XML and Relational Database

IT 4153 Advanced Database

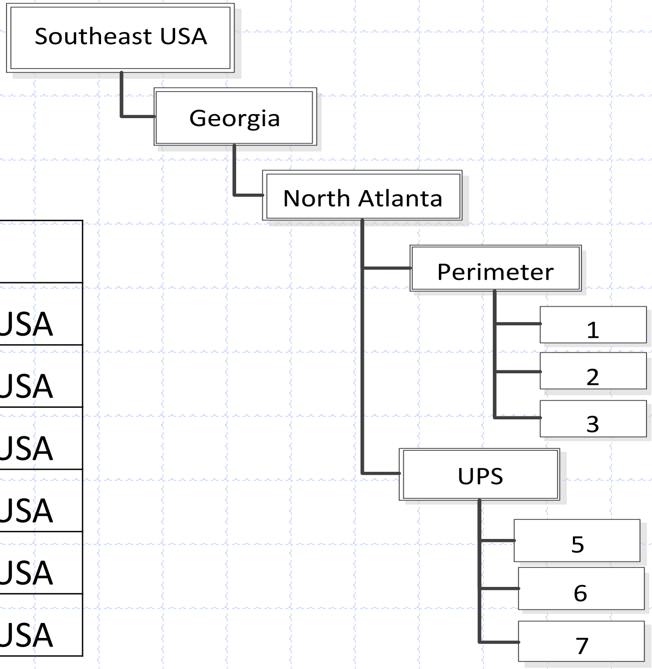
J.G. Zheng
Spring 2012

SPSU SOUTHERN POLYTECHNIC STATE UNIVERSITY

Data Structure

- Many data are hierarchical in nature, such as organizations, geographical regions, categories, etc.

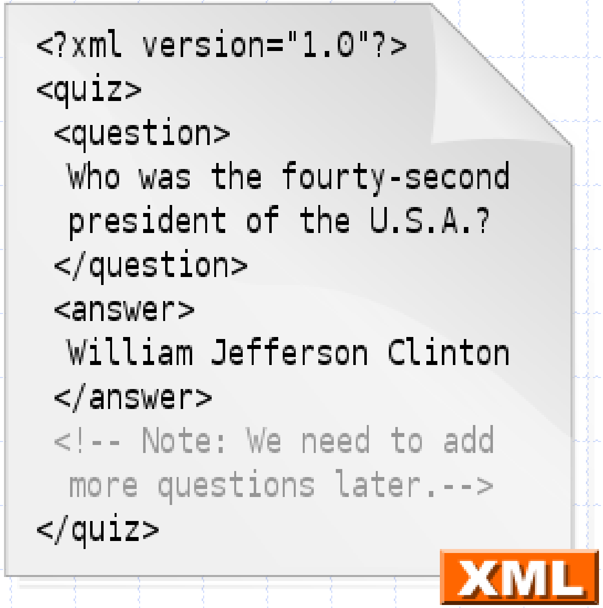
- Relational model is based on data items share common attributes; it is more complex to model hierarchical data using the relational model
  - More redundancy
  - More tables
  - More complex structure



Officer	Club	Area	Division	District	Region
1	Perimeter	North Atlanta	A	Georgia	Southeast USA
2	Perimeter	North Atlanta	A	Georgia	Southeast USA
3	Perimeter	North Atlanta	A	Georgia	Southeast USA
5	UPS	North Atlanta	A	Georgia	Southeast USA
6	UPS	North Atlanta	A	Georgia	Southeast USA
7	UPS	North Atlanta	A	Georgia	Southeast USA

XML (Extensible Markup Language)

- XML is a markup language to encode data and content using plain text
- A piece of data is coded as an XML element, which includes
  - Element name: wrapped in tags (markups), which describes the content (metadata)
  - Element content: anything go between tags.



```
<?xml version="1.0"?>
<quiz>
 <question>
  Who was the fourty-second
  president of the U.S.A.?
 </question>
 <answer>
  William Jefferson Clinton
 </answer>
 <!-- Note: We need to add
  more questions later.-->
</quiz>
```

- An XML document is in a typical hierarchical structure, consisting of hierarchical elements.

Example: Raw Data

- Consider the following data to be organized in XML format

- Course information
  - CIS 3730 Designing and Managing Data
  - CRN: 10059
  - Instructor: Dr. Jack Zheng

Example: A Sample XML Document

Notice how data are organized in hierarchical tags and text.

