Tutorial 2

Forms

Forms

You can view, add or edit data in Access by opening them in datasheet view. However, it is generally better to use Forms.

A form is an arrangement of *controls* you can use to view, add or edit data in Access.

There are two basic reasons to use forms.

- They make the job of editing , adding and viewing data easier
- They add many features and capacities over a datasheet when it comes to adding, editing and viewing data.

You can use forms for a variety of purposes

- Create a data entry form to enter data into a table
- Create a menu to open other forms or reports
- Create dialog boxes to accept user input

Most of the information in a form comes from an underlying record source: forms are usually linked to a Table or a Query.

You create the link between a form and its record source by using graphical objects called controls.

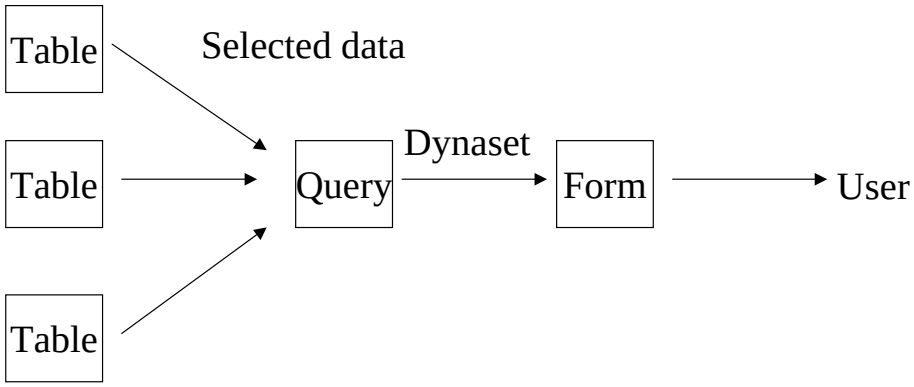
Certain conditions must be met before a form can be created:

- A form may be based on one or more tables or a query.

- If you want to produce forms that use data from several tables the relationships between those tables must be created first

- A record source is not required for specialised forms such as dialog boxes and menus.

A form displays only the information you want to see in the way you want to see it. It is usually best to produce a query on which to base a form. The query will generate a dynaset, or subset, of the desired data in the desired order.



In this example, a query extracts data from several tables according to selection criteria. The query produces a dynaset. When data is altered in the dynaset, the data in the underlying tables is changed. A form (or report) uses the dynaset as its underlying data source, showing the required data to the user. The dynaset may include data calculated in the query and derived from the data held in the tables, but not actually held in there.
Note that this whole process is outcome based - the application developer needs to analyse what the desired output of the system is in order to design the appropriate query.

Form Controls

Forms contain controls.

The commonest controls are:

Labels	to label text boxes or place text on the form
Text Boxes	to display enter and edit data from the underlying data source
Combo Boxes	to select data items
Buttons	to automate tasks such as opening other forms

A control that is linked to a field in an underlying table or query is **bound** to that field. Any changes made to the control also change the data in the field.

A control that is not linked to a field is called **unbound**. You can use an unbound control as a temporary storage location for data e.g. to show the result of a calculation on other controls.

A form may also contain a sub-form, showing related data.

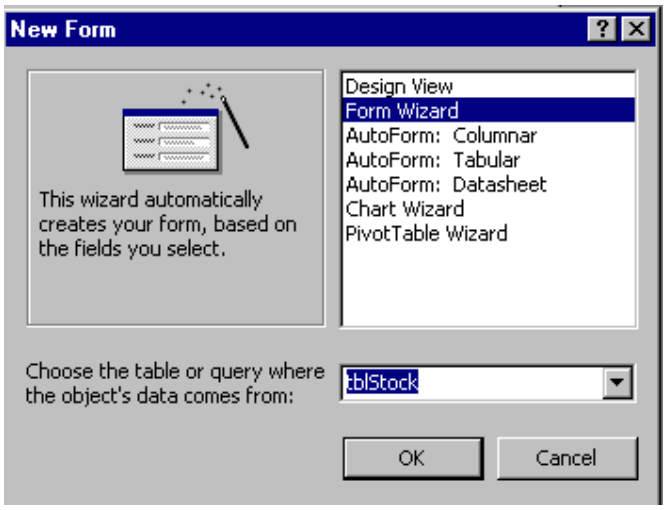
Creating a form using the Form Wizard

You can create a form on your own or you can have Microsoft Access create your form for you using a Form Wizard. A Form Wizard speeds up the process of creating a form because it does all the basic work for you. When you use a Form Wizard, Microsoft Access prompts you for information and creates a form based on your answers. Even if you've created many forms, you may want to use a Form Wizard to quickly lay out all the controls on your form. Then you can switch to Design view to customize your form.

Creating the Stock Entry form

Select the Forms tab in the Database window, and click New.

In the New Form dialogue, select the Form Wizard and choose tblStock as the table where the data comes from.



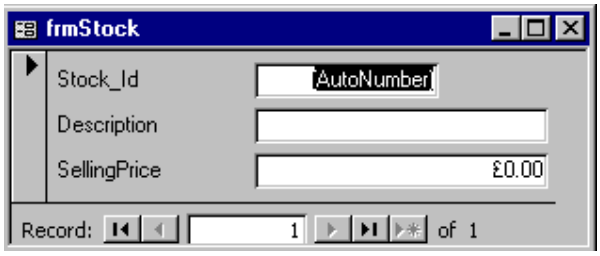
In the next dialogue, select all the available fields

In the next dialogue, select columnar layout

In the next dialogue, select standard layout

In the next dialogue, type frmStock as the title of the form, and then click Finish

The form will look like this:



The form contains Text Box Controls bound to all the fields from the underlying Stock table. Sa

This form can be used for entering, viewing or editing data.

