dmu.ac.uk
DE MONTFORT UNIVERSITY
LEICESTER

Department of
Library Services

www.library.dmu.ac.uk

IT Training
Kimberlin Library
Tel: (0116) 257 7160
Fax: (0116) 257 7046
email: itskills@dmu.ac.uk

Office 2007

Access

Intermediate

May 2010

Contents

INTRODUCTION

This training guide is aimed at those who already have experience of using Access 2007 and want to expand their knowledge of the application and databases just a little bit further.

Knowledge assumed

you will have either attended the Access 2007 Getting Started training course or have experience of using databases in Access 2007
experience of creating tables
experience of creating and modifying select queries
acquainted with labels and text boxes on forms or reports

Areas covered

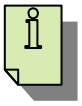
using wizards to create Access objects
building and modifying reports
using action queries
building and modifying forms
using calculated controls on forms and reports
using property sheets
introduction to macros

Document signposts

Instructions for you to type

Bold text

Shortcuts



Reminders

Notes

Exercises

DATABASE CONCEPTS

WHAT IS A DATABASE?

A database is defined as an organised collection of related data. It can be regarded as an electronic filing system that can hold thousands, even millions, of items of data.

For example, a database might be used to store data relating to the stock in a mail order firm and the orders received, or data about the students at a college, the courses they have enrolled on, and the examination marks received.

Once the data has been entered, you can retrieve specific information from it very quickly.

DATABASE OBJECTS

Every tangible aspect of a database is referred to as an object.

All data is stored in objects called **tables**. Each row of a table is called a **record**, and each record is made up of individual items of data called **fields**.

For example, each record in a 'course delegates' table will contain data about an individual delegate, such as ID number, last name, first name, department etc. In this case the fields are ID number, last name etc. When designing a database it is important to carefully consider the fields and tables that will be needed to store the data.

A **query** is used to interrogate the database and filter the data to pick out specific information from it.

A **report** is a way of presenting information from the database in a format suitable for printing.

A **form** is an on-screen interface that enables you to view, enter and edit the data in a manner that is more convenient and user-friendly than using the tables directly.

A **macro** is a sequence of commands to accomplish a particular task. When the macro is run, the commands are executed automatically and the task accomplished.

A **relationship** is a link between data in one table and data in another table. It acts like a cross-reference between the two tables.

WHAT IS A PRIMARY KEY?

A primary key is a field that uniquely identifies each record in the table.

In order to set up a relationship between two tables and show how data in one relates to data stored in the other, you need to identify a field in one of the tables where the data will be unique for every record.

In many cases it is convenient to set up an Auto Number field in the table and designate this field to be the primary key.

WHAT IS A FOREIGN KEY?

When a relationship is formed between two tables, a link is created between the primary key in one table and a field in the second table that contains values used to cross reference against the primary key values. This field in the second table is referred to as a foreign key.

WHAT IS AN INDEX?

An index is a data structure created within the database using the field(s) that you specify. This structure improves the speed of retrieving information from the table.

The primary key of a table is automatically indexed.

Access updates all of the indexes every time you add or update a record. Because of this, you should ensure that you only use indexes where you really need to, otherwise you may find that entering and editing becomes noticeably slower.

PLANNING AND DESIGNING A DATABASE

Before you create a database you must give it a lot of thought and plan how you will construct it. Here are some questions that may be useful in assisting your planning:

- What will the data be used for?
- What information will be needed from the database?
- What data needs to be stored?
- How will the data be divided into fields? What will be the data types? What field properties need to be set?
- How will the fields be organised into tables?
- If there are to be two or more tables, how will the tables be related?

When considering the fields that you need, remember that each field should only contain a single element of data. This implies, for example, that one field should not be used to store a delegate's full name, but the title should be stored in its own field, first name in its own field etc.

Amongst the many good reasons for this is the fact that it is easy to combine the contents from different fields, but very difficult to do the opposite operation and split the contents of a field into separate parts.

RELATIONAL DATABASES

A relational database is a database that consists of two or more tables linked together by the creation of relationships.

A very important decision to be made when planning a database is if and how the fields will be split into different tables. When making this decision, remember that, in order to achieve an efficient and well designed database, each table should contain data about one, and only one, entity.

For example, in a 'course delegates' table every field should be about the delegate and nothing else. This makes the database more flexible, enabling you to add other tables at a later time without altering the existing table(s), and it makes the database more efficient because each item of data is entered only once. Consider for example the following list of fields that the admin staff at a college wish to include in a database to store evening class enrolments:

student first name, student last name, student address,
student postcode, student telephone, student date of birth,
course title, room number, course leader first name,

course leader last name, course leader telephone, course leader email, exam date, and *exam mark*.

Some of the fields will store data about the students: *student first name, student last name, student address, student postcode, student telephone*, and *student date of birth*. These fields would be put in a student details table.

Course title, and *room number* would be put in a course details table.

Course leader first name, course leader last name, course leader telephone, and *course leader email* are all about the course leader and therefore will be put in a separate tutor details table.

Last of all, an exam details table would be used to store the fields for the *exam date*, and *exam mark*.

Planning for relationships

Once it has been decided what tables are needed, it is time to consider the relationships that need to be created, and include appropriate fields to accommodate these relationships.

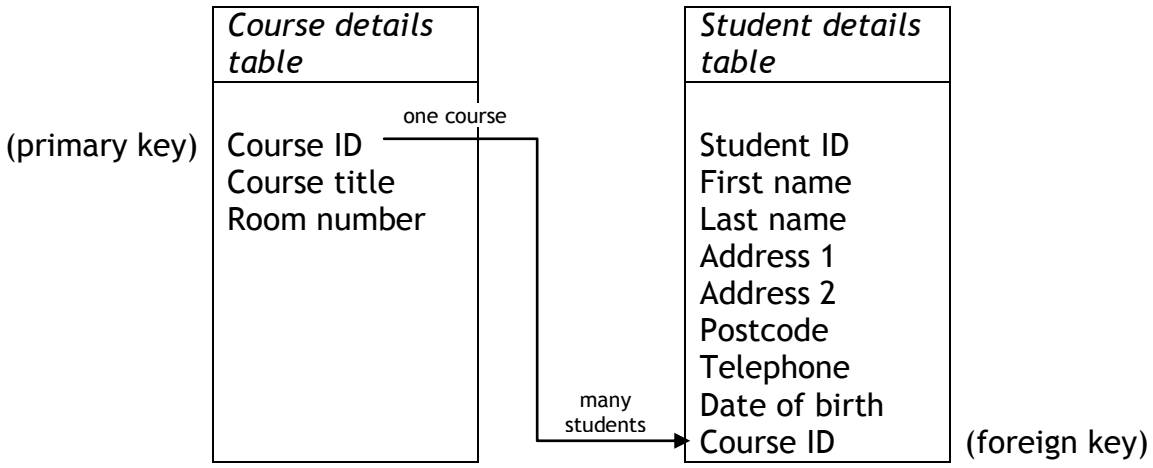
Most database designers include a primary key in every table even though some may not be needed for the relationships. This is considered to be good practice and some designers go as far as stating that each table should have a primary key.

To accommodate any relationships that may be needed, an auto number ID field will be added to each of the tables and designated to be the primary key: student ID, course ID, tutor ID, and exam ID respectively.

The next decision to be made is which of the primary keys must be added as foreign keys in other tables.

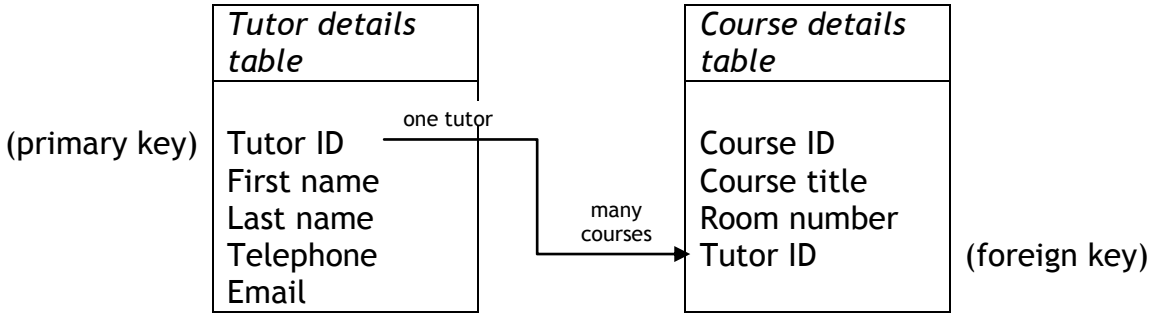
Consider the relationship between the student details and course details; it is desired that more than one student should be able to enrol on any one course. This type of relationship is called a one-to-many relationship.

To create a one-to-many relationship you link the primary key of the table on the one side of the relationship to the foreign key in the table on the many side of the relationship. The relationship must allow one course to many students, so you need to use the primary key in the course details table. Course ID must therefore be added to the student details table as a foreign key.



Similarly, it may be necessary for each tutor to be the course leader for any number of courses.

The relationship created between the course details table and the tutor details table must therefore be a one-to-many relationship, where the one end of the relationship is at the tutor details table (ie using the primary key field, tutor ID) and the many end is at the course details table. The tutor ID field must therefore be added to the course details table as a foreign key.

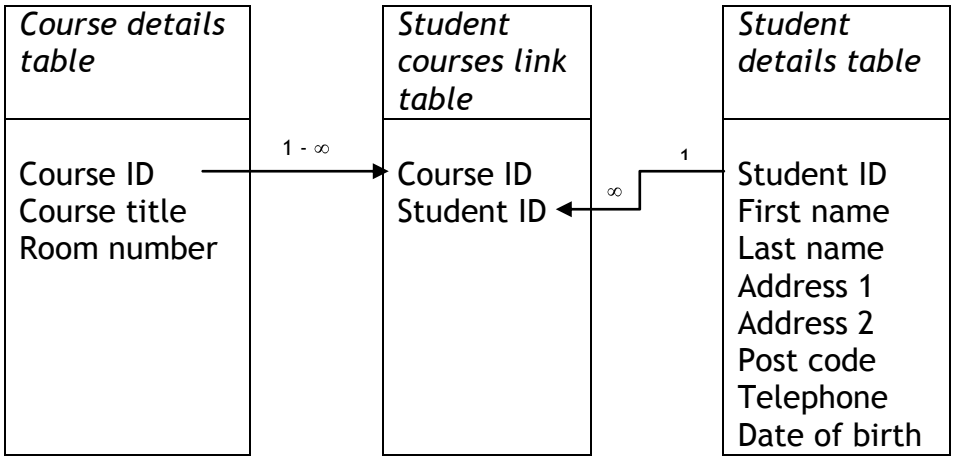


Many-to-many relationships

If the relationship between students and courses must allow each course to have more than one student and each student to do more than one course, this would be a many-to-many relationship.

It is not possible to create a many-to-many relationship directly between two tables. However, you can get around this by using an intermediate table and creating two one-to-many relationships.

So for a many-to-many relationship between the students and courses, an additional table needs to be created with the sole purpose of linking the student details table and the course details table. Although this link table can have additional fields, it will usually just have the two foreign keys that are necessary to create the relationships.



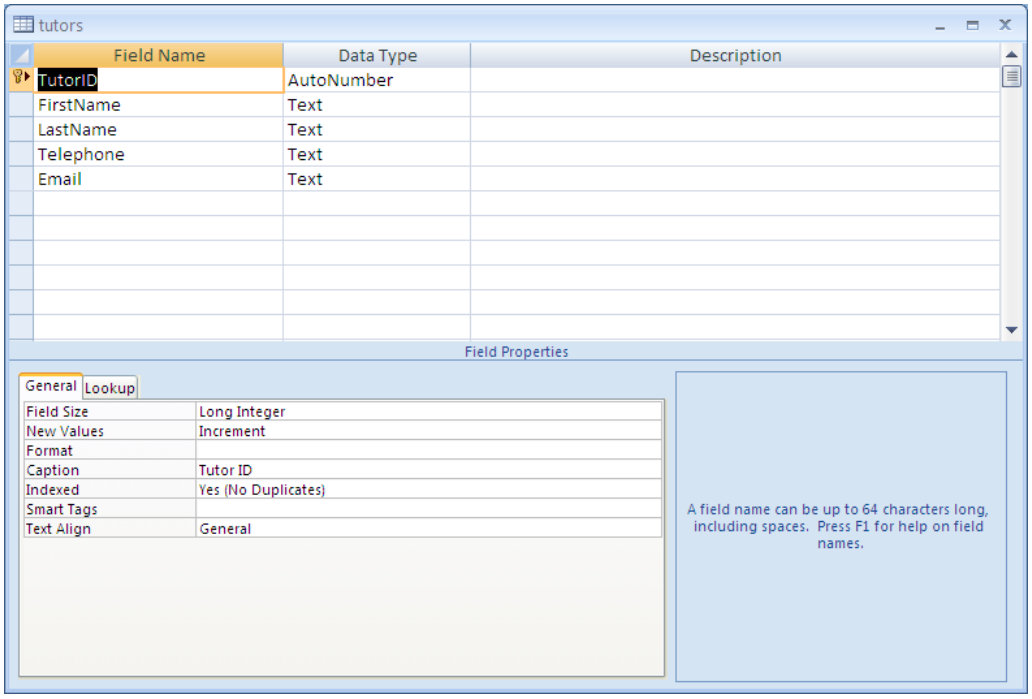
This is equivalent to a many-to-many relationship between the course details and student details tables.

RELATIONSHIPS PRACTICE

- Launch **Access**
- Open the **Enrolment** database from **C:\AccessIntermediate**

You will use this database to create two of the relationships described above. First you will create the one-to-many relationship between the tutors and courses tables allowing each tutor to lead more than one course.

- Open the **tutors** table in **Design View**



Observe that:

1. The TutorID field is the primary key
2. The Data Type is AutoNumber
3. The Field Size is Long Integer

- Close the **tutors** table

- Open the **courses** table in **Design View**

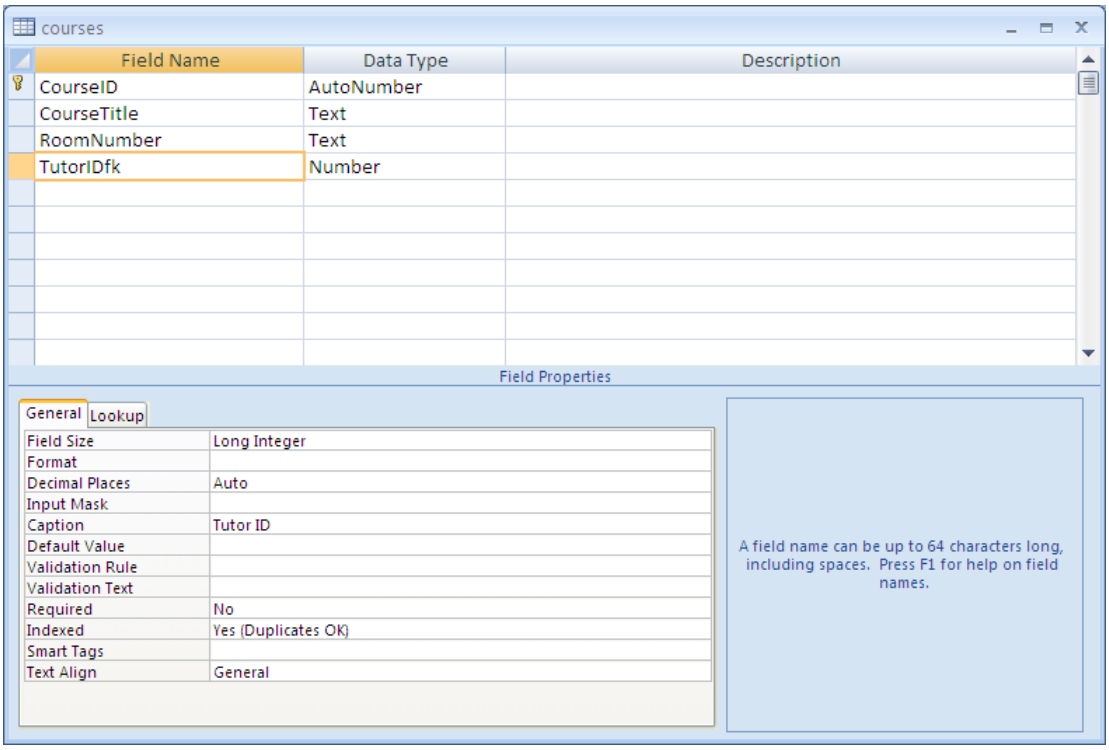
To add the TutorID field as a foreign key:

- Create a new field with Field Name **TutorIDfk** (using a slightly different name will distinguish it from the primary key in the tutors table)

- Set the Data Type to **Number** and the Field Size to **Long Integer**

Although the primary key in the tutors table has an AutoNumber data type, automatically producing a unique number for each new record, the values in the foreign key will only be assigned when the user indicates who the course leader is for a particular course and will not be unique since each tutor can lead many courses.

The data type for the foreign key, therefore, cannot be AutoNumber and must instead be Number. You must however ensure that the Field Size for the foreign key matches exactly with that of the primary key to which it will be linked, so it must be set to Long Integer.



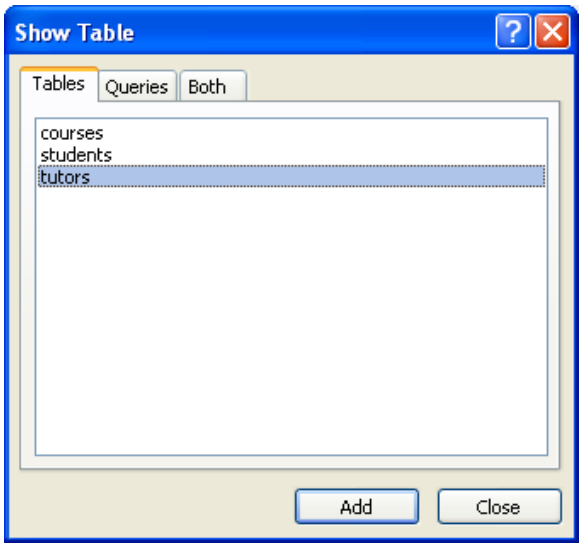
- Save and close the **courses** table

To create the relationship:

- Select **Relationships** from the **Show/Hide** group on the **Database Tools** tab

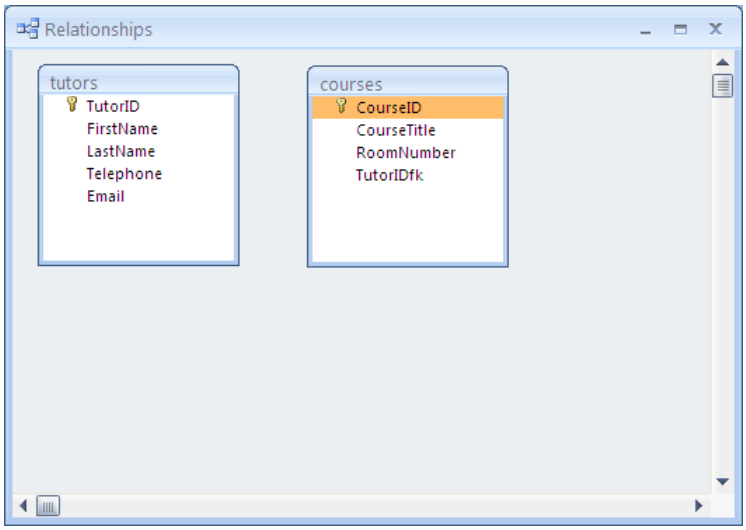
The Relationships window is displayed. No tables have been added to it as yet.

- Click **Show Table** from the **Relationships** group on the **Design** tab
- Select **tutors** in the Show Table dialog box



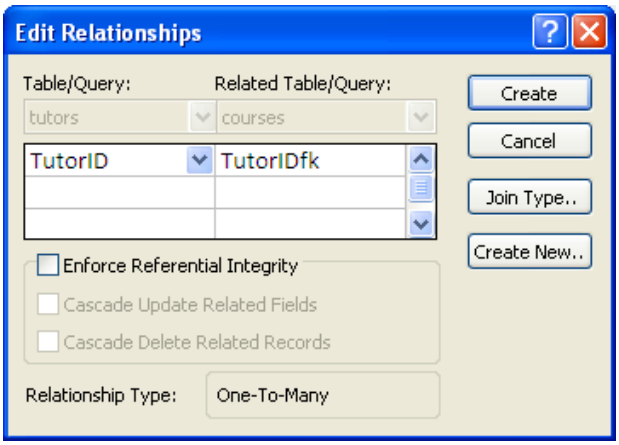
- Click **Add**
- Select **courses**
- Click **Add**
- Close the Show Table dialog box

The two tables are now displayed in the Relationships window.



- Hold down the mouse button over the **TutorID** field in the tutors table, drag the mouse pointer to the **TutorIDfk** field in the courses table, then release the mouse button

The Edit Relationships dialog box is displayed showing the tables and fields that have been selected.

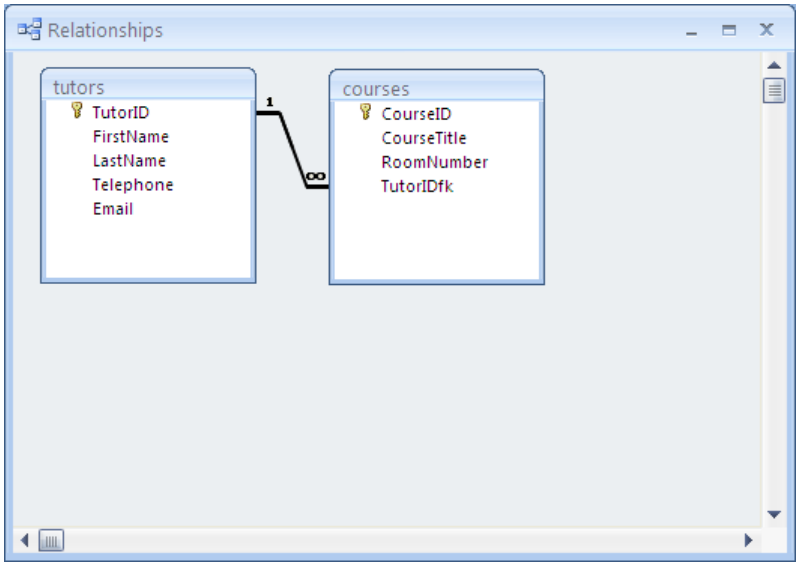


- Ensure that the **Enforce Referential Integrity** box is ticked

By selecting this option you ensure that every record on the "many" side of the relationship has a corresponding record on the "one" side. In other words, a course cannot be assigned a course leader that does not exist in the tutors table.

In addition, enforcing referential integrity prevents you from deleting records on the "one" side if related records exist on the "many" side. So for the relationship that you are creating, it will prevent you from deleting a tutor's record if there are records in the courses table referring to that tutor.

- Click **Create**



To save the changes to the Relationships window:

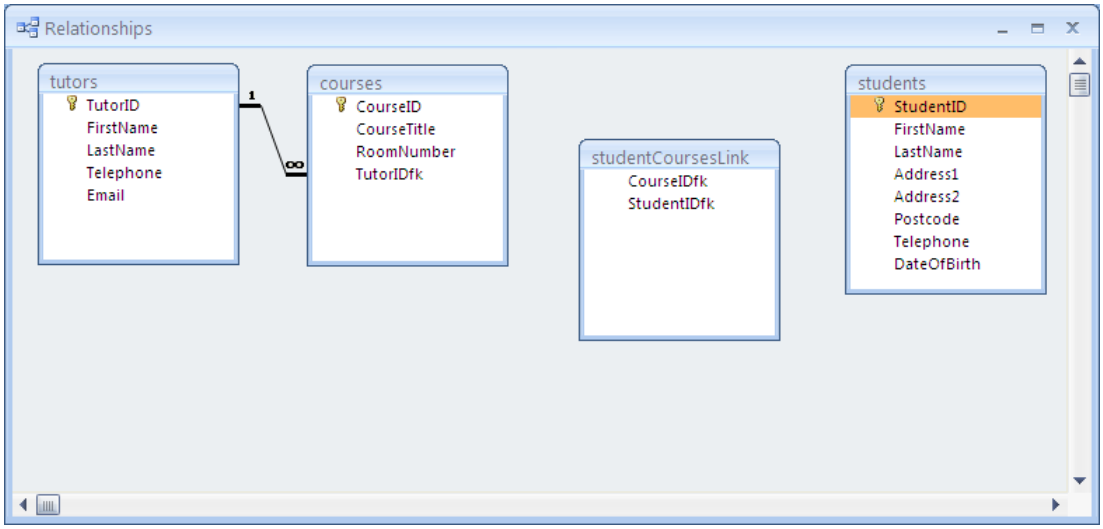
- Click the Save icon on the Quick Access Toolbar
- Close the Relationships window

You will now create a many-to-many relationship between students and courses by first creating a link table that contains the primary keys from the students and courses tables as foreign keys.

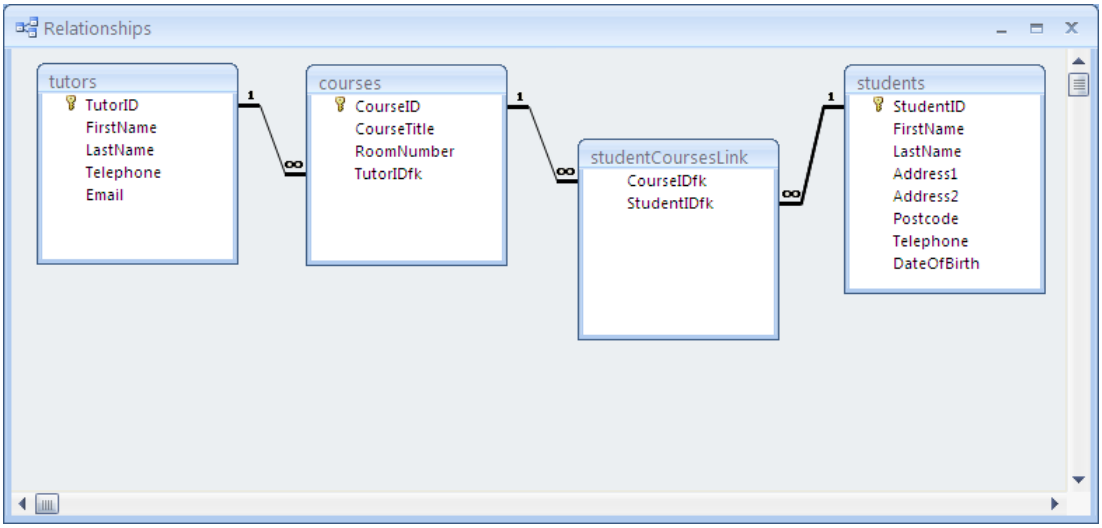
- Click **Table Design** from the **Tables** group on the **Create** tab
- Add the following fields:

Field Name	*Data Type*	*Field Size*
CourseIDfk	**Number**	**Long Integer**
StudentIDfk	**Number**	**Long Integer**

- Save the table as **studentCoursesLink** without setting a primary key
- Close the **studentCoursesLink** table
- Display the Relationships window
- Add the **studentCoursesLink** table and the **students** table to the Relationships window



- Create a one-to-many relationship between the **courses** table and the **link** table using the **CourseID** and **CourseIDfk** fields, and ensuring that Referential Integrity is enforced
- Create a one-to-many relationship between the **students** table and the **link** table using the **StudentID** and **StudentIDfk** fields, and ensuring that Referential Integrity is enforced



- Save and close the Relationships window
- Close the **Enrolment** database

NAMING CONVENTION

It is always a good idea to use descriptive names when naming database objects. Very long names should be avoided however, because they may result in typing errors when you need to refer to them later.

When choosing descriptive names it is sometimes useful to have more than one word in the name. Access allows the use of spaces in the names of some database objects, but because Access is the only database application that allows this, most database developers do not use spaces in object names.

The most popular methods employed when object names contain two or more words, is to use either the underscore character between words, or capitalise the initial character of the second and subsequent words in order to make it readable.

Apart from the sections that deal with Microsoft's templates, this training document uses the second of the two methods described above, so for example the 'first name' field in the students table is given the field name 'firstName'.

It is also useful for the name of a database object to give an indication of the type of object that it is associated with. Many database designers therefore add a prefix to the name to specify the type of object that it is.

In this document the following prefixes will be used when naming objects:

Object	*Prefix*
Table	no prefix
Query	**qry**
Form	**frm**
Sub form	**sfrm**
Report	**rpt**
Text box	**txt**
Combo box	**cbo**
Command button	**cmd**

USING DATABASE TEMPLATES

Database templates enable you to quickly create a database complete with working forms and reports.

Some templates are stored on your computer, but if your computer is connected to the internet then the online templates are also available for you to use.

