Access Tutorial 9: Advanced Forms

9.1 Introduction: Using calculated controls on forms

It is often useful to show summary information from the subform on the main form. The classic example of this is showing the subtotal from a list of order details on the main order form.

In this tutorial, you are going to explore one means of implementing this feature using calculated controls. A calculated control is an unbound control for which the *Control Source* property is set to an expression that Access can evaluate.

Clearly, calculated controls have a great deal in common with the calculated query fields you created in Section 4.3.3. Although there are no hard-and-fast rules that dictate when to use a one over the other, pushing your calculations to the lowest level (i.e., performing calculations in the query) is usually the best course of action. However, as you will see in the context of subtotals, this is not always possible.

9.2 Learning objectives

- How do I create a calculated text box?
- What is the expression builder? When is it used?
- Where can put an intermediate result of a calculation on a form so that it does not show?

9.3 Tutorial exercises

9.3.1 Creating calculated controls on forms

In this section, you are going to create a simple calculated text box to translate the `Credits` field into a dichotomous text variable [`full year`,

`half year`]. Recall that you have already implemented this feature in Section 4.3.3.2 using a calculated query field.

- Perform the steps shown in Figure 9.1 to create an unbound text box on your `fmrCoursesMain` form.
- Set the *Control Source* property of the text box using the syntax:
  `= <expression>`
  In this case, the expression should be an "immediate if" function (see Section 4.3.3.2).

By default, Access interprets text in the *Control Source* property field as the name of a variable (i.e., the name of a field or another control). As such, you must remember to include the equals sign when setting this property.

- Test your form. Note that you are prevented from editing the calculated field. If, however, you change the value of `Credits`, the value of `txtCourseLength` changes accordingly when you leave the `Credits` field.

9.3.2 Showing a total on the main form

In this section, you will create a calculated text box that displays the number of sections associated with each course. The primary motivation for this exercise is to illustrate some of the limitations of calculated controls (as they are implemented in Access) and to provide an opportunity to explore an interesting work-around.

- Create a text box call `txtNumSections` on the main form as shown in Figure 9.2.

The logical next step is to set the *Control Source* of the field to an expression that includes the `Count()` function. However, Access has a limitation in this

FIGURE 9.1: Create an unbound text box on your main form.