Add some Stock records using the form. Notice that the Escape key allows *rollback* i.e. undoing the last action. At times rollback will reinstate a field, at other times a whole record.

Notice that you cannot enter the Stock_Id. A unique value is entered automatically for each new record.

Practice using the navigation buttons at the bottom of the form.

Changing the look of the form.

The form produced by the wizard is very basic and rather unattractive. You are now going to change the properties of the form to improve its appearance.

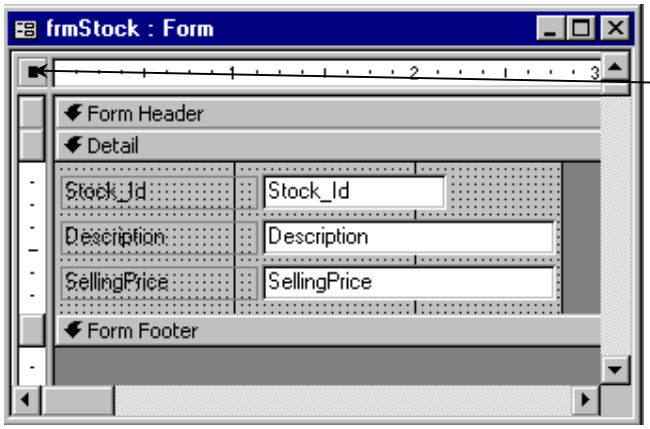
A property is an attribute of an object that defines one of the object's characteristics, such as size, colour, or screen location, or an aspect of its behaviour, such as whether it is enabled or visible. To change the characteristics of an object, you change the values of its properties in a property sheet.

To get the property sheet for an object, you select the object and select Properties from the right button menu. This will open up the Properties sheet.

Open the form in Design view. You can either do this from the database window or, if the form is already open, by clicking on the design icon:

Open the properties sheet of the form by clicking the Properties icon

or by clicking the area shown in the diagram below:



Double click on the square to view the form property sheet

Form Property sheet

Notice that the Record Source refers to the table that is supplying the data which the form will show.

Alter the following Form properties.

Scroll Bars	Neither
Record Selectors	No
Border Style	Dialog
Min Max Buttons	None
Width	8”
Caption	Stock
Dividing Lines	No

Note: if your measurements are in centimetres rather than in inches, you can change this in the Windows Settings / Control Panel / Regional Settings / Number.

Click on the Form Header bar, and change the following Form Header property

Height .5

Click on the Detail bar, and change the following Detail property

Height 4”

You do not need or want the user to have access to the Stock_Id control since it is a computer-generated Autonumber.

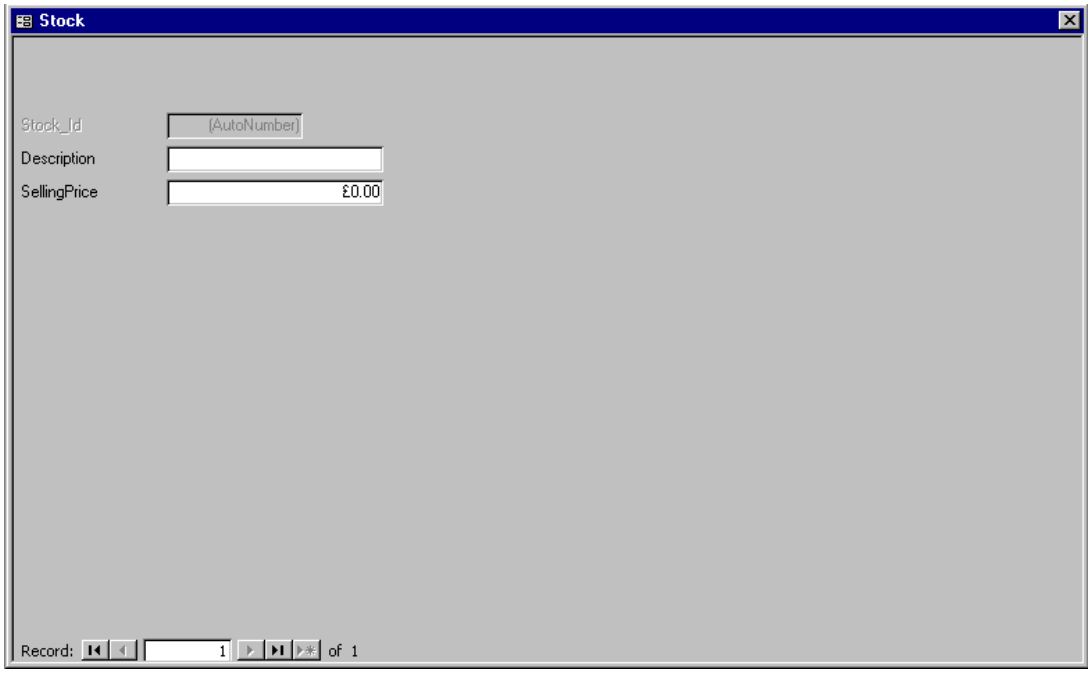
Disable the Stock_Id control by changing the following properties [double click on the Stock_Id control]:

Enabled No

Later, you can change the control's Visible property to False so that it did not appear on the form in Form view. However, when developing an application, it is often quite useful to see the primary keys just to check everything is working properly.

