# Access Tutorial 6: Form Fundamentals

6.1 Introduction: Using forms as the core of an application

Forms provide a user-oriented interface to the data in a database application. They allow you, as a developer, to specify in detail the appearance and behavior of the data on screen and to exert a certain amount of control over the user's additions and modifications to the data.

Like queries, forms do not contain any data. Instead, they provide a "window" through which tables and queries can be viewed. The relationship between tables, queries, and forms is shown in Figure 6.1.

In this tutorial, we are going to explore the basic elements of form creation using Access' form design tools. In subsequent tutorials, we will extend the functionality and ease-of-use of our basic forms with subforms (Tutorial 7), "combo box" controls (Tutorial 8), and triggers (Tutorial 13).

FIGURE 6.1: The relationship between forms, queries, and tables.

forms

queries

tables

Courses   Departments   Employees
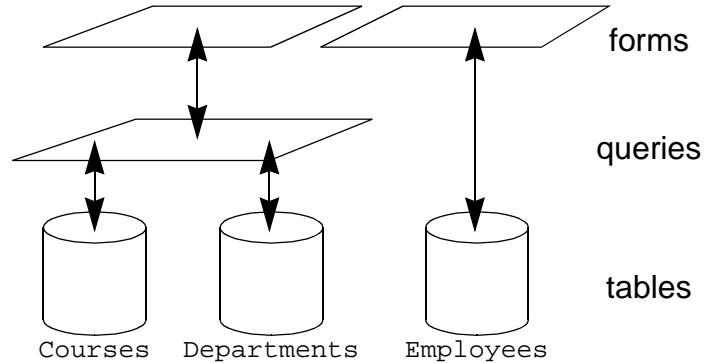
6.2 Learning objectives

☐ Do forms contain data?

☐ How do I create a form?

❒ How do I make the contents of a field on a form read-only?

❒ What is an unbound text box? How do I create one?

❒ How do I create a form using the form wizard?

❒ What is the difference between a columnar (single-column) and tabular form?

6.3 Tutorial exercises

6.3.1 Creating a form from scratch

Although Access provides an excellent wizard for creating simple forms, you will start by building a form from scratch. This will give you a better appreciation of what it is the wizard does and provide you with the basic knowledge needed to customize and refine the wizard's output.

- Create a new blank form based on the `Courses` table, as shown in Figure 6.2.
- The basic elements of the design screen are shown in Figure 6.3. Use the *View* menu to display the **toolbox** and **field list** if they are not already visible.

6.3.1.1 Adding bound text boxes

- Add a "bound" text box for the `DeptCode` field by dragging `DeptCode` from the field list to the form background, as shown in Figure 6.4.
- Reposition the `DeptCode` text box in the upper left of the form.

(?) Remember that you can always use the "undo" feature to reverse mistakes. Select *Edit > Undo* from the menu or simply press *Control-Z* (this works the same in virtually all Windows applications).

FIGURE 6.2: Create a new form to display data from the `Courses` **table.**

a Select the Forms tab from the database window.
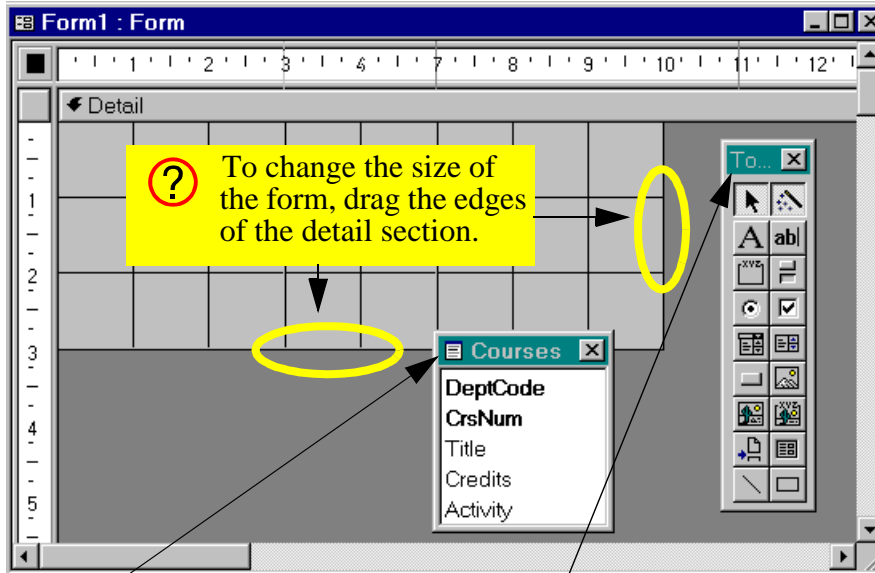
b Select Design View (do not use the wizard at this point)

c Bind the form to the `Courses` table.