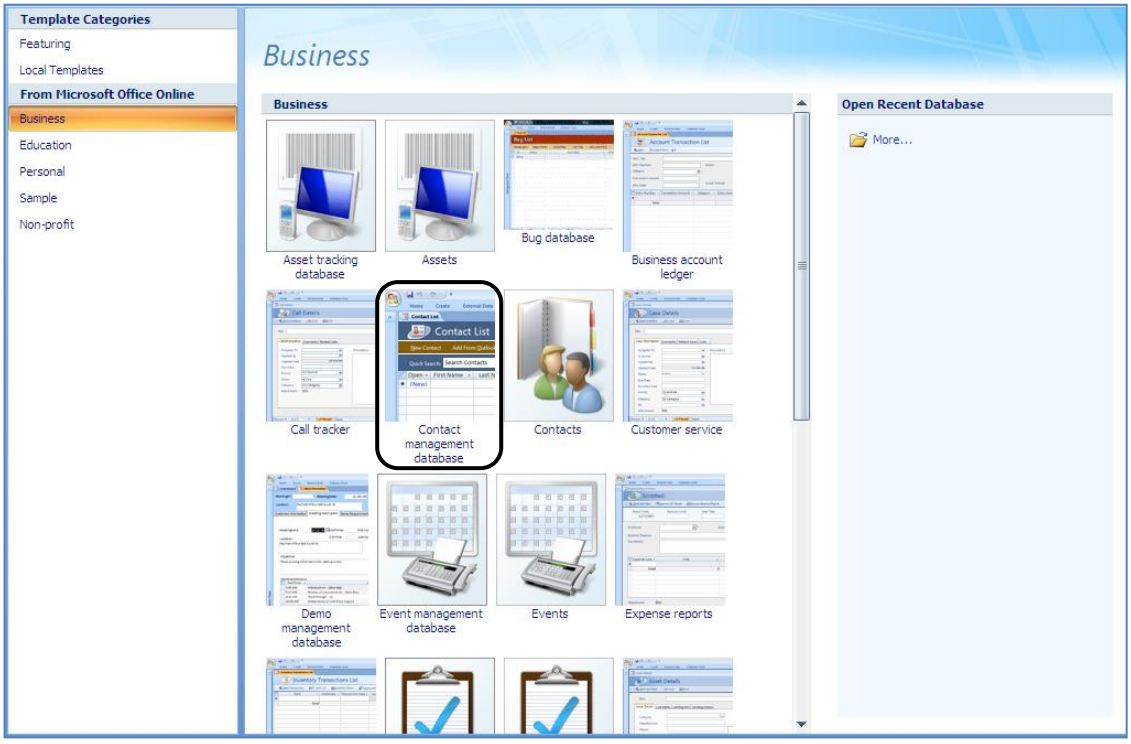
DOWNLOADING A TEMPLATE

You are going to create a database using the Contact Management online template and make a few minor alterations to the design.

The Access Getting Started Window should be on your screen. If it isn't, you will need to close any open database, or launch Access if it is not open.

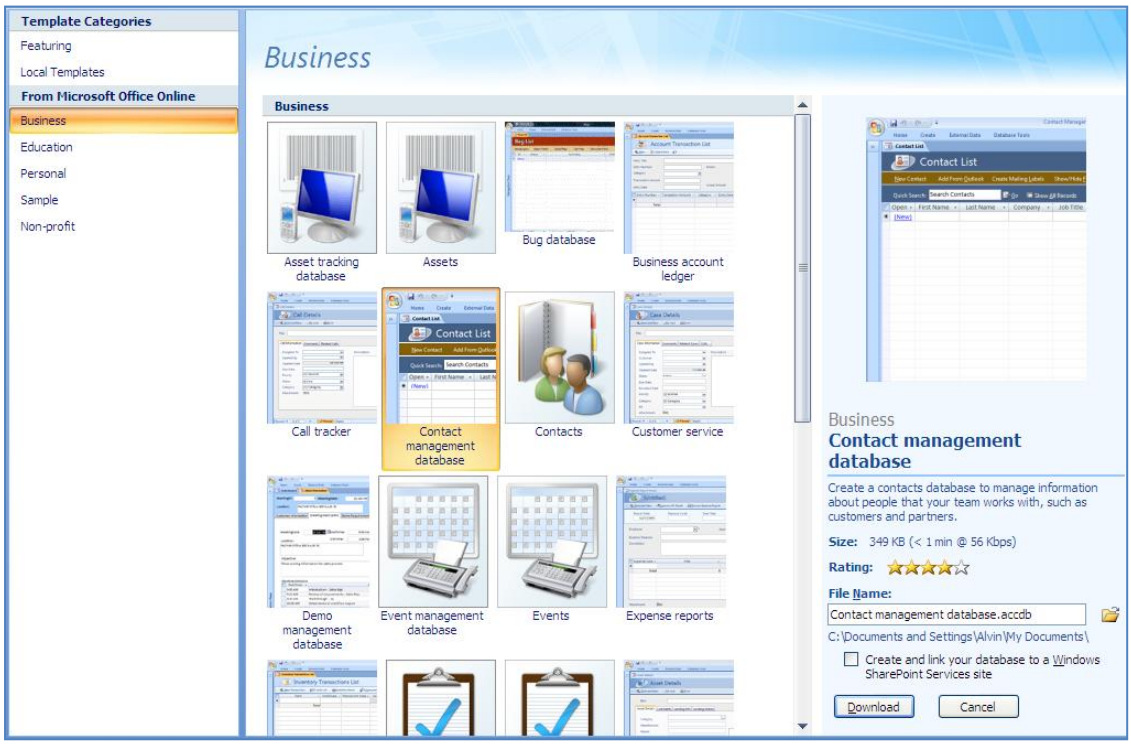
A list of Template Categories is displayed on the left.

- Select **Business**



- Select the **Contact management database** template

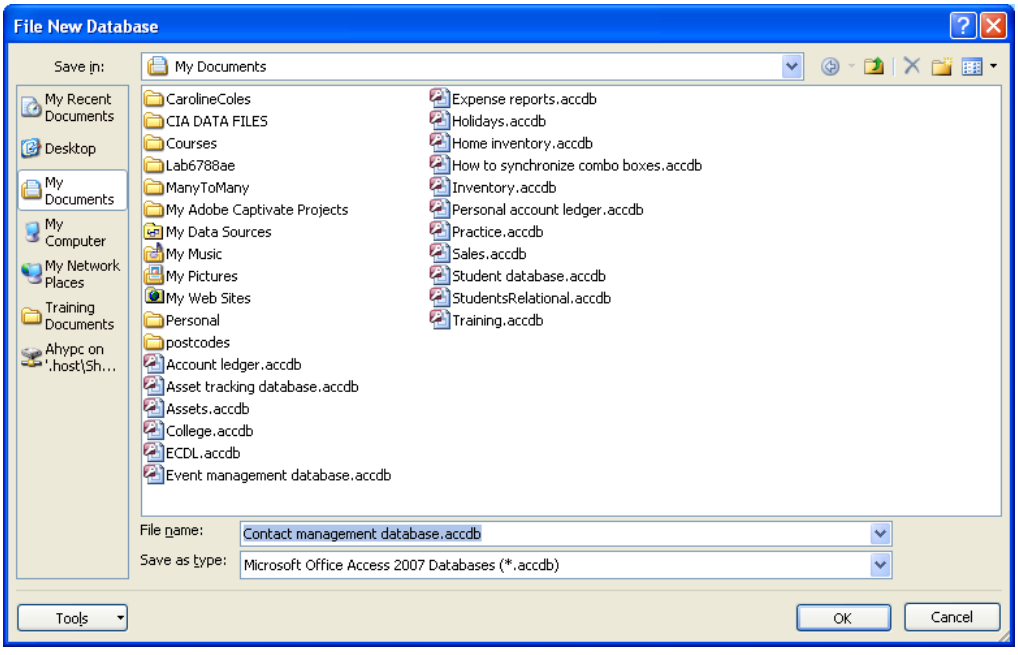
The pane on the right of the window shows the location and file name of the database being created.



You will create the database in the **AccessIntermediate** folder on drive C, accepting the file name suggested.

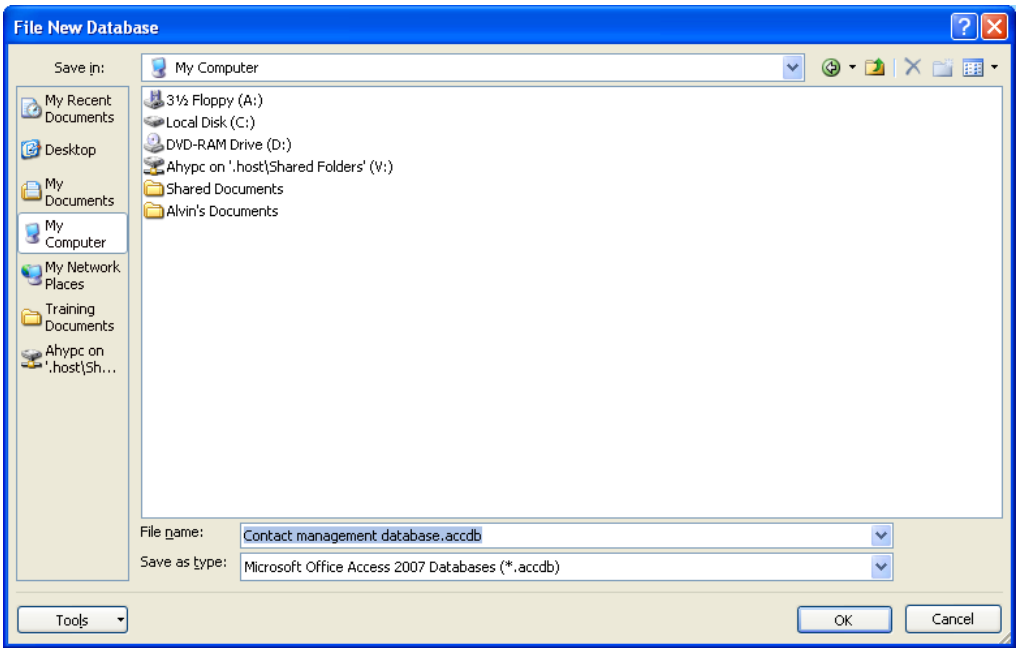
➢ Click the Browse icon next to the File Name box 

The File New Database dialog box is displayed.



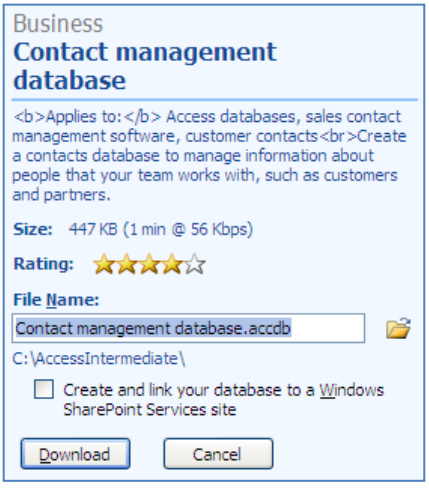
From the buttons on the left select:

- My Computer



- Select **Local Disk (C:)**
- Click **Open** to show the folders on drive C
- Select **AccessIntermediate**
- Click **Open** to open the AccessIntermediate folder
- Click **OK** to accept that folder as the location for the database

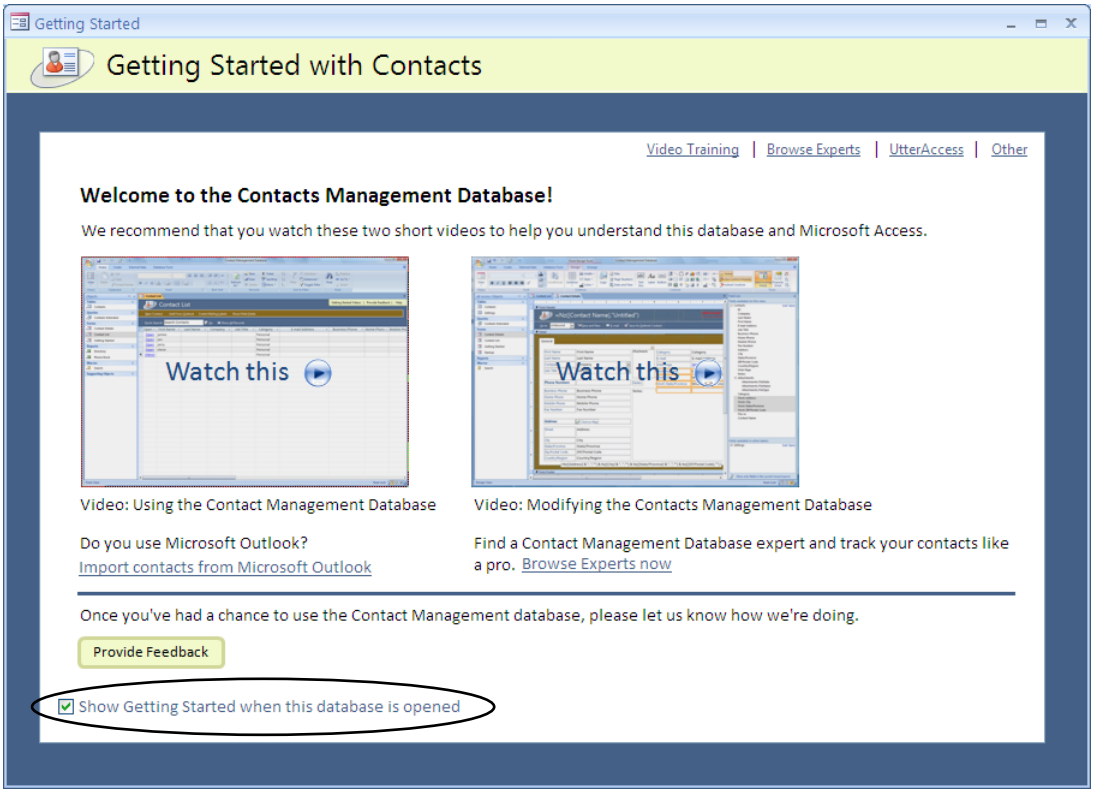
The following file name and location should now be seen in the right pane of the Getting Started window:



- Click **Download**

The database is downloaded and a Getting Started form and an Access Help window are displayed.

- Close Help



Some of the templates do not have a Getting Started form.

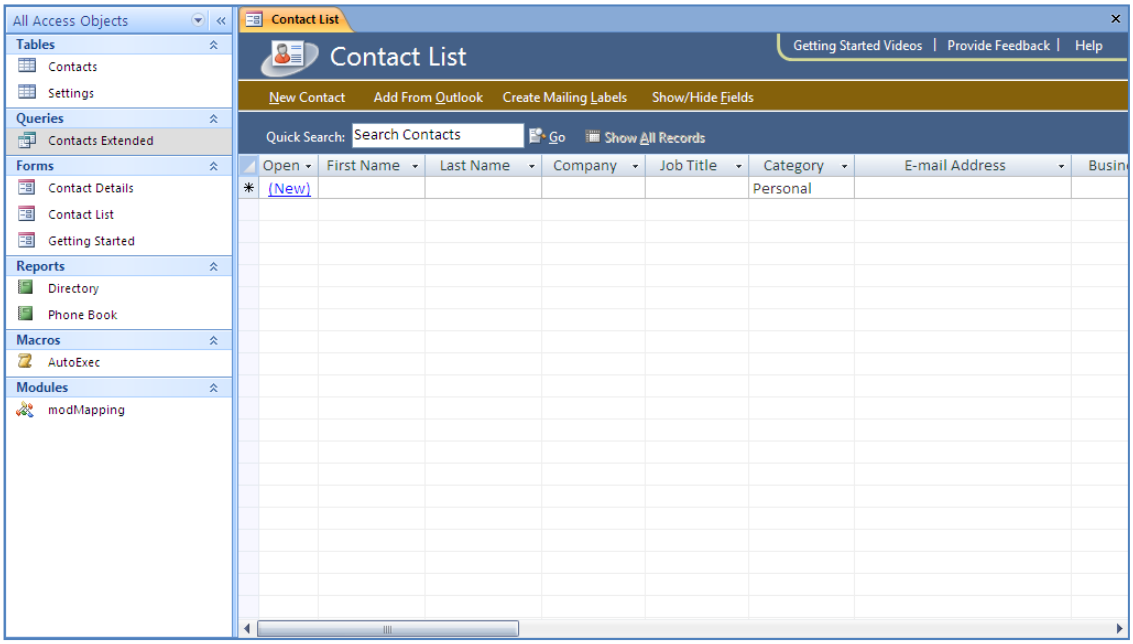
However, if you are using a template to create a database and it displays a Getting Started form, then it is advisable that you watch the training video(s) to help you to get started with the particular database.

By default the Getting Started form will be displayed every time the database is opened.

To configure the database so the form is not displayed automatically:

- Remove the tick from the check box labelled **Show Getting Started when this database is opened**
- Close the Getting Started form

The database is shown with a blank Contact List displayed.



At this point the database is ready to be used, but you will usually find that a few alterations are necessary in order to meet your specific needs.

Changing the database to fit your requirements

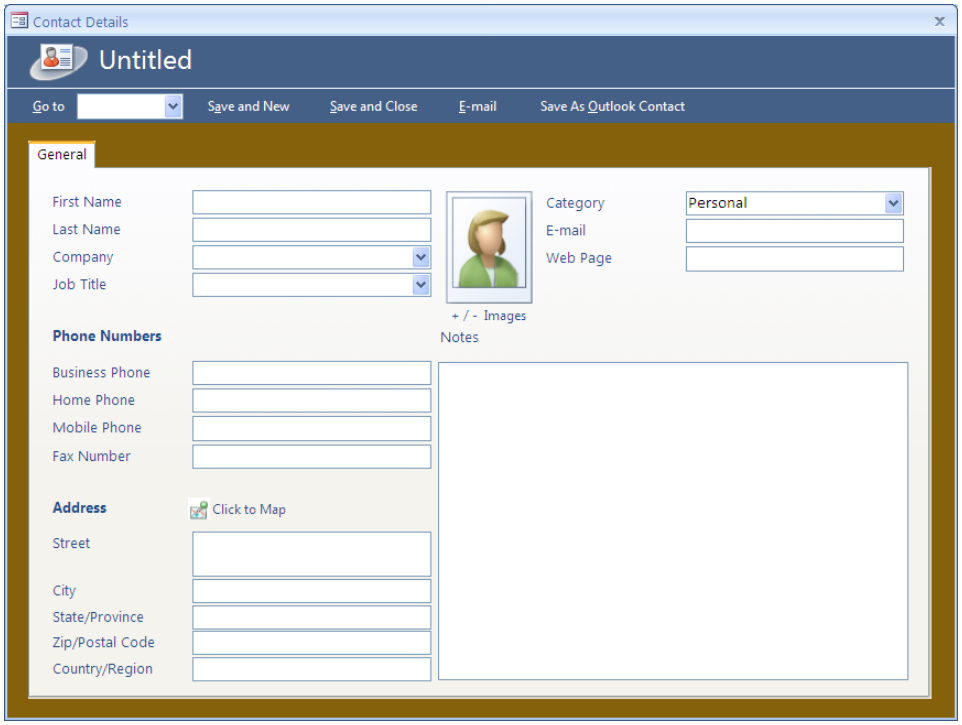
A word of warning; although you are able to change the database in any way that you wish, do remember that changing one aspect of the database may impact on other areas. Therefore give careful thought to any changes that you plan to make.

Generally speaking, you can add extra fields to the tables and create further objects and controls without undermining the existing structure. However, it is best not to delete any fields from tables, but rather remove unwanted fields from the forms and reports.

You should rename existing fields and objects with caution, ensuring that the new names are replicated to all objects as appropriate. If you are not entirely confident then choose to rename the labels on forms and reports and leave the field names unchanged.

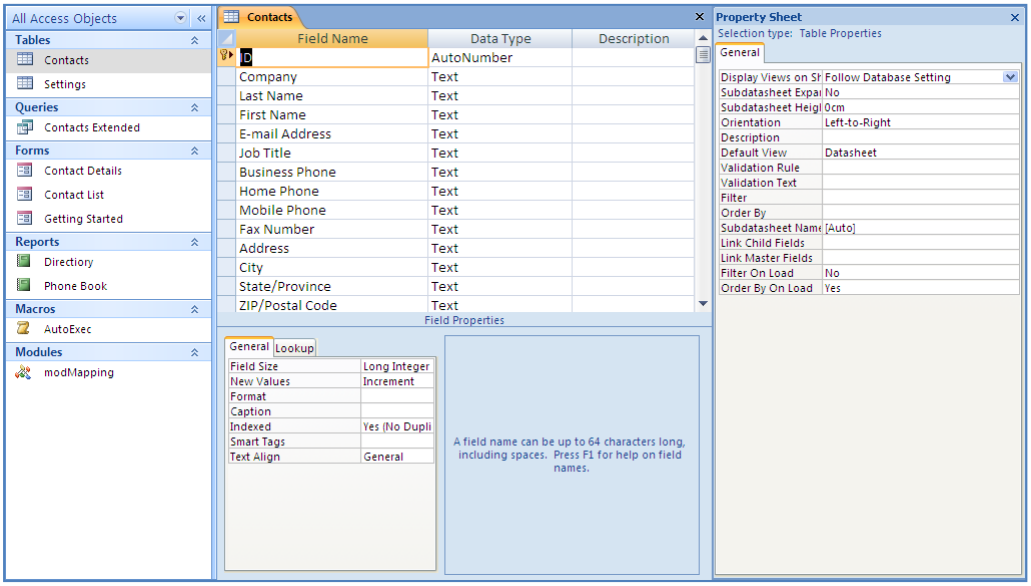
➢ Click the **New Contact** link in the form header

A blank Contact Details form is displayed.



You observe that you need to store date-of-birth but the form does not have this field. Because the data is stored in a table, you need to first insert a date-of-birth field in the Contacts table before it can be added to the form.

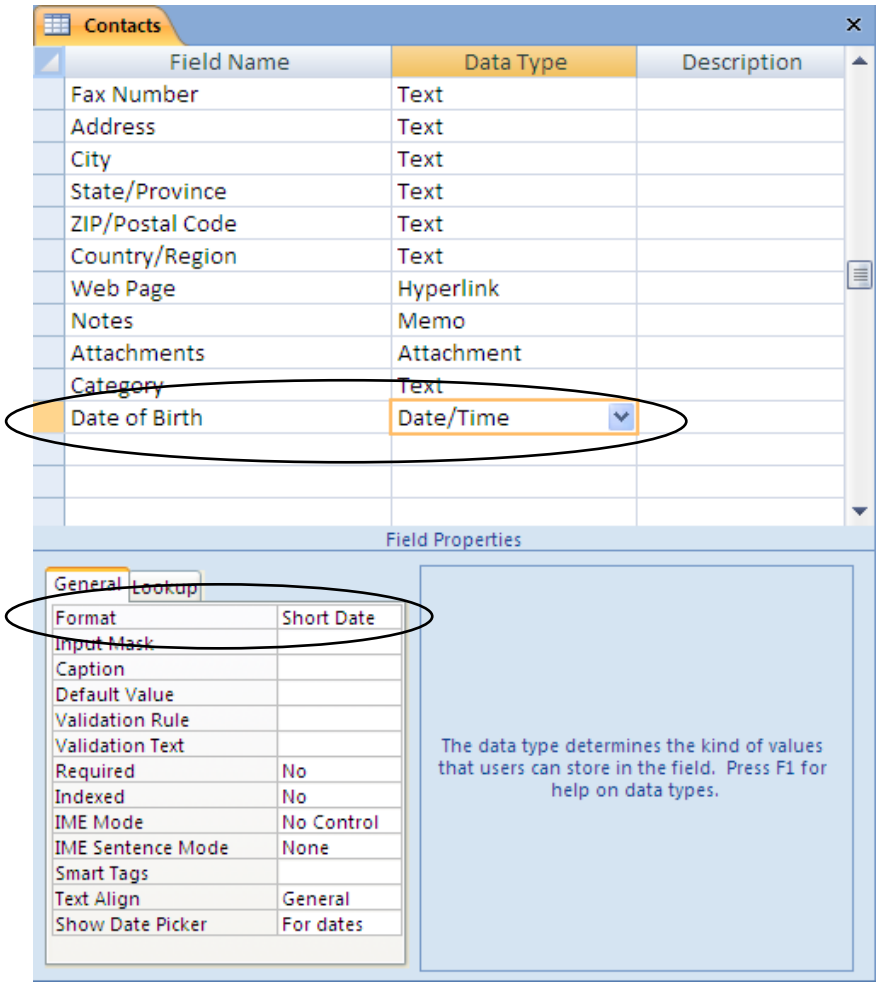
- Close the **Contact Details** form
- Close the **Contact List** form
- Open the **Contacts** table in **Design View**



Don’t worry if a Property Sheet is not displayed.

- Scroll down to the bottom of the list of fields and add the following new field:

Field Name	*Data Type*	*Format*
Date of Birth	**Date/Time**	**Short Date**



- Save and close the table

You now need to add this field to the appropriate form(s). We will assume that the only form that you want to add this field to is the Contact Details form.

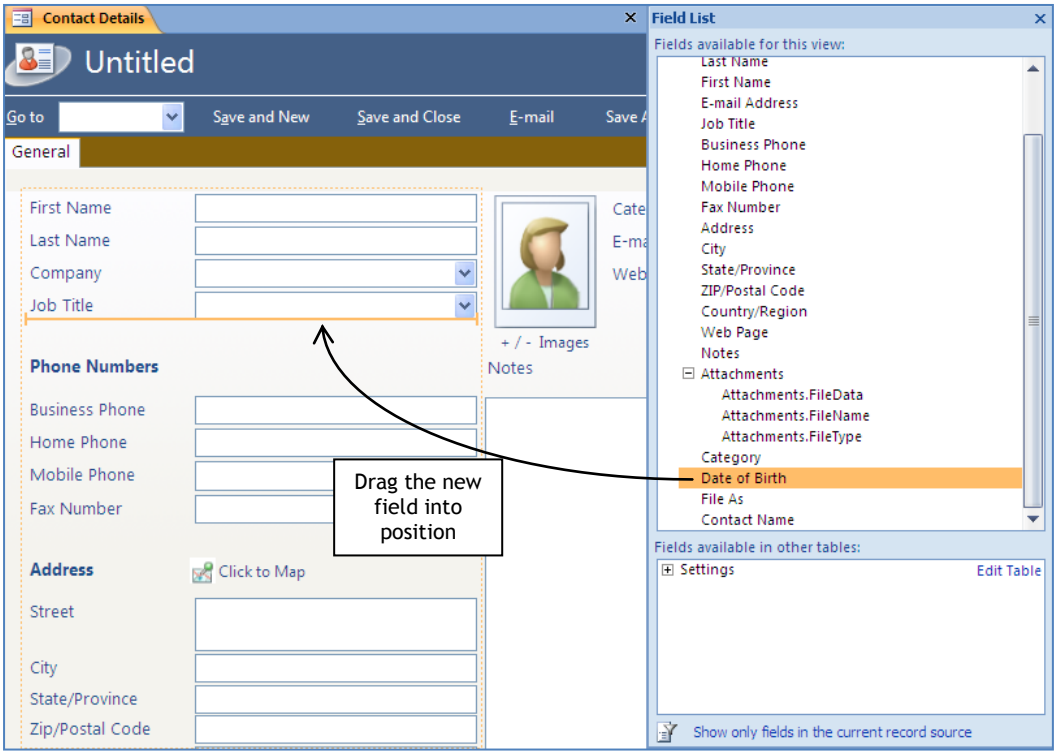
- Open the **Contact Details** form in **Layout View**

If the Field List is not displayed

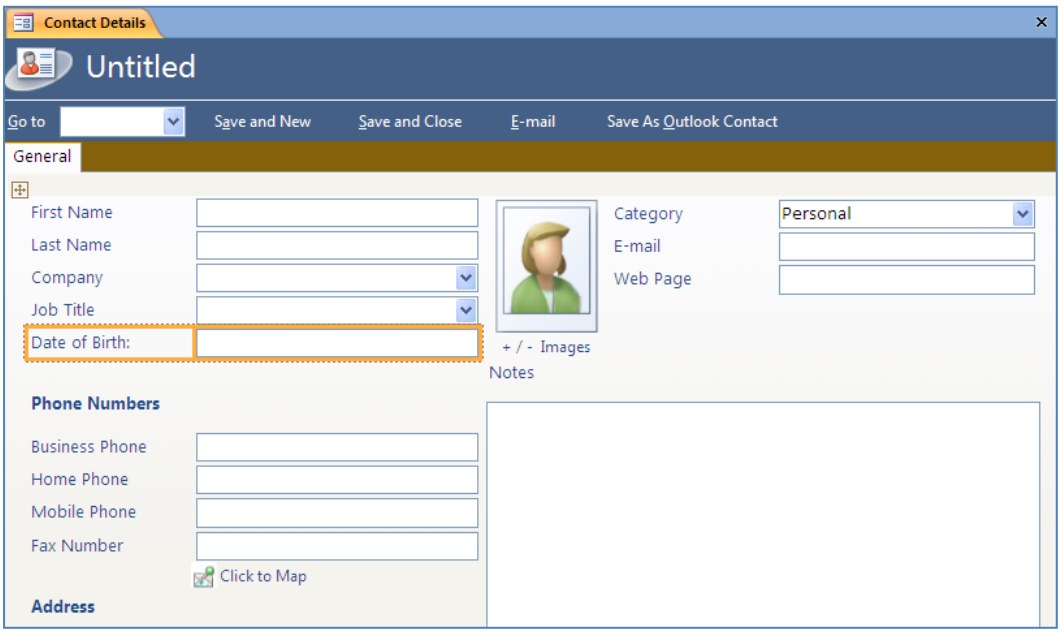
- Select **Add Existing Fields** from the **Controls** group on the **Format** tab

You will position the Date of Birth field immediately under the Job Title field.

- Hold down the mouse button over the **Date of Birth** field in the field list, drag it to the bottom of the **Job Title** combo box, then release the mouse button



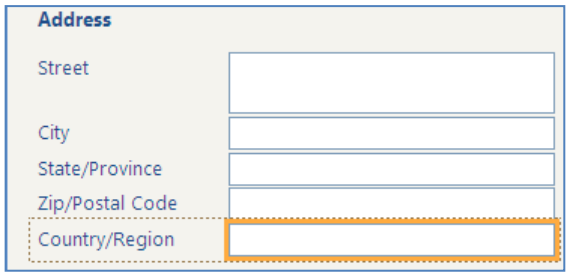
- Close the Field List



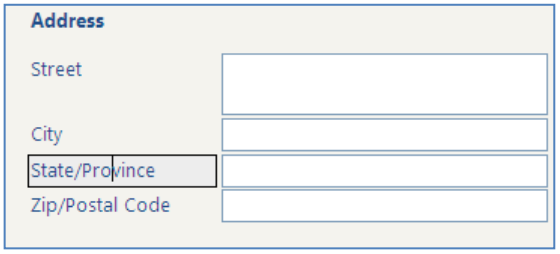
- Minimise the Office Ribbon so that you can see all the fields on the form

You want the address on the form to show the labels: Street, City, County and Postcode. You therefore need to edit two of the labels and delete the Country/Region field.

