Announcements

- Quiz canceled for this week
Announcements

• Quick Write this week on Privacy
  * (First half of chapter 13 in Fluency)
A Table with a View

Data Storage and Transfer with XML and Databases
Differences Between Tables and Databases

• When we think of databases, we often think of tables of information

• Comparing Tables
  * Database tables
    • Metadata tag identifying each of the data fields
  * Spreadsheet tables
    • Rely on position to keep the integrity of their data
  * HTML tables
    • Data as table entries with no unique identity at all
    • Concerned only with how to display the data, not with its meaning
The Database Advantage

- Metadata is key advantage of databases over other systems recording data as tables.
- Two of the most important roles in defining metadata:
  - Identify the type of data with a unique tag
  - Define the relationships of the data
XML: A Language for Metadata Tags

• Extensible Markup Language
  * Tagging scheme similar to XHTML
  * No standard tags to learn
    • Self-describing, think up the tags you need
  * Works well with browsers and Web-based applications
  * Use a simple text editor
  * XML tag names cannot contain spaces
Extensible Markup Language

XML
An Example from Tahiti

- Area in km² for Tahiti & neighboring islands

```xml
<?xml version = "1.0" encoding="ISO-8859-1" ?>
<archipelago>
  <island><iName>Tahiti</iName>       <area>1048</area></island>
  <island><iName>Moorea</iName>       <area>130</area></island>
  <island><iName>Maiao</iName>        <area>9.5</area></island>
  <island><iName>Mehetia</iName>      <area>2.3</area></island>
  <island><iName>Tetiaroa</iName>     <area>12.8</area></island>
</archipelago>
```

**Figure 16.1** XML file encoding data for the Windward Islands database. The first line states that the file contains XML tags.
An Example from Tahiti (cont'd)

- First line
  ```xml
  <?xml version="1.0" encoding="ISO-8859-1" ?>
  ```
- File should be ASCII text
- File extension should be `.xml`
### Table 16.1 Rules for writing XML.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Rule Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required first line</td>
<td><code>&lt;xml version=&quot;1.0&quot; encoding=&quot;ISO-8859-1&quot;?&gt;</code> must appear on the first line, starting in the first position.</td>
</tr>
<tr>
<td>First tag</td>
<td>The first tag encountered is the <em>root</em> element, and it must enclose all of the file’s content; it appears on the second or possibly third line.</td>
</tr>
<tr>
<td>Closing tags</td>
<td>All tags must be closed.</td>
</tr>
</tbody>
</table>
| Element naming                     | Observe these rules:  
|                                   | • Names can contain letters, numbers, and underscore characters.  
|                                   | • Names must not start with a number or punctuation character.  
|                                   | • Names must not start with the letters xml (or XML, or Xml, etc.).  
|                                   | • Names cannot contain spaces.                                                                                                                   |
| Case sensitivity                   | Tags and attributes are case sensitive.                                                                                                          |
| Proper nesting                     | All tags must be well-nested.                                                                                                                      |
| Attribute quoting                  | All attribute values must be quoted; paired single quotes (apostrophes) or paired double quotes are okay; use "dumb" quotes only; choose ‘opposite’ quotes to enclose quoted values. |
| White space                        | White space is preserved and converted to a single space.                                                                                         |
| Comments                           | XML comments have the form `<!-- This is a comment. -->`.                                                                                         |
Expanding Use of XML

- Combine encodings of two archipelagos – the Windward and the Galapagos Islands
- Root element is the tag that encloses all of the content of the XML file
  * `<archipelago>` in Fig. 16.1
  * `<geo_feature>` in Fig. 16.2
- Indenting for readability and structure
<?xml version = "1.0" encoding="ISO-8859-1"?>
<geo_feature>
  <archipelago>
    <a_name>Galapagos Islands</a_name>
    <island>
      <iName>Isabella</iName>
      <area>4588</area>
      <elevation>1707</elevation>
    </island>
    <island>
      <iName>Fernandina</iName>
      <area>642</area>
      <elevation>1494</elevation>
    </island>
    <island>
      <iName>Tower</iName>
      <area>14</area>
      <elevation>76</elevation>
    </island>
    <island>
      <iName>Santa Cruz</iName>
      <area>986</area>
      <elevation>846</elevation>
    </island>
    <island>
      <iName>Tetiaroa</iName>
      <area>12.8</area>
    </island>
  </archipelago>
</geo_feature>

