

XML & DTD

CPS 196.3
Introduction to Database Systems

From HTML to XML (eXtensible Markup Language)

2

- ❖ HTML describes the presentation of the content

```
<h1>Bibliography</h1>  
<p><i>Foundations of Databases</i>  
Abiteboul, Hull, and Vianu  
<br>Addison Wesley, 1995...
```



- ❖ XML describes only the content

```
<bibliography>  
  <book>  
    <title>Foundations of Databases</title>  
    <author>Abiteboul</author>  
    <author>Hull</author>  
    <author>Vianu</author>  
    <publisher>Addison Wesley</publisher>  
    <year>1995</year>  
  </book>  
</bibliography>
```

- ☞ Separation of content from presentation simplifies content extraction and allows the same content to be presented easily in different looks

Other nice features of XML

3

- ❖ Portability: Just like HTML, you can ship XML data across any platforms
 - Relational data requires heavy-weight protocols, e.g., JDBC
- ❖ Flexibility: You can represent any information (structured, semi-structured, documents, ...)
 - Relational data is best suited for structured data
- ❖ Extensibility: Since data describes itself, you can change the schema easily
 - Relational schema is rigid and difficult to change

XML terminology

4

- ❖ Tag names: `book`, `title`, ...
- ❖ Start tags: `<book>`, `<title>`, ...
- ❖ End tags: `</book>`, `</title>`, ...
- ❖ An element is enclosed by a pair of start and end tags: `<book>...</book>`
 - Elements can be nested:
`<book>...<title>...</title>...</book>`
 - Empty elements: `<is_textbook></is_textbook>`
 - Can be abbreviated: `<is_textbook/>`
- ❖ Elements can also have attributes: `<book ISBN="..." price="80.00">`

```
<bibliography>
<book ISBN="ISBN-10" price="80.00">
  <title>Foundations of Databases</title>
  <is_textbook/>
  <author>Abiteboul</author>
  <author>Hull</author>
  <author>Korth</author>
  <publisher>Addison Wesley</publisher>
  <year>1995</year>
</book>
</bibliography>
```

Well-formed XML documents

5

A well-formed XML document

- ❖ Follows XML lexical conventions
 - Wrong: `<section>We show that $x < 0$...</section>`
 - Right: `<section>We show that $x \leq 0$...</section>`
 - Other special entities: `>` becomes `>`; and `&` becomes `&`;
- ❖ Contains a single root element
- ❖ Has tags that are properly matched and elements that are properly nested
 - Right:
`<section>...<subsection>...</subsection>...</section>`
 - Wrong:
`<section>...<subsection>...</section>...</subsection>`

More XML features

6

- ❖ Comments: `<!-- Comments here -->`
- ❖ CDATA: `<![CDATA[Tags: <book>,...]]>`
- ❖ ID's and references

```
<person id="o12"><name>Homer</name>...</person>
<person id="o34"><name>Marge</name>...</person>
<person id="o56" father="o12" mother="o34"><name>Bart</name>...</person>
```
- ❖ Namespaces allow external schemas and qualified names

```
<book xmlns:myCitationStyle="http://.../mySchema">
  <myCitationStyle:title>...</myCitationStyle:title>
  <myCitationStyle:author>...</myCitationStyle:author>...
</book>
```
- ❖ Processing instructions for apps: `<? ...java applet... ?>`
- ❖ And more...

Valid XML documents

7

- ❖ A valid XML document conforms to a Document Type Definition (DTD)
 - A DTD is optional
- ❖ A DTD specifies
 - A grammar for the document
 - Constraints on structures and values of elements, attributes, etc.
- ❖ Example

```
<!DOCTYPE bibliography [  
  <!ELEMENT bibliography (book+)>  
  <!ELEMENT book (title, author*, publisher?, year?, section*)>  
  <!ATTLIST book ISBN CDATA #REQUIRED>  
  <!ATTLIST book price CDATA #IMPLIED>  
  <!ELEMENT title (#PCDATA)>  
  <!ELEMENT author (#PCDATA)>  
  <!ELEMENT publisher (#PCDATA)>  
  <!ELEMENT year (#PCDATA)>  
  <!ELEMENT section (title, (#PCDATA)?, section*)>  
>
```

DTD explained

8

```
<!DOCTYPE bibliography [  
  ↳ bibliography is the root element of the document  
  <!ELEMENT bibliography (book+)> ↳ One or more  
  ↳ bibliography consists of a sequence of one or more book elements  
  <!ELEMENT book (title, author*, publisher?, year?, section*)> ↳ Zero or one  
  ↳ book consists of a title, zero or more authors, an optional publisher, and zero or more sections, in sequence  
  <!ATTLIST book ISBN ID #REQUIRED> ↳ book has a required ISBN attribute which is a unique identifier  
  <!ATTLIST book price CDATA #IMPLIED> ↳ book has an optional (#IMPLIED) price attribute which contains character data  
  <bibliography>  
    <book ISBN="ISBN-10" price="90.00">  
      <title>Foundations of Databases</title>  
      <author>Abiteboul</author>  
      <author>Null</author>  
      <author>Vianu</author>  
      <publisher>Addison Wesley</publisher>  
      <year>1995</year>  
    </book>  
  </bibliography>
```

Other attribute types include IDREF (reference to an ID), IDREFS (space-separated list of references), enumerated list, etc.

DTD explained (cont'd)

9

```
<!ELEMENT title (#PCDATA)> PCDATA is text that will be parsed (<...> will be treated as a markup tag and &lt; etc. will be treated as entities);  
<!ELEMENT author (#PCDATA)> CDATA is unparsed character data  
<!ELEMENT publisher (#PCDATA)>  
<!ELEMENT year (#PCDATA)>  
  ↳ title, author, publisher, and year all contains parsed character data (#PCDATA)  
  
<!ELEMENT section (title, (#PCDATA)?, section*)>  
  ↳ Each section starts with a title, followed by some optional text and then zero or more subsections  
  
>
```

```
<section><title>Introduction</title>  
  In this section we introduce XML and DTD.  
</section>  
<section><title>XML</title>  
  XML stands for...  
</section>  
<section><title>DTD</title>  
  <section><title>Definition</title>  
  DTD stands for...  
</section>  
<section><title>Usage</title>  
  You can use DTD to...  
</section>  
</section>
```

Using DTD

10

❖ DTD can be included in the XML source file

```
<?xml version="1.0"?>
<!DOCTYPE bibliography [
  --
]>
<bibliography>
  --
</bibliography>
```

❖ DTD can be external

```
<?xml version="1.0"?>
<!DOCTYPE bibliography SYSTEM "../dtds/bib.dtd">
<bibliography>
  --
</bibliography>
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  --
</html>
```

Why use DTD's?

11

❖ Benefits of using DTD

❖ Benefits of not using DTD

XML versus relational data

12

Relational data

- ❖ Schema is always fixed in advance and difficult to change
- ❖ Simple, flat table structures
- ❖ Ordering of rows and columns is unimportant
- ❖ Data exchange is problematic
- ❖ "Native" support in all serious commercial DBMS

XML data

- ❖
- ❖
- ❖
- ❖
- ❖