- Click in the **Country/Region** text box to select it

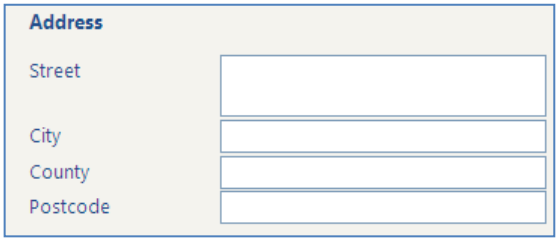


- Press the **Delete** key
- Double click on the **State/Province** label



The insertion point is in the label ready for you to edit it.

- Replace the existing text with **County** and press Enter
- Change the Zip/Postal Code label to **Postcode**



Editing labels on this form will have no effect on other forms or reports. If you were planning to use this database you would have to check each form and report and edit them individually.

- Save and close the form
- Restore the Office Ribbon

TABLES

To help you to create your tables quickly and easily, Microsoft has included a few templates for creating some of the most common tables.

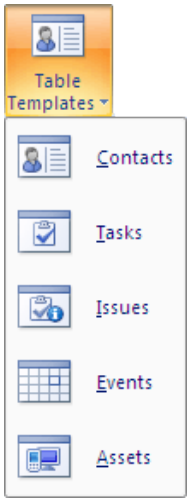
USING TABLE TEMPLATES

Many of your contacts are members of various sports clubs. You will therefore add a Sports Clubs table to the database.

You will create this table by looking for a suitable table template.

- Click **Table Templates** from the **Tables** group on the **Create** tab

A list of templates is displayed.



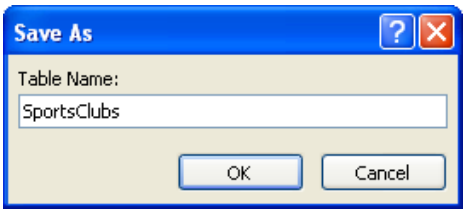
At first glance, none of these appear suitable.

However, you know that for your Sports Clubs you need to have fields such as name, phone, address, etc. This table will therefore have many fields in common with a contacts table.

- Select **Contacts** from the list of templates

A table is inserted into the database.

- Save the table using the name **SportsClubs**



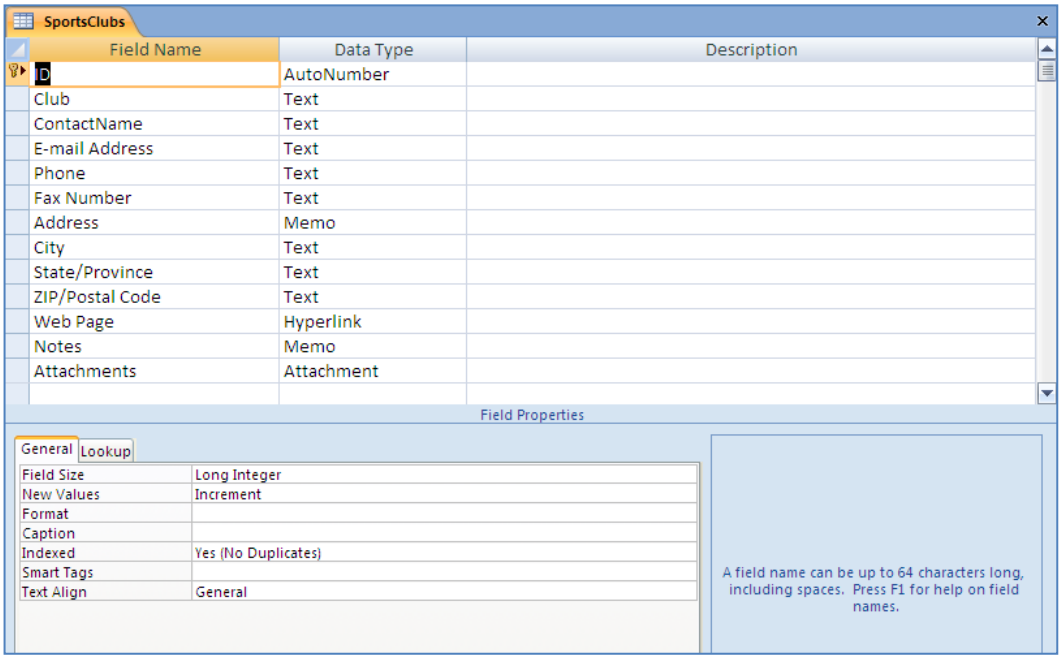
To edit the fields:

- Switch to **Design View**
- Click the **Company** field name and change it to **Club**
- Change **Last Name** to **ContactName**
- Change **Business Phone** to **Phone**

To delete the First Name field:

- Click into the **First Name** field
- Click **Delete Rows** from the **Tools** group on the **Design** tab
- Also delete the fields:

  Job Title, **Home Phone**, **Mobile Phone**, and **Country/Region**



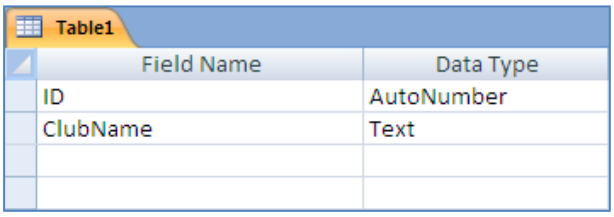
- Save the table

THE LOOKUP WIZARD

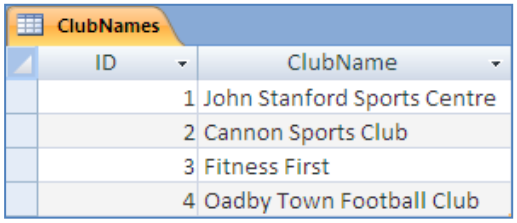
A Lookup field provides a list of values that you can choose from when you are entering data. This makes data entry easier and ensures the consistency of the data in that field. The Lookup field can get its list from a table or query.

You will use a wizard to create a lookup field for the Club names. First you need to create a table with the sole purpose of storing the Club names.

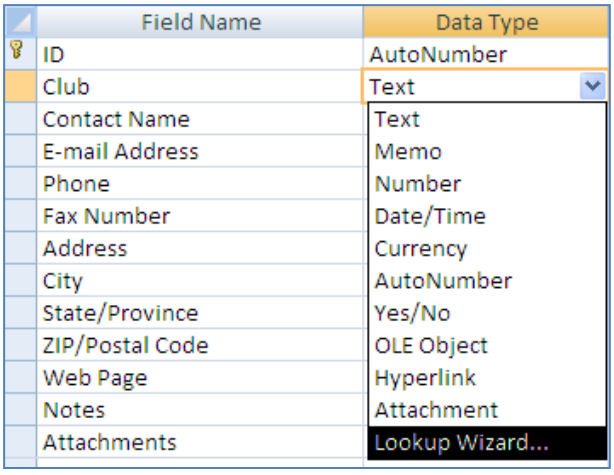
- Click **Table Design** from the **Tables** group on the **Create** tab
- Add the following fields:



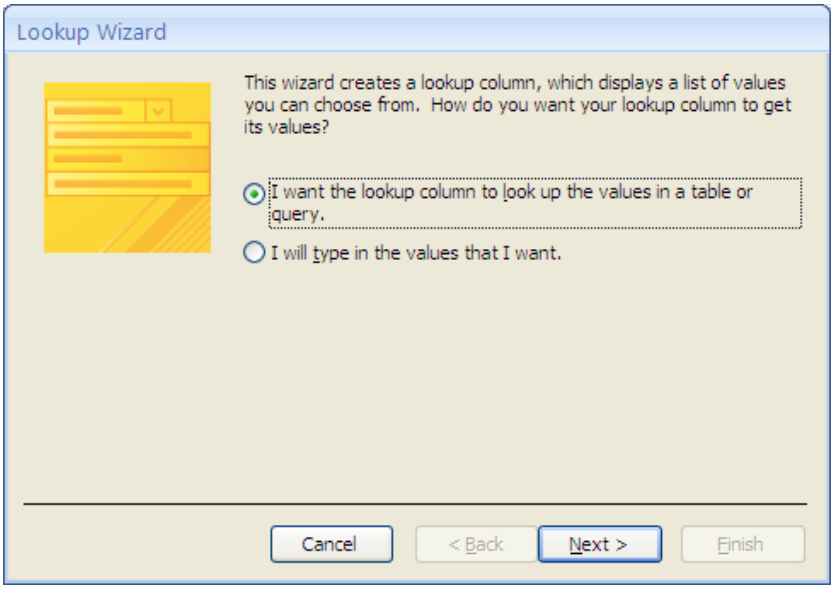
- Set the **ID** field as the Primary Key
- Save the table as **ClubNames**
- Switch to **Datasheet View**
- Enter the following names in the table (the ID field will be populated automatically):



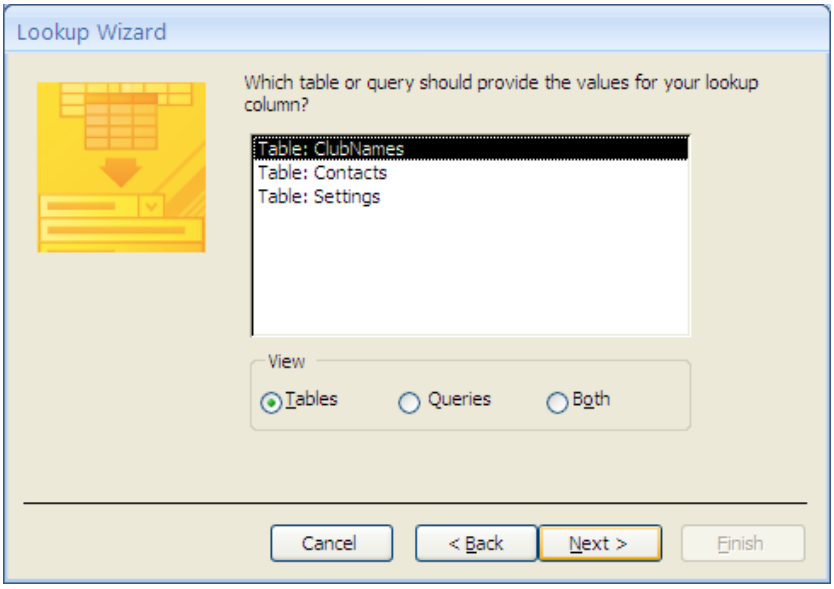
- Close the ClubNames table
- Ensure that the **SportsClubs** table is open in **Design View**
- Click in the **Data Type** box for the **Club** field
- Click the **Data Type** drop list
- Choose **Lookup Wizard** from the list



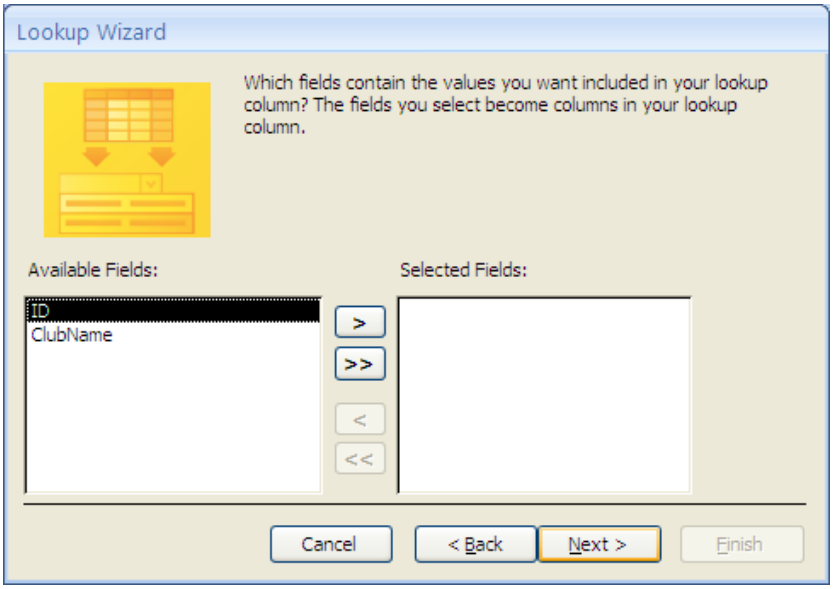
The Wizard begins.



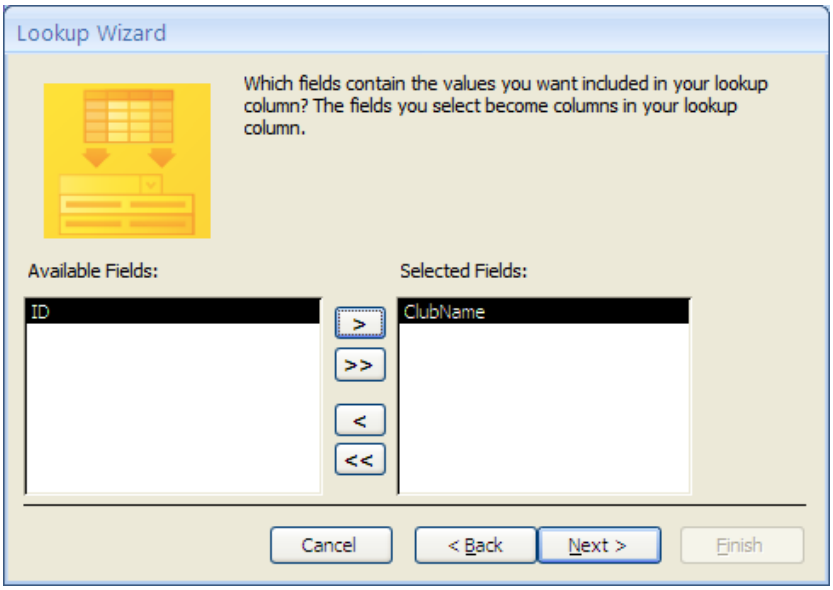
- Accept **I want the lookup column to look up the values in a table or query**
- Click **Next**



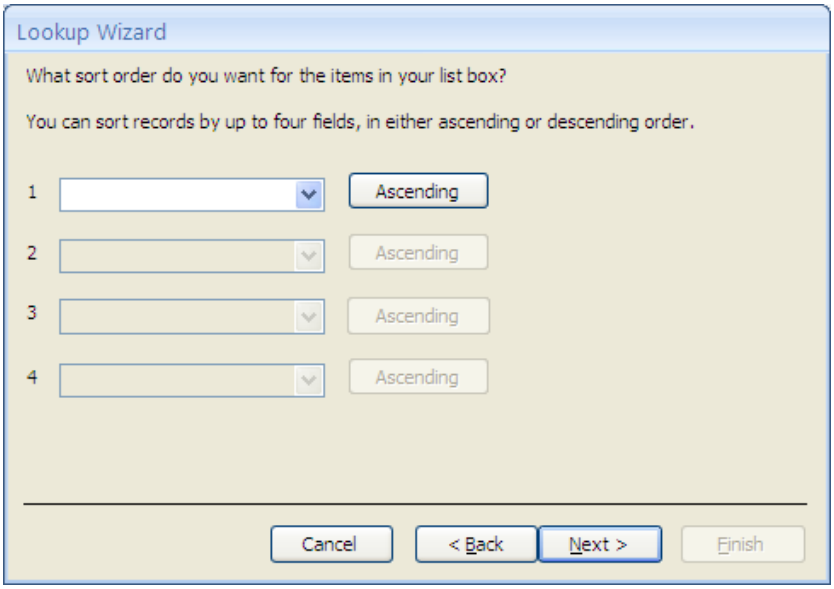
- Choose **ClubNames** as the table to provide the values to look up
- Click **Next**



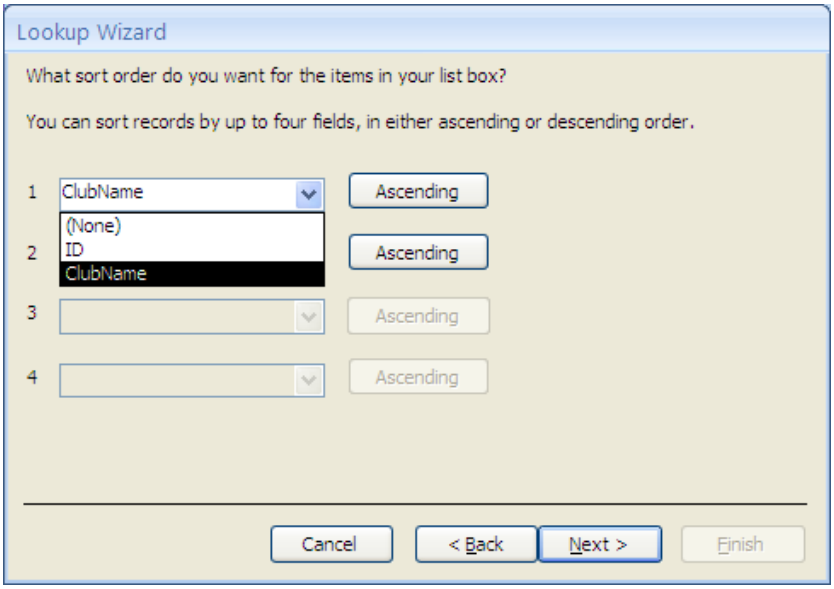
- Select **ClubName** as the field that contains the values to use in the lookup column



- Click **Next**



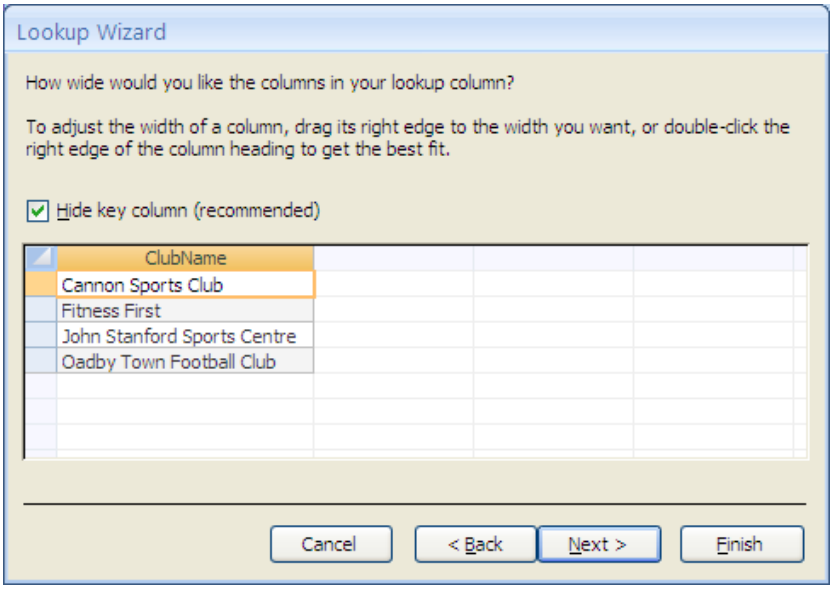
- Click the down arrow and select **ClubName** as the field to sort



- Click **Next**

The data from the lookup table is shown as a list, but the column is not wide enough to display the full text in each row.

- Follow the instruction given in the dialog box to change the width of the column, ensuring that the text in each row is fully displayed



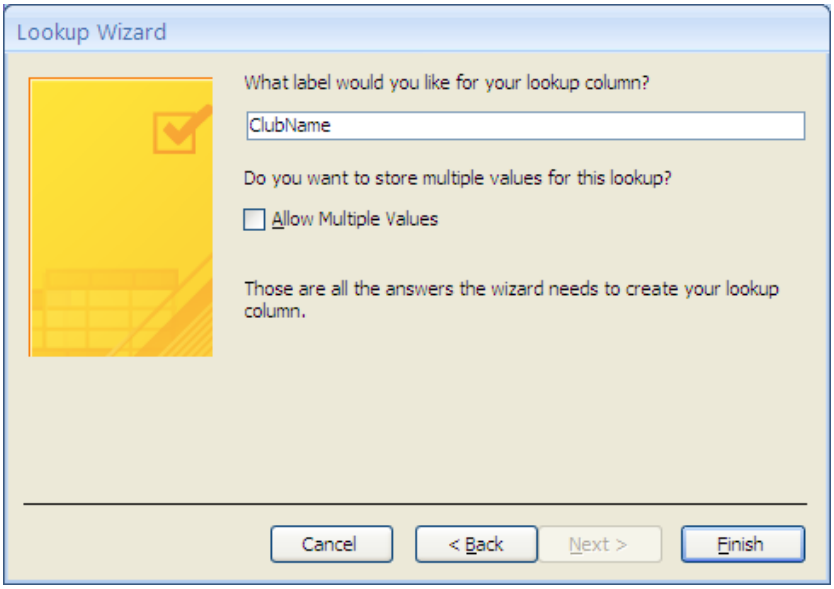
The Lookup Wizard creates a relationship between the two tables, linking the Primary Key of the lookup table with the lookup field in the other table. So although you selected the Club Name field as the field from which to get the values, Access needs the ID field also because this is used to create the relationship.

You will therefore notice a check box labelled **Hide key column**. This is ticked by default since displaying it is usually no help to the user.

- Click the **Hide key column** check box to remove the tick

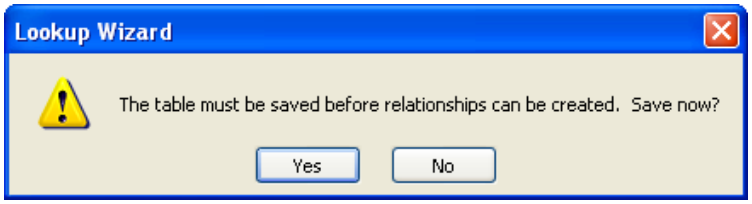
The ID field is displayed.

- Click the check box again to hide the ID field
- Click **Next**
- Type **ClubName** as the label to use for the column

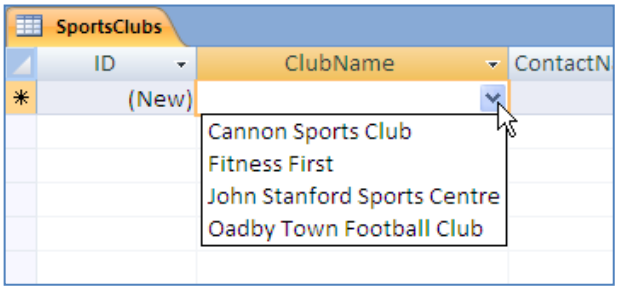


- Click **Finish**

A message appears informing you that the table must be saved before continuing. The message also verifies that the process creates a relationship between the two tables.



- Click **Yes**
- Switch to Datasheet View
- Test the Lookup in the **ClubName** field

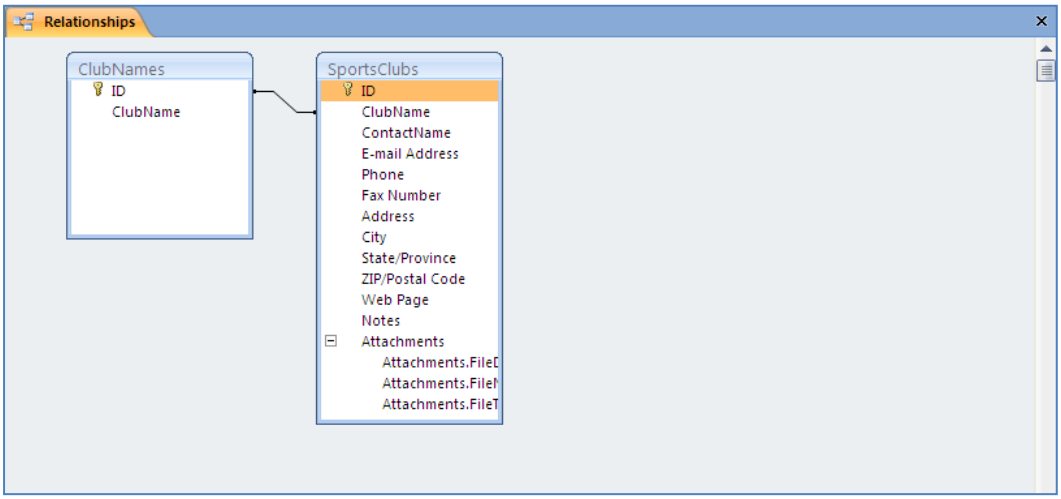


You may need to change the width of the ClubName column in order to fully display all the text.

- Save and close the table

To view the relationship that was created by the lookup wizard:

- Display the Relationships window
- If necessary, add the **ClubNames** and **SportsClubs** tables to the window



The relationship between the two tables is displayed.

- Save and close the Relationships window

The above exercise illustrates that creating a lookup field can be both a useful feature to have in a table and a convenient way to create a relationship between two tables.

SETTING FIELD PROPERTIES

Each field has an associated set of field properties, depending on the data type chosen.

Many of the properties can be set by choosing from a list of available options but others require you to type an expression or code in order to set the property.

This section discusses the **Input Mask** and **Append Only** properties. For brief explanations of all the field properties please refer to Appendix A (page 175).

Input mask

An input mask sets up placeholders in a field and limits the amount and type of data that can be entered in the field. It can also provide clues for the user regarding the structure of the input required.

You use an input mask to provide some control over what values can be entered into a field. For example, you might use an input mask in a product code field where a specific number of letters followed by a specific number of digits are required.

An input mask can consist of literal characters including spaces, dots, dashes and parentheses. The property is therefore sometimes used as an extra formatting tool.

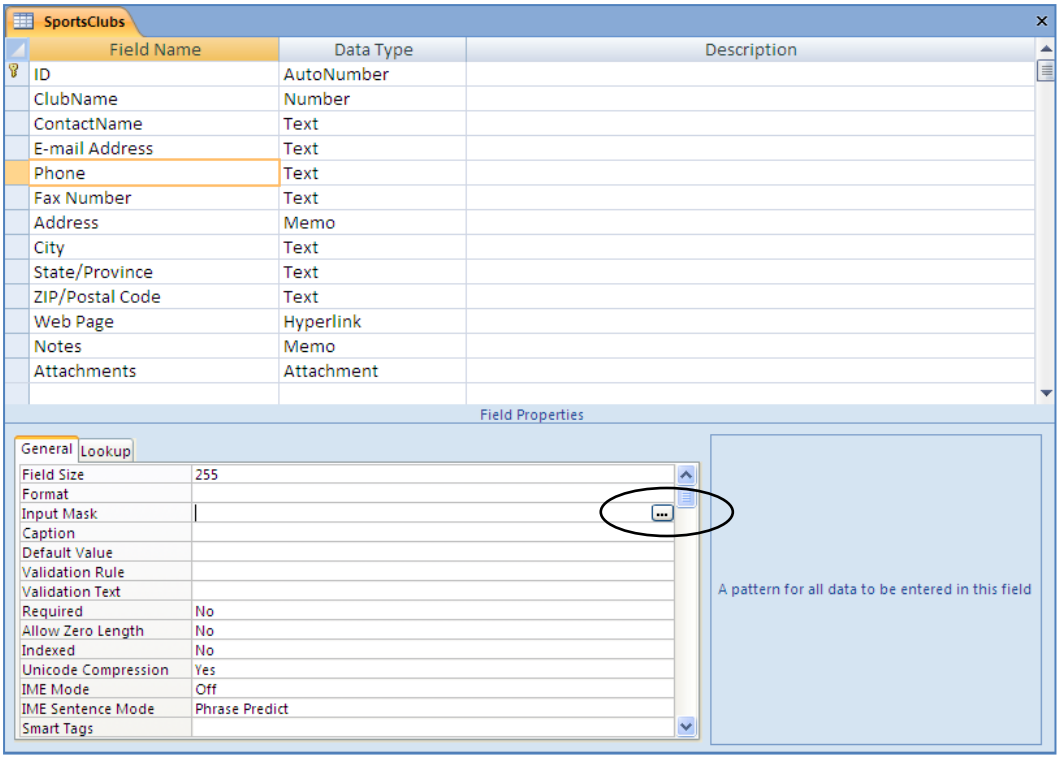
You will create input masks for the Phone and Postal Code fields in the Sports Clubs table.

- Open the **SportsClubs** table in **Design View**
- Select the **Phone** field
- Position the insertion point in the **Input Mask** box in the Field Properties section

You can create an Input Mask by typing directly into the property box. A wizard is also provided to help you with some of the most commonly used masks.

You will use the wizard to create a mask and then edit it by looking at the codes given in Appendix A.

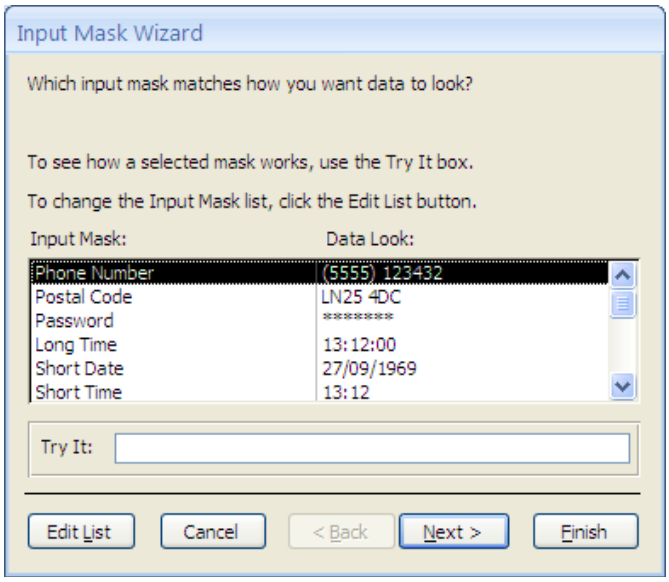
A button is displayed to the right of the Input Mask box.