Figure 16.2 XML file for the Geographic Features database. XML ignores white space, so the text in the file has been indented for easier reading.
Attributes in XML

- Use attributes for additional metadata, not for additional content
  - * Not good, name is content:
    ```xml
    <archipelago name="Galapagos">
    ```
  - * Better to give alternate form of the data
    ```xml
    <a_name accents="Gal&acute;pagos">Galapagos</a_name>
    ```
Effective Design with XML Tags

• Identification Rule: Label Data with Tags Consistently
  * You can choose whatever tag names you wish to name data, but once you've decided on a tag for a particular kind of data, you must always surround it with that tag.
• Affinity Rule: Group Related Data
  
  * Enclose in a pair of tags all tagged data referring to the same entity. Grouping it keeps it all together, but the idea is much more fundamental: Grouping makes an association of the tagged data items as being related to each other, properties of the same thing.
  
  * Groups together data for a single thing – an island
    • Association is among properties of an object
• Collection Rule: Group Related Instances
  * When you have several instances of the same kind of data, enclose them in tags; again, it keeps them together and implies that they are related by being instances of the same type.
  * Groups together data of several instances of the same thing – islands
    • Association is among the objects themselves (entities)
The XML Tree

- XML encodings of information produce hierarchical descriptions that can be thought of as trees
  * Hierarchy a consequence of how tags enclose one another and the data
Figure 16.3 The XML displayed as a tree. The encoding from Figure 16.2 is shown with the root element (geo_feature) to the left and the leaves (content) shown to the right.
DATABASES
What is a Database

• Any organized collection of data
• A collection of similar data
• Examples of databases:
  * Telephone book white pages
  * T.V. Guide
  * Airline reservation system
  * Motor vehicle registration records
  * Papers in your filing cabinet
  * Files on your computer hard drive
Data | Information | Knowledge

- **Data**
  - Can be defined in many ways
  - IS defines data as unprocessed information

- **Information**
  - Data that have been organized and communicated in a coherent and meaningful manner

- **Knowledge**
  - Knowledge—information evaluated and organized so that it can be used purposefully

Data is converted into information, and information is converted into knowledge
Data vs. Information

• We collect data
• Information is harvested from data

• Many companies are good at collecting data
• Fewer are good at harvesting information
Ultimate Purpose of a Database Management System (DBMS)

To transform

Data → Information → Knowledge → Action
Why do we need a database?

- Keep records of our:
  * Clients
  * Staff
  * Volunteers
- To keep a record of activities and interventions
- Keep sales records
- Develop reports
- Perform research
### Database Terminology

#### Phone book:

<table>
<thead>
<tr>
<th>Records (rows)</th>
<th>Fields (columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anderson Thomas A 123 Marine View Dr. 237-1234</td>
</tr>
<tr>
<td></td>
<td>Benson Karen C 1300 California Ave 237-1098</td>
</tr>
<tr>
<td></td>
<td>Casserly Rick W 12492 Rd 19 342-0502</td>
</tr>
<tr>
<td></td>
<td>Drummond Lynn M 12059 30th Ave W 931-1105</td>
</tr>
</tbody>
</table>

#### Table

| Field (the columns in a table) | · Smallest unit of information in a table  
|                               | · Sometime called “attributes”  
|                               | · First name  
|                               | · Last name  
|                               | · Middle initial  
|                               | · Street address  
|                               | · Phone number(s) |
| Record (the rows in a table)   | · All related fields are collectively called a record  
|                               | · All fields for one person are a record |
| Table                         | · A collection of records is a data table  
|                               | · Collection of everyone’s records |
| Database Management System (DBMS) | · All the related tables, queries, data entry and edit forms, reports, macros and VBA modules constitute a database |
Database Management System (DBMS)

