# Design advanced forms

7.1 Associate a query with an unbound combo or list box control.

7.2 Create an autolookup form controls.

7.3 Create a combo or list boxes based on a value list.

7.4 Design forms requiring a subform(s).

7.5 Add calculated field(s) to subforms in tabular layout.

7.6 Add tab controls to a form.

7.7 Build a Navigation User Interface.

7.8 Design a form to solve a specific business problem.

## 7.1 Associate a query with an unbound combo box or list box control

Many controls are tied (bound) to specific fields in a table or query. When the user changes the data in a bound control, the data in the underlying table is also changed.

An unbound control is useful for display purposes, or to show a set of results based on a user's selection.

### Steps to associate a query with an unbound combo box control

A combo box or list can be associated with a query so the values available will be automatically updated if new values are entered in the underlying tables.

1. Create a query
   1. Open database: Northwind\_Revised\_Mar2
   2. Create a new query based on the Category table
   3. Select the CategoryID and CategoryName fields
   4. Sort the CategoryName ascending
   5. Save the query as **qryCategories**
2. Create a new form in Form Design
   1. Turn off the “Use Control Wizards” option. To do this, expand your Controls in your ribbon and it should be an option on the bottom of that expansion. Graphical user interface, text, application

      Description automatically generated
   2. Add a combo box to the form
   3. View the properties of the combo box
   4. In the Data tab, change the Row Source to **qryCategories** (created above)
   5. Make sure the Bound Column is 1 (that is the **CategoryID** field in the query)
   6. View the form and notice the layout of the combo box.
   7. In the format, change the Column Count to 2, and adjust the widths to 0”;1” – this will hide the primary key field and only display the **CategoryName** field.
   8. Change the controls name to cboCategory
   9. Change the control’s label to lblCategory
   10. Change the control’s label’s caption to “Category”
   11. View the form
   12. Save form as **frmUnboundComboBox**

### Steps to associate a query with an unbound list box control

A list box allows a user to select a choice from a list of possible values. The user usually does not have the ability to add values.

Student exercise:

1. Create a new form in design view
2. Add a list box (Typically, we’ll adjust the height of the list box to something that works better than a single line we had in a combo box)
3. Associate it with the qryCategory query
4. Change the column count and column widths as above
5. View the form
6. Save as **frmUnboundListBox**

Note properties: Limit to List, Allow Value List Edits, List Items Edit Form

Graphical user interface

Description automatically generated with medium confidence

## 7.2 Create an autolookup query

The autolookup query control reduces the amount of information a user has to enter onto a form.

To use such a control, first create a query for tables in a one-to-many relationship. Then, when the user selects a value to appear in the many side, the corresponding fields from the one side will be filled in.

**Steps**:

1. Create a query
   1. New query
   2. Customer and Order tables (one customer places many orders)
   3. Add all fields, including CustomerID, from Order table
   4. Add all fields, **except CustomerID**, from Customer table
   5. Save the query as qryCustomerOrders2
2. Create the form
   1. New form in Design View
   2. Set form properties to use **qryCustomerOrders2**
   3. Add fields from the query
   4. Change text-boxes to combo-boxes (most may be changed already) but make sure Customer, Employee, ShipVia are combo boxes.
      1. Right-click the control, select “Change To”, Combo Box
      2. Adjust the properties of combo boxes to use query, bound column, column count, column widths
   5. Rearrange components as necessary
   6. Save as **frmCustomerOrders**

## 7.3 Create a combo box or list box based on a value list

Combo boxes and list boxes can be created to refer to a static set of values. Use this if the pre-set values do not normally change. Both combo boxes and list boxes have properties to allow the values to change or to be limited to the list.

**Steps for a Combo Box**

1. Edit frmUnboundComboBox
2. Add another combo box
3. Properties:
   1. Row Source Type: Value List
   2. Row Source: enter a semi-colon separated list or us “…” to bring up a dialog box
      1. Enter the days of the week
   3. Limit to List – “Yes” means only these values are allowed; “No” means other values can be entered.
   4. Allow value list Edits – “Yes” allows the end-user to open the list and add their own entries to be used later; “No” prevents this.

Graphical user interface, application

Description automatically generated

**Steps for a List Box (exercise)**

1. **Edit frmUnboundListBox**
2. **Add another List box**
3. **Set properties**

**Graphical user interface

Description automatically generated**

## 7.4 Design forms requiring subforms

In many situations, the data to be entered or viewed depends on two or more related tables in a one-to-many relationship (master/detail). Where such a relationship exists, one method to present the information is to use a form/subform.