Save the form and view it by clicking the View icon

Your form should now look like this:

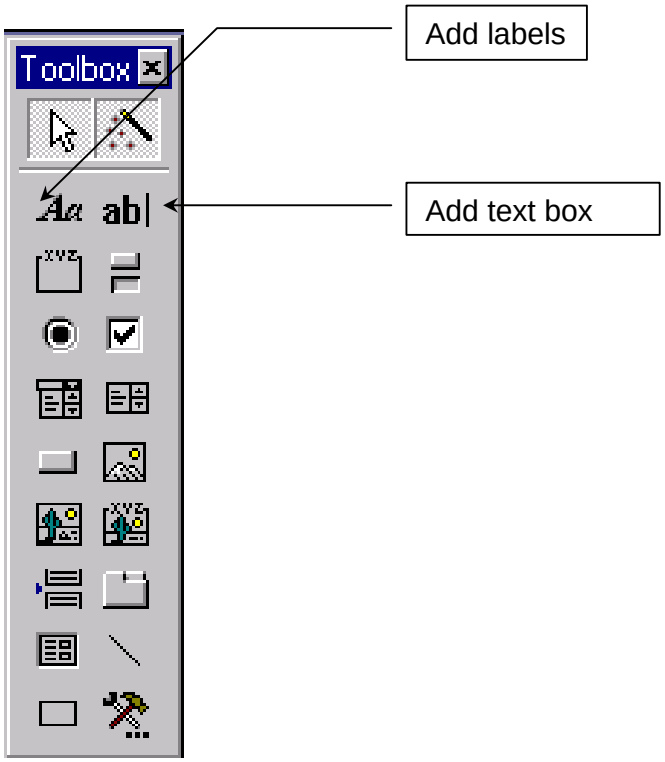


Re-designing the look of the form

Various tools are available from the Toolbox to change the look of the form.

Click on this icon on the tool bar to see the Toolbox: 

The Toolbox



Note that the layout of the textbox may be slightly different to this on your machine, since it can be customised.

Adding today's date to the Header

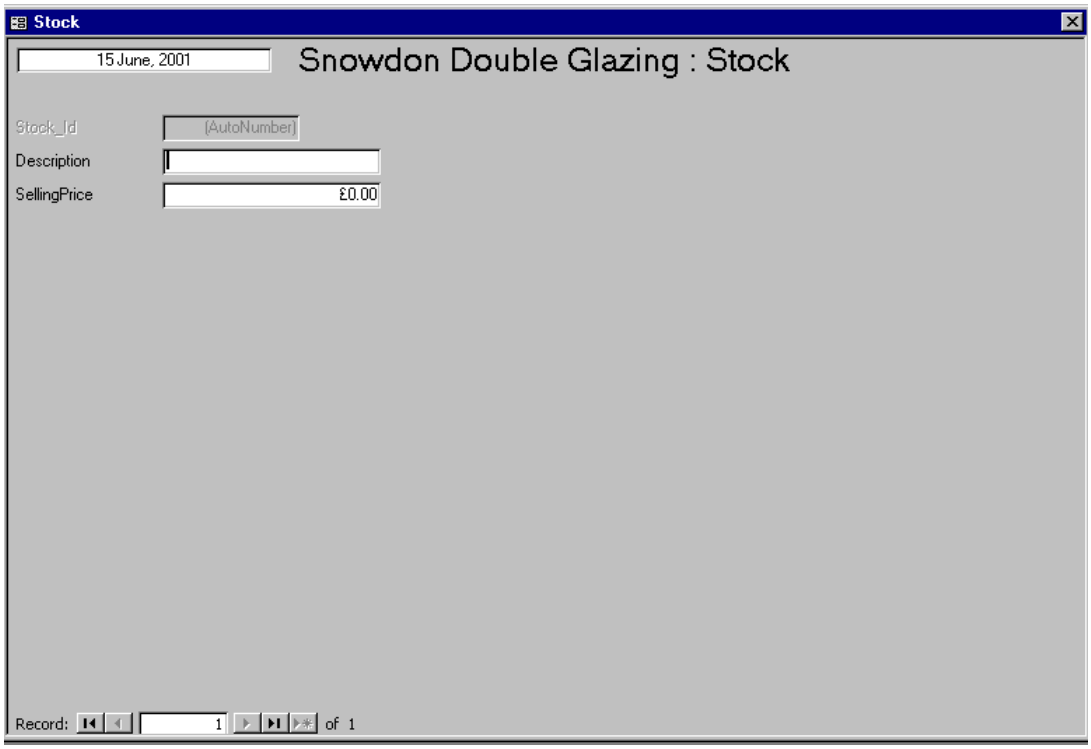
Place a new Text Box in the form Header.
Delete the Label.

Change the following properties of the Text Box:

Control Source	=Date() *This function reads the date from the computer*
Format	Long Date
Enabled	No
Locked	Yes

Adding a banner to the Header

Place a new Label in the form Header to produce a banner as in the diagram below.