• Software tools for working with data
• Designed to:
  * Store (tables)
  * Organize (sort)
  * Add, modify or delete
  * Ask questions (queries)
  * Produce forms and reports
    • Summarizing
    • Displaying details
• Toolbox is a good analogy
DBMS Examples

- Microsoft Access
- Structured Query Language (SQL)
  - Microsoft SQL Server
  - Oracle
  - MySQL
- FileMaker Pro
- Lotus Notes
- Open Office Base
Flat-File Database vs. Relational Database

- Flat-File Database
  - All relevant data in a single table, or series of unrelated tables
  - Work best for small quantities of data; where viewing and sorting the data in a single list does not create a time-consuming task
  - Typically a person’s first databases
  - Example: Excel spreadsheet or Word data list file

- Relational Database
  - Provide a solution to data entry redundancy problems
  - Linked through common fields (columns) with exactly the same data
  - Tables linked together can be queried as if one table
Flat-File Example

Staff Telephone List

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Emp ID</th>
<th>Dept</th>
<th>Location</th>
<th>Work Phone</th>
<th>M/S</th>
<th>Supervisor Name</th>
<th>Supr Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams</td>
<td>Wes</td>
<td>19589</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-1958</td>
<td>QR-07</td>
<td>Susan Buckle</td>
<td>(206) 221-2241</td>
</tr>
<tr>
<td>Alberts</td>
<td>George</td>
<td>21533</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-2153</td>
<td>QR-35</td>
<td>Marsha Mosley</td>
<td>(206) 221-1975</td>
</tr>
<tr>
<td>Allen</td>
<td>Susan</td>
<td>20256</td>
<td>PROD</td>
<td>Renton</td>
<td>(206) 393-2025</td>
<td>PB-18</td>
<td>Frank Sullivan</td>
<td>(206) 393-1000</td>
</tr>
<tr>
<td>Allert</td>
<td>Maria</td>
<td>10544</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-1054</td>
<td>QR-27</td>
<td>Lynn Jarret</td>
<td>(206) 221-1366</td>
</tr>
<tr>
<td>Andrews</td>
<td>Mike</td>
<td>22113</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-2211</td>
<td>QR-12</td>
<td>Harry Hillis</td>
<td>(206) 221-2179</td>
</tr>
<tr>
<td>Apperly</td>
<td>Ward</td>
<td>12244</td>
<td>PROD</td>
<td>Renton</td>
<td>(206) 393-1224</td>
<td>PB-14</td>
<td>Molly Goldberg</td>
<td>(206) 393-1513</td>
</tr>
<tr>
<td>Asher</td>
<td>Jane</td>
<td>11222</td>
<td>ACCT</td>
<td>Seattle</td>
<td>(206) 221-1122</td>
<td>BX-45</td>
<td>Val Johnson</td>
<td>(206) 221-1958</td>
</tr>
<tr>
<td>Astor</td>
<td>Lawrence</td>
<td>20286</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-2028</td>
<td>QR-10</td>
<td>Peggy Kramer</td>
<td>(206) 221-2083</td>
</tr>
<tr>
<td>Ayres</td>
<td>William</td>
<td>22263</td>
<td>PROD</td>
<td>Seattle</td>
<td>(206) 221-2266</td>
<td>QR-10</td>
<td>P. Kramer</td>
<td>(206) 221-2083</td>
</tr>
<tr>
<td>Baker</td>
<td>Gerald</td>
<td>19042</td>
<td>ACCT</td>
<td>Seattle</td>
<td>(206) 221-1904</td>
<td>BX-45</td>
<td>Valerie Johnson</td>
<td>(206) 221-1958</td>
</tr>
</tbody>
</table>

- Weaknesses common to flat-file systems
  - Duplicate information in the table
  - Inconsistencies in the way Supervisor Names are entered
Relational Database Example
Database Tables
Query from Two Tables
Forms
## Training Records by Employee

