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

16-2

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

16-3

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

16-4

Extensible **M**arkup **L**anguage

XML

An Example from Tahiti

- Area in km^2 for Tahiti & neighboring islands

```
<?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.

16-6

An Example from Tahiti (cont'd)

- First line
  <?xml version="1.0" encoding="ISO-8859-1" ?>
- File should be ASCII text
- File extension should be `.xml`

16-7

Table 16.1 Rules for writing XML.

Required first line	`<?xml version="1.0" encoding="ISO-8859-1"?>` must appear on the first line, starting in the first position.
First tag	The first tag encountered is the *root* element, and it must enclose all of the file's content; it appears on the second or possibly third line.
Closing tags	All tags must be closed.
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. -->`.

16-8

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

16-9

```
<?xml version = "1.0"
      encoding="ISO-8859-1" ?>
<geo_feature>
   <archipelago>
      <a_name>Windward Islands
      </a_name>
      <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>
   <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>
   </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.

16-10

Attributes in XML

- Use attributes for additional metadata, not for additional content
  - Not good, name is content:
    <archipelago name="Galapagos">
  - Better to give alternate form of the data
    <a_name accents="Galápagos">Galapagos</a_name>

16-11

Effective Design with XML Tags

- Identification Rule: Label Data with Tags Consistently
  - You can choose whatever tag names you with 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.

16-12

Effective Design with XML Tags (cont'd)

- 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

16-13

Effective Design with XML Tags (cont'd)

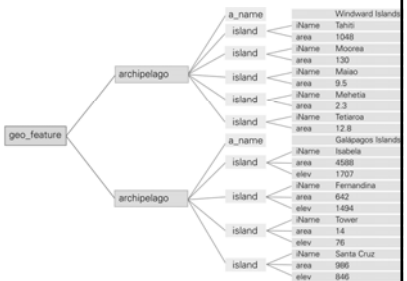
- Collection Rule: Group Related Instances
  - When you have several instances of the same kind of data, enclose them in tags; again, it keeps them together and implies that they are related by being instances of the same type.
  - Groups together data of several instance of the same thing – islands
    - Association is among the objects themselves (entities)

16-14

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

16-15



16-16

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

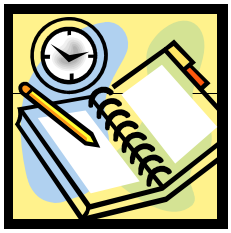
Ultimate Purpose of a Database Management System (DBMS)

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:

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

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 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

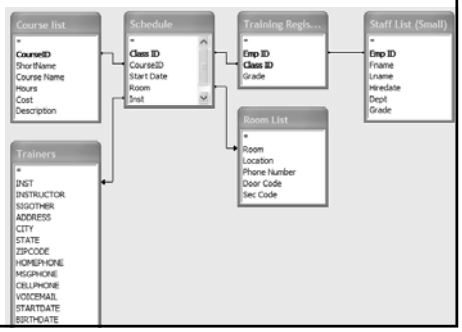
Flat-File Example

Staff Telephone List

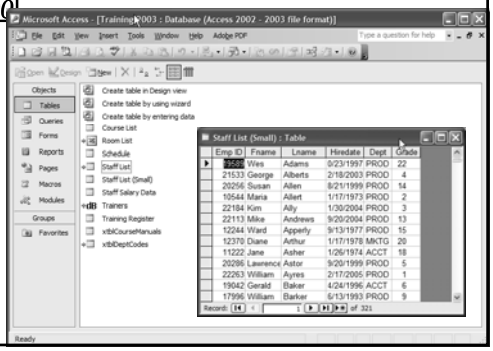
Last Name	First Name	Emp ID	Dept	Location	Work Phone	M/S	Supervisor Name	Supr Phone
Adams	Wes	19589	PROD	Seattle	(206) 221-1958	QR-07	Susan Buckle	(206) 221-2241
Alberts	George	21533	PROD	Seattle	(206) 221-2153	QR-35	Marsha Mosley	(206) 221-1975
Allen	Susan	20256	PROD	Renton	(206) 393-2025	PB-18	Frank Sullivan	(206) 393-1000
Allert	Maria	10544	PROD	Seattle	(206) 221-1054	QR-27	Lynn Jarret	(206) 221-1366
Andrews	Mike	22113	PROD	Seattle	(206) 221-2211	QR-12	Harry Hillis	(206) 221-2179
Apperly	Ward	12244	PROD	Renton	(206) 393-1224	PB-14	Molly Goldberg	(206) 393-1513
Arthur	Diane	12370	MKTG	Bellevue	(206) 882-1237	RL-27	Wes Adams	(206) 221-1958
Asher	Jane	11222	ACCT	Seattle	(206) 221-1122	BX-45	Val Johnson	(206) 221-1958
Astor	Lawrence	20286	PROD	Seattle	(206) 221-2028	QR-10	Peggy Kramer	(206) 221-2083
Ayres	William	22263	PROD	Seattle	(206) 221-2226	QR-10	P. Kramer	(206) 221-2083
Baker	Gerald	19042	ACCT	Seattle	(206) 221-1904	BX-45	Valerie Johnson	(206) 221-1958

