

13.1 Introduction: Using forms as the core of an application

Forms provide a user-oriented interface to the data in a database application. Moreover, forms permit you, as a developer, to

- specify in detail the appearance and behavior of the data on screen, and
- exert a certain amount of control over the user's ability to modify the data.

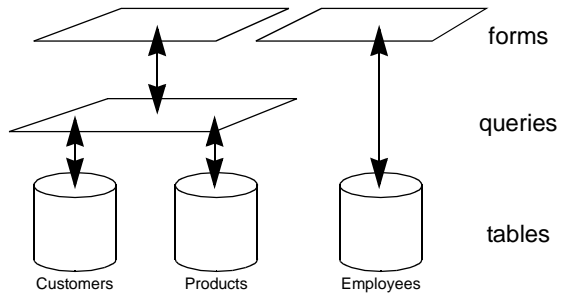
Like queries, forms do not contain any data. Instead, they provide a "lens" through which tables and queries can be viewed. The lens metaphor for describing the interaction between tables, queries, and forms is shown in Figure 13.1.

In this lesson, we are going to explore the basic elements of form creation using ACCESS' form design tools.

13.2 Learning objectives

- create a simple form
- use the properties of an object to make its contents read-only
- understand the difference between a "bound" and "unbound" text box

FIGURE 13.1: The relationship between forms, queries, and tables.



- create a form using the form wizard
- understand the difference between a "columnar" (single-column) and "tabular" form

13.3 Exercises

13.3.1 Creating a form from scratch

Although ACCESS provides an excellent wizard for creating simple forms, you will start by building a form from scratch. Working without the wizard will give you a better appreciation for



what it is that the wizard actually does and provide you with the basic knowledge needed to customize and refine the wizard's output.

(lesson13-1.avi)

- ➔ Create a new blank form based on the **Customers** table, as shown in [Figure 13.2](#).

The basic elements of the form design screen are shown in [Figure 13.3](#).

- ➔ Use the **View** menu to display the **toolbox** and **field list** if they are not already visible (see [Figure 13.3](#)).

13.3.1.1 Adding bound text boxes

(lesson13-2.avi)

- ➔ Add a “bound” text box for the **CustID** field by dragging **CustID** from the field list to the form background, as shown in [Figure 13.4](#).

FIGURE 13.2: Create a new form to display data from the **Customers** table.

The screenshot shows the 'New Form' dialog box in Microsoft Access. The 'Forms' tab is selected in the top menu. The 'Design View' option is highlighted in the list of form types. The 'Customers' table is selected in the dropdown menu at the bottom. The 'New' button is visible on the right side of the dialog box.

- 1** Select the **Forms** tab from the database window.
- 2** Press the **New** button to create a new form.
- 3** Select **Design View** (do not use the wizard at this point)
- 4** Bind the form to the **Customers** 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).

? A form can be “bound” to a table or query or be “unbound”.



FIGURE 13.3: The basic elements of the form design screen.



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

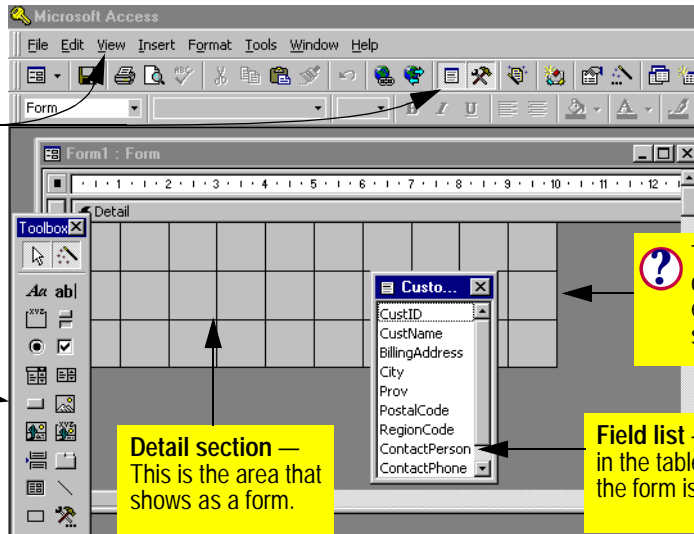
Toolbox — the icons in the toolbox are used to create graphical items and controls on the form.

Detail section — This is the area that shows as a form.



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

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



➤ Select the **CustID** text box using the mouse and reposition it in the upper left region of the form.

➤ Drag the remaining fields on to the form (do not worry about whether the fields are lined up perfectly).



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

➤ Save the form as **frmCustomers**.