This button launches the Input Mask Wizard.

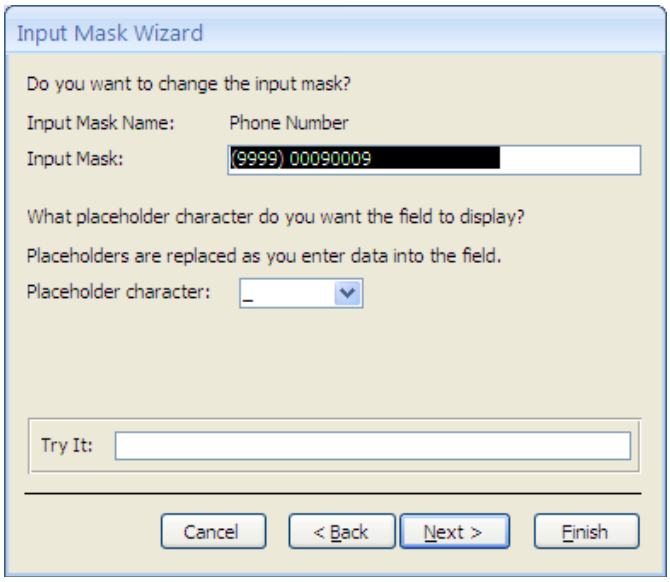
- Click the button

The Input Mask Wizard is displayed, providing options from which to choose.



- Use the scroll bar to view all the options available
- Ensure that **Phone Number** is selected
- Click **Next**

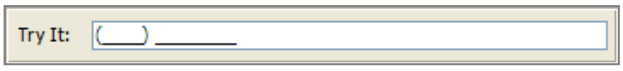
A suggested Input Mask and the character that will be displayed in the field are shown.



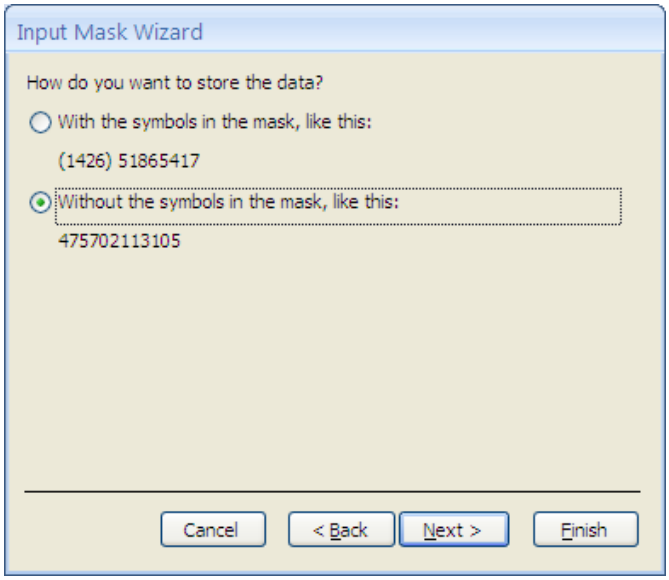
This page allows you to make changes to both the Input Mask and the placeholder character.

- Click the down arrow for the Placeholder character to see the options available
- Select the **underscore** symbol
- Click in the **Try It** box

This display is what you can expect to see in an empty Phone field when you click into it.



- Click **Next**

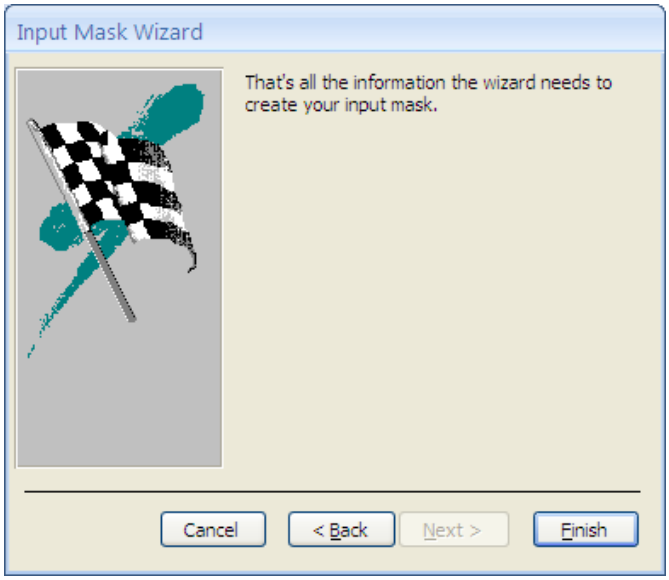


The Input Mask that you have selected, includes extra characters (parentheses and space) that will not be keyed in by the user.

On this page you can instruct Access to store the extra characters with the input, or to just store the input without the extra characters.

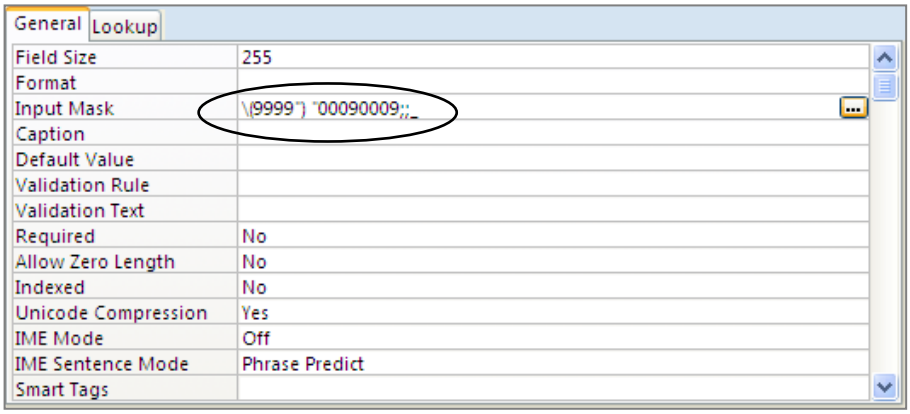
You will store the data without the symbols in the mask. This means that the parentheses and space will not take up any extra storage space in the table.

- Click **Next**



- Click **Finish**

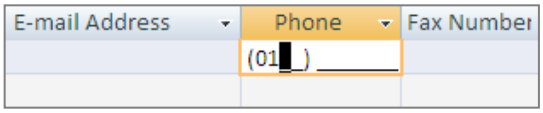
The code for the Input Mask is inserted into the Field Properties.



To see the effect of the mask:

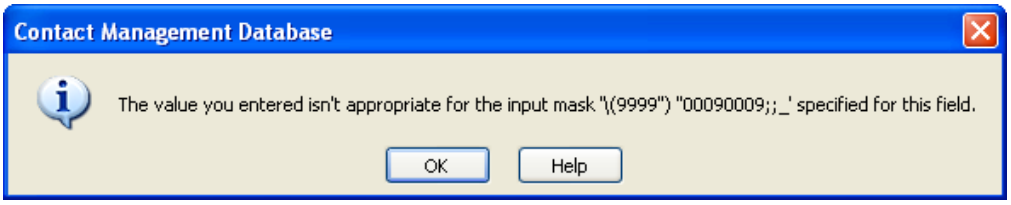
- Save the table and switch to Datasheet View
- Press the **Tab** key repeatedly until the insertion point is in the **Phone** field
- Key in the value **01161234**

As soon as you begin to type in the digits, the characters and placeholders in the mask are displayed.



- Press **Tab**

A message is displayed informing you that the value is not appropriate for the mask.

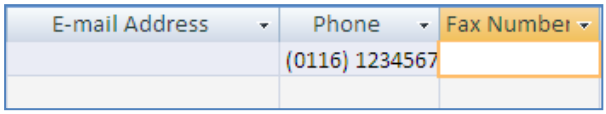


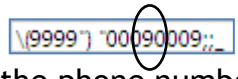
This is because you did not key in the correct number of digits.

The characters '9' and '0' in the mask represent digit placeholders. The '9' placeholder can be left blank, but the '0' placeholder must contain a digit. The mask therefore allows for four digits in an area code section followed by either seven or eight further digits.

- Click **OK** to close the message
- Key in the remaining digits for the number **01161234567**
- Press **Tab**

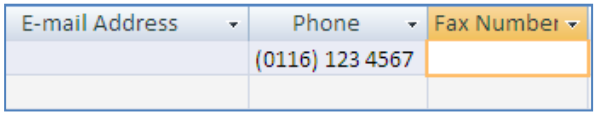
This entry satisfies the Input Mask, since the '9' at the end of the mask is a placeholder that can be left blank.



The '9' placeholder in the middle of the mask  enables the user to insert a space before the last four digits of the phone number.

- Click into the **Phone** field
- Change the entry to **0116123 4567**
- Press **Tab**

The value is displayed as follows:

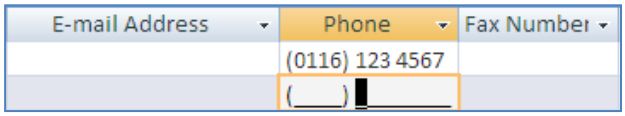


This Input Mask is not quite suitable if you wish to allow users to omit the area code when typing in the phone number.

- Click into the **Phone** field for the second record

The placeholders are displayed and the insertion point shows which placeholder is currently selected.

- Select the first placeholder outside the area code section



- Key in **251 2345**
- Press **Tab**

The value is accepted since all the '0' placeholders were filled with digits, but the final display is not what is desired.

E-mail Address	Phone	Fax Number
	(0116) 123 4567	
	(251) 2345	

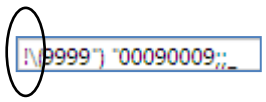
To change this, you need to refer to the table on page 176 showing Input Mask codes and their meaning.

The character '**!**' is used to cause the value entered to fill the mask from right to left. In other words, it causes the value to be right aligned within the mask.

- Switch to Design View
- Select the **Phone** field

The exclamation mark is usually inserted as the first character of the Input Mask code or just before the first semicolon.

- Insert an exclamation mark at the start of the Input Mask



- Save the change and switch to Datasheet View

The phone number is right aligned in the mask, leaving the area code blank.

E-mail Address	Phone	Fax Number
	(0116) 123 4567	
	( ) 251 2345	

- Switch to Design View and ensure that the properties for the Phone field are displayed

Observe that there are two semicolons in the code.

The Input Mask code is made up of three sections separated by semicolons.

The first section contains the appropriate number and type of placeholders, together with any characters that affect the formatting and any literal characters that must be displayed in the field.

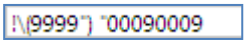
The second section has no effect on what is seen in the table or how it is displayed. This section determines whether or not any extra characters contained in the first part of the mask are stored with the data entered in the

field. Entering a '1' or leaving the section blank tells Access to store only the data entered in the field. '0' tells Access to store both the data entered by the user and the characters inserted by the mask.

The character in the third section is displayed in the field as the placeholder for the characters to be typed.

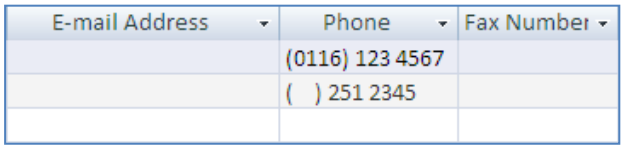
The second and third sections of the input mask are optional and can be omitted completely.

- Delete everything after the last '9' in the Input Mask

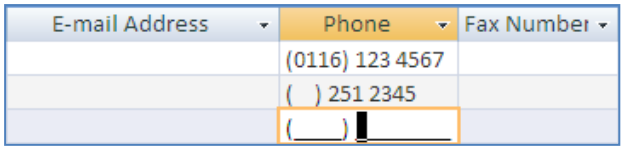


- Save the change and switch to Datasheet View

The phone numbers are displayed exactly as before.

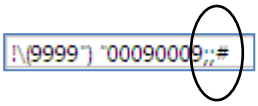


- Click into the Phone field of a blank record

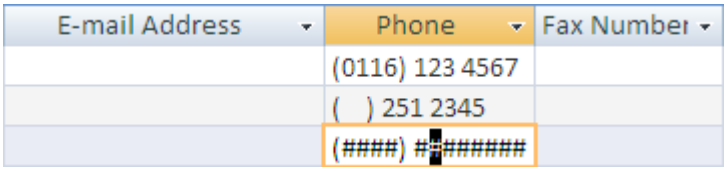


The underscore is the default character for the placeholders and is therefore still used to display the Input Mask.

- Switch to Design View
- Add **;;#** to the end on the Input Mask



- Save the change and switch to Datasheet View
- Click into the Phone field of a blank record



The hash symbol is now used for each placeholder.

Another field in the Sports Clubs table that can benefit from the use of an Input Mask is the Postal Code field.

Postcodes are usually displayed using uppercase letters. An Input Mask can be set to always display the letters in uppercase.

To begin with you will assume that all the postcodes that will be entered in your table will be of the form LE1 9BH. In other words, two letters followed by a single digit, then a space, then a single digit followed by two letters.

You will now set an Input Mask for this ensuring that the letters are always displayed in uppercase.

- Switch to Design View and select the **Zip/Postal Code** field
- Type **>LL0\ 0LL** in the **Input Mask** property

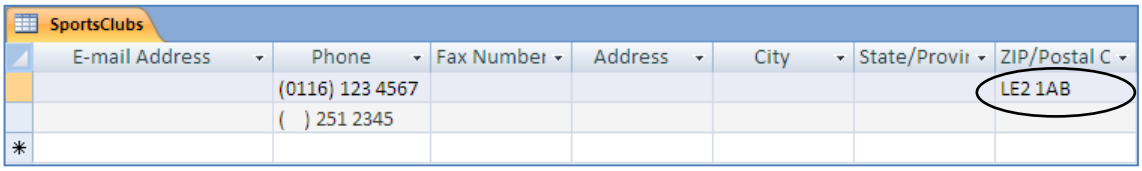
Refer to the table on page 176 for an explanation of what each of the characters in the mask represents.

- Save the changes and switch to Datasheet View
- Press **Tab** repeatedly until the insertion point is in the **Zip/Postal Code** field

It is also possible to click directly into the field, but you must ensure that the insertion point is at the first placeholder of the Input Mask.

- Type **le21ab**
- Press **Tab**

The postcode is displayed correctly formatted.



- Delete the postcode
- Try typing only letters for the Postcode
- Try typing only digits

The Input Mask restricts you to entering six characters: two letters, two digits and two letters in that order.

Assume that you will allow users to enter only the first part of the postcode.

- Edit the Input Mask to allow both LE1 9BH and SW5
  (Hint: Only the first part of the postcode must be compulsory.
  Refer to the table on page 176)

- Check that the mask works as expected

One possible solution for the mask is >LL0\ 9?? '9' means that you can enter a digit or leave the placeholder empty; '?' means that you can enter a letter or leave the placeholder empty.

You will alter the Input Mask once more, but this time you want to accommodate all UK postcodes and you want users to enter them fully.

You therefore need to accommodate these four different types: LE1 9BH, LE18 5XY, M2 6BD and M15 3LN.

- Edit the Input Mask to accommodate the above postcodes
  (Hint: first construct the mask without the space character in the code and verify that the different lengths and combinations of letters and digits are accepted in the field. Finally set the mask to fill from right to left and insert a space before the last three characters.)

- Check that the mask works by entering each of the four different postcodes

One possible solution for the mask is !>LAAA\ A?? Refer to the table on page 176 for an explanation.

Append only

This property is available for Memo and Hyperlink data types only. It is not a difficult property to set; you have two choices: Yes or No.

When the property is set to Yes, Access keeps track of the changes to the field values.

- Select the **Address** field in Design View

- Change the **Append Only** property to **Yes** by clicking the down arrow and choosing Yes

- Save the change and switch to Datasheet View

- Click into the Address field of the first record and type **56 Parliament Street**

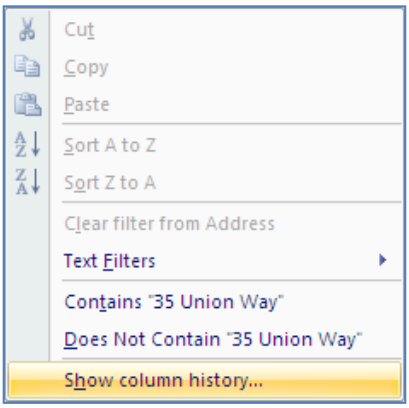
- Save the table to ensure that the value is saved

- Ensure that the Address column is wide enough to display the full text

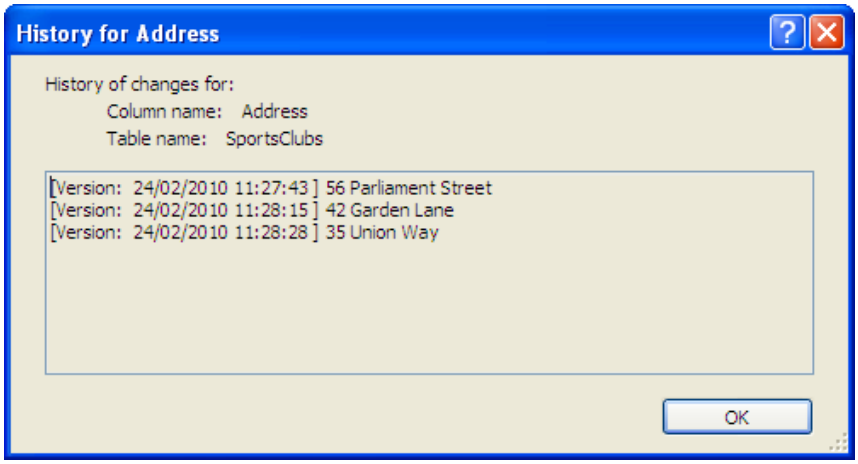
- Change the Address to **42 Garden Lane** and save the table
- Change the Address once more to **35 Union Way** and save the table

To view the history of the values in the field:

- Position the mouse pointer in the Address field of the first record and click the right mouse button
- Select **Show column history** from the shortcut menu options



A dialog box is displayed showing the history of the values.



- Click **OK**

The history information can also be displayed when viewing the data in a form.

- Close the **SportsClubs** table
- Close the **Contacts Management** Database

REPORTS

A report is used to present data in a printed format.

In the introductory course you saw how basic reports can be created automatically and you also used a wizard to create a simple report.

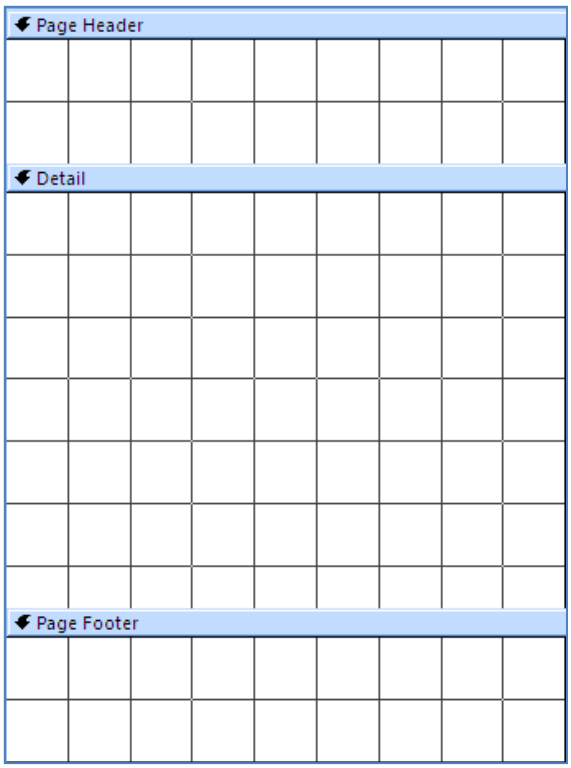
This section will demonstrate how reports can be designed and formatted to your precise individual requirements. The procedure is more labour intensive than using the wizard, but it gives you a lot more flexibility and enables you to design reports exactly as you wish.

COMPONENTS OF A REPORT

Before you get into the design of the report look at how the page is split into sections and understand how the sections are used.

Page Header

Information in this section will appear once on every page. This could include graphics, command buttons, and company address.



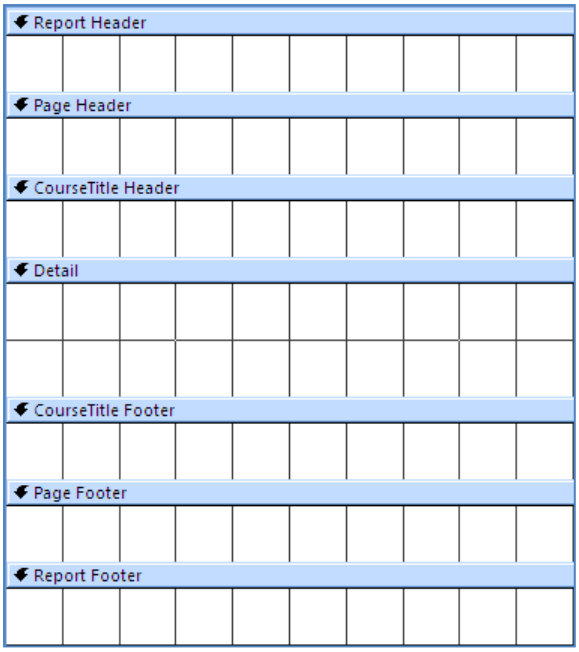
Detail

This section displays data from the database. One record can be printed on each page if you are producing a letter, or a list of data taken from several records can be printed on a page.

Page Footer

This section includes information that shows at the bottom of each page, for example a page number.

Other sections can be added to a report:



Report Header

This can include information, such as a title, for the report. It will be printed only on the first page after the Page Header.

Group Header

The group header shown in the example is 'CourseTitle Header'. This is printed at the start of each group and will usually include information to identify the group, and possibly column headings for the data that will be displayed in the Detail section.

Group Footer

The group footer is printed at the end of each group and could include a field to produce calculations based on data in individual groups, for example the total number of students on a course.

Report Footer

Anything in this section is printed at the end of the report on the last page. It could include fields that calculate overall totals for the report.

REPORT CONTROLS

All information included in a report is held in **controls**. Any database object inserted in the report is called a control. This section focuses on the controls that are used to display data from the database, and display instructions to the user. These controls are labels, text boxes, list boxes and combo boxes.

A control can be bound or unbound.

A ***bound*** control is attached to a field in the query or table that the report is based on. Bound controls display data held in a field. This is the simplest and most straightforward way to display data from the database.

An ***unbound*** control is not linked to a field. You therefore have to do more work with this type of control. However, unbound controls are more flexible and provide greater control over how data is displayed in the report.

An unbound control that contains an expression to combine data from two or more fields, or perform calculations using data from fields in the table or query is called a ***calculated*** control.

Text box

A text box is the main control used to display data. Usually it is bound to a field and therefore will display data from that single field. However, it can also be used as an unbound control, in which case you can type any text into it or use it to create a calculated control.

Label

Labels are not used to display data from the database. Rather, they are used to create titles, headings and any helpful text or instructions that you wish to provide for the users.

List box

A list box control is usually bound to a field, and used to input data into the field. When the box is clicked, a list of options is displayed. If the user selects one of the options, this value is stored in the field.

Combo box

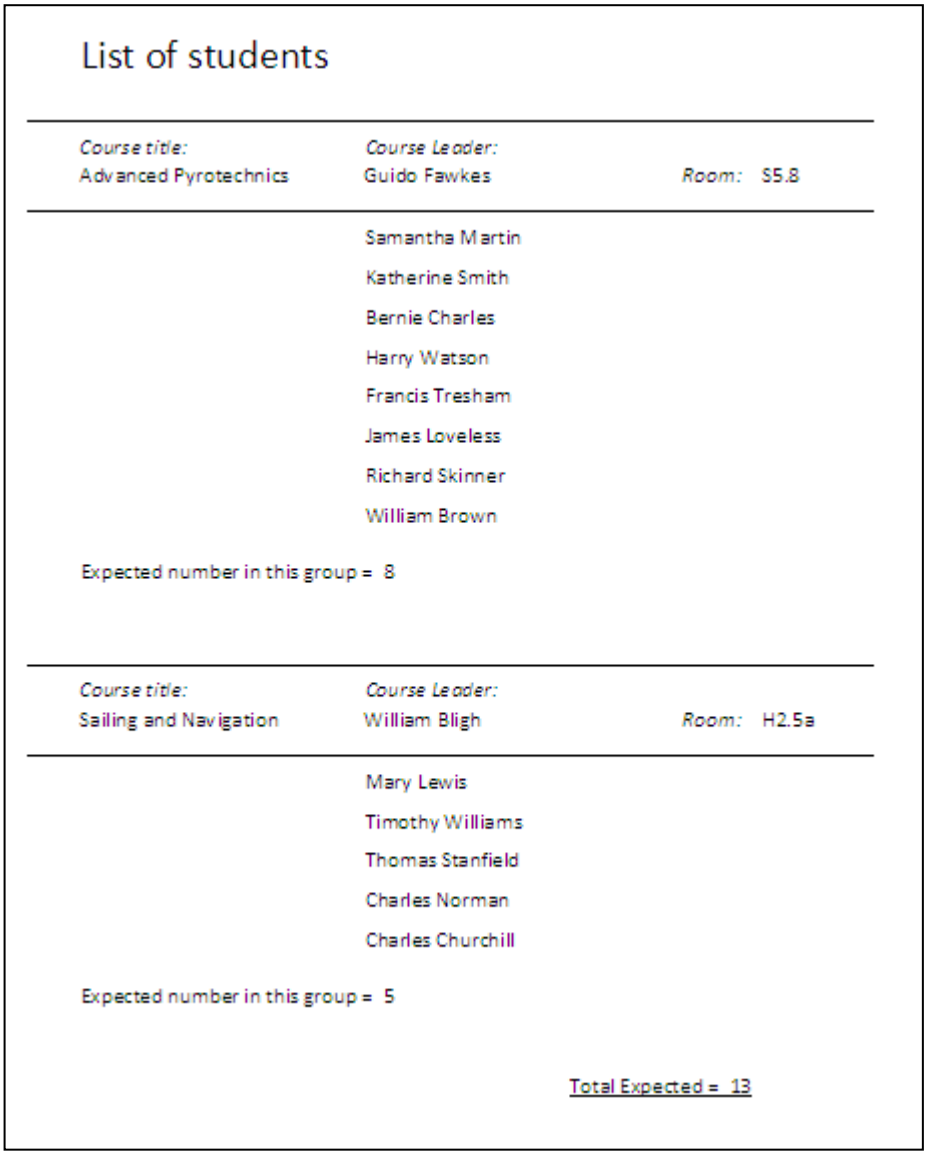
A combo box is similar to a list box and is used in the same way. It can be identified by a down arrow at the extreme right of the box. The user needs to click the down arrow to see the options available.

AN ATTENDANCE LIST

- Open the **2010 Students** database from C:\AccessIntermediate

The report that you will create is an attendance list for the Advanced Pyrotechnics and the Sailing and Navigation courses.

Here's what the finished report should look like:



List of students

Course title:	*Course Leader:*	
Advanced Pyrotechnics	Guido Fawkes	*Room:* S5.8

Samantha Martin
Katherine Smith
Bernie Charles
Harry Watson
Francis Tresham
James Loveless
Richard Skinner
William Brown

Expected number in this group = 8

Course title:	*Course Leader:*	
Sailing and Navigation	William Bligh	*Room:* H2.5a

Mary Lewis
Timothy Williams
Thomas Stanfield
Charles Norman
Charles Churchill

Expected number in this group = 5

Total Expected = 13

Create the Query for the report

You need to create a query to include just the fields that you are interested in and to select the appropriate records.

- Create a query containing the following fields from the students, courses, and tutors tables:

students	*courses*	*tutors*
FirstName	**CourseTitle**	**Forename**
LastName	**RoomNumber**	**Surname**
	StartDate	

- Include the correct criteria to select only the students on the **Advanced Pyrotechnics** or **Sailing and Navigation courses**.

When you use the word 'and' as part of the query criteria, it has a special meaning. If you type the course name 'Sailing and Navigation', the word 'and' will be used as a command telling Access to look for courses named 'Sailing' and courses named 'Navigation'.

You have two choices in a case like this:

1. Use a different field to select the records, for example the course numbers.
2. Use double quotes around the full course name so that Access will know that it is to be used together.

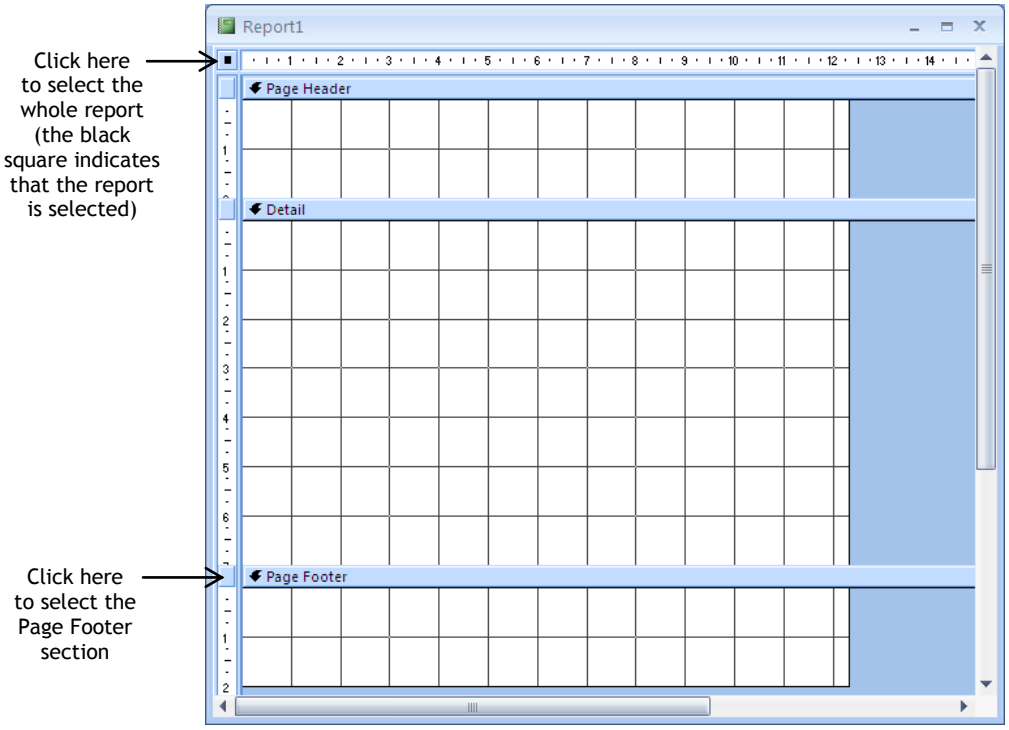
- Save the query with the name **qryGroup**
- Close the query

Begin the Report

The Report Wizard will not produce the layout that you want so you need to design the report yourself.