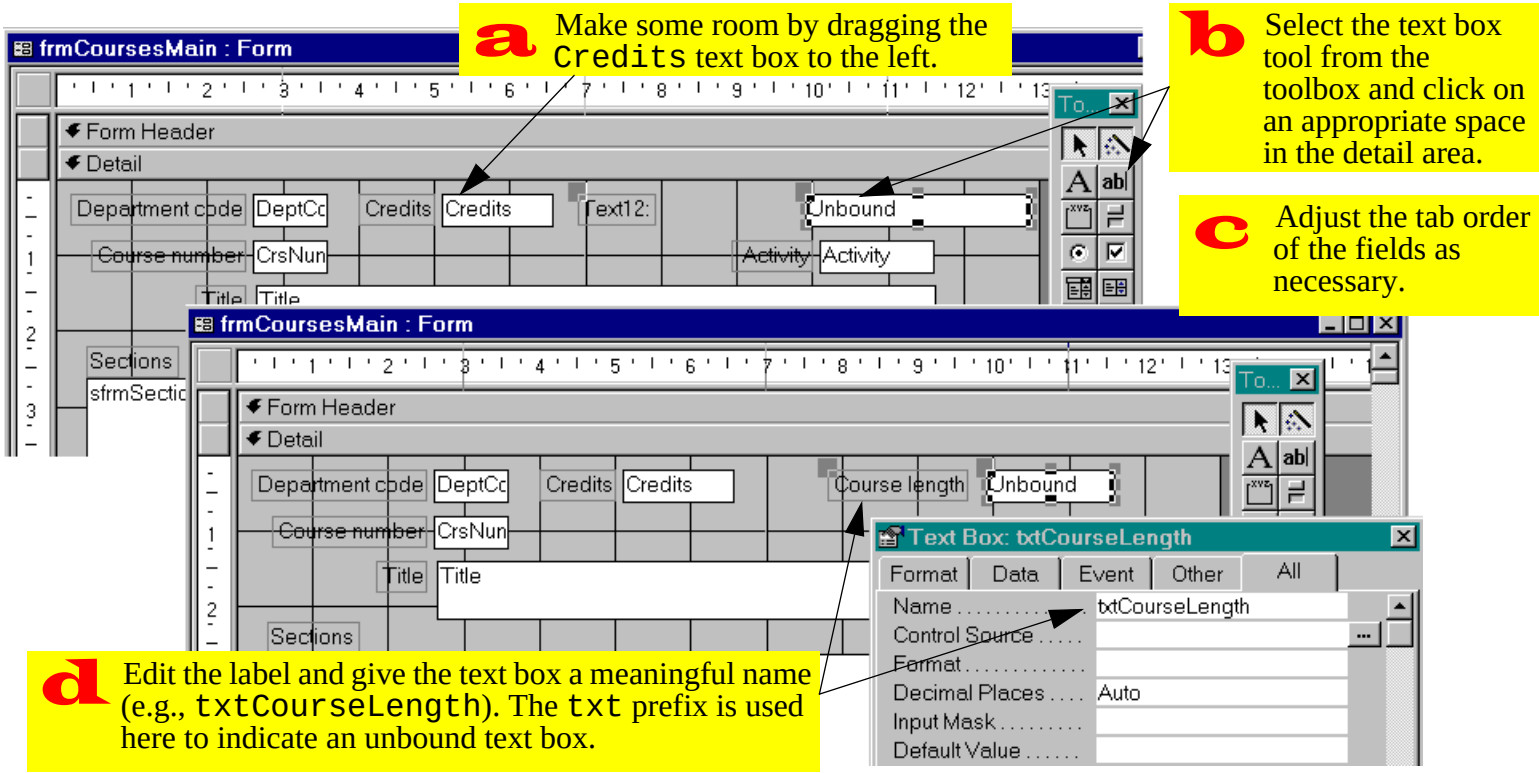
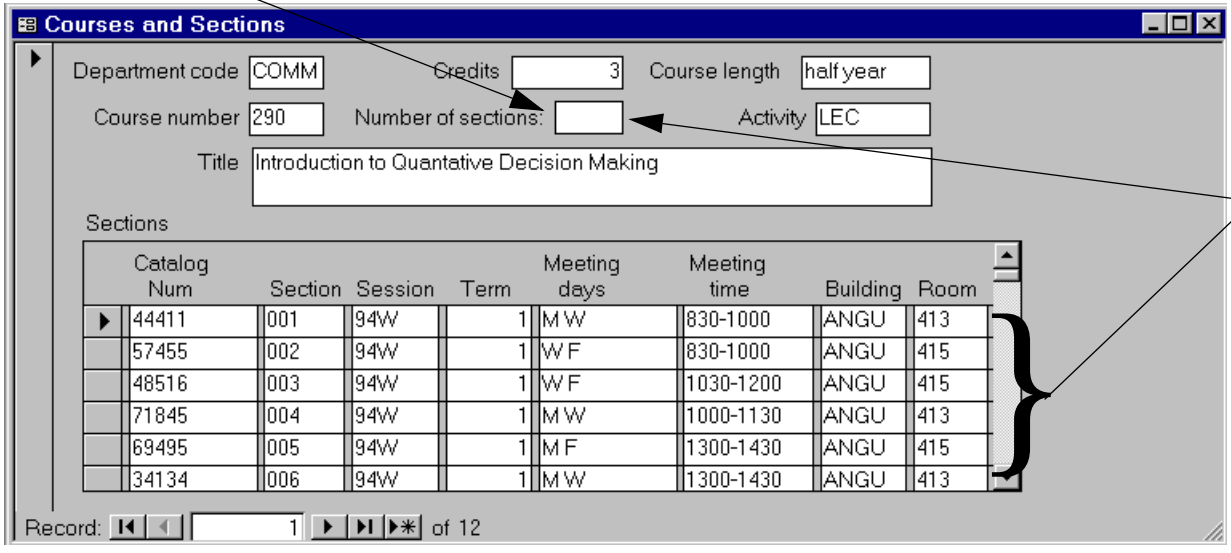


FIGURE 9.2: Create an unbound text box to show the number of sections associated with each course.

a Add an unbound text box called `txtNumSections`. Since it is currently bound to nothing, it is blank.