An XML document starts with this declaration to indicate this is an XML document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Course CRN="10059">
   <Prefix>CIS 3730</Prefix>
   <Title>Designing and Managing Data</Title>
   <Instructor><Title>Dr.</Title>
   Jack Zheng
   </Instructor>
</Course>
```

"CRN" is an attribute node of an element. The value of an attribute is always enclosed within double quotes.

Tags are used as metadata to describe data and content.

All elements should have starting tag (<tagname>) and closing tag</tagname>. Note there should not be spaces in tag names.

This is a text node, indicating the content value of the element "Instructor".

XML Element

- Elements are the fundamental units of XML content.
  - Element name: wrapped in tags (markups), which describes the content (metadata).
  - Element content: anything go between a pair of opening and closing tag.
- Simple element: a simple element has name (tag) and text content.

```
<Instructor>Dr. Jack Zheng</Instructor>
```

- Complex element: a complex type element can have a mixture of
  - Child elements (element nodes)
  - Plain texts (text nodes)
  - Attributes (attribute nodes)

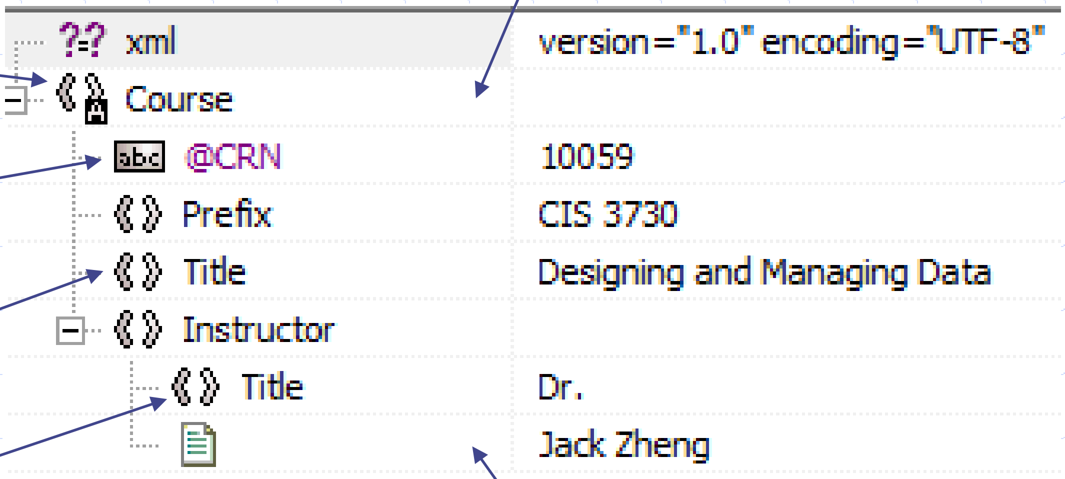
```
<Instructor Title="Dr.">Jack Zheng</Instructor>
<Instructor Type ="Visiting"><Title>Dr.</Title>Jack Zheng</Instructor>
```

- Every XML document must have a single root (top level) element (with opening and closing tags).

XML Nodes Hierarchy

- An XML document consists of hierarchical elements and nodes.
- Major node types
  - Element node (element)
  - Attribute node (attribute)
  - Text node (text content)

A graphic tree view of the document on the last slide (created in XMLPad 3). Hierarchies are clearly visualized.



?=? xml	version="1.0" encoding="UTF-8"
Course	
abc @CRN	10059
Prefix	CIS 3730
Title	Designing and Managing Data
Instructor	
Title	Dr.
	Jack Zheng

"Course" is the root element.

"CRN" is an attribute node of an element.

These are second level elements under "Course"

This is a third level element under "Instructor"

This is a text node under "Instractor".

XML’s Features

- It is a textual data format, with strong support via Unicode.
  - Easy to exchange information between different types of programs in different computers.

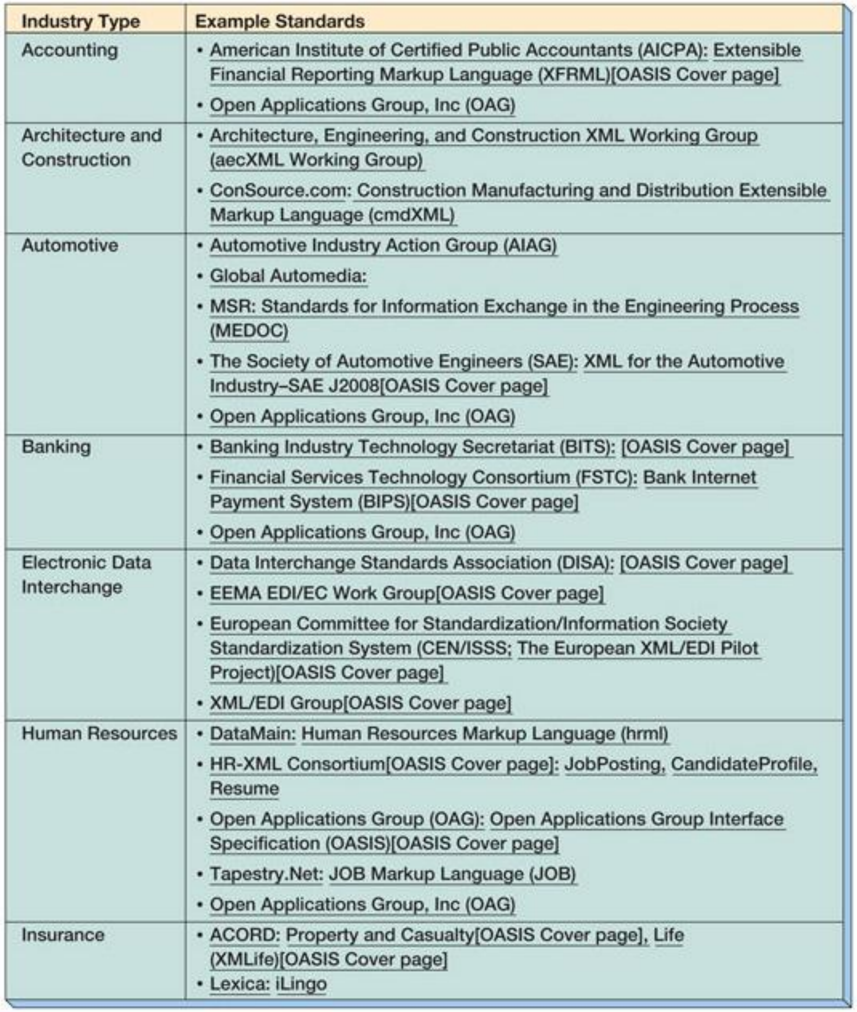
- XML documents are semi-structured
  - Markups provide simple description of the content, and can be easily used for processing instructions.
  - The tree (hierarchical) structure is easy to understand, and already used commonly in data structures and computing (generality).

- Standardized
  - Standards have been developed to represent various types of documents.
  - Standard programs have been developed to read, transform, and generate XML documents.

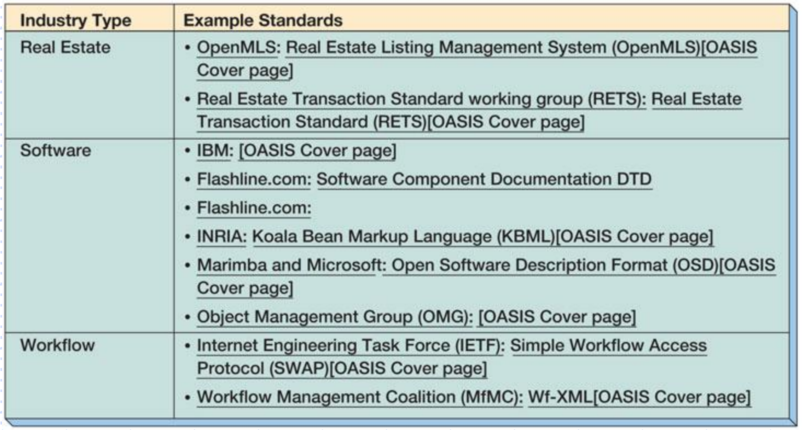
XML’s Major Uses

- Document representation
  - Document: OOXML, XHTML
  - Resource: OWL, RDF
  - Content: OPML
  - Vector graphics: SVG
- Data structure/storage
  - Configuration file: web.config, httpd.conf
  - Database: XML database, XML data type
  - Object serialization
- Exchanging data/message