Consider the following examples:

* Orders placed with a supplier – one supplier has many orders placed with it.
  + Show the supplier information (master, or FORM), and the orders placed with that supplier (detail, or SUBFORM)
* Maintenance of an asset – one asset item has many maintenance records
  + Display the asset (master form), and the maintenance performed on the asset (detail subform)
* Order details – One order has many detail lines/records
  + Display the order (master form), and the products ordered for that order (detail subform)

Some options to create a form/subform:

* Form Wizard
* Design View

### Using form wizard to create form/subform with datasheet

Use the Kelsey Vets database for the next few examples:

1. Click Create Tab
2. More Forms > run the Form Wizard
3. Select tblVisits, and select all fields (>>)
4. Select tblVisitDetails, and select all fields
5. Next
6. How to view? “By tblVisits” gives options for form with a subform
7. Next
8. Layout for subform? **Datasheet**
9. Next – names of the forms:
   1. frmVisits
   2. subFrmVisitDetails (naming convention)
   3. Modify the form’s design
10. Finish

Adjustments:

1. Potentially, you many need to “Remove Layout” from different controls. If that option is available to you when select the label for the subreport, go ahead and remove it. Also, remove the label.
2. Align the subform with the main form (drag it to the left)
3. View form – note that it doesn’t fit
4. Select a field in the subform and use the interface to select all fields.
5. Remove layout if necessary
6. Swith to layout view and resize the subform box and individual labels

**Create a form called frmCustomer to be used later:**

**Graphical user interface, application

Description automatically generated**

### Using form wizard to create form/subform with tabular layout

1. Click Create Tab
2. Launch the Form Wizard
3. Select tblPet, and select all fields (>>)
4. Select tblVisit, and select all fields (>>)
5. Next
6. How to view? “By tblPets” gives options for form with subform
7. Next
8. Layout for subform? Tabular.
9. Next – names for forms:
   1. frmPets
   2. subFrmVisits
   3. Modify form’s design
10. Finish

Adjustments:

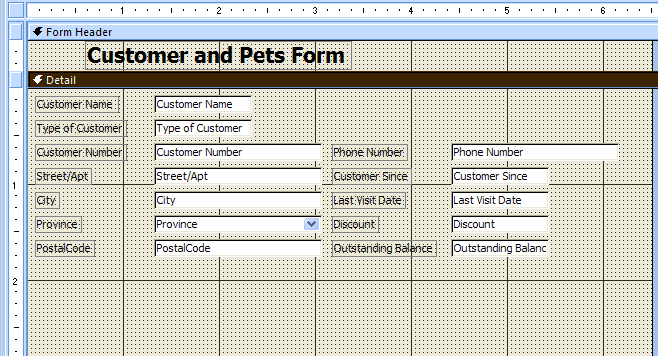
1. Remove layout (if necessary) from subFrmVisits label, and remove the label
2. Align the subform with the main form (drag it to the left)
3. View form – note doesn’t fit
4. Resize the subform box and individual labels (layout view)
5. **Add** Days Since Last Visit textbox to Main
6. **Add** Sum of Total Amount to footer of subform
   1. Maybe, you want to close the frmPets and open the subform by itself to adjust these settings and add the new value. It’s a little easier than trying to work in the limited space the master form gives you for the subform.

### Add a subform to an existing form

A master form may be designed in one step, and the subform gets added at a later time. This can be accomplished in two ways, using the subform wizard:

* Attach a pre-existing Form
* Attach a subform based on a table or a query

Example of attaching a subform based on a table: (Customers and their Pets)

1. Select tblCustomer, and use Form Wizard to create a form with all fields from this table
2. Adjust the layout to resemble the following:
3. 
4. Save the form as **frmCustomerAndPetsOrig**
5. Make a copy of frmCustomerAndPetsOrig and rename it to frmCustomerAndPetsDatasheet. Open it in design view
6. Enable the “Use Control Wizards”
7. Click on the Subform/Subreport control button
8. Drag a rectangle about 4.5” wide x 1” tall (11.5x2.5cm)
9. The SubForm wizard appears:
10. Graphical user interface, application

    Description automatically generated
11. Accept Use existing Tables and Queries, and select Next
12. Choose the table tblPets and select all fields and click Next:
13. Graphical user interface, application

    Description automatically generated
14. Identify how the subForm is connected with the main form: in this case, we use Customer Number, Click Next:
15. Graphical user interface, text, application

    Description automatically generated
16. Name the subform subFrmPetsDatasheet, and click Finish
17. Adjust the width and height of the subform, as necessary
18. View the form, Save your changes.

