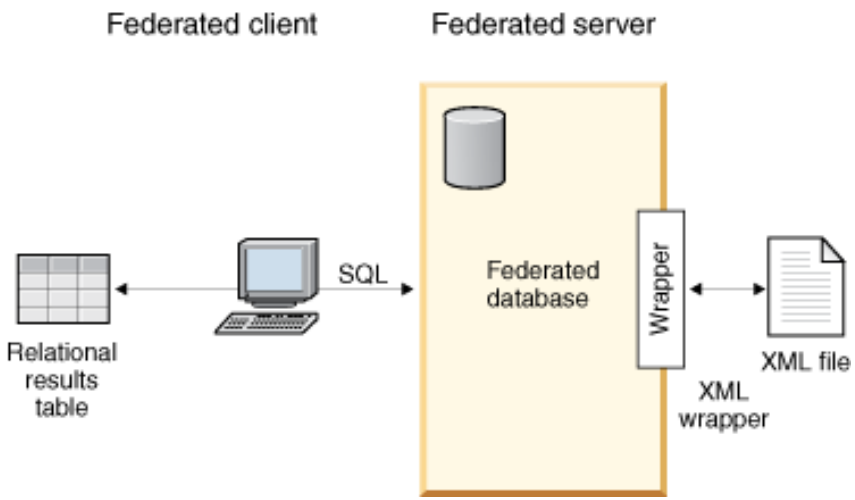
- Data model is based on the Document Object Model (DOM).
- Queries and operations on the XML data are fast.
- E.g., BaseX, eXist-db, and Sedna.

Types of XML Data Adoption/Support (2)

XML Wrappers and Mappers

XML Wrappers/Mappers

- To access XML data in relational way, e.g., to query in SQL.
- E.g., an XML wrapper as a component of the DB2 federated arch.



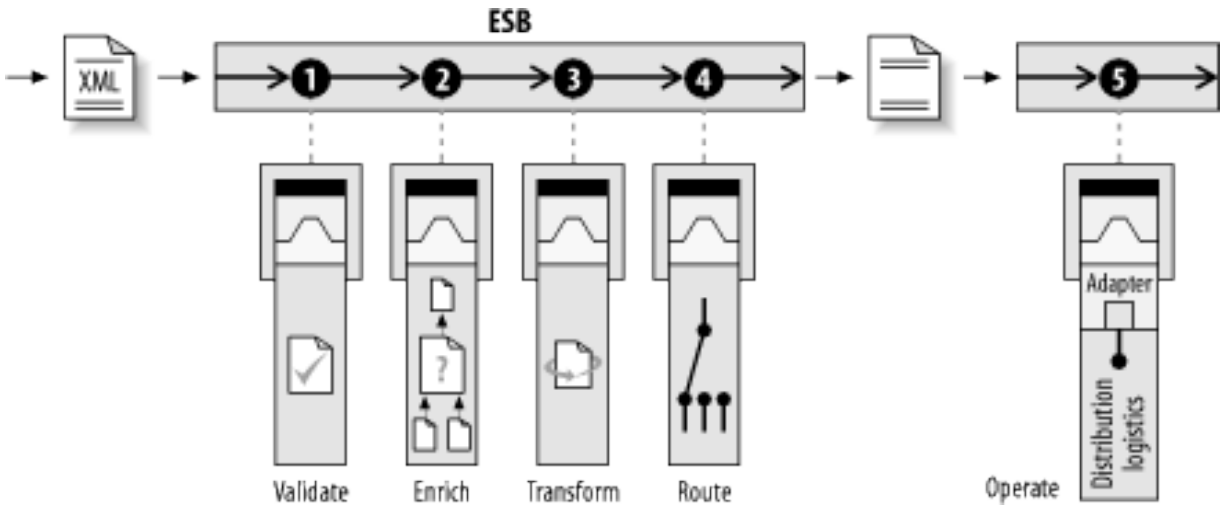
(adopted from “Db2 11.5: XML wrapper”)

Types of XML Data Adoption/Support (3)

XML Middleware

Middleware Utilizing XML Technologies

- To communicate between application components in XML format.
- The XML can be utilized as platform-independent data format.
- The ability to implement VETRO features.
  (Validity, Enrichment, Transformations, and content-based ROuting of XML msgs.)
- E.g., SOAP/RDF/RSS-based Web-services, ESB.



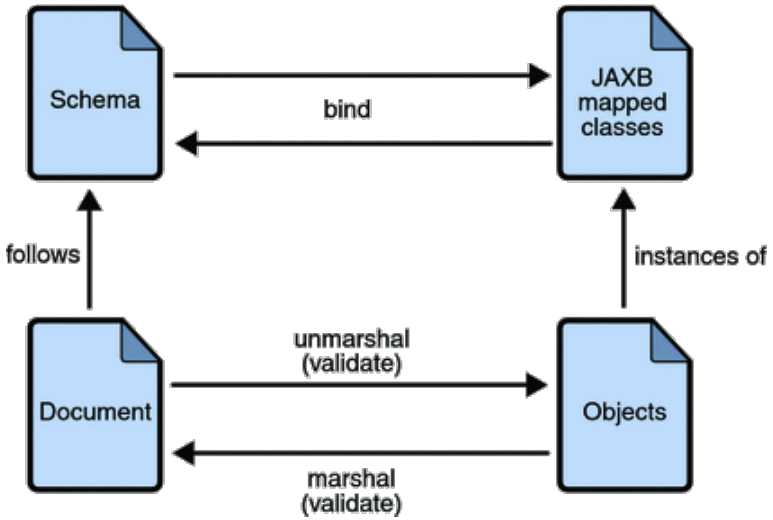
(adopted from “David A. Chappell: Enterprise Service Bus, 2004”)

Types of XML Data Adoption/Support (4)

XML Binding

XML Data Binding

- The mapping of XML to object data and vice versa.
- To provide a persistence of the objects.
  (i.e., serialization and deserialization to/from XML data)
- E.g., Java Architecture for XML Binding.