(?) Since you can build a form on top of a table or a query, both are shown in this list (here is where a meaningful naming convention starts to pay off)
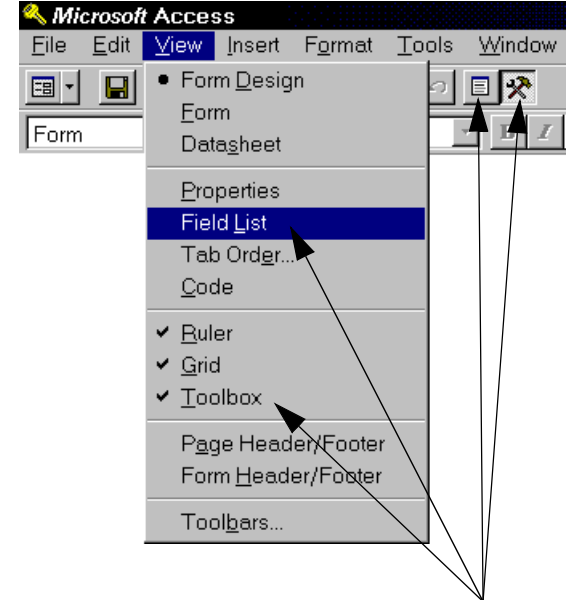
FIGURE 6.3: The basic elements of the form design screen.

To change the size of the form, drag the edges of the detail section.

The field list — shows the fields in the table or query to which the form is bound.

The toolbox — the icons in the toolbox are used to create graphical items and controls on the form.

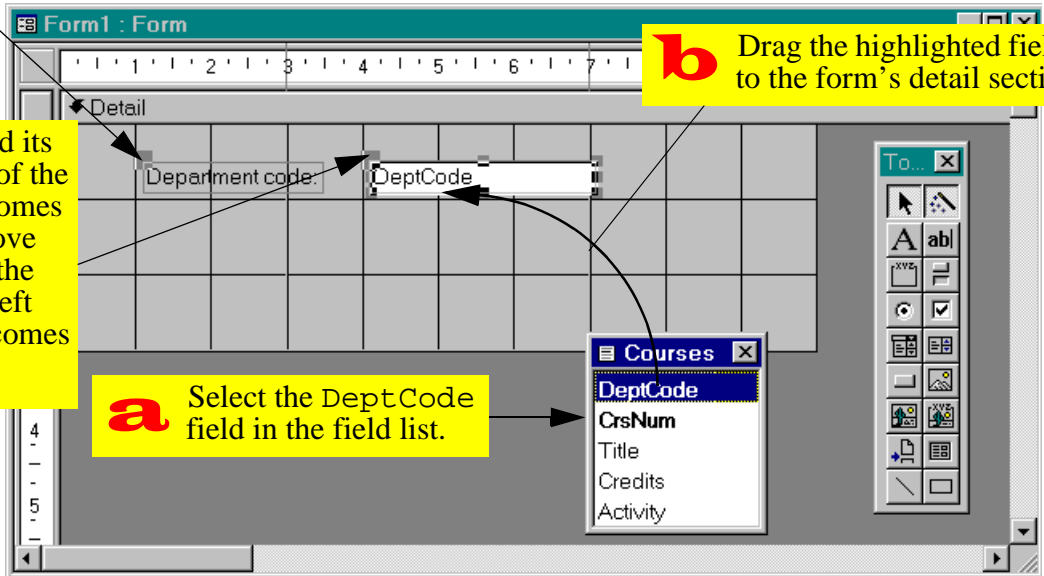If the field list and toolbox are not displayed, use the View menu or toolbar icons.

FIGURE 6.4: **Create a bound text box for the** `DeptCode` **field.**

(?) Access uses the field's caption property as the default label for the text box. If no caption is specified, the field name (e.g., `DeptCode`) is used. To save time editing labels, choose your captions with this feature in mind.

b Drag the highlighted field on to the form's detail section.