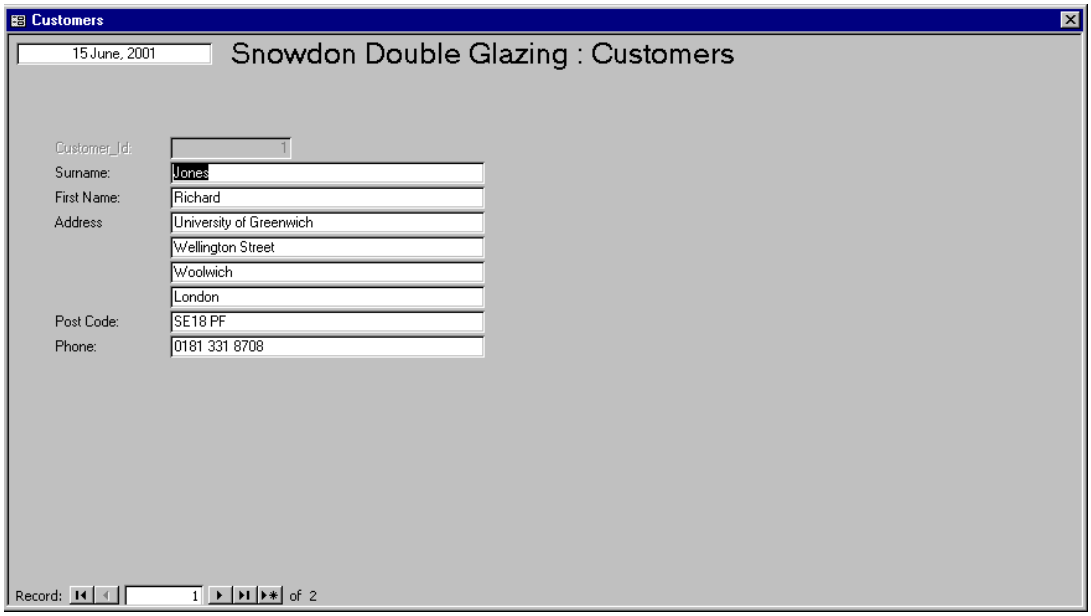


Use this Stock form to populate (add records to) the Stock table.

Customer Entry Form

Create a Customer Entry form based on tblCustomer.

Final Version - getting it to look exactly like this takes time!



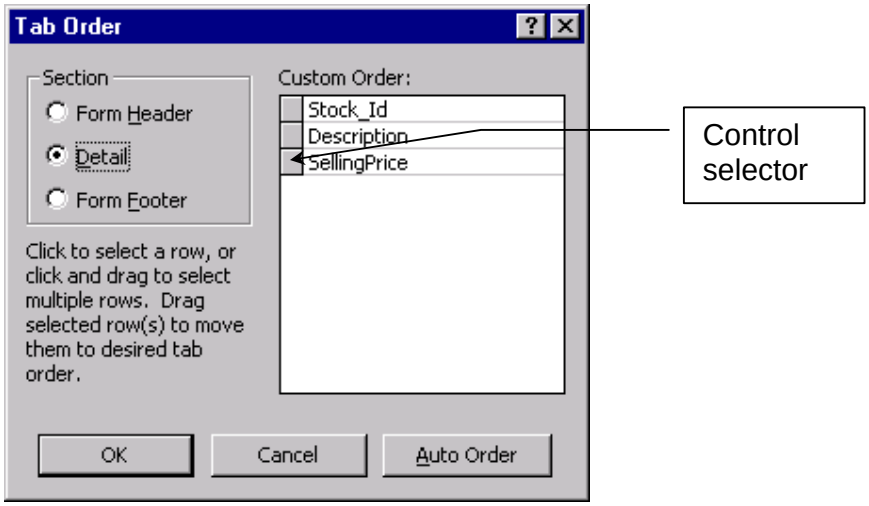
Use the form to populate the Customer table i.e. put some customers in your database.

Changing Tab Order

Sometimes you have to re-arrange text boxes on the form. You will find that you will then want to change the order in which controls are entered when the tab key is pressed

Either go to
View Menu
Tab Order

Or right click on the form and choose Tab Order

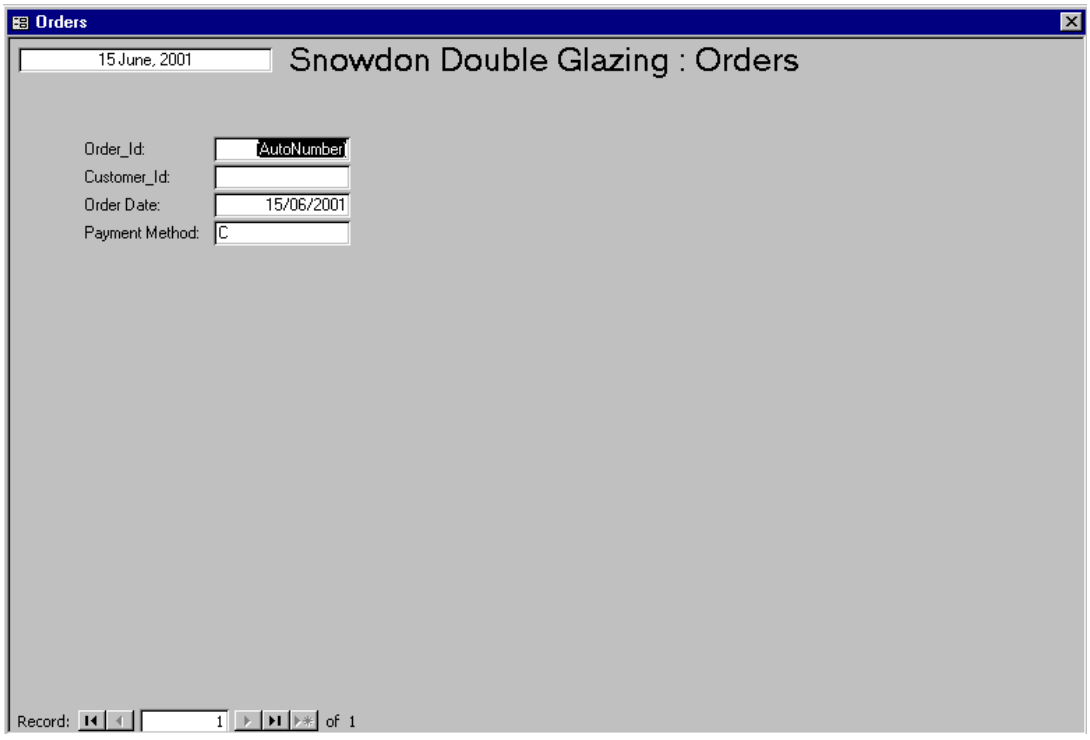


Click the selector for the control you want to move. (Click and drag to select more than one control at a time.) Click the selector again and drag the control to the desired location in the list. When you have finished, click OK. Don't forget to test your form to see that it works correctly.