  - Content syndication: RSS, Atom
  - Commutation protocol: SOAP, WSDL, WAP
- Representational language
  - Style: XSLT
  - Interface representation: User interface: XUL, XAML

XML in Industries



Industry Type	Example Standards
Accounting	• American Institute of Certified Public Accountants (AICPA): Extensible Financial Reporting Markup Language (XFRML)[OASIS Cover page] • Open Applications Group, Inc (OAG)
Architecture and Construction	• Architecture, Engineering, and Construction XML Working Group (aecXML Working Group) • ConSource.com: Construction Manufacturing and Distribution Extensible Markup Language (cmdXML)
Automotive	• Automotive Industry Action Group (AIAG) • Global Automedia: • MSR: Standards for Information Exchange in the Engineering Process (MEDOC) • The Society of Automotive Engineers (SAE): XML for the Automotive Industry–SAE J2008[OASIS Cover page] • Open Applications Group, Inc (OAG)
Banking	• Banking Industry Technology Secretariat (BITS): [OASIS Cover page] • Financial Services Technology Consortium (FSTC): Bank Internet Payment System (BIPS)[OASIS Cover page] • Open Applications Group, Inc (OAG)
Electronic Data Interchange	• Data Interchange Standards Association (DISA): [OASIS Cover page] • EEMA EDI/EC Work Group[OASIS Cover page] • European Committee for Standardization/Information Society Standardization System (CEN/ISSS; The European XML/EDI Pilot Project)[OASIS Cover page] • XML/EDI Group[OASIS Cover page]
Human Resources	• DataMain: Human Resources Markup Language (hrml) • HR-XML Consortium[OASIS Cover page]: JobPosting, CandidateProfile, Resume • Open Applications Group (OAG): Open Applications Group Interface Specification (OASIS)[OASIS Cover page] • Tapestry.Net: JOB Markup Language (JOB) • Open Applications Group, Inc (OAG)
Insurance	• ACORD: Property and Casualty[OASIS Cover page], Life (XMLife)[OASIS Cover page] • Lexica: iLingo



Industry Type	Example Standards
Real Estate	• OpenMLS: Real Estate Listing Management System (OpenMLS)[OASIS Cover page] • Real Estate Transaction Standard working group (RETS): Real Estate Transaction Standard (RETS)[OASIS Cover page]
Software	• IBM: [OASIS Cover page] • Flashline.com: Software Component Documentation DTD • Flashline.com: • INRIA: Koala Bean Markup Language (KBML)[OASIS Cover page] • Marimba and Microsoft: Open Software Description Format (OSD)[OASIS Cover page] • Object Management Group (OMG): [OASIS Cover page]
Workflow	• Internet Engineering Task Force (IETF): Simple Workflow Access Protocol (SWAP)[OASIS Cover page] • Workflow Management Coalition (MfMC): Wf-XML[OASIS Cover page]

Basic XML Family Standards

- XML Schema: an XML-compliant language for defining the structure of an XML document.

- XSL (Extensible Stylesheet Language), including
  - XSLT (XSL Transformation): an XML language for transforming XML documents between different schemas.
  - XPath: a non-XML language used by XSLT, and other non-XSLT contexts, for addressing the parts of an XML document.
  - XSL-FO (XSL Formatting Objects): an XML language for specifying the visual formatting of an XML document.

- XPointer
