Semi Structured Data

Semi-structured Data Explained

- Missing attributes:

```
<person>   <name> Ali</name>
           <phone>1234</phone>
 </person>

<person>  <name>Jamal</name>
</person>
```

← no phone !

- Could represent in a table with nulls

name	phone
Ali	1234
Jamal	-

Semi-structured Data Explained

- Repeated attributes

```
<person> <name> Mariam</name>
         <phone>2345</phone>
         <phone>3456</phone>
</person>
```

← two phones !

- Impossible in tables:

name	phone	
Mariam	2345	3456

???

Semistructured Data Explained

- Attributes with different types in different objects

```
<person> <name>  <first> Ali </first>
                  <last> Salem </last>
         </name>
         <phone>1234</phone>
</person>
```

← structured name !

- Nested collections (no 1NF)
- Heterogeneous collections:
  - <db> contains both <book>s and <publisher>s

XML

- eXtensible Markup Language
- XML 1.0 – a recommendation from W3C, 1998
- Roots: SGML (Standard Generalized Markup Language). SGML is both a language and an ISO standard for describing information embedded within a document.
  - HyperText Markup Language (HTML) is based on the SGML standard. (used in publishing).
- After the roots: a format for sharing *data*

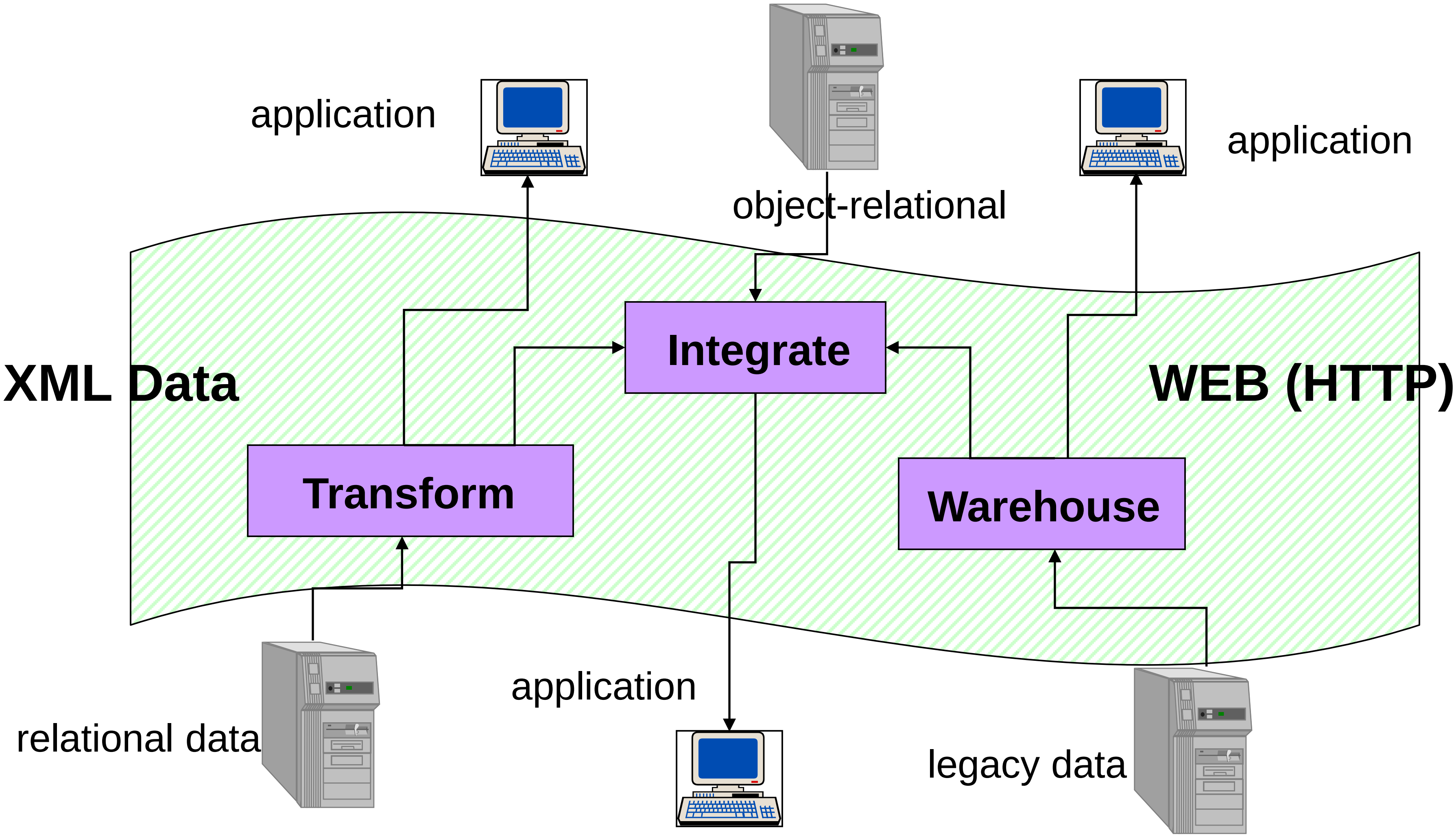
XML Data

- XML is self-describing
- Schema elements become part of the data
  - Relational schema: persons(name,phone)
  - In XML <persons>, <name>, <phone> are part of the data, and are repeated many times
- Consequence: XML is much more flexible
- XML = semistructured data

XML Data

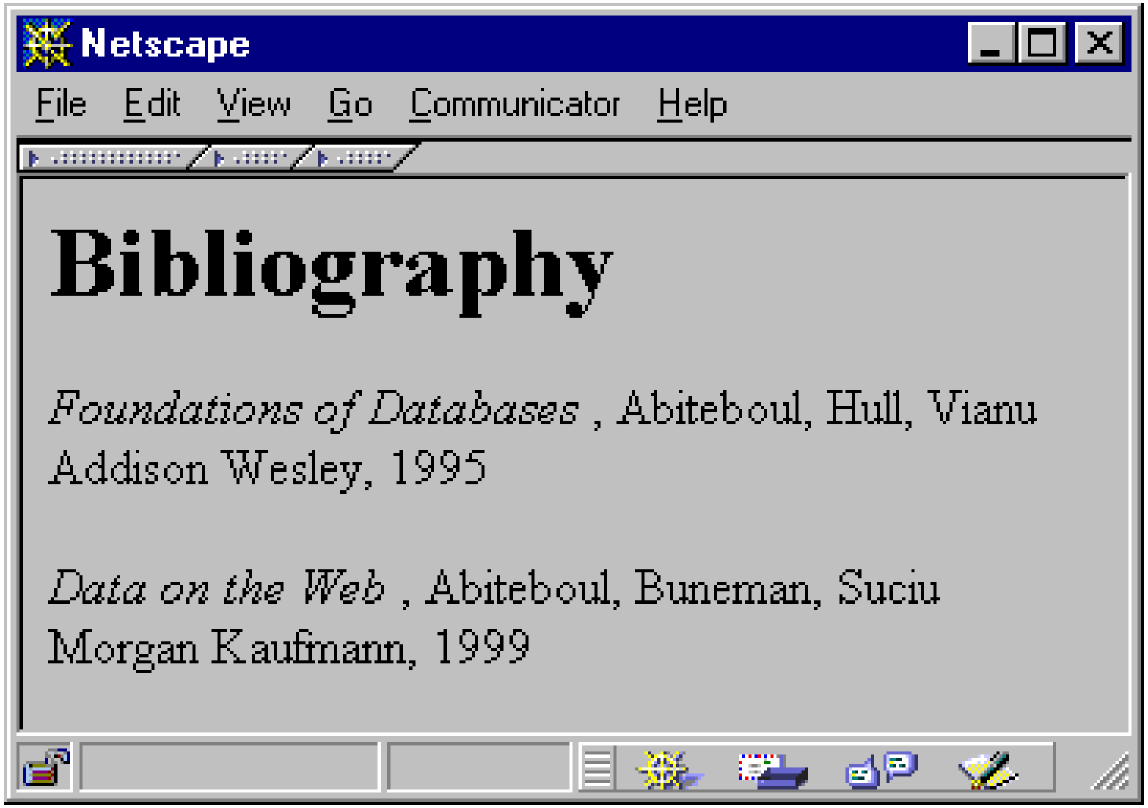
- Relational data does not have a syntax
  - I can’t “give” you my relational database
  - Need to import it from other syntax, like CSV (comma-separated-values)
- XML = rich syntax for data
  - But XML is not relational: *semistructured*
- Usage:
  - Map any data to XML
  - Store it in files, exchange on the Web, etc.
  - Even query it directly, using XPath, XQuery

