More Forms

INFO/CSE 100, Spring 2005
Fluency in Information Technology

http://www.cs.washington.edu/100
Readings and References

• Reading
  » Fluency with Information Technology
    • Chapter 15, Case Study in Database Design

• References
  » MS Access Help files
    • keyword “form”
Link one book with many authors?

- We DO want:
  » to link each book to one or more authors

- We DON'T want
  » to specify extra fields (author1, author2, author3, …)
    • this is wasteful and limits the max number of authors
  » to specify each book entry several times, naming a different author in each row
    • this duplicates all the other information about the book
Add a cross-reference table!

- Refine the design so that it includes another table that is a book-author cross reference
  - Each entity in the table is a single cross reference
    - Attribute: ISBN
    - Attribute: Author ID
  - No primary key
- Now we can break the many-to-many relationship into two 1-to-many relationships that we already know how to implement
Define new cross-reference entities
**book-author table**

![Database screenshot with tables](image)

The screenshot shows a database with tables for authors, books, and author-book relationships. The author table includes columns for ID, Name, and Phone. The books table has columns for ISBN, Title, and Price. The book-author table links the two through ISBN and AuthorID.
Define the new relationships
Define a query that uses the relationship

```
SELECT books.ISBN, books.Title, authors.Name, publishers.Name
```
Get the new view of the data

- Notice that this view has redundant data
  - That's okay, because we are not storing it this way, just presenting it
  - The redundant items (Alex, Another Press) came from a single entry in a table – they are guaranteed to be identical
Now we've implemented this entire schema.
View: All Books from “Another Press”
View: All Books by Alex

SELECT authors.Name, [book-author].ISBN, books.Title
WHERE (((authors.Name)="Alex");
View: All info about a given ISBN

```
SELECT books.ISBN, books.Title, books.Price, authors.Name, publishers.Name
WHERE (((books.ISBN)=[ISBN?]));
```
Views as Tables

- Recall that the result of a query is a table
- We have been presenting the table to the user in simple tabular form
But tables are not pretty …

Users need help understanding what they are looking at and what they can do with it
Front end and Back end

• Front end
  » We present the data to the user with some sort of Graphical User Interface
    • Simple tabular display as we have been doing
    • MS Access provides *Forms* and *Reports* for GUIs
    • Web pages

• Back end
  » The database stores the data in tables
  » We use queries to construct new "virtual" tables
Forms

A form is primarily used to enter or display data in a database.

The designer controls what it looks like and how it works, so it can be tailored to specific needs.
A Form is just a Face for a table

- The form lets the designer arrange the data, label it, provide some control over events, etc
  » the presentation
  » multiple presentations are possible depending on the specific needs of each user

- Underlying data comes from a table or a query
  » the content
  » single source of data ensures consistency
How does a form get built?

The Form wizard can help get you started.
But you probably want to tweak it …

Design
Explore the Design capabilities

• Properties of the various controls can be set
• Controls and labels can be moved around
• Images and patterns can be applied
• Event handlers can be written just like on HTML pages with onClick, etc
  » these are written in Basic, not JavaScript
Displaying an image

• In general, images are not stored directly in the database
  » This would mean copying the image and storing it as part of the database file
    • The resulting database is very big
    • The image files are not available outside of the database program

• But we can easily store a link to the image file
  » a text field containing the path to the image file
  » use the path to find, load, and display the image
Simple Display Form

ImagePaths table
To display a linked image

• Create a form based on a table or query that includes the path attribute
  » include a text field on the form to hold the path
• Create an image control on the form
  » this is where the image is actually displayed
• Create event handlers to load the image when something changes
  » associated with the form event OnCurrent
  » associated with the text field event AfterUpdate
text field that holds the value of the primary key for the ImagePaths table, the ID attribute

text field that holds the value of the attribute ImagePath

image field that displays the image pointed to by ImagePath
How do we change the image? Event Handlers

double click here to get this

then click here
OnCurrent event handler for the form

**ImagePath** is the name of the text field that holds the path to the image on your form.

**ImageFrame** is the name of the Image control that displays the image on your form.
1. Select the text field that holds the path

2. Click the properties button

3. Click the code generator button
AfterUpdate event handler for the field

ImagePath is the name of the text field that holds the path to the image on your form.

ImageFrame is the name of the Image control that displays the image on your form.

```vba
Private Sub ImagePath_AfterUpdate()
    If IsNull(Me!ImagePath) Then
        Exit Sub
    End If

    If (IsRelative(Me!ImagePath) = True) Then
        Me!ImageFrame.Picture = CurrentProject.path & "\" & Me!ImagePath
    Else
        Me!ImageFrame.Picture = Me!ImagePath
    End If
End Sub

Function IsRelative(fName As String) As Boolean
    ' Return false if the file name contains a drive or UNC path
    IsRelative = (InStr(1, fName, ":") = 0) And (InStr(1, fName, "\"\") = 0)
End Function
```