  - A standard for linking one document to another.

- XML Namespaces: A standard for allocating terminology to defined collections and revolving naming conflicts.

XML Tools and Editors

- Notepad
  - Most primitive but it works!
- XMLPad
  - a pretty powerful XML/XSD/XSLT editor; providing multiple views for XML content.
  - http://www.wmhelp.com/xmlpad3.htm
- Visual Studio 2010
  - Great to visualize XML schema
- Internet Explorer
  - Convenient to view XML files
- More tools:
  - http://cubicle-h.blogspot.com/2009/09/free-xmlxslt-tools.html

XML and Database

- Can relational database and hierarchical XML work together?
  - XML provides a standardized yet customizable way to describe the content of documents.
  - Database is a matured, sophisticated, and commonly accepted technology.

- Three basic strategies for XML and relational database
  - Native XML database
  - Transformation to/from relational data
  - XML enabled database (hybrid)

Native XML Database

- Use XML format as the fundamental storage unit (logical level)
  - Can be implemented on any physical level models

- Use XML specific query and procedure languages
  - XPath
  - XQuery

XML Enabled Database

- Many relational DBMS add capabilities to store and process XML data
  - Defining XML data type and storing XML data just like text, number, date, etc.
  - Generating XML data/document from tables and queries
  - Querying XML data/document using XML specific query methods
  - Validating XML data/document

Transformation to/from Relational Data

- Store data in relational databases.
- Transform into XML format to external programs
  - XML documents can automatically be generated from database data, and vice versa.
- Load external XML data into the database, and transform it to relational data.

Relational Data and XML

- XML Document vs. XML Data
  - Document centric XML file
    - Focus on content
    - Fewer tags, less structured
  - Data centric XML file
    - Focus on data and structure
    - More tags, more structured

- Relational data (table) can be transformed to XML format (data centric XML file)

Simple Relation-to-XML Guideline

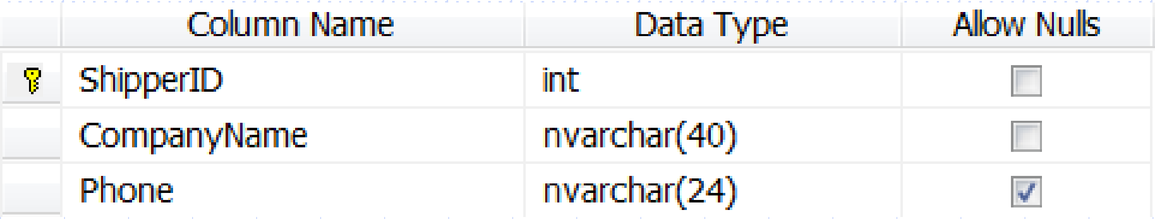
- The table becomes the root element (a complex type): may use the table name as the root element name.

- Each row (record) becomes direct child elements (complex types) under the root element.

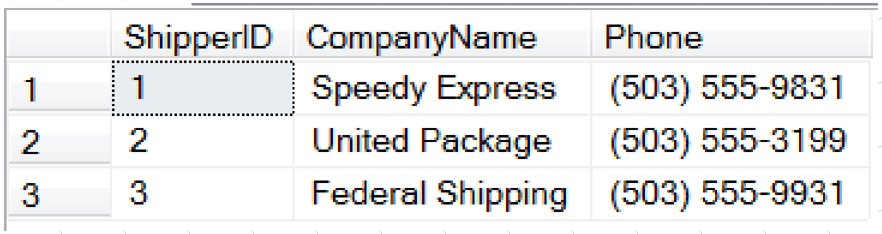
- Each value in the row becomes (two choices)
  - an attribute of the row element (the column name becomes the attribute name, and the data becomes the attribute value), or
  - an third level child element (simple type) under the row element: the column name becomes the element name and the data becomes the text node under the element.

Example: Shippers Table

- Transforming a single table
  - The “Shippers” table in the “Northwind” database.



	Column Name	Data Type	Allow Nulls
🔑	ShipperID	int	□
	CompanyName	nvarchar(40)	□
	Phone	nvarchar(24)	☑



	ShipperID	CompanyName	Phone
1	1	Speedy Express	(503) 555-9831
2	2	United Package	(503) 555-3199
3	3	Federal Shipping	(503) 555-9931

XML Data File 1