What you want is a means of counting the records in the subform and displaying the count in the new text box.

regard: you cannot use an **aggregate function** (`Sum()`, `Avg()`, `Count()`, etc.) on a main form that refers to a field in a subform. As a consequence, you have to break the calculation into two steps:

1. use the aggregate function to create a calculated text box on the subform (i.e., a “dummy” field to hold an intermediate result);
2. create a calculated control on the main form that references the dummy text box created in the first step.

It is important that you realize that this procedure does not involve any immutable, fundamental information systems knowledge. Rather, it is merely an example of the type of work-around (hack, kludge, etc.) that is routinely used when using a tool like Access to create a custom application.

9.3.2.1 Calculating the aggregate function on the subform

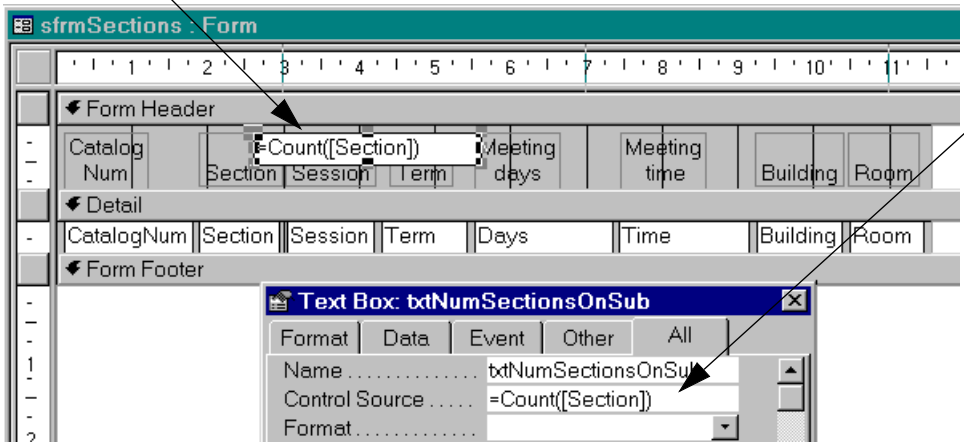
- Create an unbound text box on the subform as shown in Figure 9.3.
- Save the subform but do not close it.
- Return to the main form and set the *Control Source* of `txtNumSections` to equal the value of `txtNumSectionsOnSub`. Since the naming conventions for objects on forms and subforms can be tricky, use the **expression builder** (as shown in Figure 9.4) to build the name for you.

The expression builder organizes all the elements of the database environment into a hierarchical structure. You build an expression by “drilling down” to the element you need and double-clicking to copy its name into the text area.

The expression builder takes some practice. One problem is that it is easy to double-click

FIGURE 9.3: Perform the count on the subform.

a Create a calculate control called `txtNumSectionsOnSub` and place it in the form header (do not worry about its location, you will move it later).