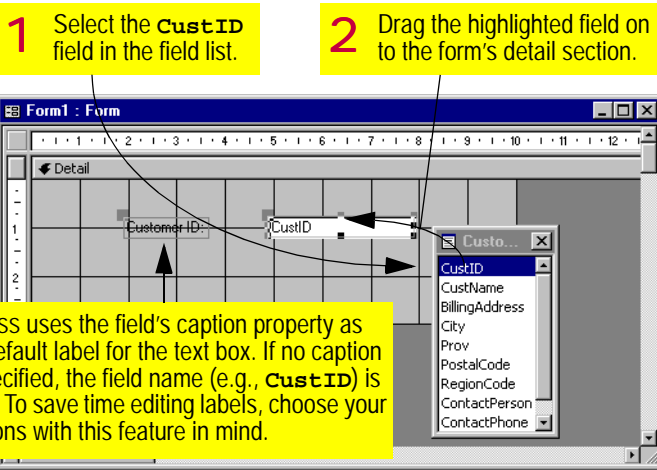

➤ Select **View** → **Form** to see the resulting form. Alternatively, press the form view icon (.

FIGURE 13.4: Create a bound text box for the *CustID* field.

? By default, ACCESS creates a "text box control" when you drag a field onto the form. A text box is simply a window into the underlying field in the table.

➔ Select **View** → **Form Design** or press the design view icon () to return to design mode.

properties to control the user's ability to change the information in a field.

(lesson13-3.avi)

13.3.1.2 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**

➔ Select the **CustName** text box and right-click to bring up its property sheet, as shown in [Figure 13.5](#).

➔ Scroll down the property sheet to the **Locked** property and set it to Yes, as shown in [Figure 13.6](#).

FIGURE 13.5: Bring up the property sheet for the **CustName** text box.

1 Select the object (e.g., the **CustName** text box) for which you wish to see the properties.

2 Right-click once on the selected object to get the context menu.

3 Select **Properties** to get the property sheet for the object.

The screenshot shows the Microsoft Access interface. The main window is titled 'FormCustomer : Form'. The design view shows a form with fields: Customer ID (CustID), Customer name (CustName), Billing address (BillingAddress), City (City), and Province (Prov). The 'CustName' text box is selected, and its context menu is open, showing the 'Properties' option. The 'Text Box: CustName' property sheet is also visible, showing various property groups like Name, Control Source, Format, etc.

? The name and object type of the selected object shows in the title bar of the property sheet.

? When an object has been selected, it is bordered by eight dark "sizing handles".

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

? The properties are broken down into four groups. To see all the properties, select the **All** tab.

- ➔ Switch to the form view and attempt to change the contents of the **CustName** field. The contents of the field are indeed locked.

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

- ➔ Return to design mode and make the following changes: reset the **Locked**

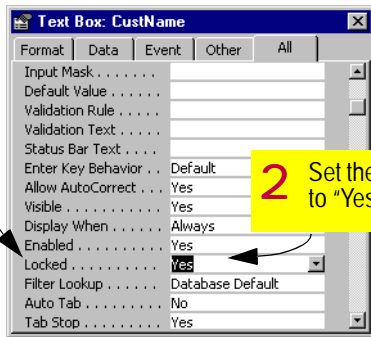
property to **No** and set the **Enabled** property to **No**.

- ➔ Attempt to change the contents of the **CustName** field in form view, as shown in [Figure 13.7](#).



FIGURE 13.6: Change the **Locked** property of **CustID** to Yes.

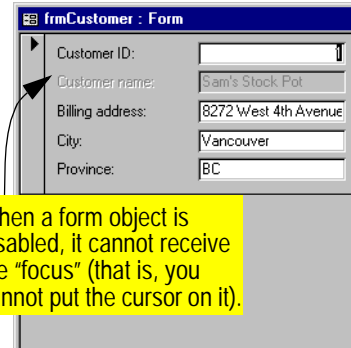
1 Use the scroll bar to find the **Locked** property.



2 Set the property to "Yes".

FIGURE 13.7: Disable **CustName** and attempt to change the value in the field.

1 Set **Enabled=No**, **Locked=No**, and view the form.



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

? A trick: if you set **Enabled=No** and **Locked=Yes**, the field will be disabled, but will not be greyed out.

13.3.1.3 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 making 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

your kitchen supply project, it is instructive to see how they work.

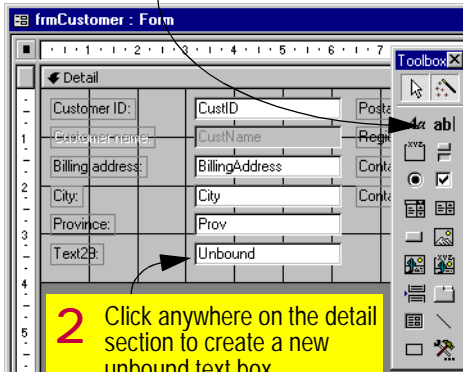
(lesson13-4.avi)

➔ Select the text box tool (**ab|**) from the toolbox and create an unbound text box, as shown in [Figure 13.8](#).