Order Entry Form

Create an Order form based on tblOrder.

To begin with, you will get a result something like this:



Why is the date appearing in Order Date, and C in Payment Method?

The main problem with this as an entry form is that it would require the user to know the ID of the customer making the order. This of course would be impractical.

Instead, we would like the user to able to select a Customer by looking at the customer's Surname and Forename.

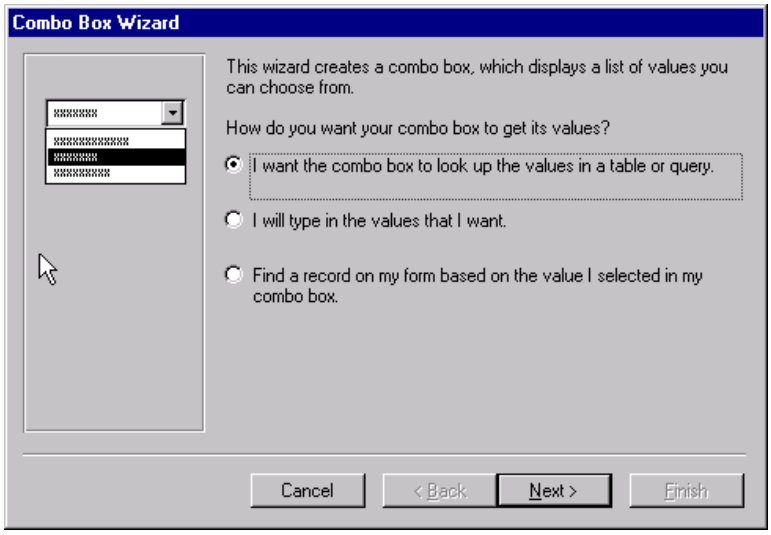
This can be accomplished with a Combo box.

Using a Combo Box to populate the Customer_Id

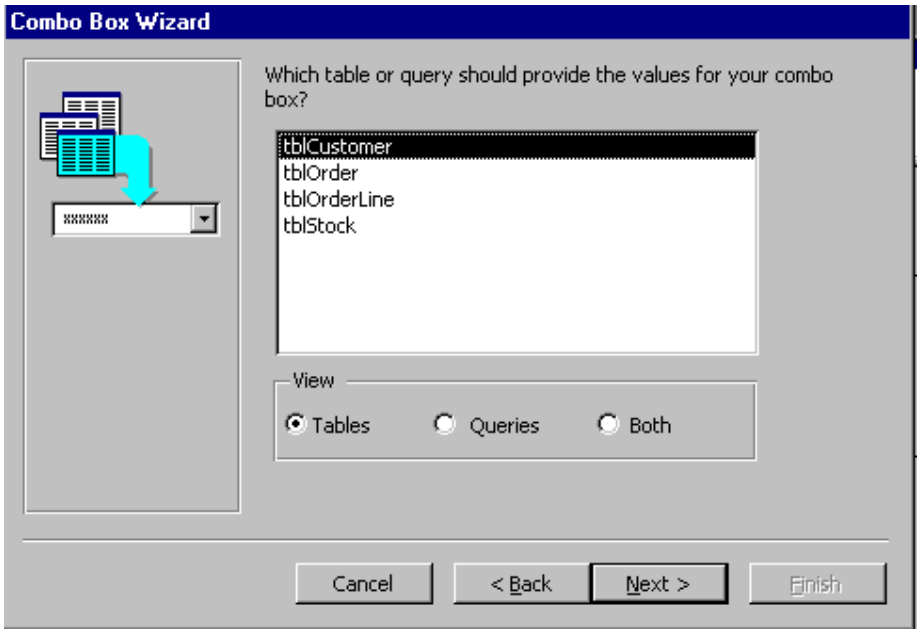
Delete the *Customer_Id* label and text box controls from the *Order Entry* form. They are going to be replaced by a Combo Box.

Make sure the Control Wizard button is on in the ToolBox
Select the Combo tool from the ToolBox,