### To change the subform's view from datasheet to tabular:

1. Copy subFrmPetsDatasheet and rename to **subFrmPetsTabular**
2. Copy frmCustomerAndPetsDatasheet and rename to **frmCustomerAndPetsTabular**. Open this form in design view
3. Click on the subform object. Change the **Name** and **Source Object** Properties to subFrmPetsTabular
4. Click on the form-select button for the subform object
5. Change the Default View from Datasheet to Continuous Forms
6. Select all objects in the subform, remove layout if necessary, apply tabular layout
7. Adjust the height of labels, rows, detail section
8. Switch to Layout view
9. Adjust column widths as needed
10. Save changes

## 7.5 Add calculated field(s) to subforms in tabular layout

Adding a calculated field to a subform can depend on multiple factors:

* Is the subform based on a table or query?
* Is the subform design based on a datasheet or tabular layout?
* Where is the calculated field located? On the main form or on the subform. If on the subform, is it located in the subform detail or the subform footer?

We will look at two methods of adding a calculated field:

1. Tabular layout with subform based on a table
2. Tabular layout with subform based on a query

### Add a calculated field to a form based on a table

Setup forms:

1. Copy frmVisits and subFrmVisitDetails to **frmVisitsCalc1, subFrmVisitDetailsCalc1.**
2. Open frmVisitsCalc1 in design view
3. Change the subform association to subFrmVisitDetailsCalc1

Add calculated fields:

1. Click on the subform
2. Add a new text box:
   1. Name: txtExtendedPrice
   2. Control Source: **=[Treatment Price] + [MedicationPrice]**
   3. Format: Currency
   4. Decimal Places: 2
   5. Label’s Name: lblExtendedPrice
   6. Label’s Caption: Extended Price
3. Select all items in the subform and change to tabular layout using Arrange > Tabular
4. Subform’s form property – Continuous Forms
5. Switch to Layout view, and adjust widths.

Add calculated field in subform’s footer section:

1. Expand the subform’s footer, add a text box to the footer.
   1. Name: txtSubtotal
   2. Control Source: **=sum([Treatment Price] + [MedicationPrice])**
   3. Format: Currency
   4. Decimal places: 2
   5. Label’s Name: lblSubtotal
   6. Label’s Caption: Subtotal

Add calculated field to the main (parent) form’s footer section

1. Expand the main form’s footer area, add text box to the footer
   1. Name: txtTaxes
   2. Control Source: **=[Tax Rate]\*[subFrmVisitDetailsCalc1].[Form]txtSubtotal]**
   3. Format: Currency
   4. Decimal Places: 2
   5. Label’s Name: lblTaxes
   6. Label’s Caption: Taxes

Adjust Layout

1. Remove Treatment Code and Medication Code
2. Adjust subform width to just under 7.5”
3. Adjust form width to 7.5”
4. Line up txtTaxes with the txtExtendedPrice
5. View, save, close

### Add a calculated field to a form based on a query

Create query:

1. Table: tblVisitDetails
2. Fields: All
3. Calculated field: Extended Price: [TreatmentPrice] + [MedicationPrice]
4. Save as **qryVisitDetails**

Setup forms:

1. Copy frmVisitsCalc1 and subFrmVisitDetailsCalc1 to **frmVisitsCalc2 and subFrmVisitDetailsCalc2**
2. Open frmVisitsCalc2 in Design View
3. Change the subform association to subFrmVisitDetailsCalc2
4. Change subform Form Record Source property to qryVisitDetails

Add calculated fields:

Calculated field in subform’s detail section

* + Change txtExtendedPrice properties
    - Control source: Extended Price (instead of the original formula)