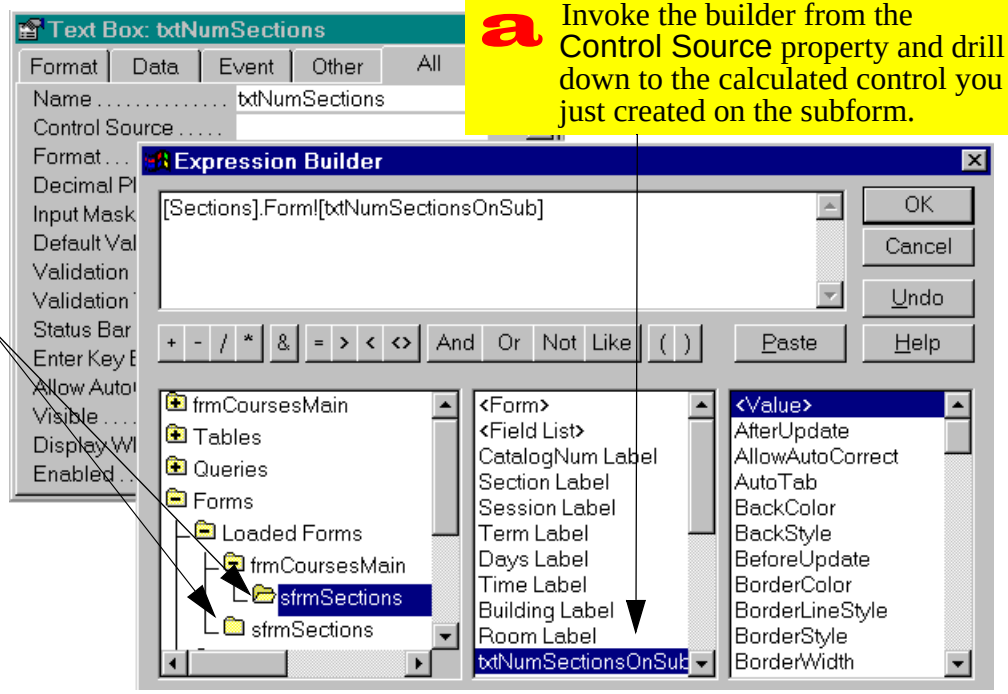


b Set the Control Source property to `=Count([Section])`. Note that any field can be used as the argument for the `Count()` function.

FIGURE 9.4: Use the builder to drill down to the calculated control on the subform.

a Invoke the builder from the Control Source property and drill down to the calculated control you just created on the subform.

? Note that when the main form and the subform are both open, the subform appears twice in the builder: once as a "stand-alone" form (under "Loaded Forms") and once *as a component* of the main form (press the + sign on the `frmCoursesMain` folder). You want to use the latter (you will never access the subform in stand-alone mode).



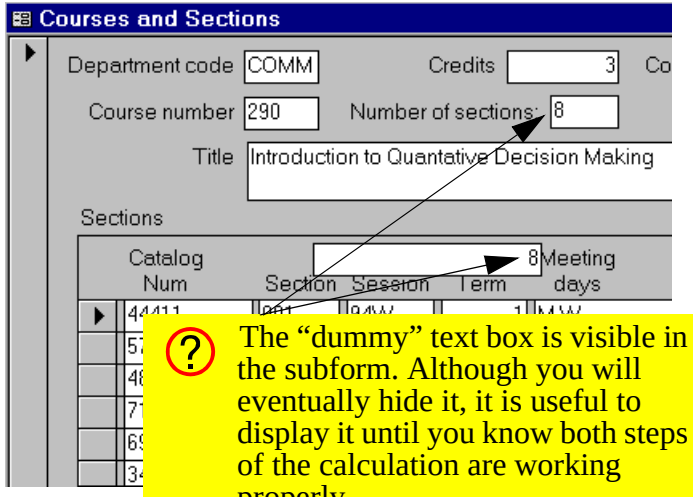
on the wrong thing. Another problem is that Access attempts to guide you by inserting «Expr» place-holders all over the place. The solution to both problems is to click on the text window and make liberal use of the *Delete* key.

The point made about “stand-alone” and “component” subforms in Figure 9.4 is extremely important. The reason you use the `sfrm` prefix is so you know that the form is designed to be a component of another form. If you select the stand-alone version the form in the builder, the name created by the builder will be incorrect and an error will result.

- Close the subform (in version 7.0 and 8.0, the main form and subform cannot be open at the same time).

- Test the form. The value of `txtNumSections` and `txtNumSectionsOnSub` should be identical, as shown in Figure 9.5.

FIGURE 9.5: The number of sections on the main form.



9.3.2.2 Hiding the text box on the subform

The obvious problem in Figure 9.5 is that the dummy text box shows on the subform. There are at least two ways to get around this: one is to set the *Visible* property of the text box to `No`; a slightly more elegant approach is to use the **page header** or **page footer** to hide the text box.

The page header and footer are areas on the form that only show when the form is printed. Since you will never print a form (reports are used for printed material), these areas can be used to hide intermediate results, etc.

- In design mode, select *View > Page Header/Footer* from the menu.

In version 2.0, the menu structure is slightly different. As such, you must select *Format > Page Header/Footer*.