(?) To move an object and its label, drag the center of the object (the cursor becomes a white arrow). To move just the object or just the label, drag the upper left handle (the cursor becomes a pointing finger).

a Select the `DeptCode` field in the field list.

Form1 : Form

Detail

Department code: DeptCode

Courses
DeptCode
CrsNum
Title
Credits
Activity

- Drag the remaining fields on to the form, as shown in Figure 6.5 (do not worry about whether the fields are lined up perfectly).
- Select *View > Form* to see the resulting form. Alternatively, press the form view icon (⊞).
- Select *View > Form Design* or press the design view icon (✏) to return to design mode.

6.3.1.2   Using a field's properties to protect its contents

Every object on an Access form (e.g., text box, label, detail section, etc.) has a set of properties that can be modified. In this section, you are going to use the *Locked* and *Enabled* properties to control the user's ability to change the information in a field.

- Select the DeptCode text box and right-click to bring up its property sheet, as shown in Figure 6.6.

- Scroll down the property sheet to the *Locked* property and set it to Yes, as shown in Figure 6.7.
- Switch to the form view and attempt to change the contents of the DeptCode field.

A stronger form of protection than locking a field is "disabling" it.

- Return to design mode and make the following changes: reset the *Locked* property to No; set the *Enabled* property to No.
- Attempt to change the contents of the DeptCode field in form view, as shown in Figure 6.8.
- Save the form as frmCourses.

6.3.1.3   Adding an unbound text box

All the text boxes created in the previous section were "bound" text boxes—that is, they were bound to a field in the underlying table or query. When you change the value in a bound text box, you are mak-

FIGURE 6.5: **Add the text boxes and switch to form view to see the resulting form.**

(?) Text boxes are simply "windows" on to the fields in the underlying table.

a Add the remaining fields to the form.

b Select View > Form from the main menu to view the form.

(?) You can add more than one field to the form with one drag-and-drop operation by holding down the Control button when selecting the fields from the field list.

FIGURE 6.6: **Bring up the property sheet for the** `DeptCode` **text box.**

b Right-click once on the selected object to get the pop-up menu.

c Select Properties to get the property sheet.

a Select the object (e.g., the `DeptCode` text box) for which you wish to see the properties. When an object has been selected, it is bordered by six dark "handles".
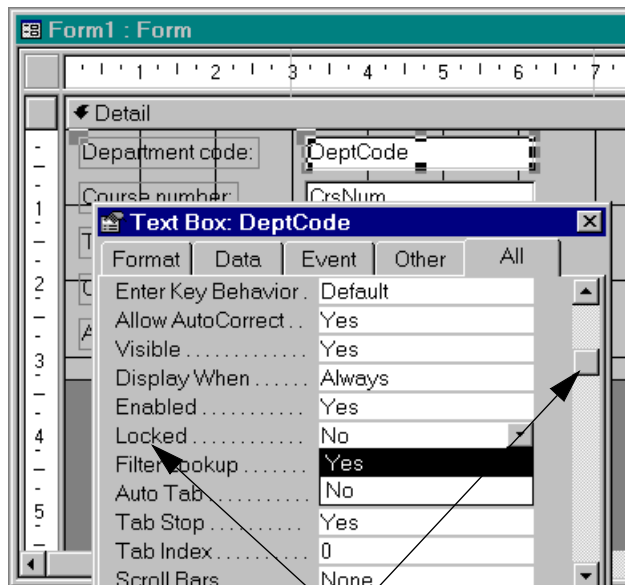
(?) The properties are broken down into four groups. To see all the properties, select the All tab.

(?) Some properties of the text box (such as input mask) are inherited from the field to which the text box is bound.

FIGURE 6.7: **Change the *Locked* property of**
`DeptCode` **to** `Yes`**.**

a Use the scroll bar to find the Locked property.

ing the change directly to the data in the underlying table.

It is possible, however, to create objects on forms that are not bound to anything. Although you will not use many "unbound" text boxes in the assignment, it is instructive to see how they work.

- Create a new empty form bound to the `Courses` table and save it using the name `frmCoursesUB`.
- Select the text box tool (`abl`) from the toolbox and create and unbound text box, as shown in Figure 6.9.

6.3.1.4   Binding an unbound text box to a field

The only difference between a bound and an unbound text box is that the *Control Source* property of a bound text box is set to the name of a field. In this section, you are going to change the unbound text box shown in Figure 6.9 to a bound text box.