<table>
<thead>
<tr>
<th>Emp ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Course Name</th>
<th>Start Date</th>
<th>Grade</th>
<th>Room</th>
<th>Instructor</th>
<th>Dept</th>
<th>Hours</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>12650</td>
<td>Adams</td>
<td>Wei</td>
<td>Microsoft Access, Level 1</td>
<td>2/1/2001</td>
<td>3.4</td>
<td>G210</td>
<td>Term-Urbis</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Access, Level 2</td>
<td>2/1/2001</td>
<td>3.4</td>
<td>G210</td>
<td>Term-Urbis</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Access, Level 3</td>
<td>2/1/2001</td>
<td>3.4</td>
<td>G210</td>
<td>Term-Urbis</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Access, Level 4</td>
<td>2/1/2001</td>
<td>3.4</td>
<td>G210</td>
<td>Term-Urbis</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Access, Level 5</td>
<td>2/1/2001</td>
<td>3.4</td>
<td>G210</td>
<td>Term-Urbis</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td>215311</td>
<td>Alberta</td>
<td>George</td>
<td>Microsoft Windows 95, Level 1</td>
<td>9/2/2001</td>
<td>3.1</td>
<td>R121</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Excel, Level 1</td>
<td>7/1/2002</td>
<td>2.6</td>
<td>G210</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Excel, Level 2</td>
<td>7/1/2002</td>
<td>2.6</td>
<td>G210</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Excel, Level 3</td>
<td>7/1/2002</td>
<td>2.6</td>
<td>G210</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Excel, Level 4</td>
<td>7/1/2002</td>
<td>2.6</td>
<td>G210</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Excel, Level 5</td>
<td>7/1/2002</td>
<td>2.6</td>
<td>G210</td>
<td>Bob Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td>20256</td>
<td>Allen</td>
<td>Susan</td>
<td>Microsoft Word, Level 1</td>
<td>2/2/2000</td>
<td>4.0</td>
<td>R123</td>
<td>Sally Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Word, Level 2</td>
<td>2/2/2000</td>
<td>4.0</td>
<td>R123</td>
<td>Sally Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Word, Level 3</td>
<td>2/2/2000</td>
<td>4.0</td>
<td>R123</td>
<td>Sally Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Word, Level 4</td>
<td>2/2/2000</td>
<td>4.0</td>
<td>R123</td>
<td>Sally Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Word, Level 5</td>
<td>2/2/2000</td>
<td>4.0</td>
<td>R123</td>
<td>Sally Larsen</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td>10541</td>
<td>Allen</td>
<td>Mara</td>
<td>Microsoft Windows 95, Level 1</td>
<td>8/2/2005</td>
<td>3.0</td>
<td>R121</td>
<td>Doug Hidman</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Microsoft Windows 95, Level 2</td>
<td>8/2/2005</td>
<td>3.0</td>
<td>R121</td>
<td>Doug Hidman</td>
<td>PR03</td>
<td>7</td>
<td>$145.00</td>
</tr>
</tbody>
</table>

*Thursday, January 24, 2007*
RELATIONAL DATABASES
Video

- Relational databases and tables
Relational Database Model

• Introduced by E. F. Codd in 1970

• A Logical View of Data
  * Enables developer to view data logically rather than physically
  * Greater logical simplicity tends to yield simpler and more effective database design methodologies
Tables

- Cornerstone of Relational DBMS
- Advantages – structural and data independence
- Conceptually Resembles a file
  * Note a file is actually a physical structure
- Easier to understand than its hierarchical and network database predecessors
Table Characteristics

1. A table is perceived as a two-dimensional structure composed of rows and columns.
2. Each table row (tuple) represents a single entity occurrence within the entity set.
3. Each table column represents an attribute, and each column has a distinct name.
4. Each row/column intersection represents a single data value.
5. All values in a column must conform to the same data format. For example, if the attribute is assigned an integer data format, all values in the column representing that attribute must be integers.
6. Each column has a specific range of values known as the **attribute domain**.
7. The order of the rows and columns is immaterial to the DBMS.
8. Each table must have an attribute or a combination of attributes that uniquely identifies each row.