Calculated field in subform’s footer section

1. Change txtSubtotal properties
   * Control source: =sum([Extended Price])

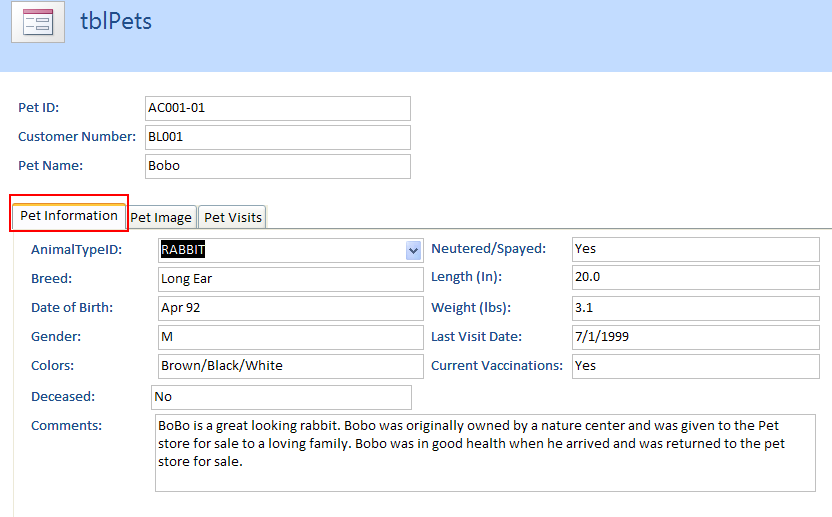
Calculated field in the main form’s footer section

1. Change txtTaxes properties
   * Control source: = [Tax Rate] \* [subFrmVisitDetailsCalc2].[Form]![txtSubtotal]

## 7.6 Add a tab control to a form

A tab control is useful for organizing information on the form by separating the information into tabbed "panes", where only one pane is visible at a time. A tab control can conserve space onscreen and show information from one or more tables.

Example to create a three-pane tab

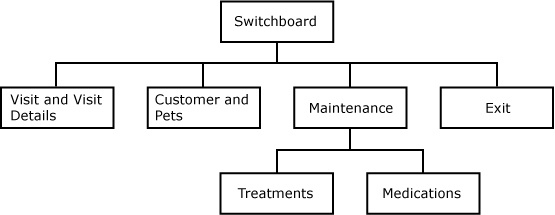
1. Create a new form using the Form tool based on the tblPets table.
2. Click on Tab Control tool and drop a tab control onto the form design grid
3. Click the first tab and change properties:
   * Name: pgPetInformation
   * Caption: Pet Information
4. To move text boxes and labels from the main form onto the first tab:
   * Selected the following fields: (text boxes and labels):
     + Animal Type ID, Breed, Date of Birth, Gender, Colors, Neutered/spayed, Length, Weight, Last Visit Date, Current Vaccinations, Deceased, Comments
     + That is, do not select: Pet ID, Customer Number, Pet Name, and Picture
   * Cut to remove the fields
   * Click on the first tab
   * Past to add the fields
   * Re-arrange as per Image:
   * 
5. Click second tab and change properties:
   * Name: pgPetImage
   * Caption: Pet Image
6. We can cut paste the image information, but we already cut that so it’s no longer on the form. Let’s go to our field selector and add it from there.
   * Change size of the Picture text box, and adjust the Size Mode to Zoom
     + Clip: Displays the object at actual size. If the object is larger than the control, its image is clipped on the right and bottom by the control’s borders
     + Stretch: Sizes the object to fill the control. This setting may distort the proportions of the objects
     + Zoom: Displays the entire object, resizing it as necessary without distorting the proportions of the object. This setting may leave extra space in the control if the control is resized
     + Use the Clip setting for the fastest display. You can use the Stretch setting for bar graphs and line graphs without concern for size adjustments. The Stretch setting can distort circles and photos.
7. Right-click on the tab-control and choose Insert Page
8. Click third tab and change properties:
   * Name: pgPetVisits
   * Caption: Pet Visits
9. View, test, save, alter if needed.

## 7.7 Build a Navigation User Interface

To provide the user a more user-friendly interface, and to restrict the user’s ability to edit the database design, Access has a built-in switchboard manager to help create a switchboard interface. The switchboard is the applications starting point to give the user the ability to do specific tasks in the application.

A switchboard is usually a hierarchical collection of small forms with command buttons (or other controls) to execute common tasks, such as opening forms, printing reports, etc.

Prior to designing the switchboard, the designer should plan and layout a navigation diagram to illustrate the links between the application screens.



In this example, four buttons exist on the switchboard. Three link to a form while the fourth exits the application. Before using the switchboard manager, the objects that are linked to the switchboard buttons should already exist.