```
<?xml version="1.0" encoding="utf-8"?>
<Shippers>
  <Shipper ShipperID="1" CompanyName="Speedy Express"
   Phone="(503)555-9831" />
  <Shipper ShipperID="2" CompanyName="United Package"
   Phone="(503)555-3199" />
  <Shipper ShipperID="3" CompanyName="Federal Shipping"
   Phone="(503)555-9931" />
</Shippers>
```

The root element has the table name as the element name. It is a complex type.

Each row (record) becomes a direct child element under the root element. There are 3 records hence 3 “Shipper” elements.

Values of the row (record) become attributes of the row element: the column name becomes the attribute name. The data becomes the attribute value.

XML Data File 2

```
<?xml version="1.0" encoding="utf-8" ?>
<Shippers>
    <Shipper>
        <ShipperID>1</ShipperID>
        <CompanyName>Speedy Express</CompanyName>
        <Phone>(503) 555-9831</Phone>
    </Shipper>
    <Shipper>
        <ShipperID>2</ShipperID>
        <CompanyName>United Package</CompanyName>
        <Phone>(503) 555-3199</Phone>
    </Shipper>
    <Shipper>
        <ShipperID>3</ShipperID>
        <CompanyName>Federal Shipping</CompanyName>
        <Phone>(503) 555-9931</Phone>
    </Shipper>
</Shippers>
```

The root element has the table name as the element name.

Each row (record) becomes a direct child element under the root element. There are three occurrences.

Third level child elements under the row element: the column name becomes the element name; the data becomes the text node.

Generating XML from SELECT

- SQL Server 2008
  - Directly format data into XML format using the “FOR XML” clause in SQL SELECT queries

- Example

```
SELECT * from Shippers FOR XML AUTO
```

Summary

- Key concepts
  - XML
  - Tree (hierarchical) structure
  - Markup, Tag
  - XML Element, Node, Attribute
  - XML and relational database: differences and how they can work together
- Key skills
  - Use XML format to represent hierarchical data and relational data.