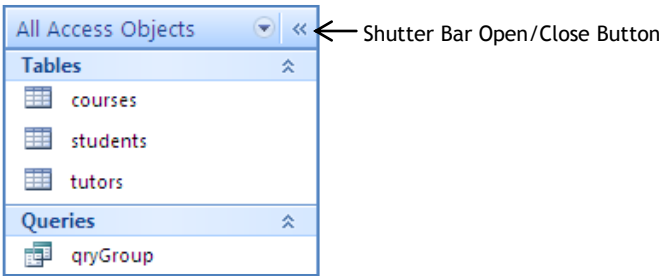
➢ Select **Report Design** from the **Reports** group on the **Create** tab

A blank report is displayed.



You may have to scroll down in order to see the Page Footer section.

➢ Minimise the Navigation pane by clicking on the **Shutter Bar Open/Close Button**, to provide more space for working on the report

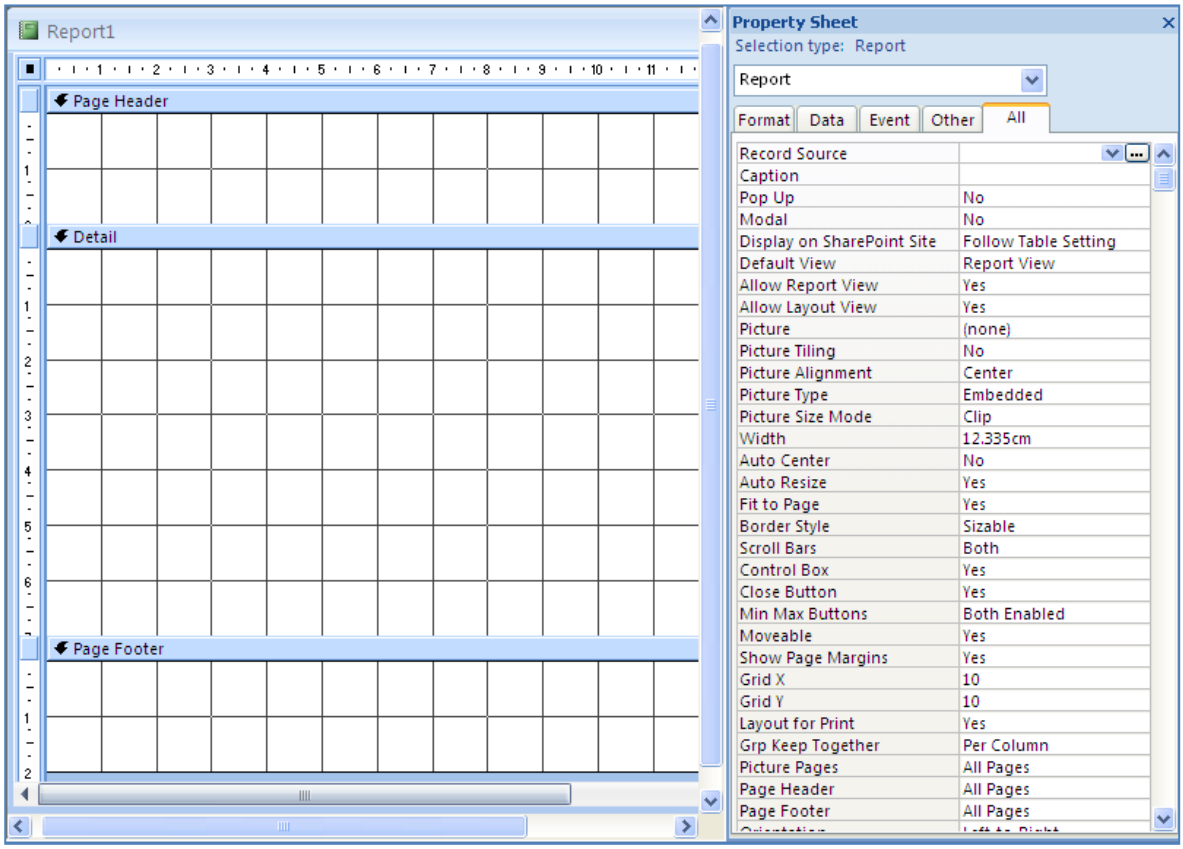


The Properties Sheet

You can affect the look, size and behaviour of the report by changing some of its properties.

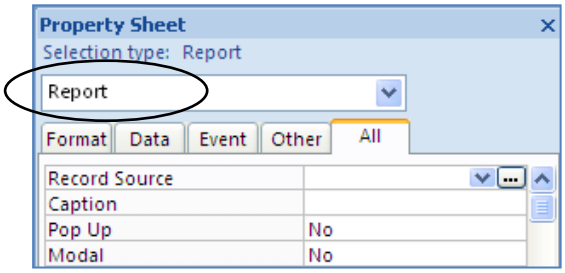
➢ Click **Property Sheet** from the **Tools** group on the **Design** tab

The Property Sheet is displayed.

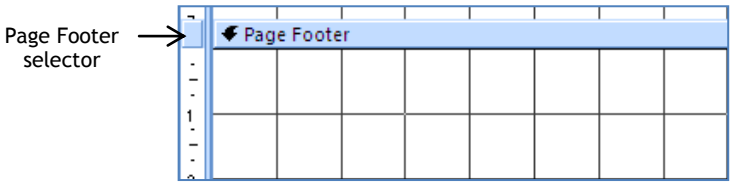


Properties exist for the whole report, for each section of the report, and for each individual control on the report. The Property Sheet pane is used to display the properties for any of these objects.

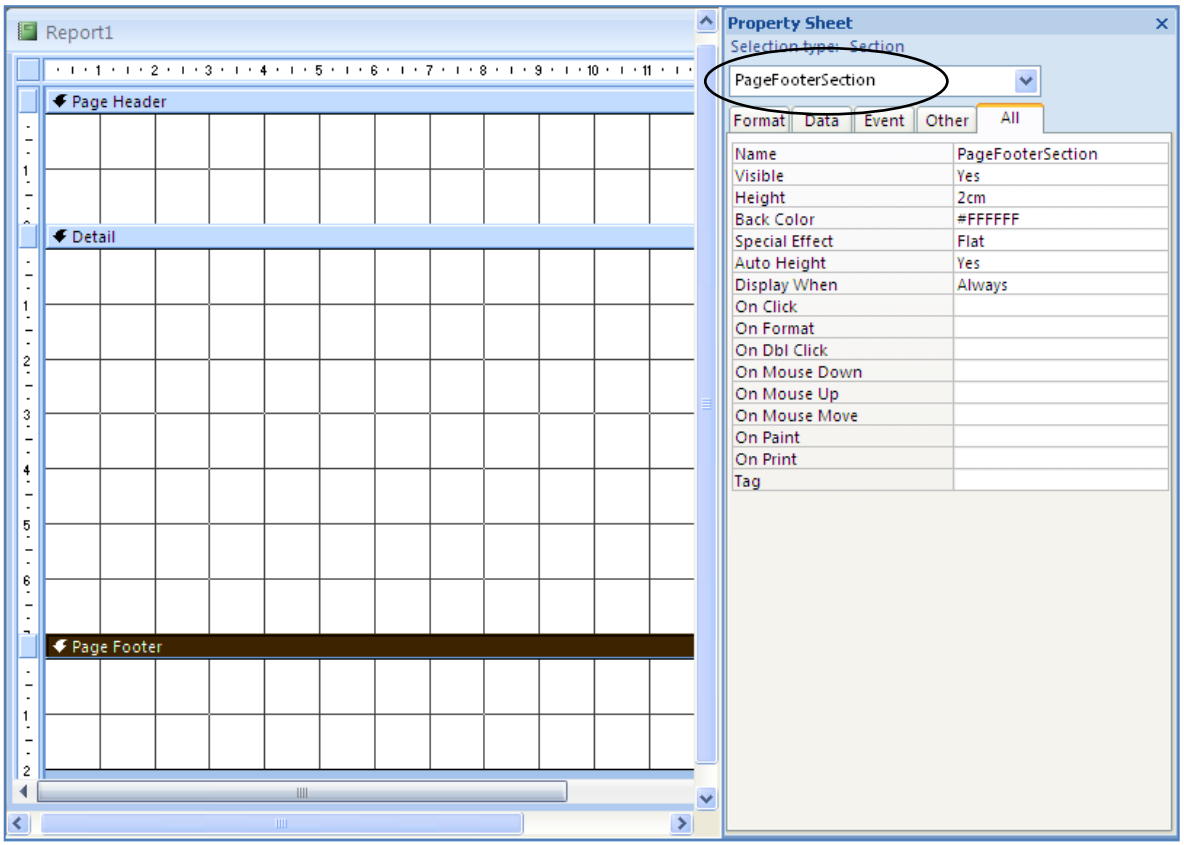
The whole report is selected, so the properties displayed are for the report, as indicated by the word **Report** at the top of the pane.



➢ Click the **Page Footer** selector



The Page Footer section is selected and the Property Sheet now displays the properties for this section of the report.



➢ Click the Report Selector (the square button just above the Page Header selector)

Alternatively

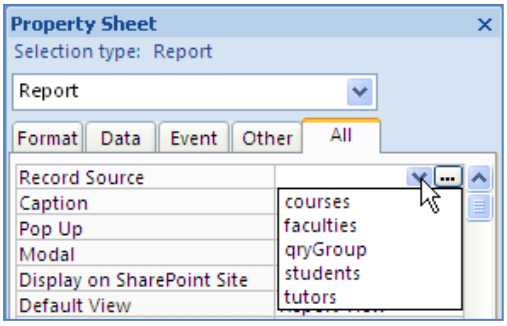
- Click the down arrow at the top of the Property Sheet
- Select **Report**

The Property Sheet displays the properties of the Report.

To indicate where the report should look for the data you need to set the Record Source property.

The **All** tab on the Property Sheet shows all the properties for the selected object. Choosing one of the other tabs will display only the properties in that category. It is sometimes easier to find an individual property from the shorter lists on the category tabs. For this exercise, however, there is no advantage in selecting another tab because the Record Source property is at the top of the list.

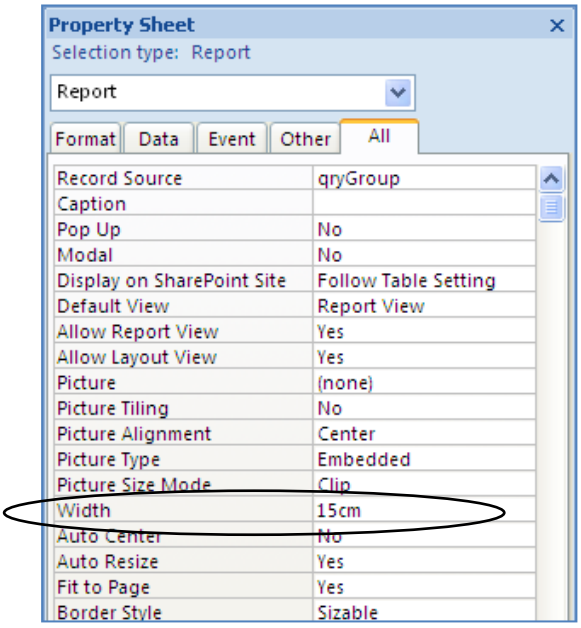
- Click the down arrow in the **Record Source** property



- Select **qryGroup**

The width of the report also needs to be changed. This can be done by dragging the edge of the report, but the value can be set accurately in the properties.

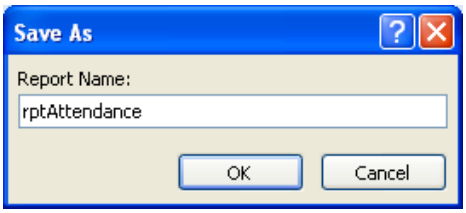
- Click into the **Width** property
- Change the value to **15cm**



Fifteen centimetres is the approximate width to fit an A4 page in portrait orientation, when the default margin size is taken into consideration. If you make the page wider than 15 cm the document is likely to print on two pages instead of one.

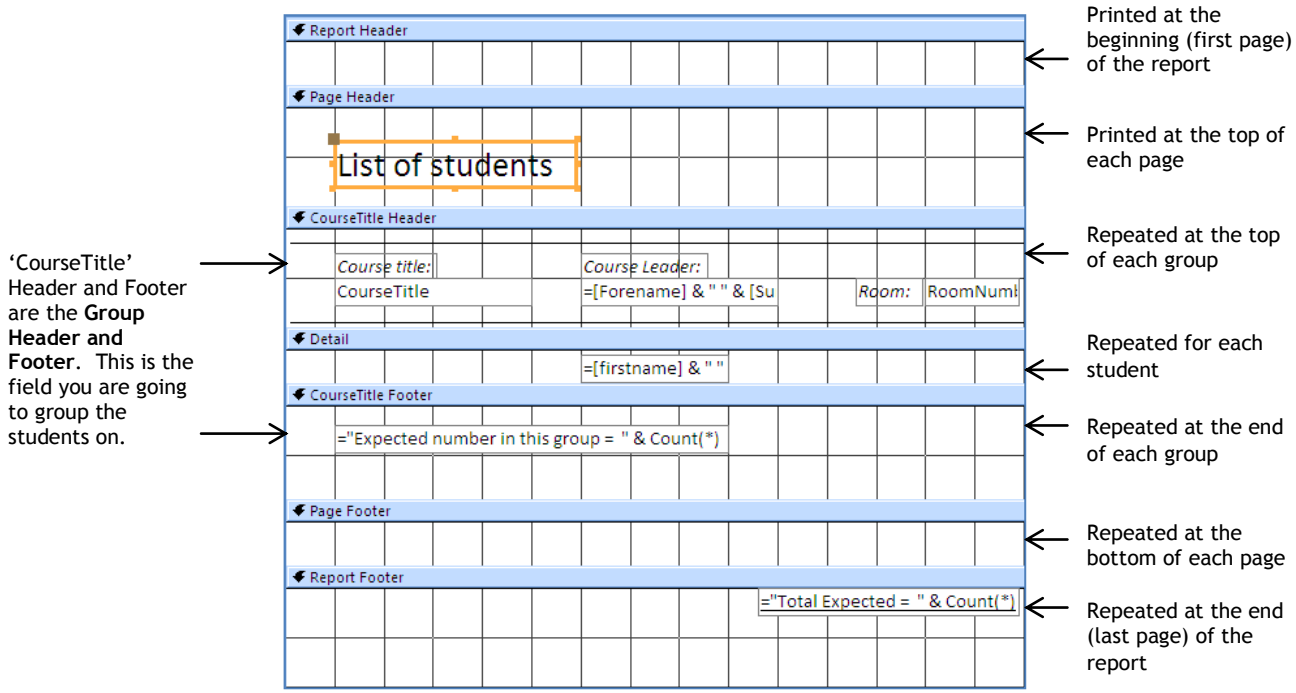
To save the report:

- Click the **Save** button
- Change the suggested Report Name to **rptAttendance**



- Click **OK**

The Design View of the report will look like this when all the controls are in place:



- Compare the above diagram with that on page 47 to see how the design relates to the printed report

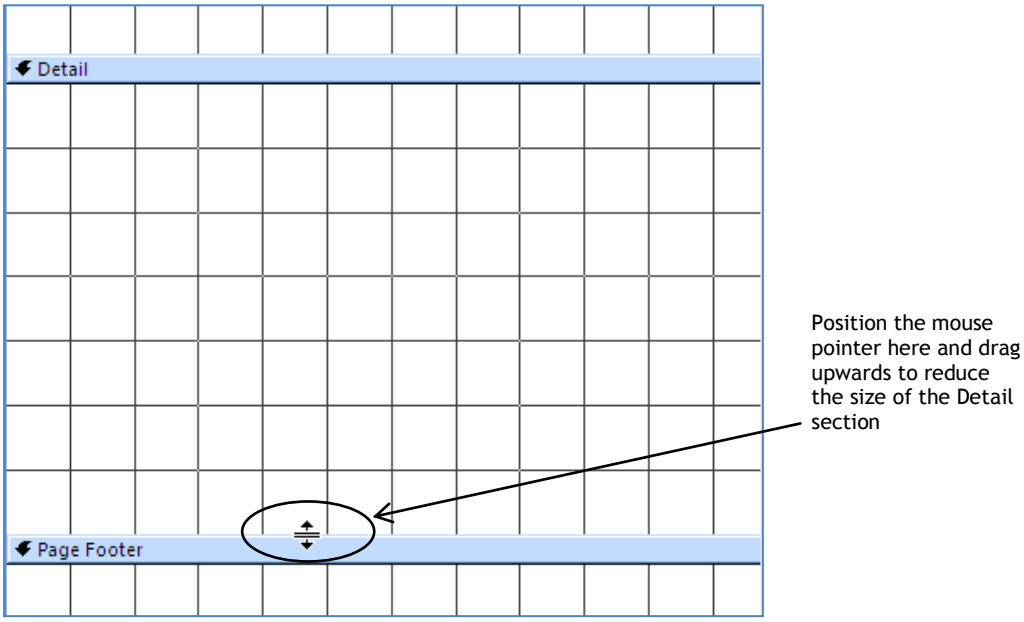
Your design does not have all the header and footer sections needed for the report, so these will have to be added.

First though, you will change the size of the Detail section.

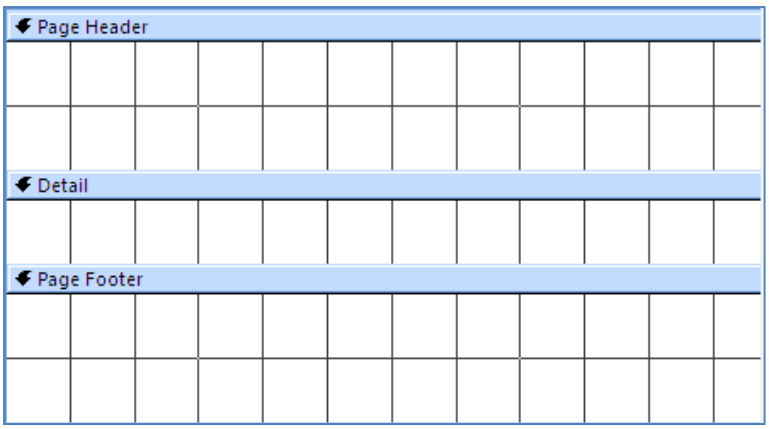
The only control that will be added to this section is a text box to display a student's name. Also, the height of the Detail section will determine the space used to display each name in the list; the greater the height, the larger the space between consecutive names in the list.

- Position the mouse pointer to the top of the Page Footer bar

The pointer changes to a double arrow.

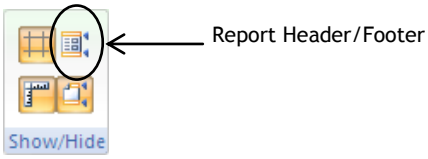


- Hold the mouse button down and drag upwards
- Reduce the Detail section to one row of squares

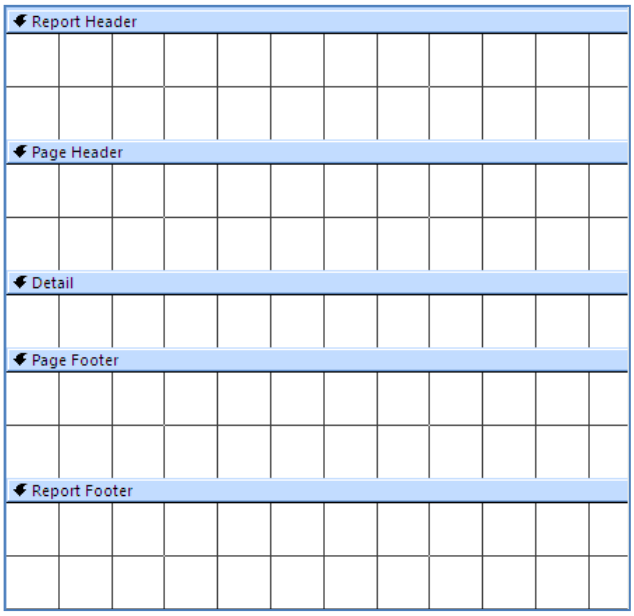


To add the Report Header and Footer:

- Click the **Report Header/Footer** button from the **Show/Hide** group on the **Arrange** tab



The Design page now looks like this:

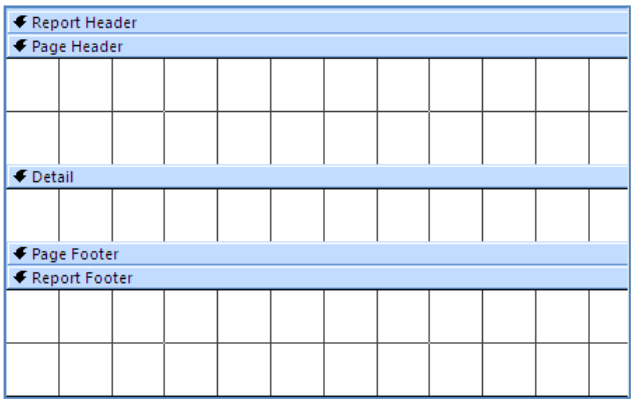


You will be using the Report Footer but not the Report Header.

- Drag the Page Header bar up to meet the Report Header bar

The Page Footer section will also not be needed.

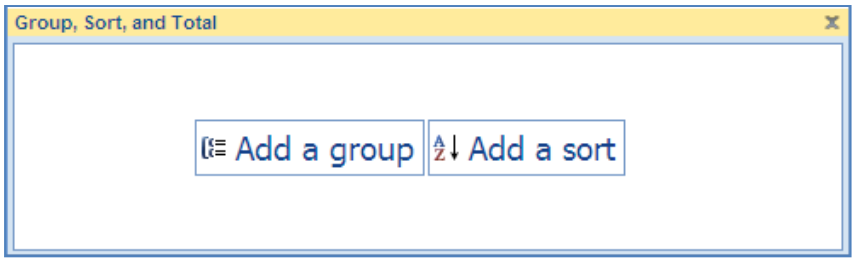
- Drag the Report Footer bar up to meet the Page Footer bar



To add the Group Header and Footer for the CourseTitle field:

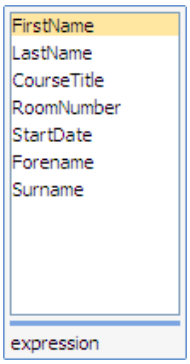
- Click the **Group & Sort** button from the **Grouping & Totals** group on the **Design** tab

The Group, Sort, and Total pane is displayed at the bottom of the Design page.



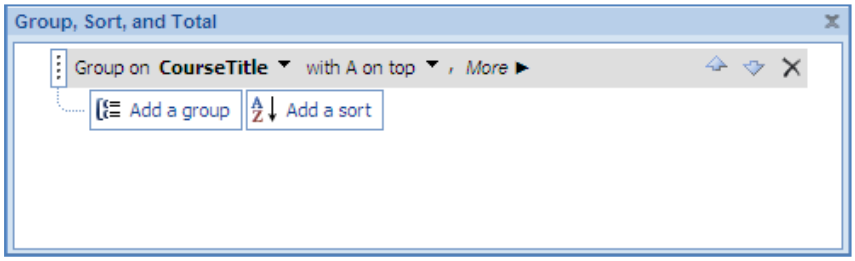
- Click **Add a group**

A list of available fields is displayed.



- Click **CourseTitle**

A CourseTitle Header section is added to the design and a CourseTitle options bar is displayed in the Group, Sort, and Total pane.

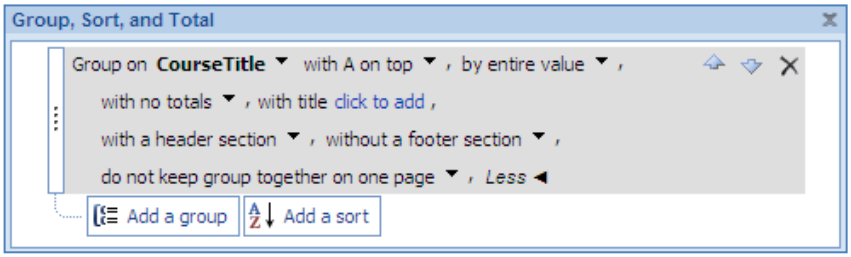


Sorting is automatically defined for the group. The text 'with A on top', on the options bar, indicates that the CourseTitle values will be sorted in ascending order. The Group Footer, however, is not displayed by default.

To display the CourseTitle Footer:

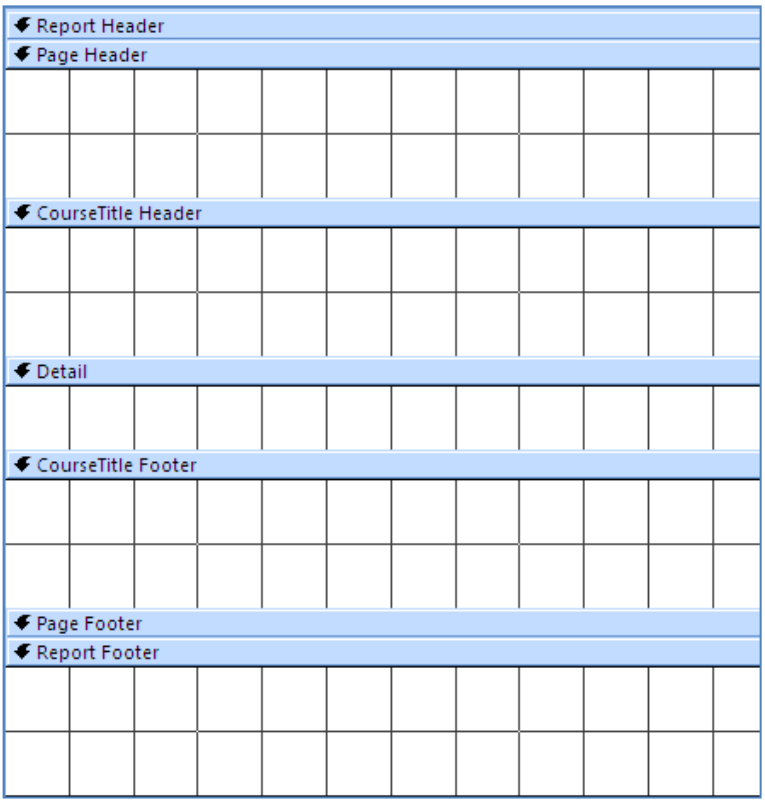
- Click the **More** button on the CourseTitle options bar

The bar expands to display all the possible options:



- Click the down arrow to the right of 'without a footer section'
- Select **with a footer section**

The CourseTitle Footer is added to the design.

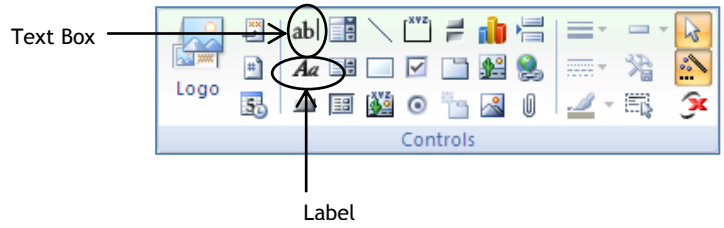


- Click the **Group & Sort** button to close the pane

You are now ready to put controls on the report.

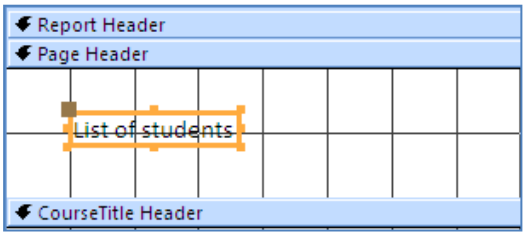
You will start with the title. The title **List of students** needs to be put in a Label in the Page Header section.

- Click the **Label** button from the **Controls** group on the **Design** tab



- Click in the Page Header at the point where you want the Label control to begin
- Type **List of students** in the label and press the **Enter** key

As you type, the Label expands to accommodate the text.



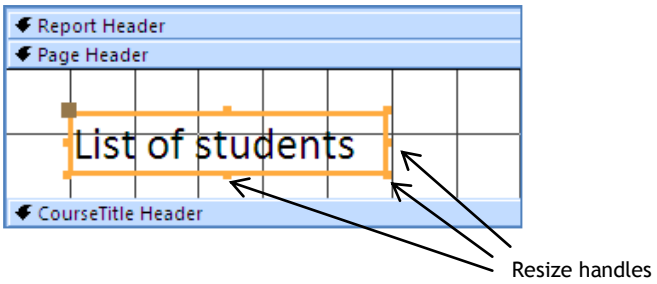
Notice that pressing **Enter** does not take you to a new line in the label. To go to a new line you would press **Ctrl** + **Enter**. Pressing enter takes you out of editing the text in the label and selects the label, highlighting the border in orange.