(adopted from “The Java Tutorials”)

XML in Relational Databases

1. XML data in binary/text CLOB/BLOB SQL data types
   - \+ fast and easy to implement; strictly respects/keeps the content of the original XML document
   - – difficult to query or update as a relational databse engine cannot access the XML elements inside the stored text/binary values
2. XML data is parsed and individual elements-content and attribute-values are stored in a relation table as column-values
   - \+ the extracted data is relational and can be queried/modified by common means
   - – the original XML document is lost and it may be difficult to reconstruct from the relational data
3. XML data is stored in SQL XML data type as object-rel. data
   - \+ XML objects have attributes&methods to update/query the XML data
   - – the content of the XML document can be integrated with relational data into SQL statements and constraints

XML and SQL

- SQL:2003-14 defines the basic XML datatype,
  (later updated in SQL:2006-14, SQL:2008-14, and SQL:2011-14)
  - including mappings of XML and SQL data types and meta-data,
  - predicates to check XML content (CONTENT), match to a XQuery expression (XMLEXISTS), and validity (VALID),
  - and functions such as XMLQUERY to extract values from XML fields.
- However, many DB vendors do not fully support this standard.
  (e.g., not in MySQL; partially in Oracle, IBM DB2, or MS SQL Server)
- XML data supported in Oracle 9.2 and later (since 2002).
  (XMLType similar to SQL:2003 XML data type; by Oracle XML DB component)

XML Data in Column and XPath Querying in SQL

```
CREATE TABLE resolutions_xml (
  id number PRIMARY KEY, legis_num NUMBER,
  resolution XMLType
);

SELECT id,
  extract(resolution,
    '/resolution[@public-private="public"]/action')
  AS action,
  extractValue(resolution,
    '/resolution[congress="108"]/official-title')
  AS title
FROM resolutions_xml
WHERE existsNode(resolution,
 '/resolution[legis-num="558"]') = 1;
```

(adopted from “Querying XML, An Oracle White Paper”)

Querying XML Data in SQL by XQuery

```
SELECT xmlquery('
for $r in $res/resolution
let $a := $r/action
where $a/action-date="20040311"
order by $r/legis-num ascending
return
<all-sponsors>
{$a/action-desc/sponsor}
{$a/action-desc/cosponsor}
</all-sponsors>'
  passing resolution as "res"
  returning content
)
FROM resolutions_xml;
```

(adopted from “Querying XML, An Oracle White Paper”)

JavaScript Object Notation (JSON)

- A language-independent data format to store and transmit data.
  (derived from JavaScript, so any valid JSON file is a valid JavaScript file)
- Usually smaller data representation than in the case of XML.
  (and more human-readable and easily parsable than XML; structures, arrays, etc.)
- No data-metadata separation, less customizable, more relaxed[1].
  (vs. XML validation by a schema w/user-defined types, predefined tags, etc.)

```
[ {
  "name": "Pilsner Urquell", "price": 50.0,
  "description": "A pale lager beer brewed in Plzen.",
  "calories": 215
}, {
  "name": "...", "price": 123.0,
  "description": "...",
  "calories": 123
} ]
```

[1] There is JSON Schema is based on the concepts from XSD; not standardized.

JSON and SQL

- SQL:2016 introduced JSON functions,
  (no JSON data type, just functions to process JSON in text-based SQL types)
  - json_object and json_array to create JSON data as SQL string,
  - json_exists, json_value, and json_query to query JSON data,
  - json_table to transform a JSON data into corresponding SQL table.
- SQL can utilized JSON path language from ECMAscript.
  (using $ for the current context element, . for an object member, and [] for an array)

JSON Data in Column and JSON Querying in SQL

```
CREATE TABLE t (
  jcol CLOB CHECK (jcol IS JSON)
);

INSERT INTO t (jcol) VALUES (json_array(
  json_object('id' value 10, 'name' value 'Anna'),
  json_object('id' value 20, 'name' value 'Bob')
));

SELECT jt.* FROM t, JSON_TABLE (
  jcol, '$[*]' COLUMNS (id NUMERIC PATH '$.id',
    name VARCHAR(255) PATH '$.name')
) jt
WHERE json_exists(jcol, '$.id')
  AND json_exists(jcol, '$.name');
```

(adopted from “What’s New in SQL:2016”)

JSON in NoSQL Databases

- JSON is well-supported and utilized by NoSQL databases.
- The utilization is both
  - to store JSON value in a key-value NoSQL db for parsing in app.
  - and to serve as a data representation in document NoSQL dbs.
- See MongoDB in the NoSQL lecture and labs.
- Also another NoSQL databases, such as Couchbase, MarkLogic, OrientDB, Riak, etc.

Thank you for your attention!

Marek Rychlý

<rychly@fit.vut.cz>