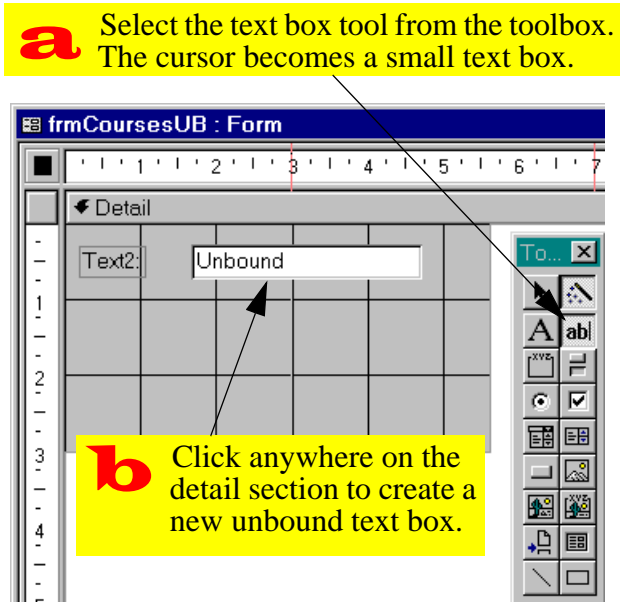**FIGURE 6.8: Set the *Enabled* property of `DeptCode` to `No` and attempt to change the value in the field.**
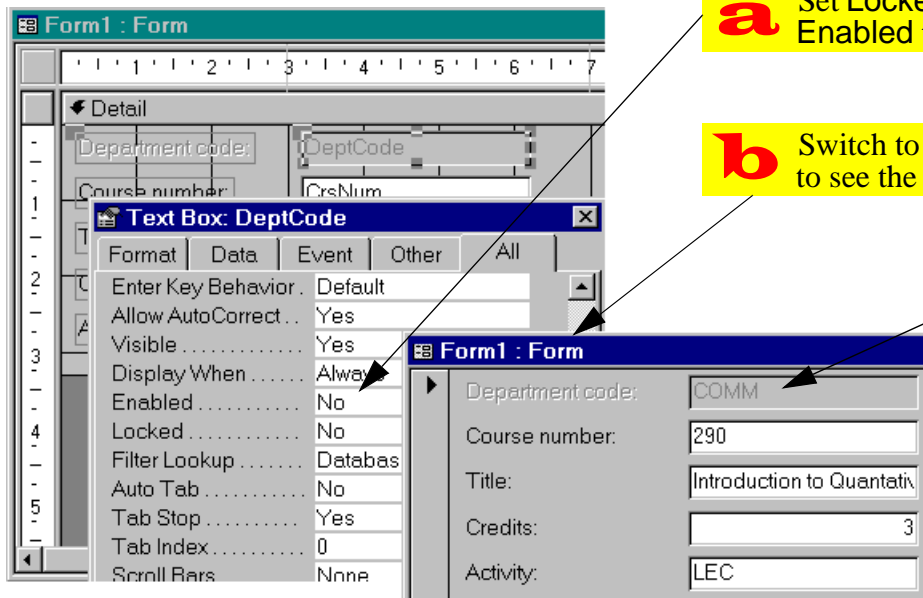
a Set Locked to `No` and Enabled to `No`.

b Switch to form view to see the result.

(?) When a form object is disabled, it cannot receive the "focus" (that is, you cannot put the cursor on it).

(?) By default, disabled form objects are greyed out. To override this feature, set the Locked property to `Yes` and the Enabled property to `No`.

FIGURE 6.9: Create an unbound text box.

a Select the text box tool from the toolbox. The cursor becomes a small text box.

b Click anywhere on the detail section to create a new unbound text box.

• Bring up the property sheet for the unbound text box. Change its *Control Source* property from null to DeptCode, as shown in Figure 6.10.

6.3.2 Creating a single-column form using the wizard

Now that you understand the basics of creating and modifying bound text boxes, you can rely on the form wizard to create the basic layout of all your forms.