XML Data Sharing and Exchange



Specific data management tasks

From HTML to XML



Bibliography

Foundations of Databases , Abiteboul, Hull, Vianu
Addison Wesley, 1995

Data on the Web , Abiteboul, Buneman, Suciu
Morgan Kaufmann, 1999

HTML describes the layout

HTML

```
<h1> Bibliography </h1>
<p> <i> Foundations of Databases </i>
        Abiteboul, Hull, Vianu
        <br> Addison Wesley, 1995
<p> <i> Data on the Web </i>
        Abiteoul, Buneman, Suciu
        <br> Morgan Kaufmann, 1999
```

XML

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

XML describes the structure

XML Terminology

- tags: book, title, author, …
- start tag: <book>, end tag: </book>
- elements: <book>…</book>,<author>…</author>
- elements are nested
- empty element: <red></red> abbrv. <red/>

well formed XML document

- if it has matching tags
- tags are properly nested
- single root element
- and more constraints, e.g. on names

More XML: Attributes

```
<book price = “55” currency = “USD”>
   <title> Foundations of Databases </title>
   <author> Abiteboul </author>
    …
   <year> 1995 </year>
</book>
```

attributes are alternative ways to represent data

More XML: IDs and References

```
<person id=“o555”>  <name> Ali </name> </person>


<person id=“o456”>  <name> Mariam </name>
                   <children idref=“o123 o555”/>
</person>


<person id=“o123” mother=“o456”><name>Fatima</name>
</person>
```

Scope of IDs and references is the document

More XML: CDATA Section

- Syntax: <![CDATA[.....any text here...]]>
- CDATA section instructs the parser to ignore most markup characters.
- Example:

```
<example>
   <![CDATA[ some text here </notAtag> <>]]>
</example>
```

More XML: Entity References

- Syntax: &entityname;
- Used like macros
- Example:
  <element>
  this is less than &lt;
  </element>

&lt;	<
&gt;	>
&amp;	&
&apos;	‘
&quot;	“
&#38;	Unicode char

some predefined entities

complete list: http://www.w3.org/TR/xhtml-modularization/dtd_module_defs.html

More XML: Processing Instructions

- Syntax: <?target argument?>
- Example:

```
<product> <name> Alarm Clock </name>
          <?ringBell 20?>
          <price> 19.99 </price>
</product>
```

- Processed by external applications, e.g. php (bad style)