FIGURE 13.8: Create an unbound text box.

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



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

13.3.1.4 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 13.8](#) to a bound text box.

- Bring up the property sheet for the unbound text box. Change its **Control Source** property from NULL to **CustID**, as shown in [Figure 13.9](#).

13.3.2 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.

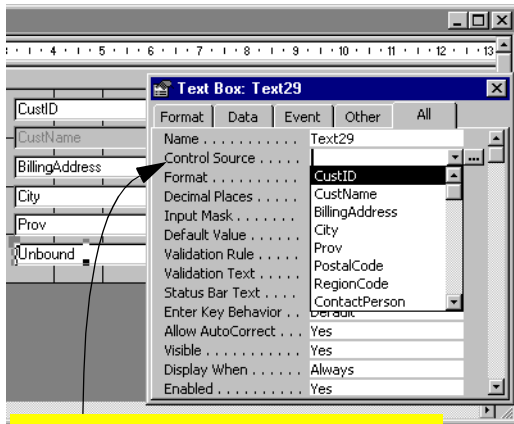
(lesson13-5.avi)

- Switch to form view and enter a value in the new unbound text box.
- Use the record navigation buttons to step through the different customers. Notice that unlike the bound text boxes for **CustID**, **BillingAddress**, and so on, the value in the unbound text box does not change.

- Create a new form bound to the **Products** table using the form wizard, as shown in [Figure 13.10](#).
- Use the form wizard to specify the fields you want on your form and the order in which they appear. Select "columnar" when prompted for the form type.



FIGURE 13.9: Set the **Control Source** property of an unbound text box.

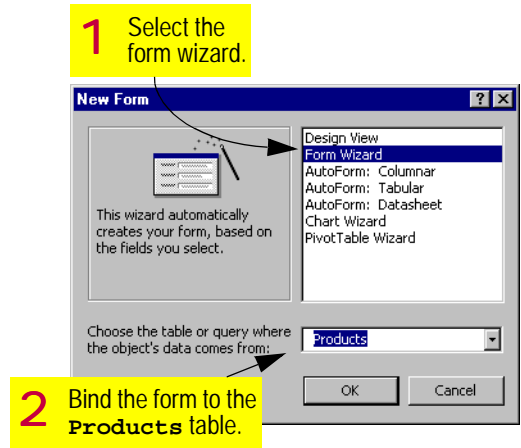


1 Use the pull-down list to set the **Control Source** property to **CustID**.

2 “Columnar” forms are called “single column” forms in version 2.0.

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.

FIGURE 13.10: Create a new form using the form wizard.



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



13.4 Discussion

13.4.1 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 (subforms are discussed in [Lesson 14](#)). There is also 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 13.11](#).

frmProducts

Product ID: 57 4966
 Description: Mixing bowl, 16 qt.
 Unit: EA
 Unit Price: \$12.50
 Quantity on hand: []

A **columnar** form displays one record per page.

FIGURE 13.11: The same information displayed as a columnar, tabular, and datasheet form.

frmProductsTabular

Product ID	Description	Unit	Unit Price	Quantity on hand
51 5012	Water jug, s.s. w/ice guard, 2 litre	EA	\$23.50	[]
57 3828	Spatula, 6" "Cuisipro"	EA	\$4.00	[]
57 3828	Spatula, 8" "Cuisipro"	EA	\$4.25	[]
57 4966	Mixing bowl, 16 qt.	EA	\$12.50	[]

A **tabular** form displays more than one record per page.

frmProductsTabular

ProductID	Description	Unit	UnitPrice	QtyOnHand
51 5012	Water jug, s.s. w/ice guard, 2 litre	EA	\$23.50	36
57 3828	Spatula, 6" "Cuisipro"	EA	\$4.00	65
57 3828	Spatula, 8" "Cuisipro"	EA	\$4.25	[]
57 4966	Mixing bowl, 16 qt.	EA	\$12.50	[]
57 551	S.S. salad server set	2PC	\$3.15	[]
71 12101	S.S. soup ladle	EA	\$5.25	[]
71 12110	S.S. skimmer	EA	\$5.00	[]
71 12111	S.S. sauce ladle	EA	\$5.25	[]
71 12114	S.S. grave ladle with spout	EA	\$4.75	[]
74 4042	Snail plate w/white handle	EA	\$3.15	[]
74 4321	Pasta fork, 1"	EA	\$4.00	[]

A **datasheet** form is identical to the datasheet view of a table or query. Since the datasheet view gives the designer less control over the format of the data than the other two form types, it is generally inappropriate for use in applications.



13.5 Application to the project

- ➔ Use the wizard to create columnar forms for all your *master tables*. Note that in some cases (e.g., **Customers**) you will want to base the form on a join query rather than table in order to show information such as the name of the region.