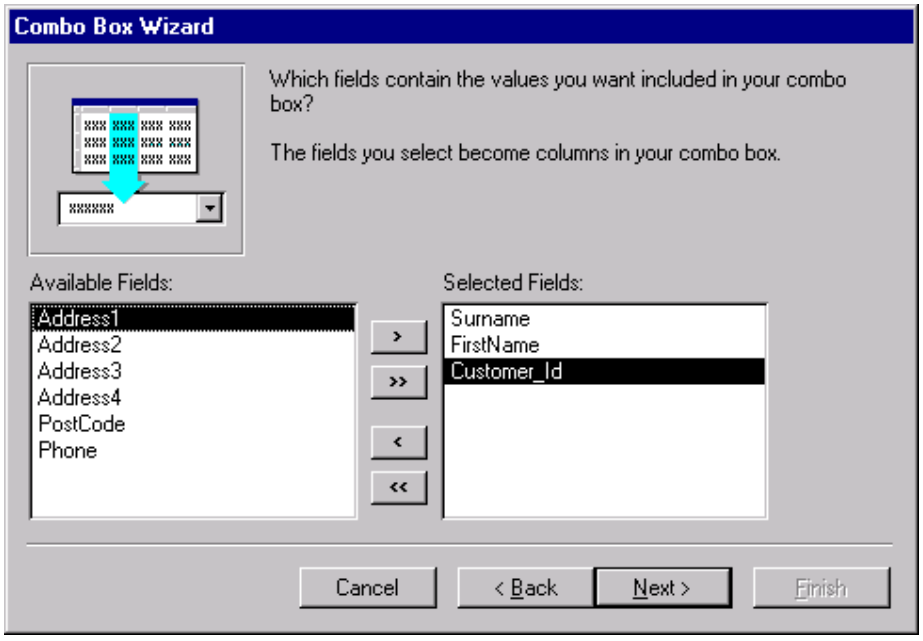
and position it on the form



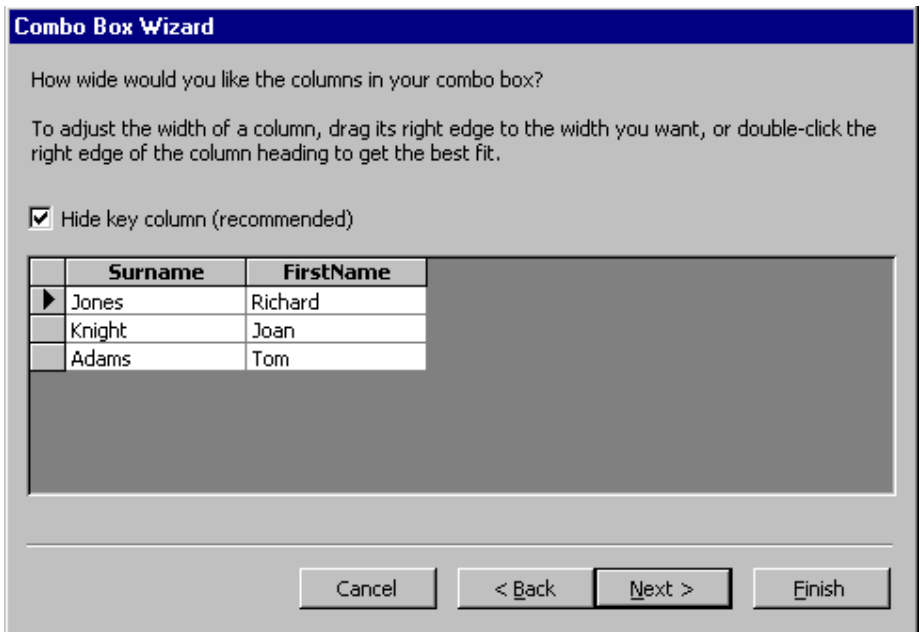
- Select the Combo box to look up its values from tblCustomer



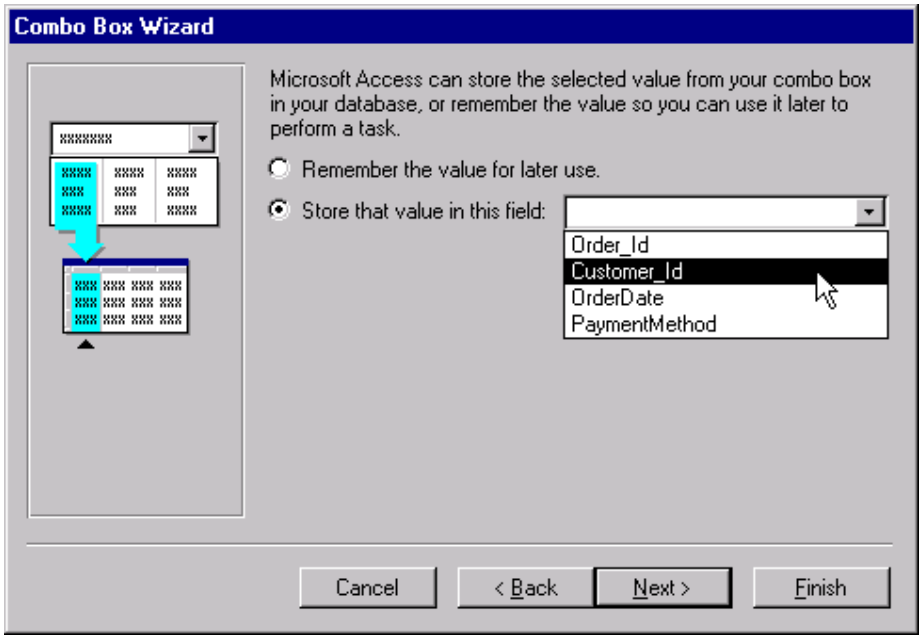
- Select the following fields IN THIS ORDER to include in the Combo Box - Surname, FirstName, Customer_Id. The first field [Surname] is the one which will eventually appear in the combo box on the form.
-
- You must include the Customer_Id field, since this is the data we are trying to capture in order to store in the Order table.



