XML & DTD

CPS 196.3
Introduction to Database Systems

From HTML to XML (eXtensible Markup Language)

- HTML describes the presentation of the content
- XML describes only the content

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

- 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

- 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/>
  - Can be abbreviated: <is_textbook/>
  - Elements can also have attributes: <book ISBN="..." price="80.00"/>

Well-formed XML documents

A well-formed XML document

- Follows XML lexical conventions
  - Wrong: <section> We show that x < 0... </section>
  - Right: <section> We show that x &lt; 0... </section>
  - Other special entities: > becomes &gt; and & becomes &amp;
- 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>... </subsection> </section>

More XML features

- Comments: <!-- Comments here -->
- CDATA: <![CDATA[Tags: <book>...]]>
- ID’s and references
  - 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

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:

```xml
<!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

```xml
<!DOCTYPE bibliography [
  <!ELEMENT bibliography (book+)>
  <!ELEMENT book (title, author*, publisher?, year?, section*)>
  <!ATTLIST book ISBN ID #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 (cont’d)

```xml
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT section (title, (#PCDATA)?, section*)>
```