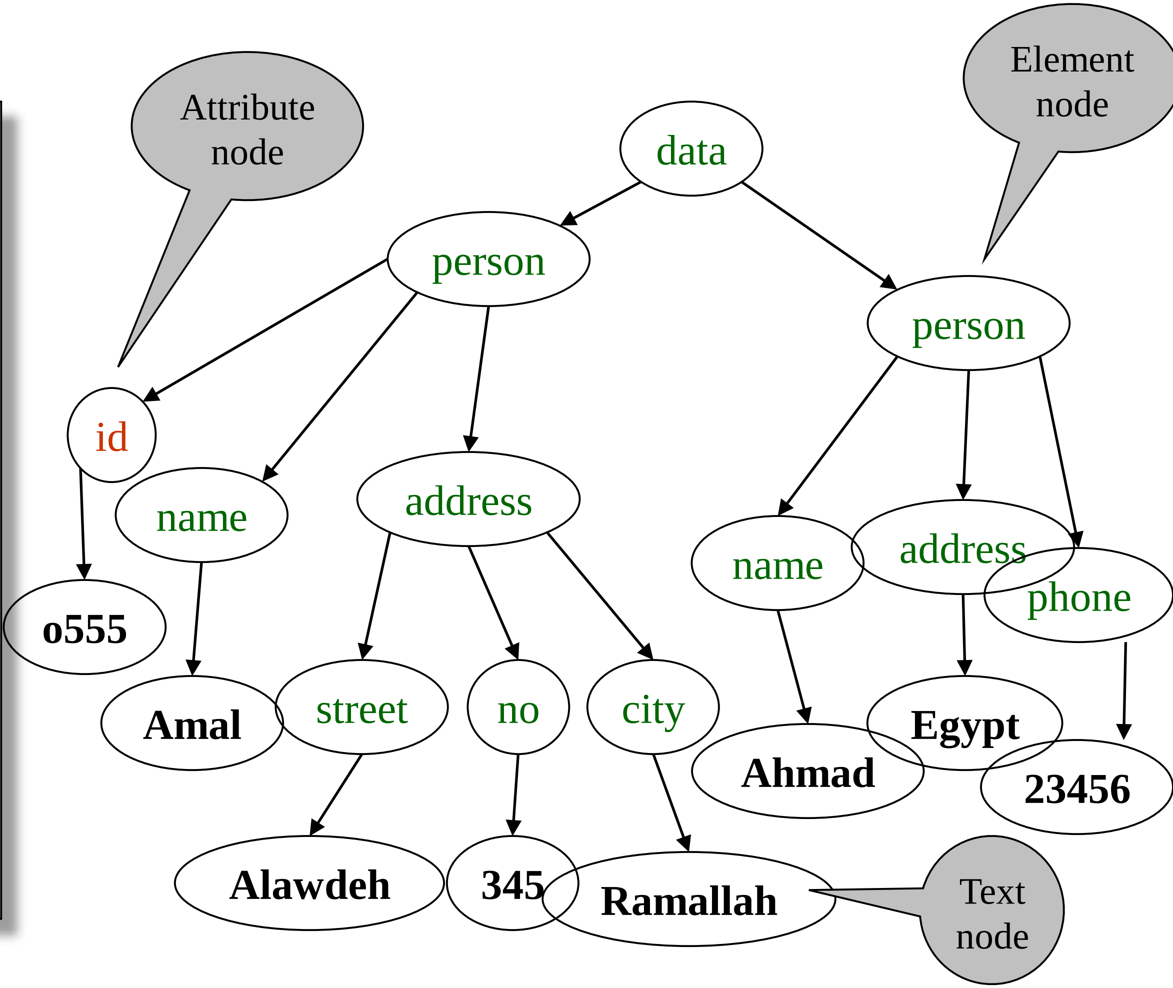
More XML: Comments

- Syntax <!-- .... Comment text... -->

- Yes, they are part of the data model !!!

XML Data: a Tree !

```
<data>
        <person id="o555" >
                <name> Amal </name>
                <address>
                        <street> Alawdeh </street>
                        <no> 345 </no>
                        <city> Ramallah </city>
                </address>
        </person>
        <person>
                <name> Ahmad </name>
                <address> Egypt  </address>
                <phone> 23456  </phone>
        </person>
</data>
```



Order matters !!!

Jason Formatter

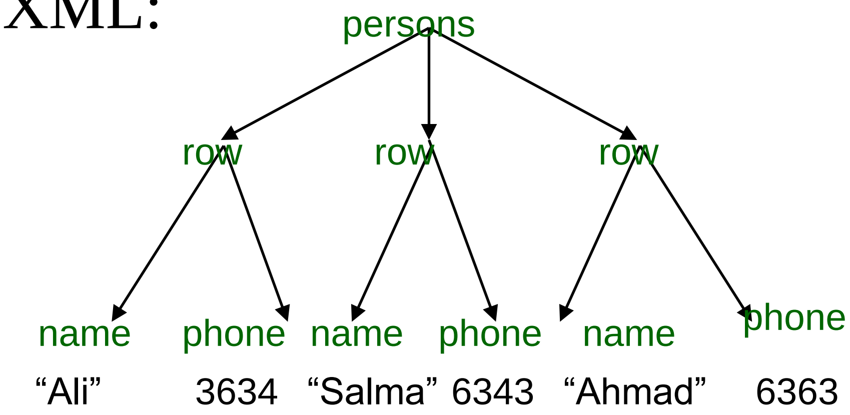
- https://jsonformatter.org/xml-viewer

From Relational Data to XML Data

persons

name	phone
Ali	3634
Salma	6343
Ahmad	6363

XML:



```
<persons>
      <row> <name>Ali</name>
                 <phone> Ahmad</phone>
      </row>
       <row> <name>Salma</name>
                  <phone> 6343</phone>
       <row> <name>Ahmad</name>
                  <phone> 6363</phone>
      </row>
</persons>
```

Additional Readings on XML

- XML
  - http://www.w3.org/XML/1999/XML-in-10-points
  - www.zvon.org/xxl/XMLTutorial/General/book_en.html
  - http://db.bell-labs.com/galax/
  - http://www.w3.org/TR/REC-xml-names
- Xpath
  - http://java.sun.com/webservices/docs/ea2/tutorial/doc/JAXPXSLT2.html
- Xquery
  - http://www.w3.org/TR/xmlquery-use-cases/
  - http://www.xmlportfolio.com/xquery.html
- Main source: www.w3.org (hard to read !!!!)