Now that the label is selected you can change its format.

- Using either the **Font** group of commands or the **Property Sheet**, change the **Font Size** to **20**

The text is now too large to be displayed fully in the label.

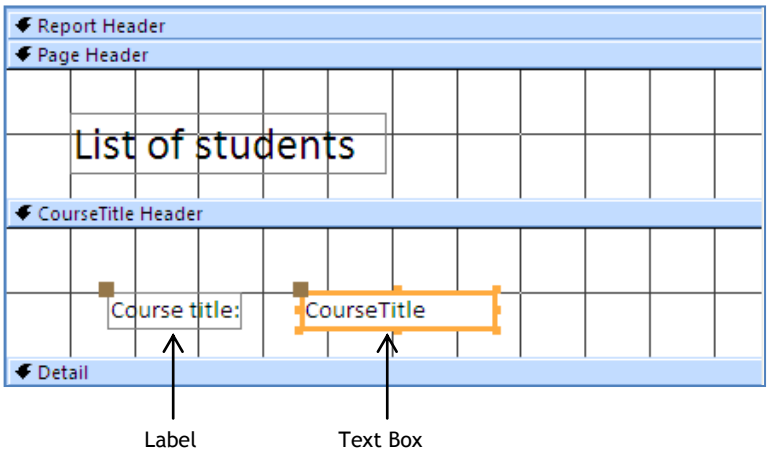
- Drag the resize handles to make the label long enough and wide enough to fully display the text



For each group, you want to display the title of the course, the course leader, and the room number. This involves displaying data from these three different fields.

To display the Field List:

- Click **Add Existing Fields** from the **Tools** group on the **Design** tab
- Drag the **CourseTitle** field into the middle of the CourseTitle Header section



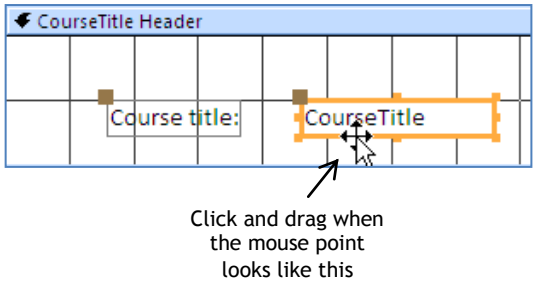
Two controls are added to the section: a text box in which individual course titles will be displayed, and a label to inform the user that the data in the text box is a 'Course title'.

The controls can be repositioned by dragging.

- Move the mouse pointer over the border of the text box, but avoiding the resize handles

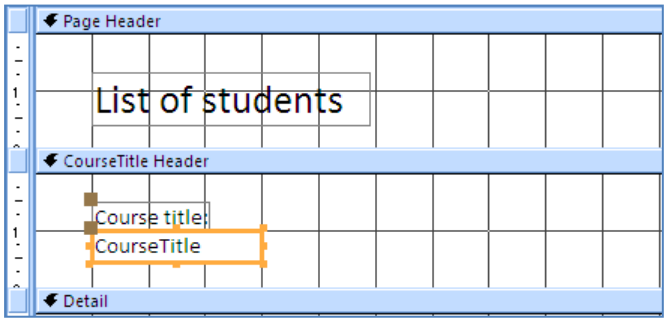
The mouse pointer changes to cross arrows.

- Click and drag in any direction

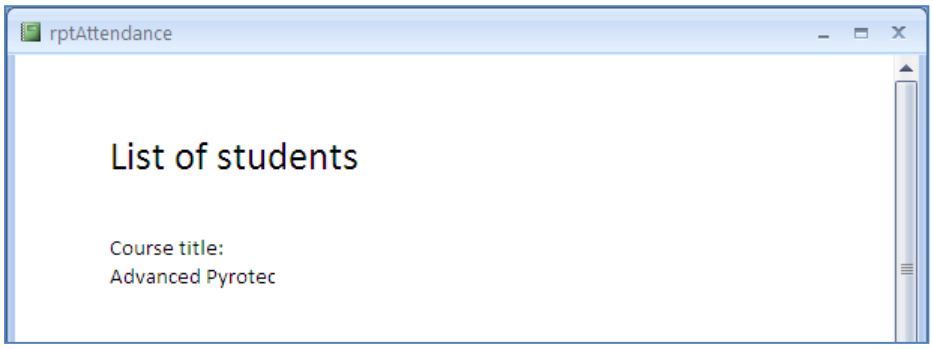


Both the text box and the label move together. To move each control individually you need to position the mouse pointer over the square in the top left corner of the control before dragging.

- Drag each control by the handle at the top left corner to reposition them near the left side of the page, with the label above the text box as shown below



- Change the view to **Report View**

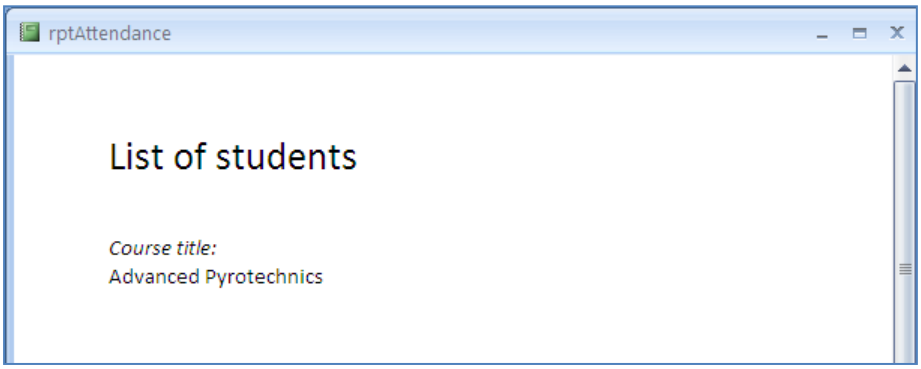


The title is not fully displayed, meaning that the text box needs to be resized.

- Switch to **Layout View**

You can resize the control in Design View, but Layout View is ideal for this job since you can see when the control is wide enough to display the text.

- Click on the Course title name to display the text box
- Drag the right border until the text box fully displays the Course title
- Select the label and format it as **Italic**



Detail section

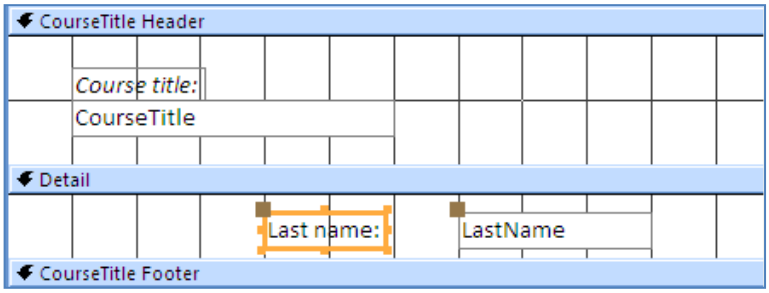
The controls in this section repeat for each record taken from the database.

To create a list of names for each course, you need to insert data from the firstName and lastName fields from the students table into the Detail section.

- Switch to **Design View**
- Drag the **lastName** field into the **Detail** section

For this report you do not need labels in the Detail section. To delete the label:

- Select the label

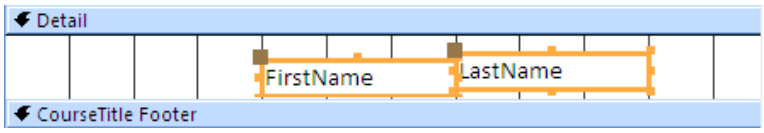


- Press the **Delete** key
- Drag the **firstName** field on to the report to the left of lastName
- Delete the firstName label

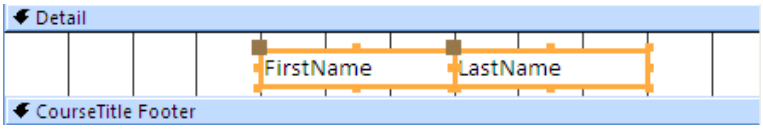
To align the two text boxes:

- Select the **firstName** text box
- Hold down the **Shift** key and select the **lastName** text box

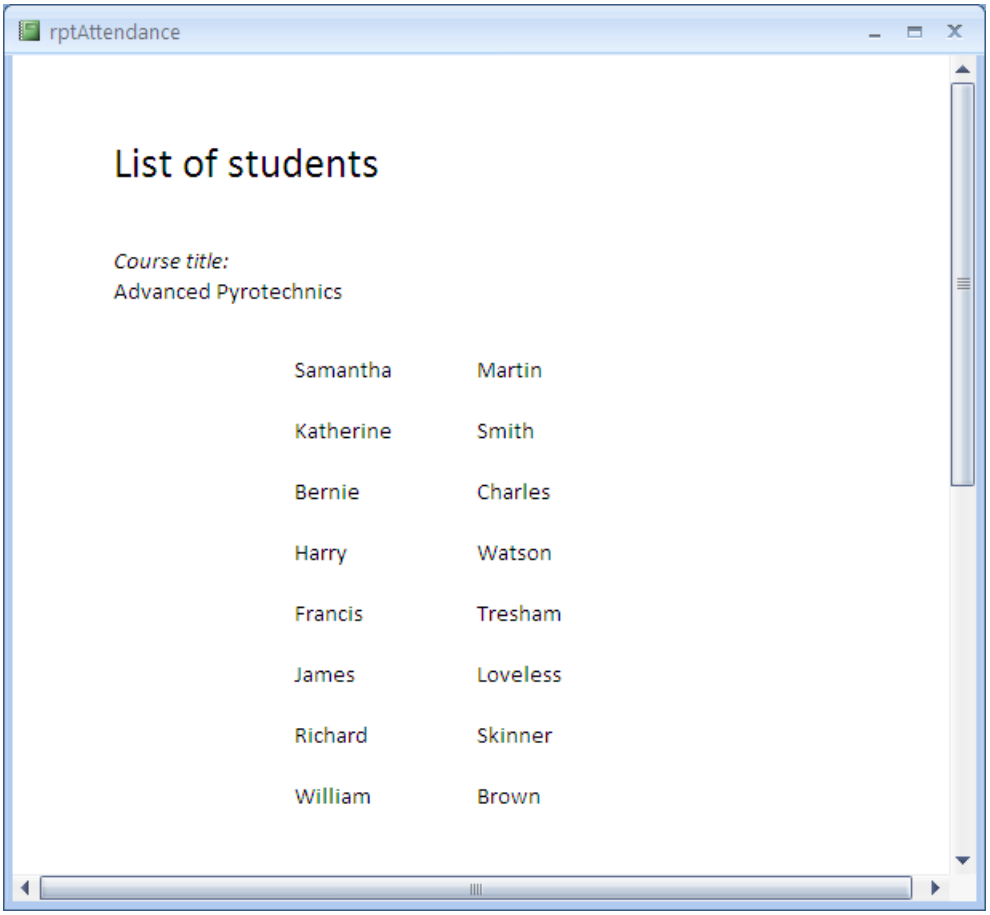
Both text boxes are selected.



- Click **Top** from the **Control Alignment** group on the **Arrange** tab



➢ Switch to **Report View**



A list of names is displayed.

➢ Scroll down the page to view the list for the second course

Calculated control

A list of names is displayed on the report, but the presentation would be improved by removing the large spaces between the first names and last names.

This can be done by using a calculated control in place of the two text boxes.

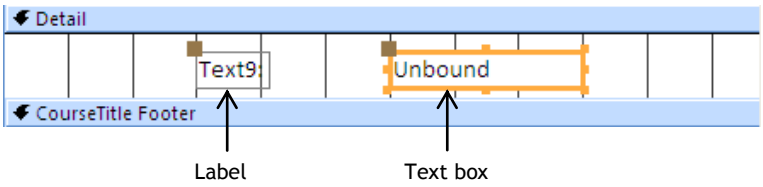
A calculated control may be used to perform calculations on data stored in the database, combine data from two or more fields, display text, or a combination of these.

You use an unbound text box to create a calculated control.

- Switch to **Design View**
- Delete the two text boxes in the Detail section
- Click the **Text Box** button from the **Controls** group on the **Design** tab

- Click into the middle of the Detail section

A label and an Unbound text box are inserted.



- Delete the label
- Click into the text box and type the following:

  =[FirstName] & " " & [LastName]

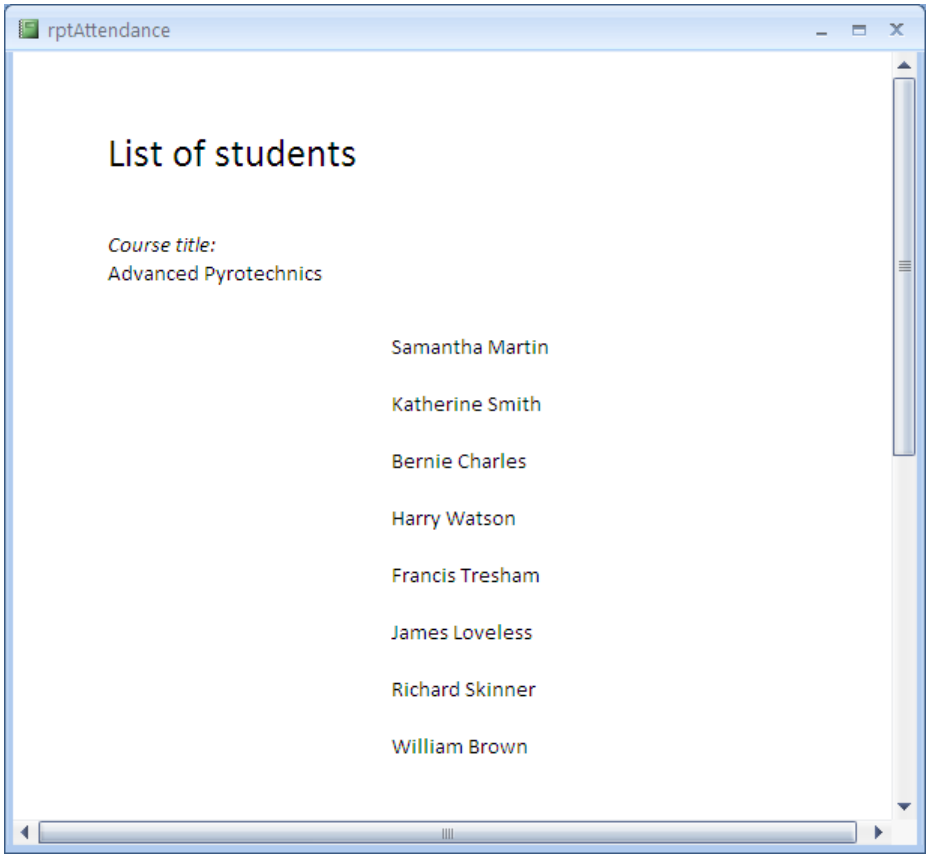
- Press Enter

This 'expression' will insert the value of the firstName field in the current record, add a space and then insert the value of the lastName field.

Rules for creating expressions:

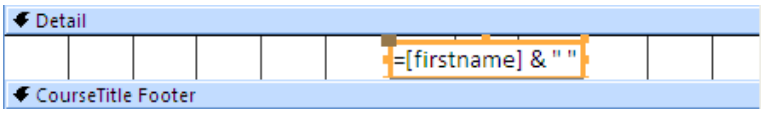
- An expression in a calculated control **always** begins with =.
- Field names **must** be enclosed in square brackets.
- Text **must** be enclosed within speech marks. In the example above, speech marks are used to insert a space between the first name and last name.
- Different items within the expression: field names, text, etc. **must** be joined by an ampersand.

- Save the report
- Switch to **Report View**



To reduce the space between the lines you need to reduce the size of the Detail section.

- In Design View, ensure that the calculated control is selected
- Use the arrow keys on the keyboard to move the control as close as possible to the top of the Detail section
- Drag the **CourseTitle Footer** bar up to the bottom of the control

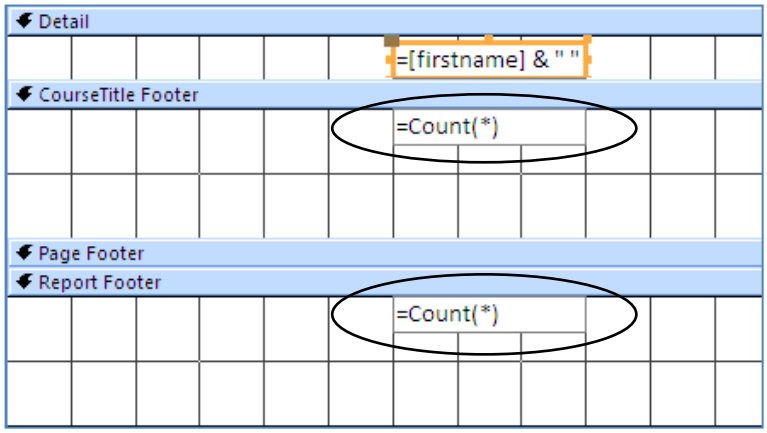


Counting records

To add a count for each group and for the whole report:

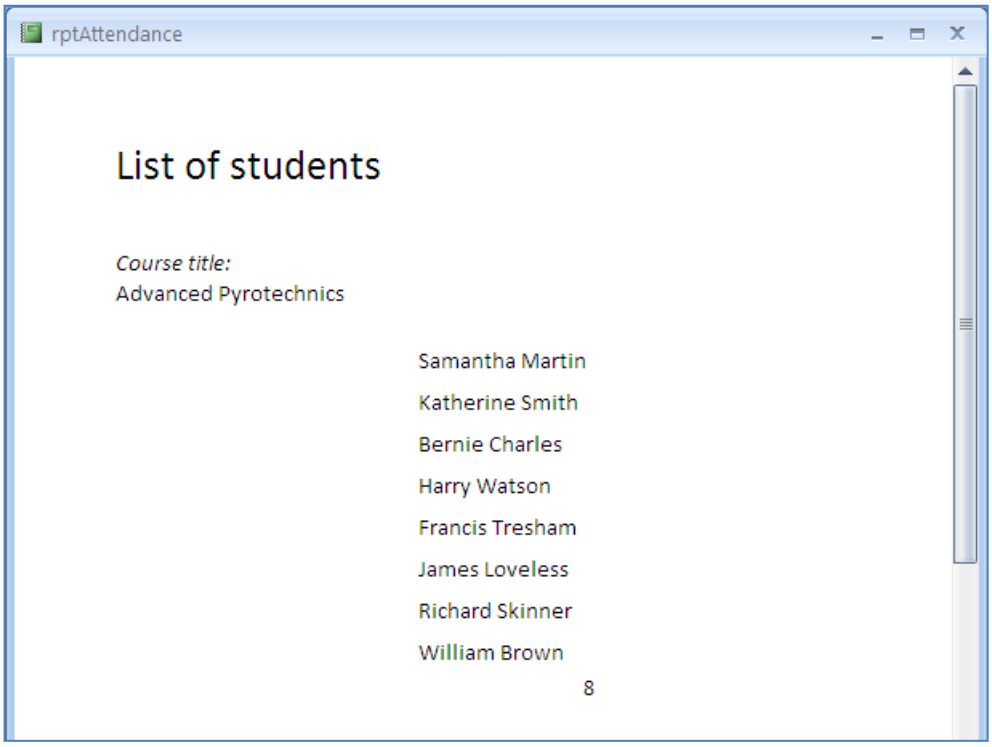
- Click **Totals** from the **Grouping & Totals** group on the **Design** tab
- Select **Count Records** from the list of options

Calculated controls are inserted into the CourseTitle Footer and Report Footer sections, with the expression **=Count(*)**.



Count is one of the many functions available in Access. The asterisk instructs Access to count all records.

- View the results by switching to Report View



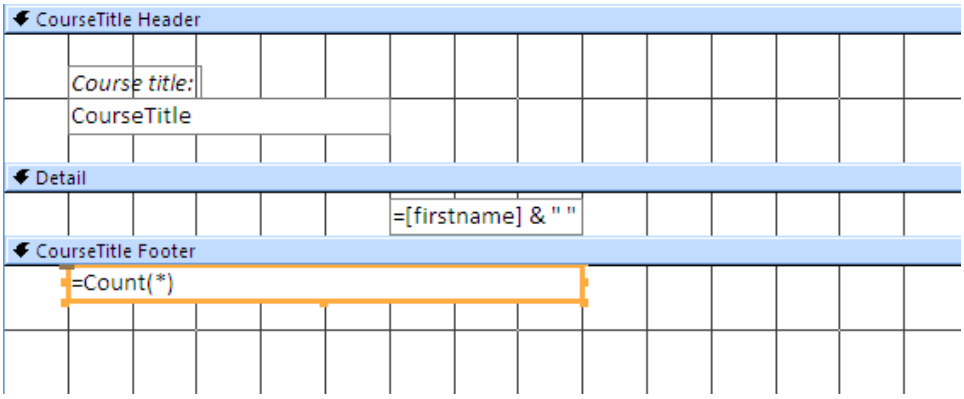
- Scroll down to see the count for the second group and for the whole report

More space between the list of names and the count value would be useful, but of more importance, you need to include text to explain what these numbers are.

- Switch to Design View

To add helpful text to the count, you could insert a label to the left of the calculated control, or you could extend the **=Count(*)** expression to include appropriate text. For this exercise, you will add text to the calculated control.

- Click the calculated control in the CourseTitle Footer
- Use the resize handle to extend the left border of the control so it aligns with the controls in the CourseTitle Header



- Click in the selected control to put the insertion point in the control
- Change the expression to the following:

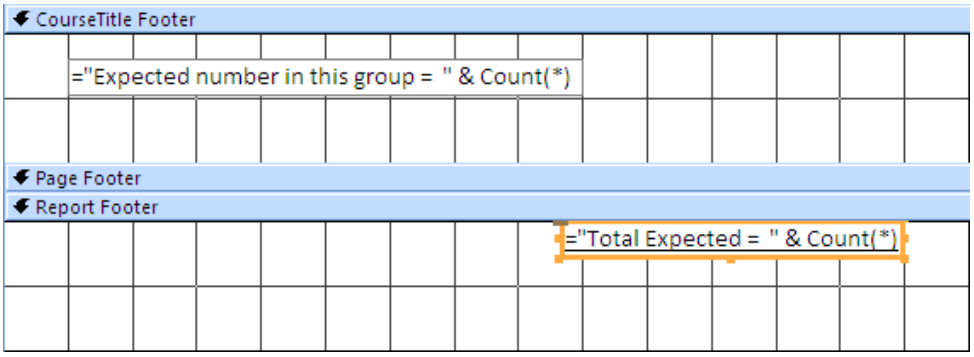
  =“Expected number in this group = ” & Count(*)

- Press Enter
- To create some space between the count value and the list of names above it, use the down arrow key to move the control down within the CourseTitle Footer (see the diagram below)
- Change the expression in the control in the Report Footer to:

  =“Total Expected = ” & Count(*)

- Resize the control so that the full expression is displayed
- Move the control to the right of the report

- Use the **Underline** tool in the **Font** group to underline the contents of the control



- View the results in Report View

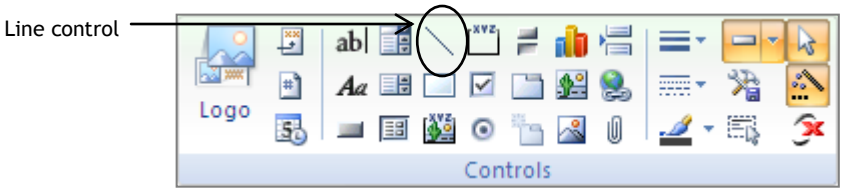
Adding horizontal lines

To improve the design of the report and separate the course title from the list of students, you will draw two horizontal lines in the group header.

- Ensure that you are in Design View

To draw a line above the controls in the CourseTitle Header:

- Click the **line** button from the **Controls** group on the **Design** tab

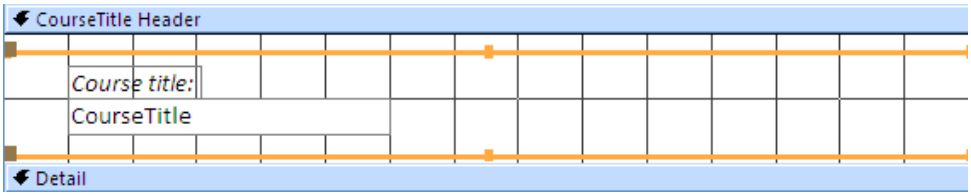


- Move the mouse pointer to the point in the CourseTitle Header above the controls where you want the line to start
- Hold down the mouse button, drag the mouse to the right and release the button where you want the line to end

While dragging the mouse, the line will be invisible if it is perfectly horizontal and will only become visible when you release the mouse button.

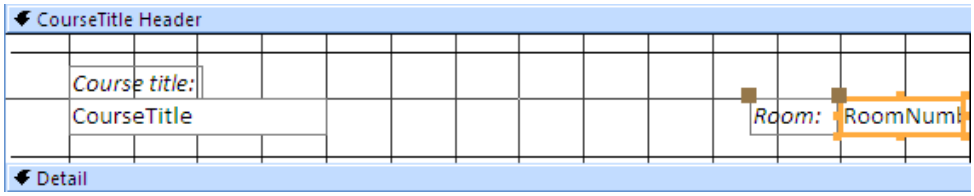
If the line that you have drawn is not perfectly horizontal, to straighten it, change the **Height** property to **0cm**.

- Draw a second horizontal line, the same length as the first, below the controls in the CourseTitle Header

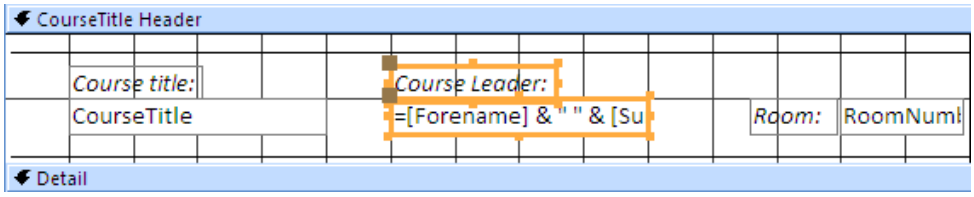


To complete the report:

- Drag the **RoomNumber** field to the **CourseTitle Header**
- Edit the label control to display **Room:**
- Italicise the label
- Resize and reposition the controls as shown below



- Create a calculated control in the **CourseTitle Header** to display the **Forename** and **Surname** of the course tutor (ensure that there is a space between the two values)
- Edit the label control to display **Course Leader:**
- Italicise the label
- Reposition the controls as shown below



- View the results in **Report View** (your report should be similar to the one shown on page 47)
- Save and close the report

FILTERING

You can filter a report to display only the records that satisfy the particular filter criteria.

- Open **qryGroup** in **Design View**
- Delete the **CourseTitle** criteria
- Save and close the query
- Double click the report **rptAttendance** in the Navigation pane

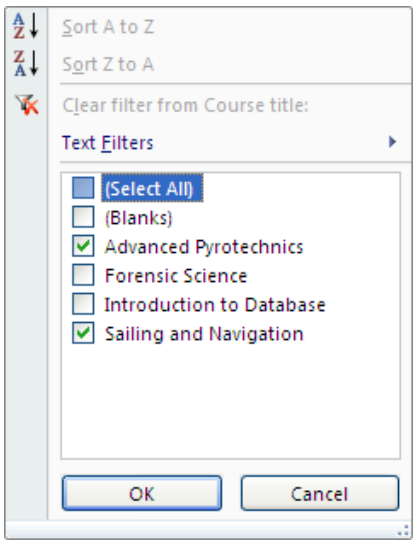
This opens it in Report View.

Now that you have deleted the criteria from the query on which the report is based, it displays lists for four courses.

- Scroll down the report to check that all lists are displayed and that the total number of students expected is 31

To filter the report to display only the lists for Advanced Pyrotechnics and Sailing and Navigation:

- Click on any Course title in the report
- Click **Filter** from the **Sort & Filter** group on the **Home** tab
- Click the **(Select All)** check box to remove all the ticks
- Select the **Advanced Pyrotechnics** and **Sailing and Navigation** check boxes



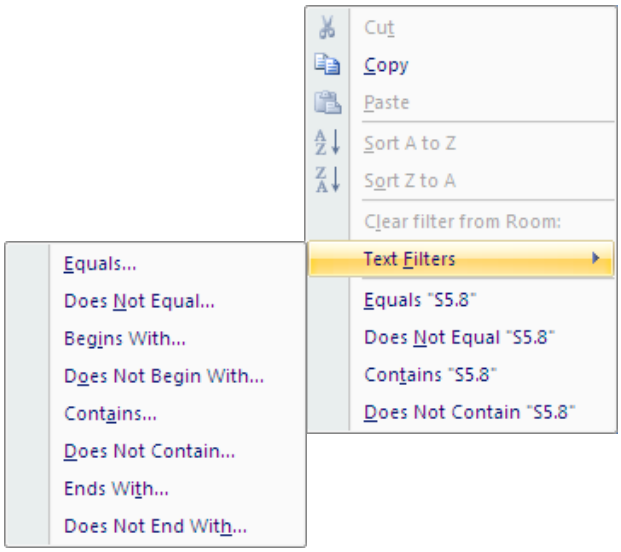
- Click **OK**
- Check that only these two lists are displayed

To remove the filter:

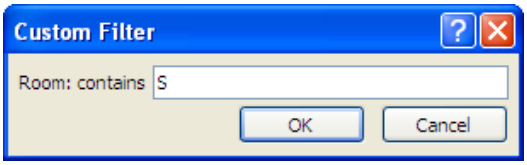
- Click **Toggle Filter** from the **Sort & Filter** group

To filter the report to display only the lists for which the **Room number** includes the letter **S**:

- Right-click on any Room number value
- Point to **Text Filters** from the shortcut menu



- Click **Contains**
- Type **S** in the Custom Filter box



- Click **OK**
- Check that the report displays lists for rooms **S5.8** and **S0.55** only
- Close the report

Saving a filtered report does not save the filter.