- Weaknesses common to flat-file systems

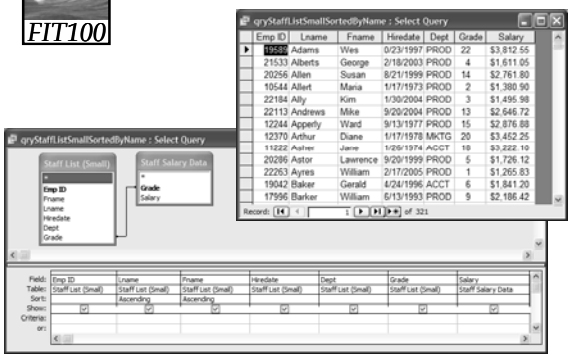
Relational Database Example



Database Tables

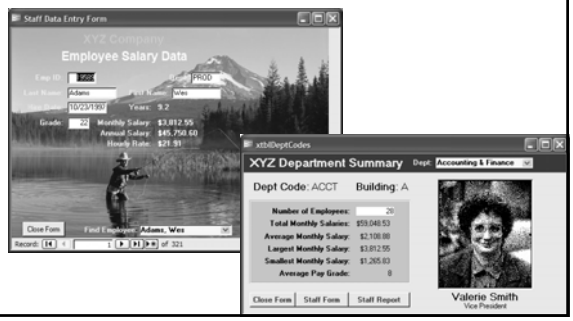


Query from Two Tables



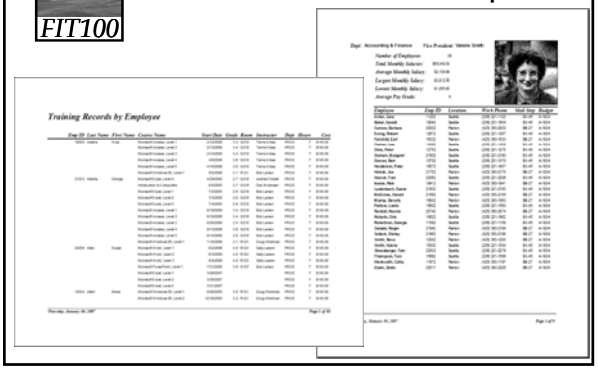
FIT100

Forms



FIT100

Reports



FIT100

RELATIONAL DATABASES

FIT100

Video

- Relational databases and tables

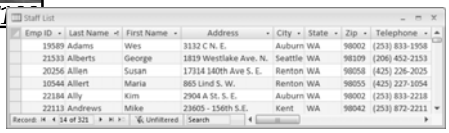
FIT100

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

FIT100

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

Staff List

Emp ID	Last Name	First Name	Address	City	State	Zip	Telephone
19589	Adams	Wes	3132 C N. E.	Auburn	WA	98002	(253) 833-1958
21533	Alberts	George	1819 Westlake Ave. N.	Seattle	WA	98109	(206) 452-2153
20256	Allen	Susan	17314 140th Ave S. E.	Renton	WA	98058	(425) 226-2025
10544	Allert	Maria	865 Lind S. W.	Renton	WA	98055	(425) 227-1054
22184	Ally	Kim	2904 A St. S. E.	Auburn	WA	98002	(253) 833-2218
22113	Andrews	Mike	23605 - 156th S.E.	Kent	WA	98042	(253) 872-2211

Record: 14 of 321 | Unfiltered | Search

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.