- Drag (or cut and paste) `txtNumSectionsOnSub` from the form header to the page header, as shown in Figure 9.6.
- Test the result.

9.4 Discussion

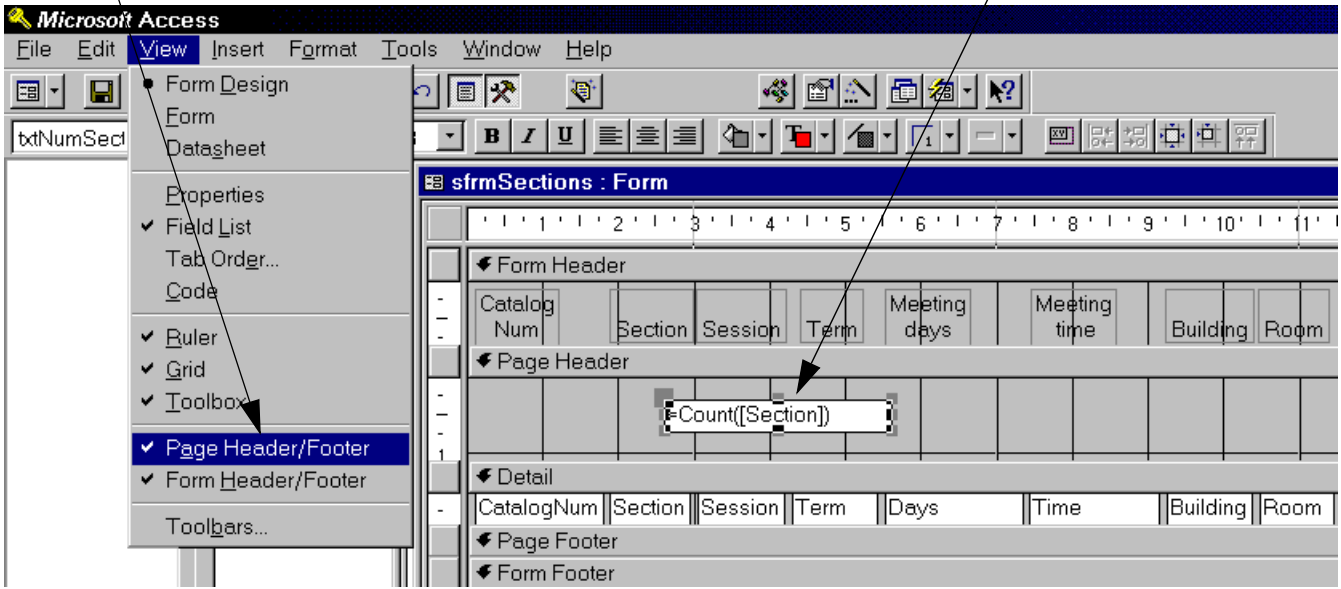
In Section 4.3.3.2 and Section 9.3.1, you accomplished the same thing (showing `half year` or `full year`) using different techniques. The advantage of implementing this as a calculated query field is that you can use this field repeatedly in other forms. On the other hand, if you do the transformation on the form, you have to repeat the calculation on every form that requires the calculated field.

In the case of the aggregate function, the situation is slightly different. Although you can use the **totals** feature of QBE (see on-line help) to count the number of sections for a particular course within a query, the resulting recordset is non-updatable (and hence

FIGURE 9.6: Hide the intermediate result in the page header.

a Select View > Page Header/Footer from the menu (Format > Page Header/Footer in version 2.0) to show the page header and footer.

b Drag (or cut and paste) the field you want to hide into the page header.



not much use for editing course names, etc.). As a result, you are forced to do the calculation on the form rather than in the query.

9.5 Application to the assignment

To show the subtotal, tax, and grand total on your order form, you use the same techniques illustrated here. The only difference is that you use the `Sum()` function instead of the `Count()` function to get the subtotal for the order.

- Create a dummy field on your `OrderDetails` subform to calculate the subtotal for the order.
- Calculate the tax (G.S.T. only for wholesale) and grand total on the main form (traditionally, this information is located near the bottom of the form—but *not* in the form footer).