Introduction

- Database processing and document processing need each other.
  - Database processing needs document processing for transmitting database views.
  - Document processing needs database processing for storing and manipulating data.
- As Internet usage increases, organizations want to make their Web pages more functional by displaying and updating data from organizational databases.

XML

- XML, or Extensible Markup Language, was developed in early 1990s:
  - XML is a subset of SGML, or Standard Generalized Markup Language.
- Today XML is a hybrid of document processing and database processing.
  - It provides a standardized yet customizable way to describe the content of documents.
  - XML documents can automatically be generated from database data, and vice versa.
- SOAP (which formerly stood for Simple Object Access Protocol but is now just a name instead of an acronym) is an XML-based standard protocol for sending messages of any type, using any protocol over the Internet.
XML (Continued)

- XML is used for describing, representing, and materializing database views.
- XML is better than HTML because:
  - It provides a clear separation between document structure, content, and materialization.
  - It is standardized but allows for extension by developers.
  - XML tags accurately represent the semantics of their data.
- **Document Type Declarations (DTDs)** and **XML Schemas** can be used to describe the content of XML documents.
- Both Oracle and SQL Server can produce XML documents from database data.

**Document Type Declarations (DTDs) and XML Schemas** can be used to describe the content of XML documents.

XML DTD

- An XML document consists of two sections:
  - **Document Type Declaration (DTD).**
    - The DTD begins with `DOCTYPE <document_type_name>`
  - **Document data.**
- An XML document can be:
  - **Type-valid** if the document conforms to its DTD.
  - Well-formed and not type-valid, if
    - It violates the structure of its DTD, or
    - It has no DTD.
- A DTD may be stored externally so many documents can be validated against the same DTD.

**XMLSpy**

- The diagrams in the text and in the following slides were prepared with Altova’s **XMLSpy**.
- There is a free Home version of XMLSpy 2005 available at: [http://www.altova.com/support_freexmlspyhome.asp](http://www.altova.com/support_freexmlspyhome.asp)
XSLT

- XSLT, or the Extensible Style Language may be used to materialize (transform) XML documents using XSL document.
  - From XML documents into HTML or into XML in another format.
- XSLT is a declarative transformation language:
  - Declarative: create rules, not procedures, to materialize the document.
  - Transformational: transforms the input document into another document.
- XSLT uses stylesheets to indicate how to transform the elements of the XML documents into another format.
External DTD for CustomerList

```xml
<!DOCTYPE CustomerList SYSTEM "CustomerList.dtd" Chapter 13>
<?xml-stylesheet type="text/xsl" href="CustomerList.xsl"?>

<CustomerList>
  <Customer name="John Smith" address="#123 Main St., USA"
    city="New York" state="NY" zip="10001"/>
  <Customer name="Jane Doe" address="#456 Park Ave., USA"
    city="Los Angeles" state="CA" zip="90210"/>
</CustomerList>
```

XML Document with Two Customers

```xml
<!DOCTYPE CustomerList SYSTEM "CustomerList.dtd" Chapter 13>
<?xml-stylesheet type="text/xsl" href="CustomerList.xsl"?>

<CustomerList>
  <Customer name="John Smith" address="#123 Main St., USA"
    city="New York" state="NY" zip="10001"/>
  <Customer name="Jane Doe" address="#456 Park Ave., USA"
    city="Los Angeles" state="CA" zip="90210"/>
</CustomerList>
```

Example XSL Stylesheet

```xml
<?xml-stylesheet type="text/xsl" href="#customerlist.xsl"?>
```

Example HTML in Browser

David M. Kroenke's
Database Processing
Fundamentals, Design, and Implementation
(10th Edition)

End of Presentation:
Chapter Thirteen Part One