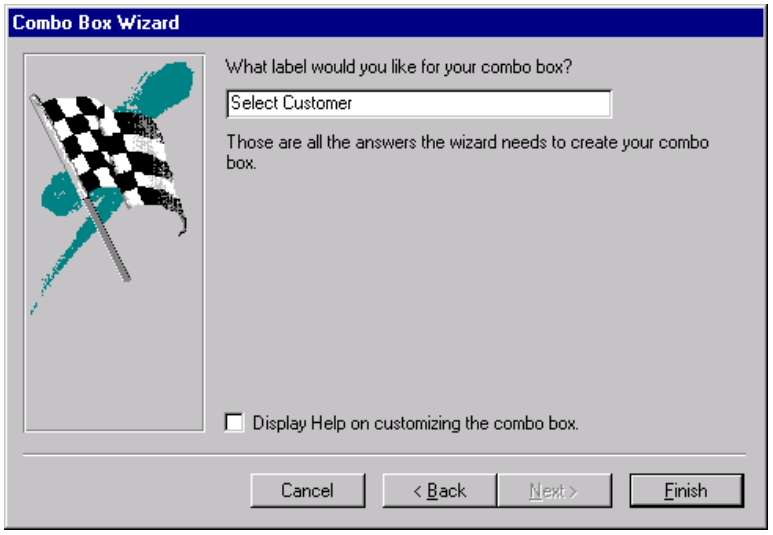
If necessary, adjust the width of the columns to accommodate the data. Notice that the primary key (Customer_Id) is automatically hidden from the user, even though this is the actual data being sought.



- Customer_Id is the selected value which you want to use in the database
- You want to store that value in the field : Customer_Id in the underlying tblOrder Table

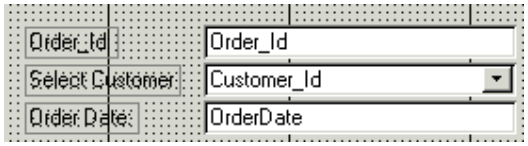


- Choose a suitable label for the combo box e.g. Select Customer



and Finish

The end result: [you may need to adjust the label size, alignment and tab order]

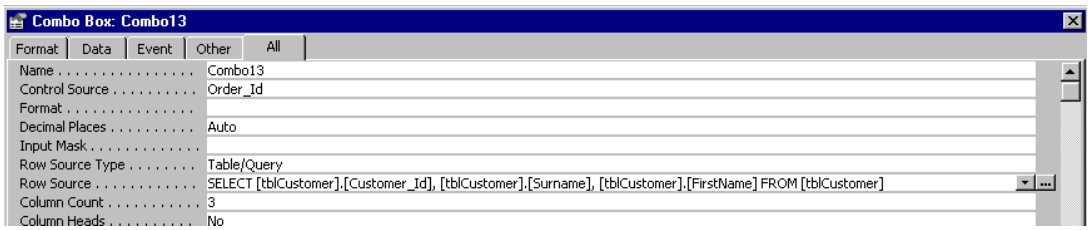


The Combo Box is

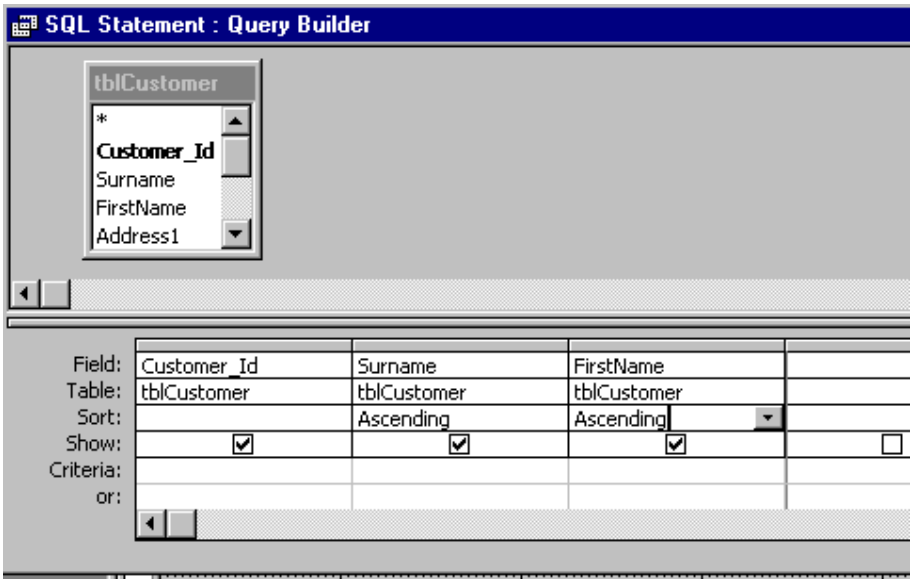
- **bound** *to the Customer_Id field in the tblOrder table*
- *getting its values from the tblCustomer table*
- *showing the user the Customer surname*

Open the form and try using the combo box. You will see that the customers do not appear in alphabetical order. You can change this.

Open the form in design view. Open the property sheet of the combo box.



Click your mouse in the entry Row Source and then click on the button with the three dots [the Expression Builder]. You are now in the Query Builder. Change the Sort row as shown:



Close the Query Builder and answer Yes to the Save changes dialogue.