To save a filtered report you must create a query with the appropriate criteria and base the report on the query.

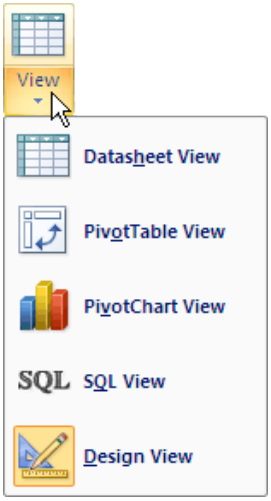
- Open the query **qryGroup** in Design View
- For the **CourseTitle** criteria, type **"Advanced Pyrotechnics" or "Sailing and Navigation"**

A query is a specific request for the retrieval, creation, modification, or deletion of data.

The lower pane in the Query Design window is called the QBE (Query by Example) grid. This is where you specify the fields and criteria upon which the query should act. This interface makes it easier for you to create queries, but all Access queries use SQL (Structured Query Language). Access converts what you specify in the Query Design window into SQL code.

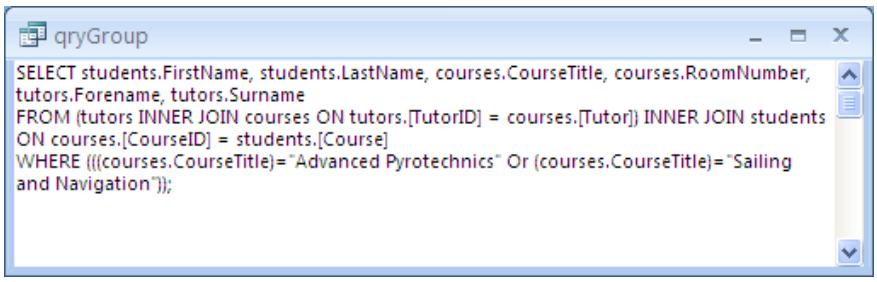
To view the SQL code for the current query:

- Click the down arrow on the **View** button



- Select **SQL View**

The SQL code is displayed in the query window.



If you know SQL you can use this view to build, edit, or troubleshoot your queries.

ACTION QUERIES

There are two basic types of queries: Select Queries and Action Queries.

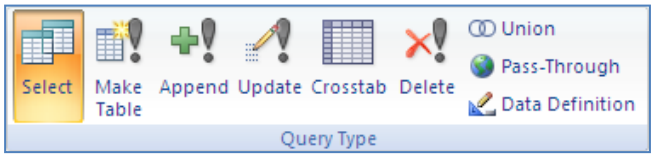
A Select query does not alter the data stored in the database, but simply retrieves data from one or more tables for the purpose of displaying it.

Action queries alter the data stored by performing additional operations on the data selected. They are used to insert, modify or delete data from the database.

To see the different types of queries:

- Display the query in Design View

The **Query Type** group is shown on the **Design** Query Tools tab.



Notice that the **Select** type is highlighted, indicating that this query is a Select query.

Query Type	*Explanation*
Make Table	Creates a new table and copies the results of the query into it
Append	Copies the results of the query into an existing table as new records
Update	Changes values of individual fields in one or more existing records in a table
Delete	Deletes the selected records from a table
Crosstab	Summarises data by separating it into groups based on one or more fields
Union	Combines the results of two or more similar Select queries
Pass-Through	Used to send commands directly to a database server
Data Definition	Used to create or edit database objects eg tables, indexes, etc.

In general, there are two steps involved in creating an Action query: you first create a Select query that identifies specific records and fields, and then convert the query to an Action query. However, the last three types listed above are SQL specific and must be written directly using SQL code.

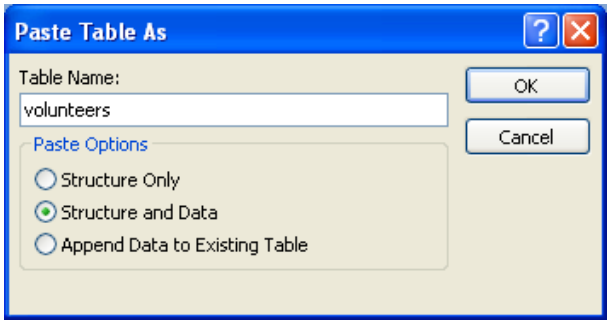
- Save and close **qryGroup**

You are going to look at two Action queries. First you will create a new table to use for the exercises.

- Select the **tutors** table in the navigation pane (do not open it)
- Click **Copy**
- Click **Paste**

The Paste Table As dialog box is displayed.

- Type **volunteers** in the Table Name box



- Ensure that **Structure and Data** is selected as the Paste Option
- Click **OK**
- Open the **volunteers** table in Design View
- Change the **TutorID** Field Name to **ID** and the Caption to **Volunteer ID**
- Save the changes
- In Datasheet View, delete all the entries from the **Email** field

Volunteers

Volunteer ID	Forename	Surname	Telephone	Email	Add New Field
1	Guido	Fawkes	2543875		
2	William	Bligh	2546238		
3	George	Loveless	2546489		
4	Alison	Sparks	2542836		
5	Mary	Hopkins	2547621		
(New)					

- Close the table

Append queries

An Append query is used to copy records from one table to another. The fields that you intend to copy do not need to have the same names, but they must have the same, or compatible, data types.

All the students in the students table from Leicestershire (with postcode LE....) need to be added to the volunteers table.

You will create and run an Append query to complete this task.

The first step is to create a Select query to pick out the appropriate fields and records from the students table. The only data you are interested in are the names, telephone numbers, and postcodes.

- Create a Select query in Design View with the fields **FirstName**, **LastName**, **Telephone**, and **Postcode** from the **students** table

To filter all the records with a postcode beginning with LE:

- Type **LE*** for the Postcode criteria

The asterisk character is used as a 'wildcard' to represent any one or more characters.

A question mark is used to represent any one character.

- View the query in Datasheet View to check that it has filtered the correct records

Query1

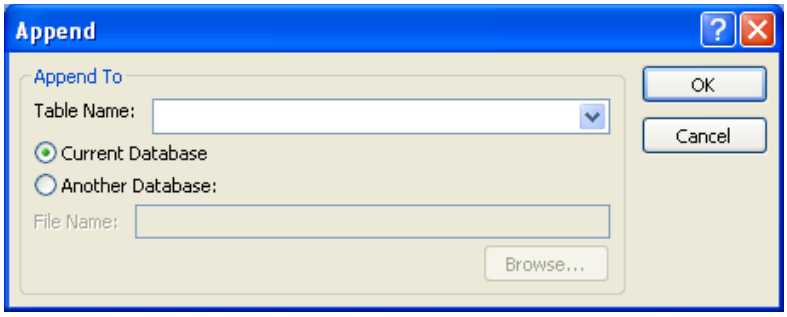
First name	Last name	Telephone	Postcode
William	Brown	2567657	LE7 3PS
Christopher	Wright	2652462	LE20 3MN
Harry	Watson	2105874	LE4 6JF
Bernie	Charles	2743928	LE12 3DS
William	Brown	2567657	LE7 3PS
Peter	Harrison	2716849	LE17 3BH
*			

- Switch to Design View

The next step converts the Select query to an Append query.

- Select **Append** from the **Query Type** group on the **Design** tab

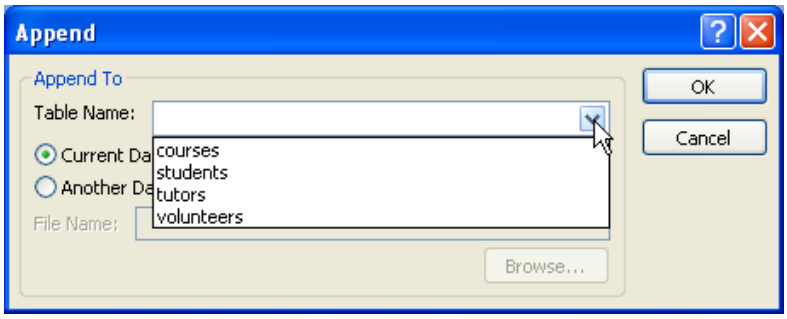
The Append dialog box is displayed.



- **Current Database** is selected by default

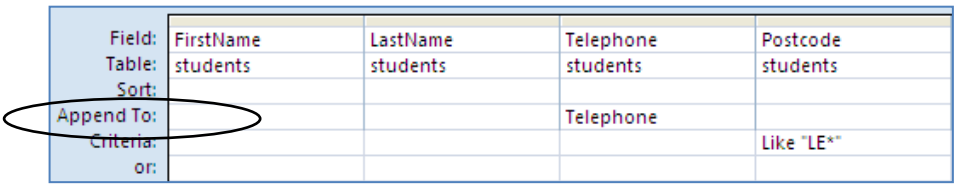
To choose the table that the records will be added to:

- Click the down arrow in the **Table Name** box



- Select **volunteers**
- Click **OK**

An **Append To:** row is added to the QBE grid.



Notice also that **Append** is highlighted in the Query Type group on the Ribbon, indicating that this is now an Append query.

Warning

The two commands in the Results group on the Design tab perform different functions for an Action query.

The **Run** command is used to perform the action specified by the query – in this case append the selected records to the specified table. So use this only when you are ready to add the records to the table.

The **View** command is used to switch between views and is useful to check which fields and records will be affected by the query.

- Use the **View** command button to switch to Datasheet View

Only the Telephone field is displayed.

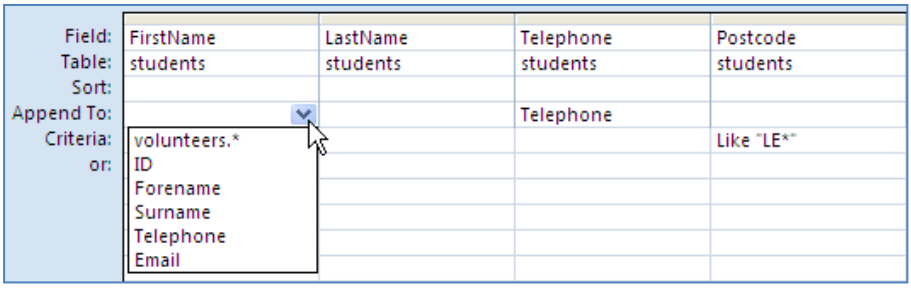
- Switch to Design View

Notice that Telephone is the only field identified in the Append To row.

This is because this field name exists in both the students table, where the data is being taken from, and the volunteers table, where the records will be added.

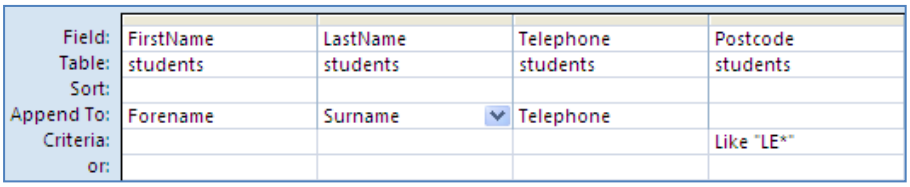
You need to indicate where the data from the other fields should be stored.

- Click in the **Append To** box for the **FirstName** field
- Click the down arrow



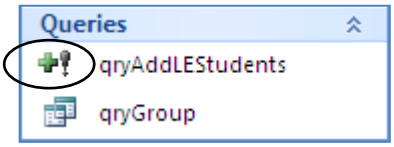
The list of fields in the volunteers table is displayed.

- Select **Forename**
- Select **Surname** as the field to append the **LastName** data to



- Save the query using the name **qryAddLEStudents**

Notice in the Navigation pane that the Append icon is used to indicate the type of query.

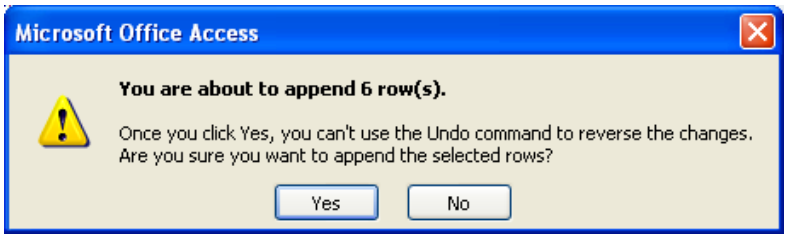


- Check in Datasheet View that data from three fields are now displayed

There are two ways that you can *Run* the query to perform the specified action.

- Ensure that the query is displayed in Design View
- Click **Run**

A message box is displayed, informing you that 6 rows are about to be added, and warning that you cannot use the Undo command to reverse the action.



- Click **Yes** to append the records
- Close the query
- Open the **volunteers** table

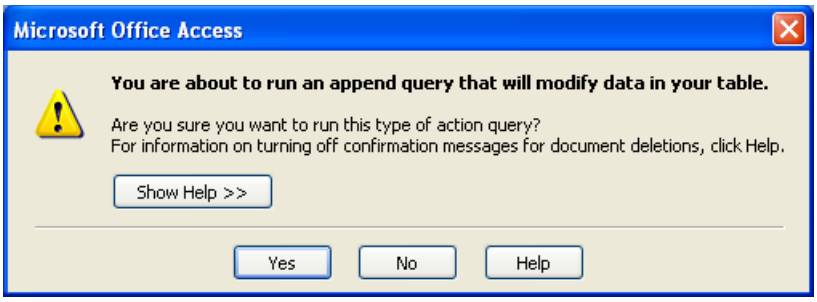
The six student records have been added to the table.

volunteers

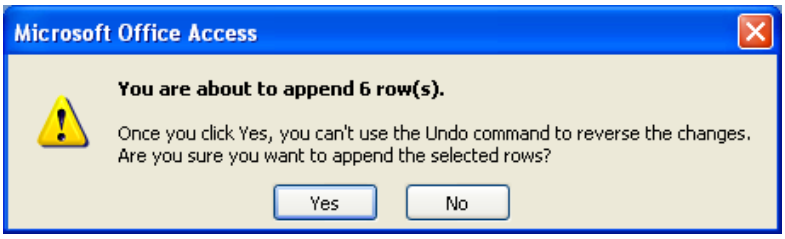
Volunteer ID	Forename	Surname	Telephone	Email	Add New Field
1	Guido	Fawkes	2543875		
2	William	Bligh	2546238		
3	George	Loveless	2546489		
4	Alison	Sparks	2542836		
5	Mary	Hopkins	2547621		
6	William	Brown	2567657		
7	Christopher	Wright	2652462		
8	Harry	Watson	2105874		
9	Bernie	Charles	2743928		
10	William	Brown	2567657		
11	Peter	Harrison	2716849		
*	(New)				

- Close the table
- Double click the query **qryAddLEStudents** in the Navigation pane

A message is displayed informing you that this will run the append query, and seeking confirmation that you wish to continue with the action.



- Click **Yes**



- Click **Yes**
- Open the **volunteers** table

The records from the students table are added a second time.

volunteers

Volunteer ID	Forename	Surname	Telephone	Email	Add New Field
1	Guido	Fawkes	2543875		
2	William	Bligh	2546238		
3	George	Loveless	2546489		
4	Alison	Sparks	2542836		
5	Mary	Hopkins	2547621		
6	William	Brown	2567657		
7	Christopher	Wright	2652462		
8	Harry	Watson	2105874		
9	Bernie	Charles	2743928		
10	William	Brown	2567657		
11	Peter	Harrison	2716849		
12	William	Brown	2567657		
13	Christopher	Wright	2652462		
14	Harry	Watson	2105874		
15	Bernie	Charles	2743928		
16	William	Brown	2567657		
17	Peter	Harrison	2716849		
(New)					

- Close the table

Update queries

An Update query can be regarded as a powerful and flexible *Find and Replace* tool.

Using an Update query you can change the values of specific fields in all records or selected records in a table. The new values for the fields can be taken from other tables in the database or can be typed directly into the query.

You will use an Update query to add email addresses to the volunteers table.

- Create a new query in Design View
- Add the volunteers table to the Query Design

All the student volunteers will share the same email. The address is 'dsu@dmu.ac.uk'. The student volunteers start with record ID 6.

For an Update query, you need to add to the query grid only the fields that are to be updated and the fields that will be used to select the appropriate records.

- Add the **Email** and **ID** fields to the query grid
- In the **ID** column, set the criteria **>5**
- Click **Update** in the **Query Type** group on the **Design** tab

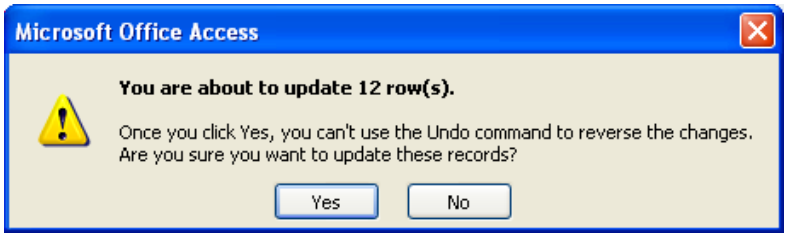
An *Update To* row is added to the QBE design grid.

- In the **Update To** row of the **Email** column, type **"dsu@dmu.ac.uk"**

Field:	Email	ID
Table:	volunteers	volunteers
Update To:	"dsu@dmu.ac.uk"	
Criteria:		>5
or:		

- Click **Run**

A message is displayed informing you of the number of records that are about to be updated, and warning that you that the procedure cannot be reversed.



- Click **Yes**

- Open the **volunteers** table

The email address has been added to all but the first five records.

volunteers

Volunteer ID	Forename	Surname	Telephone	Email	Add New Field
1	Guido	Fawkes	2543875		
2	William	Bligh	2546238		
3	George	Loveless	2546489		
4	Alison	Sparks	2542836		
5	Mary	Hopkins	2547621		
6	William	Brown	2567657	dsu@dmu.ac.uk	
7	Christopher	Wright	2652462	dsu@dmu.ac.uk	
8	Harry	Watson	2105874	dsu@dmu.ac.uk	
9	Bernie	Charles	2743928	dsu@dmu.ac.uk	
10	William	Brown	2567657	dsu@dmu.ac.uk	
11	Peter	Harrison	2716849	dsu@dmu.ac.uk	
12	William	Brown	2567657	dsu@dmu.ac.uk	
13	Christopher	Wright	2652462	dsu@dmu.ac.uk	
14	Harry	Watson	2105874	dsu@dmu.ac.uk	
15	Bernie	Charles	2743928	dsu@dmu.ac.uk	
16	William	Brown	2567657	dsu@dmu.ac.uk	
17	Peter	Harrison	2716849	dsu@dmu.ac.uk	
(New)					

- Close the table

You will now use the Update query to copy the email addresses of the tutors from the tutors table to the volunteers table.

- Delete the **ID** field from the QBE grid

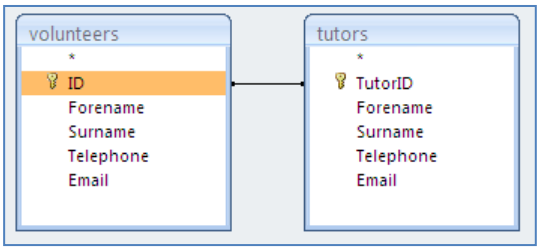
- Add the **tutors** table to the query design

Access needs to know what the relationship is between these two tables. You initially copied the tutors from the tutors table into the first records of the volunteers table, so the ID numbers of the tutors should be the same in both tables. The ID field will therefore be used to define a relationship between the tables for the purpose of this query.

- Position the mouse pointer on the **ID** field in the **volunteers** table and click and hold down the left mouse button

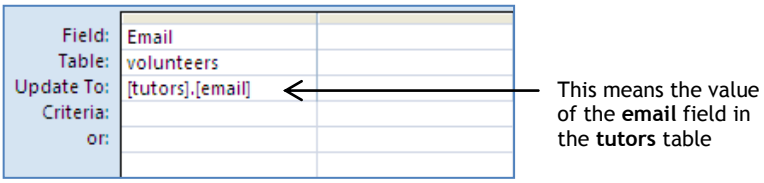
- Drag to the **TutorID** field in the **tutors** table and release the mouse button

A relationship line is displayed connecting the tables via the ID fields.



You now want to instruct the query to update the email field to the email values from the tutors table.

- In the **Update To** row, replace the existing text with **[tutors].[email]**



- Save the query as **qryEmailUpdate**
- Click **Run**
- Click **Yes** to close the warning and complete the update
- Open the **volunteers** table

The five tutors' email addresses are added to the table.

- Ensure that the email addresses are fully displayed

volunteers

Volunteer ID	Forename	Surname	Telephone	Email	Add New Field
1	Guido	Fawkes	2543875	gfawkes@uni.ac.uk	
2	William	Bligh	2546238	wbligh@uni.ac.uk	
3	George	Loveless	2546489	gloveless@uni.ac.uk	
4	Alison	Sparks	2542836	asparks@uni.ac.uk	
5	Mary	Hopkins	2547621	mhopkins@uni.ac.uk	
6	William	Brown	2567657	dsu@dmu.ac.uk	
7	Christopher	Wright	2652462	dsu@dmu.ac.uk	
8	Harry	Watson	2105874	dsu@dmu.ac.uk	
9	Bernie	Charles	2743928	dsu@dmu.ac.uk	
10	William	Brown	2567657	dsu@dmu.ac.uk	
11	Peter	Harrison	2716849	dsu@dmu.ac.uk	
12	William	Brown	2567657	dsu@dmu.ac.uk	
13	Christopher	Wright	2652462	dsu@dmu.ac.uk	
14	Harry	Watson	2105874	dsu@dmu.ac.uk	
15	Bernie	Charles	2743928	dsu@dmu.ac.uk	
16	William	Brown	2567657	dsu@dmu.ac.uk	
17	Peter	Harrison	2716849	dsu@dmu.ac.uk	
*	(New)				

- Save and close the table
- Close the query

Before running any action query in a working database, it is advisable to first make a backup of the database because the action cannot be reversed.

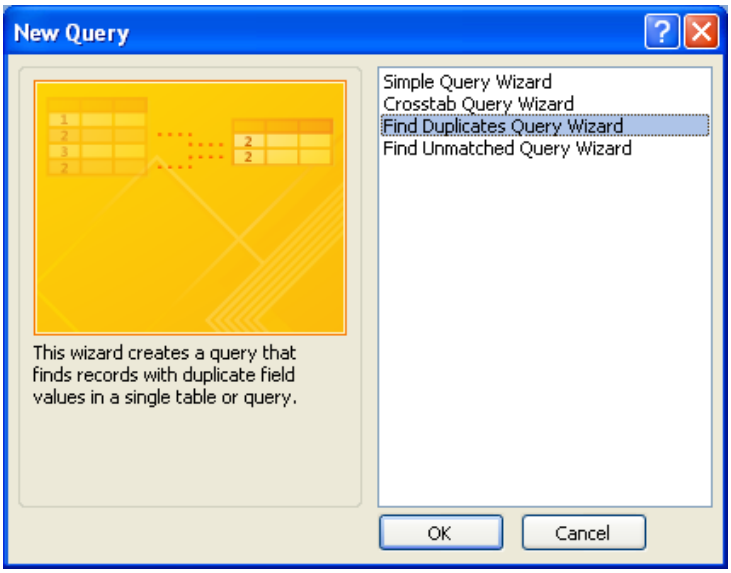
SELECT QUERIES

Access includes wizards to enable you to find records that have been duplicated within a table, and to select records in one table that have no matching records in a related table.

Finding duplicate records

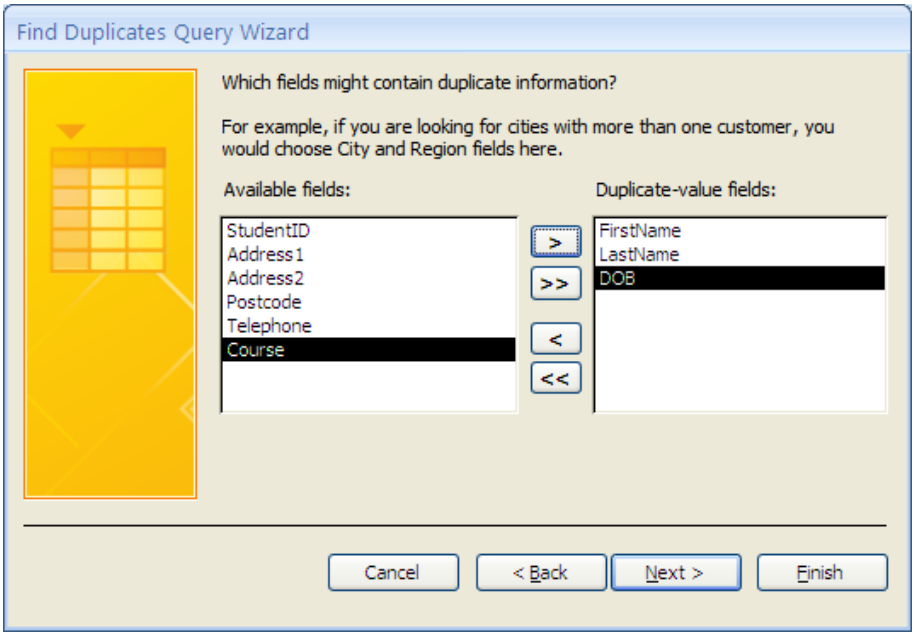
You will use the Find Duplicates Query Wizard to check for duplicate records in the students table. You will consider a record to be a duplicate if it has the same first name, last name, and date of birth as another record.

- Click **Query Wizard** from the **Other** group on the **Create** tab
- Select **Find Duplicates Query Wizard**

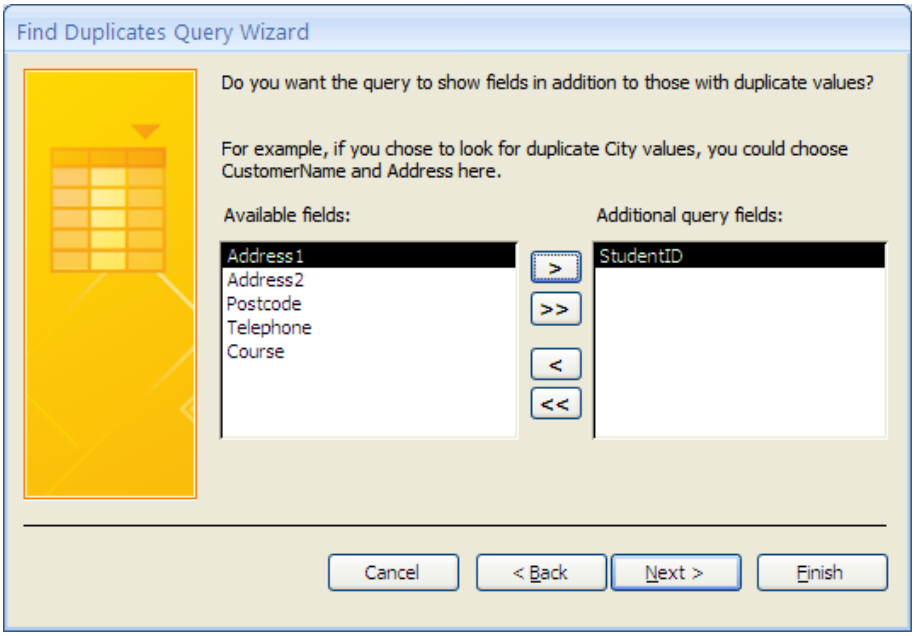


- Click **OK**
- Select the **students** table and click **Next**
- Choose **FirstName**, **LastName**, and **DOB** as the **Duplicate-value fields**

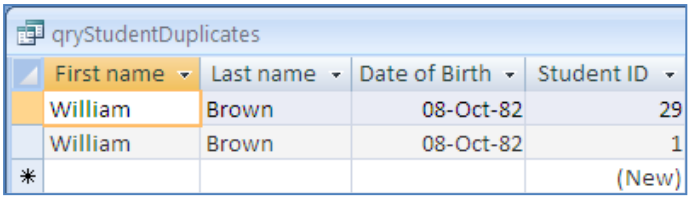
This defines a duplicate record to be a record where the values in these three fields are the same as another record.



- Click **Next**
- Select **StudentID** as an **additional query field**



- Click **Next**
- Name the query **qryStudentDuplicates**
- Click **Finish**



The results show that there are two records with the same first name, last name and date of birth.

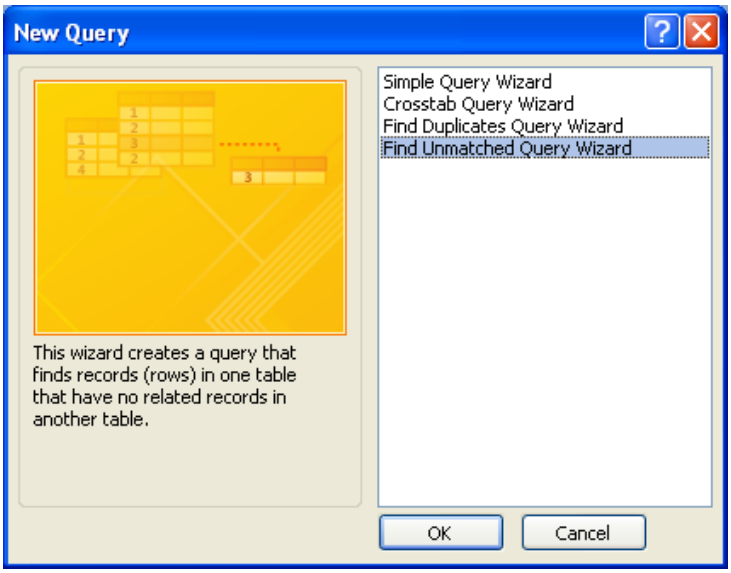
This is a select query. It therefore does not alter the table. If you wish to investigate further or delete one of the records, then this will need to be done as a separate action.