• Create a new form bound to the Courses table using the form wizard, as shown in Figure 6.11.
• Use the form wizard to specify the fields you want on your form and the order in which they appear, as shown in Figure 6.12. Select "columnar" when prompted for the form type.
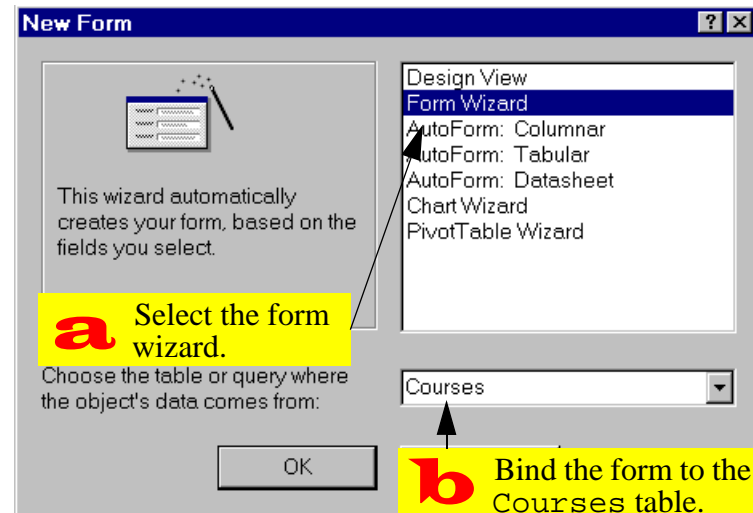
⚠ **2** "Columnar" forms are called "single column" forms in version 2.0.

FIGURE 6.10: Set the Control Source property of an unbound text box.
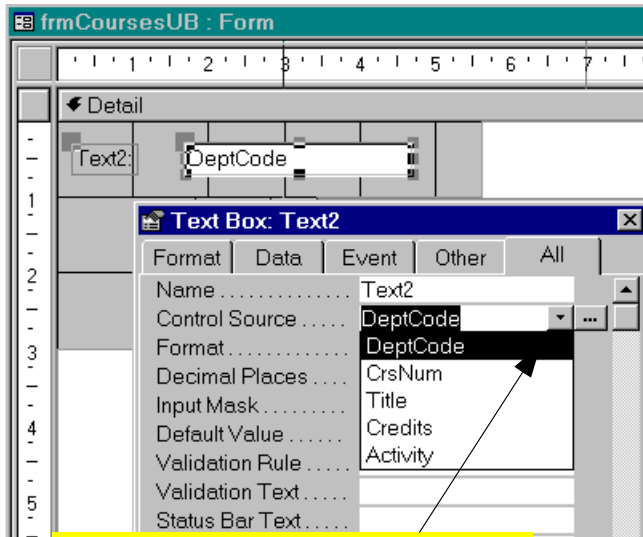
a Use the pull-down list to set the Control Source property to DeptCode.

FIGURE 6.11: Create a new form using the form wizard.

a Select the form wizard.

b Bind the form to the Courses table.

FIGURE 6.12: **Use the form wizard to determine the order of fields on your form.**

Form Wizard

Which fields do you want on your form?

You can choose from more than one table or query.

Tables/Queries:

Table: Courses

Available Fields:

Title
Credits
Activity

Selected Fields:

DeptCode
CrsNum

(?) to show a field, either double-click it or press the > button.

(?) To show all the fields, press the >> button.

(?) The order in which the fields appear in this pane is the order in which they will appear on the form. Use the < and << buttons to move fields back to the pane on the left.

Cancel   < Back   Next >   Finish

The primary advantage of the wizard is that it automatically creates, formats, and aligns the bound text boxes. Of course, once the wizard has created a form, you are free to modify it in any way.

(?) If you make a mistake when creating a form (e.g., you put the fields in the wrong order) it is often easier to use the wizard and start over than to fix the problem manually.

6.4 Discussion

6.4.1 Columnar versus tabular versus datasheet forms

Columnar forms show one record per page. **Tabular** forms, in contrast, show many records per page and are used primarily as subforms. There is also a a **datasheet** form type, but it is seldom used since it gives the developer relatively little control over the

look and behavior of the data. The three different types of forms are shown in Figure 6.13.

6.5 Application to the assignment

• Use the wizard to create columnar forms for all your master tables. Note that in some cases (e.g., `BackOrders`) you will want to base the form on a join query rather than table in order to show important information such as `CustName` and `ProductName`.

FIGURE 6.13: The same information displayed as a columnar, tabular, and datasheet form.

A columnar form displays one record per page.

A tabular form displays more than one record per page.

A datasheet form is identical to the datasheet view of a table or query. Since it gives the designer very little control over the format of the data, it is generally inappropriate for use in an end-user application.