How to:

1. Under the Database Tools tab, click the Switchboard Manager button.
   * If this tool is not available, open up Access Options -> Customize Ribbon -> Create a custom group under Database Tools -> Select All Commands on left pane and find Switchboard Manager -> Click the add button.
2. If a switchboard does not already exist in your database, you will be prompted to create one: Click **Yes**
   * ![Graphical user interface, application

     Description automatically generated
3. A “Switchboard Items” table is created, and the Switchboard Manager window is displayed. Click the **Edit** button
   * Graphical user interface, text

     Description automatically generated
4. The Edit Switchboard window is displayed:
   * Graphical user interface, text, application, email

     Description automatically generated
5. To add a button to the switchboard, click the New… Button
6. The Edit Switchboard Item window is displayed. Enter the following information about the first button:
   * Graphical user interface, application, email

     Description automatically generated
7. Add another:
   * Graphical user interface, application, email

     Description automatically generated
8. And another to exit the application:
   * Graphical user interface, application

     Description automatically generated
9. Close the switchboard manager

*Note: Access 2013 has several features that lessen the need for switchboards, including editing the navigation pane and creating custom ribbons. Customizing the navigation pane will be covered in the next step of this outcome. Custom ribbons require some advanced skills that are not covered in this outcome.*

Let’s create a navigation form instead. We can use the Navigation Form Tool in the forms group in the Create ribbon. The Navigation Tool will create a form with tabs along the top or along the sides which can be linked to other forms we have already created. We will select a simple layout with **one row horizontal tabs** which will be used to open the following forms:

1. Navigation Tab 1: frmCustomer
2. Navigation Tab 2: frmPets
3. Navigation Tab 3: frmVisits
4. Open the Kelsey Vets database and click on Create tab
5. Click on the Navigation Control tool from the Forms grouping and select **Horizontal Tabs**
6. Add the 3 forms by dragging them from the Navigation Pane on the left to the **(Add New)** area. This adds a new tab with your desired form.
7. Graphical user interface, application, table, Word

   Description automatically generated
8. Save as **frmNavigation**
9. Once we created an appropriate Navigation user interface, we may wish to limit what the users can interact with to only the user interface. This can be done by opening the **File** menu, **Options**, and selected **Current Database**. Set the **frmNavigation** form as the **Display Form** and remove the **Display Navigation Pane** check box to prevent the default navigation pane from being displayed.
10. After the DB has been closed and reopened, only our navigation form will be displayed to the users to interact with. The default navigation pane will no longer be available on the left hand side.
11. **IN THE EVENT THAT YOU NEED TO CHANGE THE DB SCHEMA, and the navigation pane is removed, HOLD DOWN THE SHIFT KEY WHILE OPENING THE DATABASE WHICH WILL OPEN IT IN ADMIN MODE.**
12. Graphical user interface, text, application, email

    Description automatically generated

# Add Command Buttons

To add command buttons that (for example) open the Treatments and Medications forms from the Maintenance form, complete the following steps:

1. Create forms for tblTreatment and tblMedications using the Form Wizard tool after selecting the appropriate table.
2. Save the two forms as frmTreatment and frmMedications
3. Open a new blank form in Design View
4. Select the Use Control Wizards button from the Controls group on the Design ribbon.  (You will need to expand the Controls group using the pull down on the right side.)
5. Click on the Command Button tool and draw a button on the new form grid.
6. The first Command Button Wizard window is opened and displays two list boxes. The first list box displays the Categories of actions while the second list box displays specific Actions associated with each category. Select the Form Operations category in the first list box to display the list of available Form Actions in the second list box.
7. Click on the Open Form action in the second list box and click Next.
8. Select the frmMedications from the list box objects to open and click Next.
9. Accept the default display Open the form and show all records. Click Next.
10. Click on the radio button adjacent to the text option and type in Open Medications Form. Click Next.
11. Change the command button name to cmdOpenMedicationsForm. Click Finish.
12. Repeat the Steps 6-11 to add a button that will open the Treatments form.
13. If you have to resize the form: Change the Maintenance Form:
14. Properties:
    * Caption: Maintenance Form
    * Pop Up: Yes
    * Modal: Yes
    * Border Style: Dialog
    * Record Selectors: No
    * Navigation Buttons: No
    * Scroll Bars: Neither
    * Open the form, and resize the window. Close and save the form with the new size
15. View and test the completed Maintenance form.
16. It is also possible to change the style, shape and colour of the **Control Button** using the various options in the **Control Formatting** group on the **Forma**t ribbon of the **Form Design Tools**.

## 7.8 Design a form to solve a specific business problem

Do assignment ☺