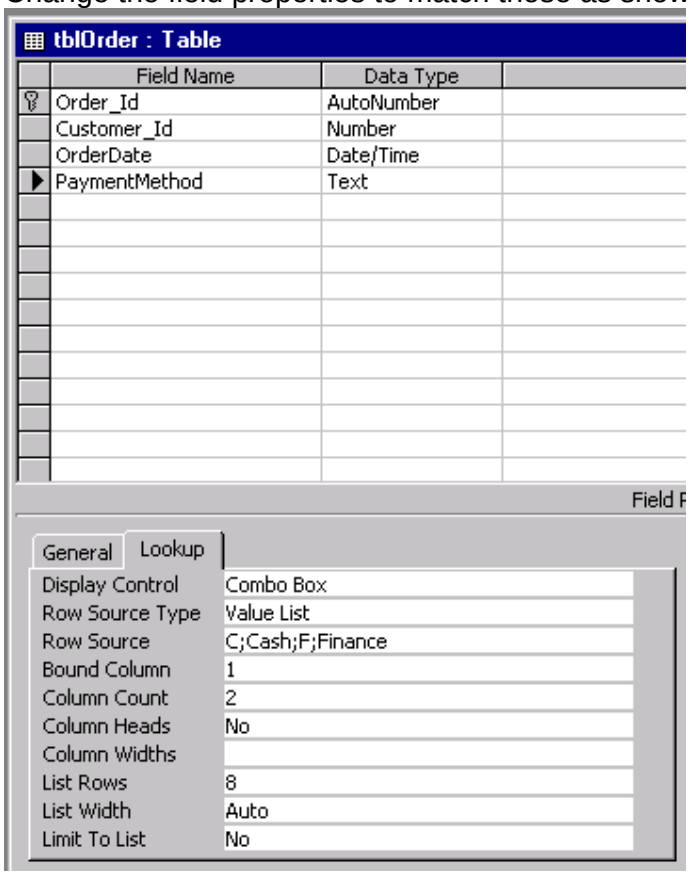
Close the properties sheet. Open the form and try out the combo box again.

We will do more with Queries later.

Using Lookup to produce a Combo box.

We are now going to use another way to create a combo box - this time for PaymentMethod.

Open the Order table in design mode.
Select the PaymentMethod field and click the Lookup tab.
Change the field properties to match those as shown :

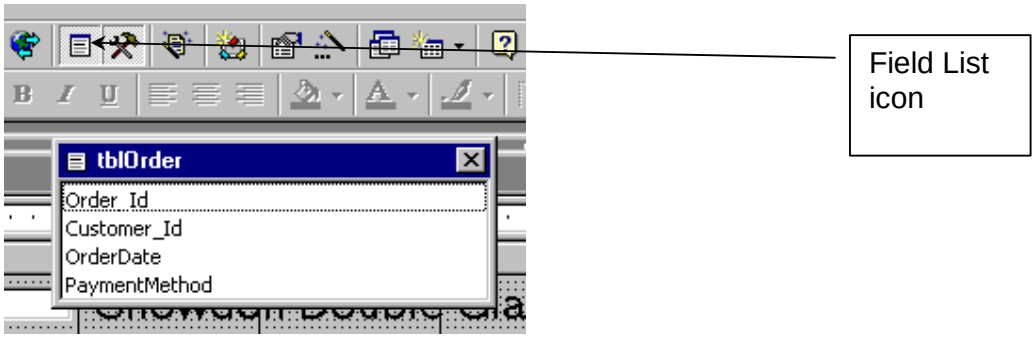


When a control associated with this field is placed on a form, it will automatically become a combo box. Rather than getting its data from a table or query, this combo box will get its data from the list of values typed in the property *Row Source*.

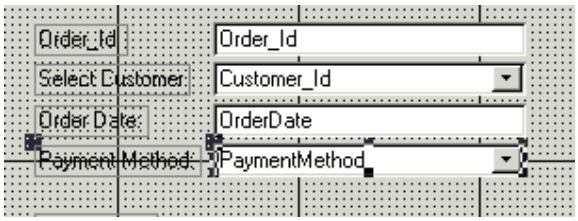
Notice that 2 columns will be shown to the user. The first column contains the value which is Bound to the data field i.e. the data which will be saved. The second column shows data helping the user to make the choice.

Open the Order Entry form in design mode.
Delete the current PaymentMethod control.

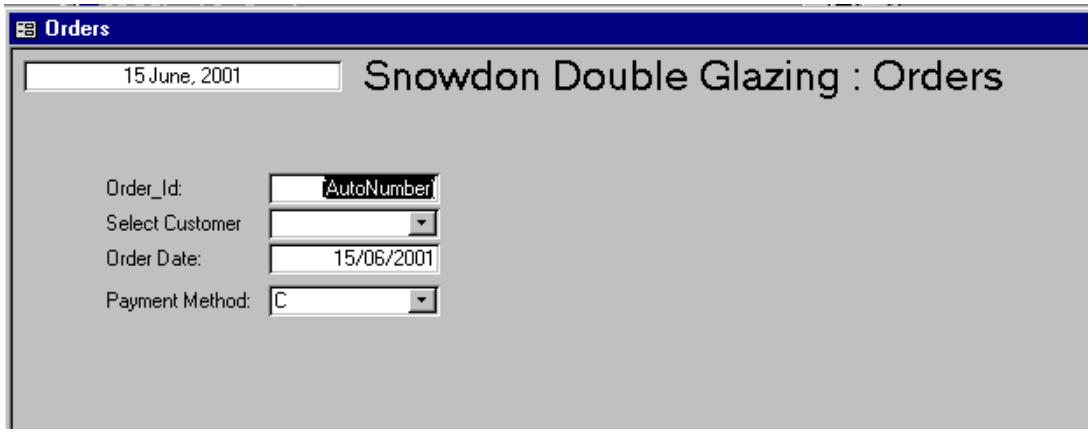
Click the Field List icon to show the list of fields in the underlying query / table



Select PaymentMethod from the Field List and drop it onto the form.



The end result on screen should look something like this :



You can adjust the column widths for legibility.

CHECKPOINT: SDGCheck2