- Close the query
- Open the **students** table
- Delete the record with **Student ID** equal to **29**
- Close the table

Finding unmatched records

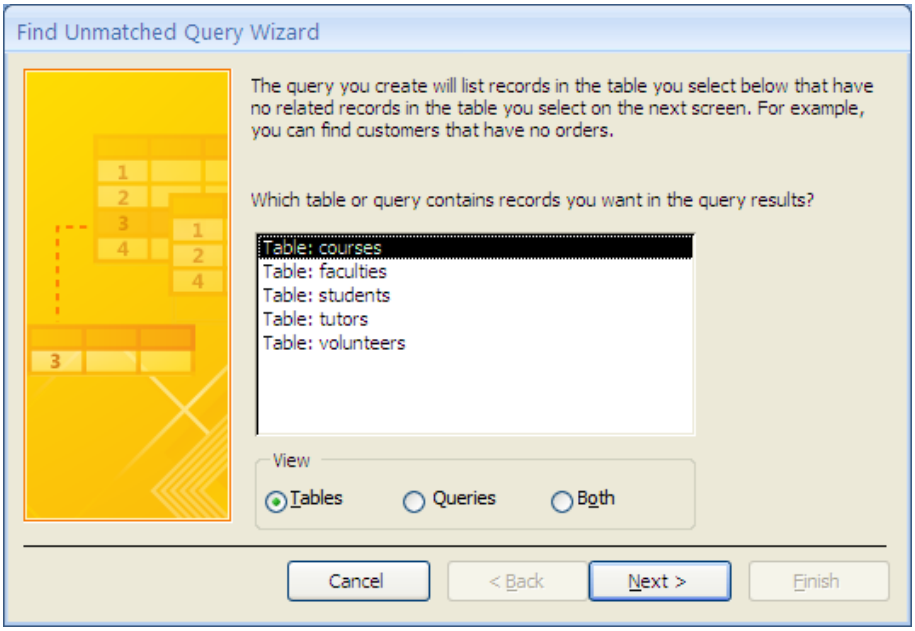
You will use the Find Unmatched Query Wizard to find out if there are any courses that currently have no student bookings.

- Click **Query Wizard** from the **Other** group on the **Create** tab
- Select **Find Unmatched Query Wizard**



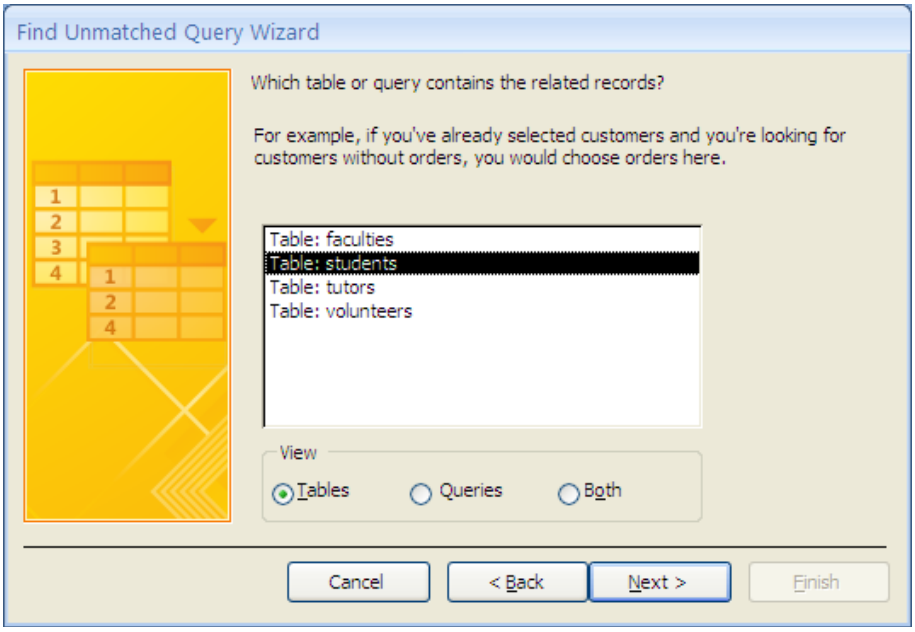
- Click **OK**

- Select the **courses** table as the table containing the records that you want to be displayed in the query results



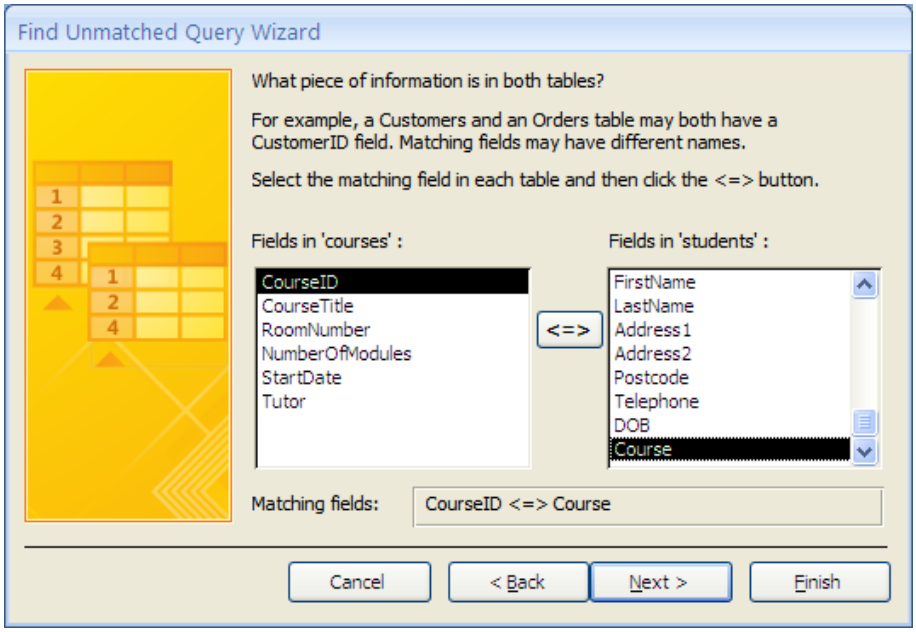
- Click **Next**

- Select the **students** table as the table containing the related records

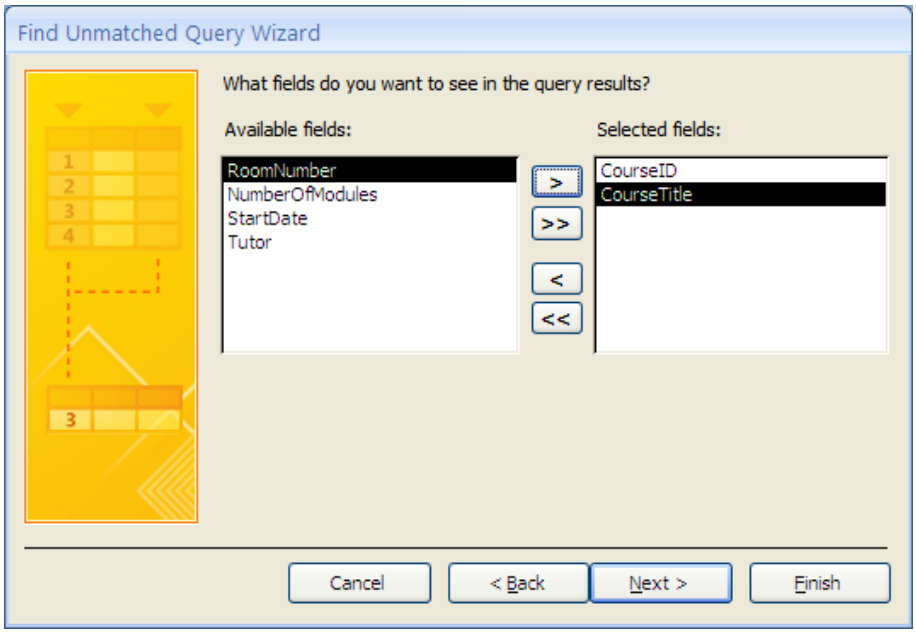


- Click **Next**

A relationship already exists between the two tables, so this is reflected in the wizard. If no relationship existed, you would need to follow the instructions to create a relationship between the tables.



- Click **Next**
- Select **CourseID** and **CourseTitle** as the fields to be displayed in the results



- Click **Next**
- Name the query **qryNoBookings**
- Click **Finish**

The results indicate that no students are enrolled on the Trade Unionism & Its History course.

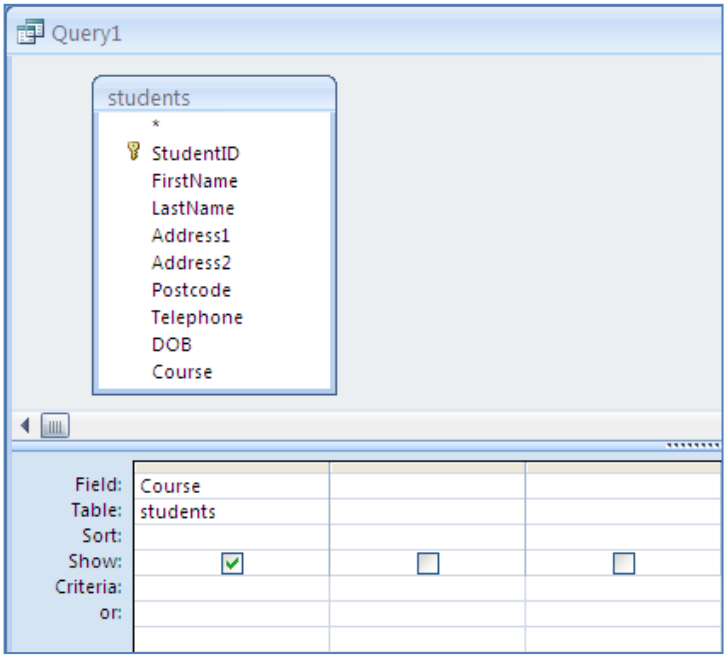
- Close the query

Grouping

Queries can be used to group data from tables and calculate summary information.

As an example, you will create a query to produce a count of how many students are enrolled on each of the available courses.

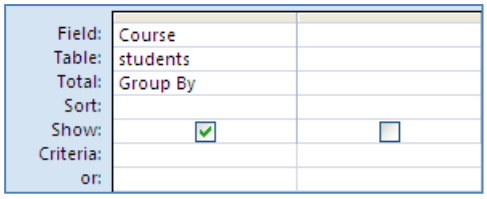
- Create a Select query in Design View projecting only the **Course** field from the **students** table on to the QBE grid



To create a grouped display from the query:

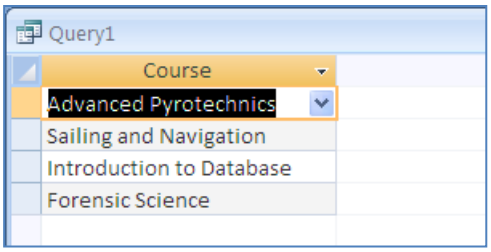
- Click **Totals** from the **Show/Hide** group on the **Design** tab

A **Total** row is added to the QBE grid, with the default entry **Group By**.



- Run the query

Each course is listed just once.

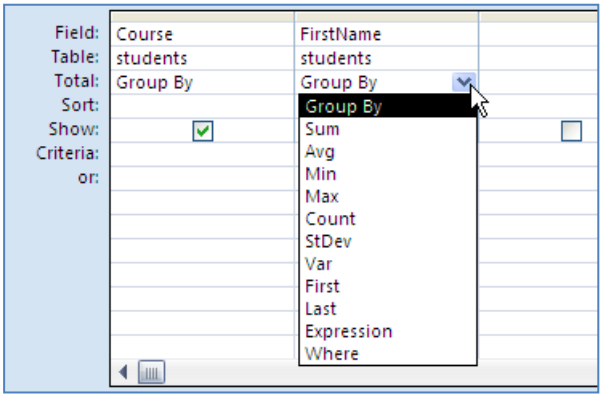


This idea is sometimes used to display only unique values in a field. To be used, for example, when creating combo boxes or list boxes.

- Switch to Design View

To count the students on each course you need to add a second field to the grid. It doesn't matter which field.

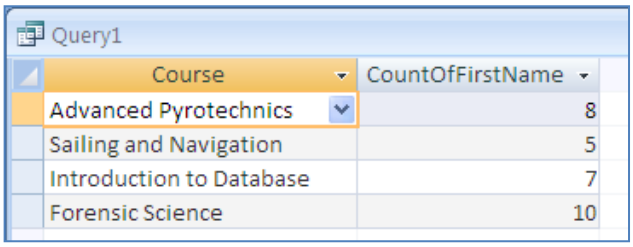
- Add the **FirstName** field to the QBE grid
- Click into the **Total** row for the **FirstName** field
- Click the down arrow to see the options available



- Click **Count**

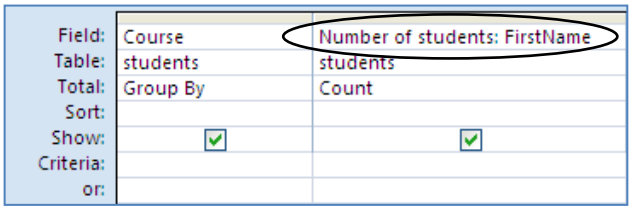
This will produce a count of first names for each course group.

- Run the query

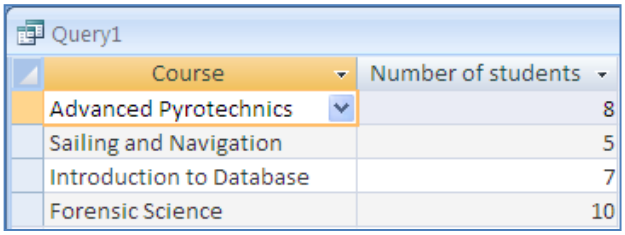


To customise the name of the column:

- Switch to Design View
- Click into the **Field** row for the **FirstName** field in the QBE grid
- Type **Number of students:** to the left of **FirstName**



- Run the query



- Save the query as **qryStudentNumbers**
- Close the query

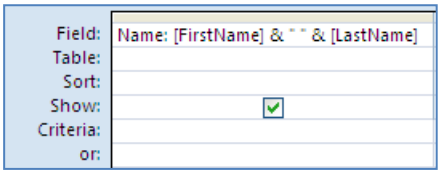
Calculated fields

Earlier in the course you used a calculated control to combine first name and last name values. Performing calculations on fields and combining fields can be achieved within queries by creating new fields called Calculated Fields.

- Create a new query in Design View and add the **students** table to the design

Instead of projecting the FirstName and LastName fields, you will create a calculated field called Name, which contains both the FirstName and LastName values with a space between them.

- In the first column of the QBE grid, for the **Field** type:
  Name: [FirstName] & " " & [LastName]



- Run the query

When you base reports and forms on queries with calculated fields, you can use these fields in the same way as the fields defined in the tables.

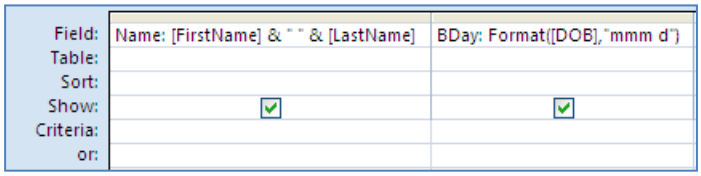
If you need to use a particular calculated control in a number of reports, it will usually be quicker to create a calculated field in a query and base the reports on the query.

- Switch to Design View

Formatting a field

It is possible to format a field within a query. This enables you, for any given field, to use different formats within the database.

- In the second column of the QBE grid, for the **Field** type:
  BDay: Format([DOB], "mmm d")



This takes the value from the DOB field and displays it using the format specified by the code *mmm d*.

- Run the query

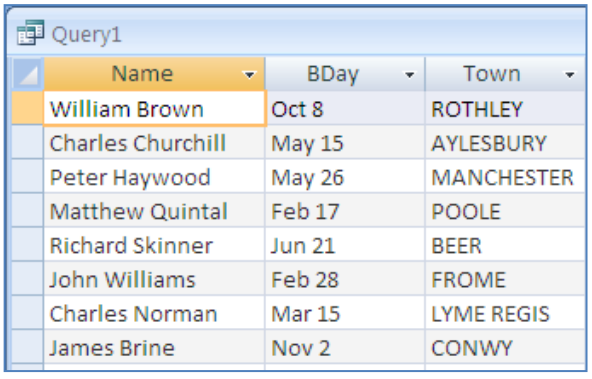
Date format codes include:

Code	*Description*	Examples		
		Format	*Value*	*Display*
m	Displays the month as ##	**m**	25/12/2008 7/4/1998	12 4
mm	Displays the month as 00	**mm**	7/4/1998	04
mmm	Displays the month abbreviated to three letters	**mmm**	25/12/2008	Dec
mmmm	Displays the name of the month in full	**mmmm**	7/4/1998	April
d	Displays the day as ##	**d**	25/12/2008 7/4/1998	25 7
dd	Displays the day as 00	**dd**	7/4/1998	07
ddd	Displays the name of the day abbreviated to three letters	**ddd**	25/12/2008	Thu
dddd	Displays the name of the day in full	**dddd**	7/4/1998	Tuesday
yy	Displays the year as 00	**dd/mm/yy**	7/4/1998	07/04/98
yyyy	Displays the year as 0000	**mmm-yyyy**	25/12/2008	Dec-2008

- Switch to Design View
- In the next blank column of the QBE grid, for the **Field** type:
  Town: Format([address2], ">")

This formats the text as all uppercase (see page 176).

- Run the query

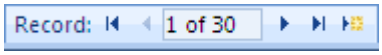


- Save the query as **qryFormatExamples**
- Close the query

Null values

- Create a query in Design View, projecting the **FirstName**, **LastName** and **Postcode** fields from the **students** table on to the QBE grid
- Run the query

Observe that there are 30 records in total.



- Switch to Design View
- Enter the Criteria **LE*** in the **Postcode** column
- Run the query

Observe that 5 records starting with LE were found.

- Switch to Design View
- Change the **Postcode** Criteria to **Not Like "LE*"**
- Run the query

The query has found 22 records. This may be surprising since you found that 5 out of a total of 30 records have postcodes starting with LE, implying that 25 of the records do not have postcodes starting with LE.

The reason why only 22 records were found is that **Not Like "LE*"** finds all the records with a value in the Postcode field that do not start with LE. It does not look for records that do not have a value in the field.

Access uses the term **Null** for empty. To find all the records where the postcode field does not start with LE, including null values:

- Change the **Postcode** Criteria to **Is Null or Not Like "LE*"**
- Run the query

The query has now found 25 records, including 3 with no postcode values.

- Save the query as **qryNullValues**
- Close the query

FORMS

Much of the terminology and techniques used in creating and editing Forms is the same as that used for Reports. This section assumes that you have worked through the Reports section and are therefore acquainted with the basic techniques involved in working with labels and text boxes.

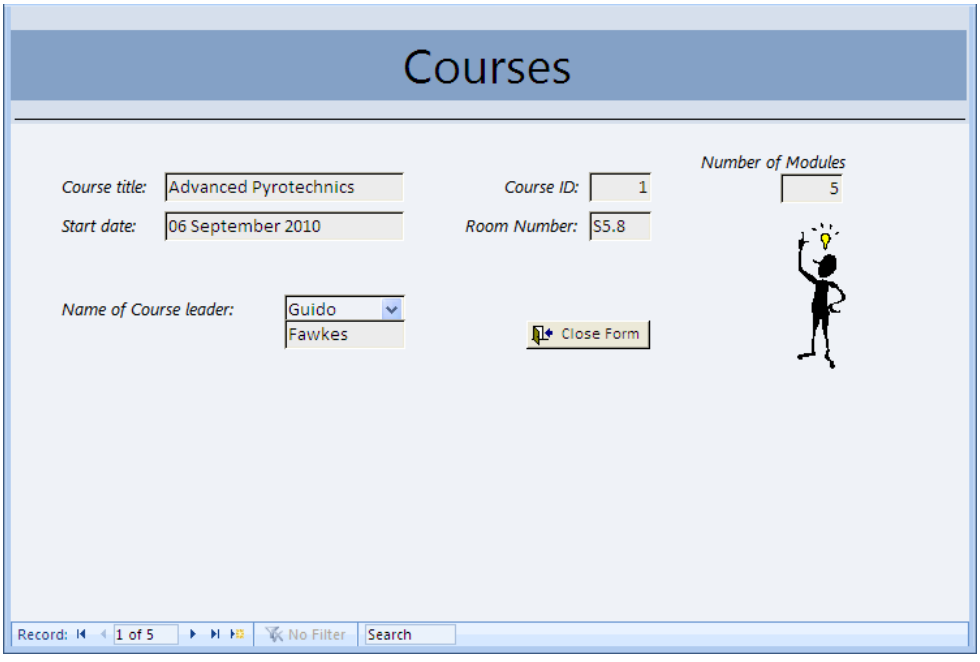
While data may be entered directly into tables, users will find it much easier to use a Form for entering and editing data. Forms make your database more user-friendly.

Like reports, standard forms based on a table or query can be created easily and automatically, and edited if necessary. However, if you will need to make lots of changes to an automatically created form, you may find it more useful to start with a blank design and create the form to your desired specification.

This section takes you through the process of creating forms without using the wizards.

CREATING A FORM USING DESIGN VIEW

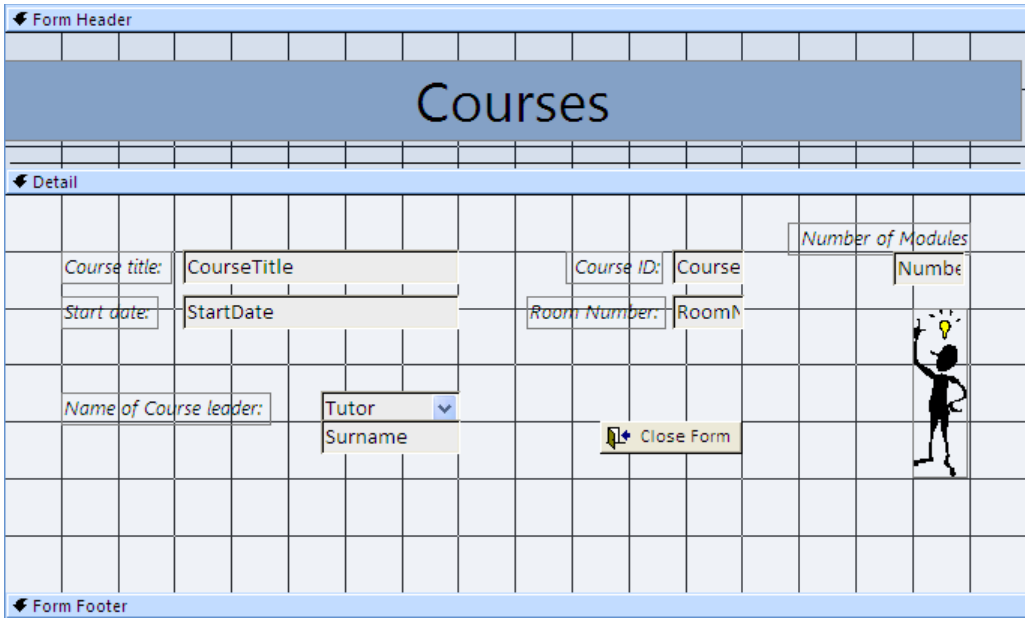
You will be creating the form below by starting with a blank form and inserting each control individually on to the form.



Apart from viewing and editing existing data, the form is intended to allow the user to add new courses and enter course leaders for the courses. The course leaders' names are stored in the tutors table; therefore a Combo Box will be

employed to allow users to quickly select the course leader from the list of tutors, instead of having to type in the name each time.

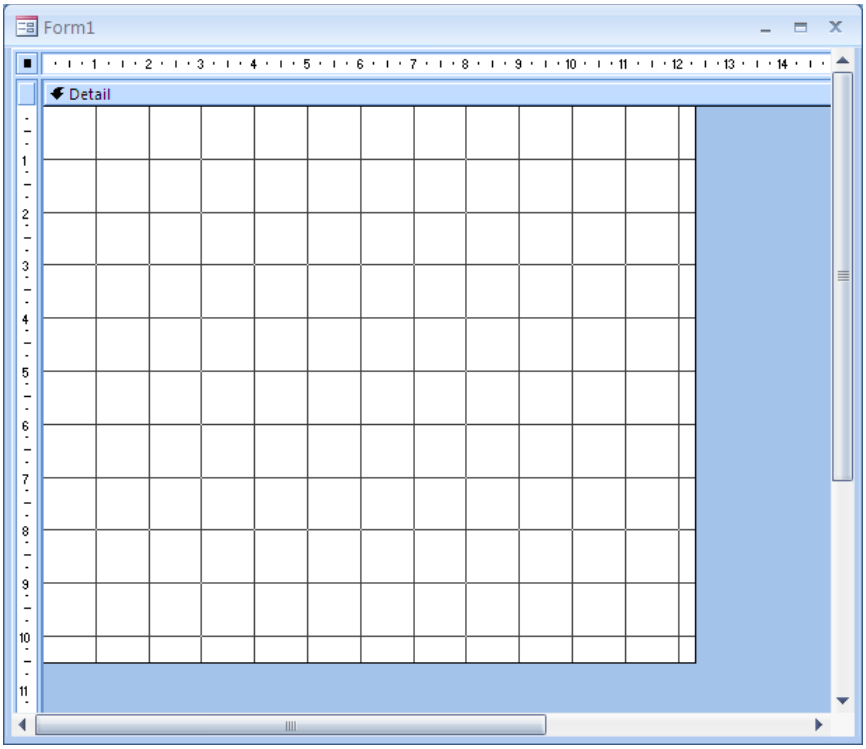
In Design View the form looks like this:



To create the form:

- Click **Form Design** from the **Forms** group on the **Create** tab

A blank form showing the Detail section is displayed.



- Minimise the Navigation pane
- If necessary, maximise the Form window
- Adjust the width of the form to between 18 and 20 cm (I used 18)

The height of the form will be adjusted later.

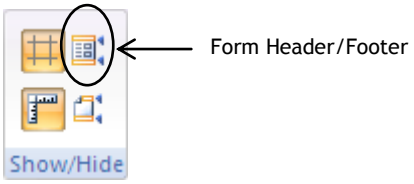
It is possible to add Page Header and Footer sections and Form Header and Footer sections to the form.

Page Headers and Footers are not normally used with forms. Forms are usually displayed in Form View but Page Headers and Footers are not displayed in this View. Designers only include these sections if the form is to be printed, or if a control needs to be positioned where it cannot be seen.

The Form Header section is often used to add a title to the form, or a logo, or navigation buttons, or simply for design effect since this section can be coloured differently from the Detail section.

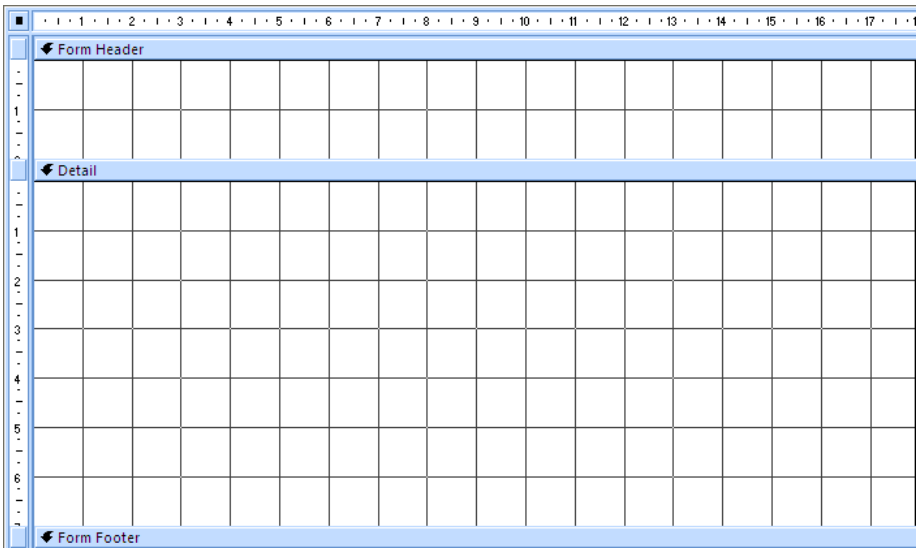
To add a Form Header:

- Click the **Form Header/Footer** button from the **Show/Hide** group on the **Arrange** tab

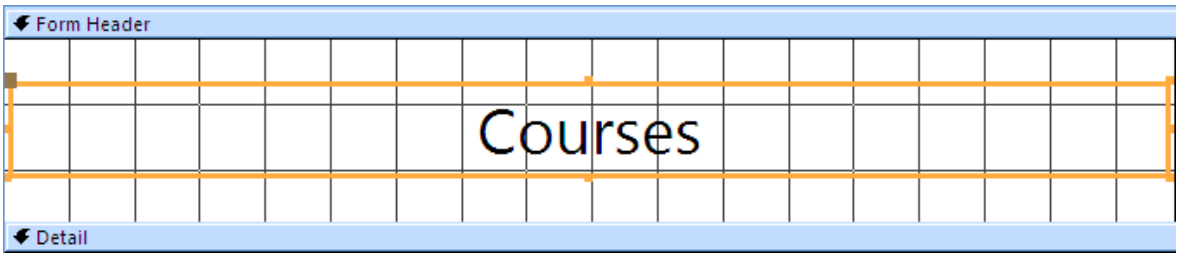


You will not need the Form Footer.

- Drag the bottom edge of the grid to meet the Form Footer bar



- Add a label to the Form Header section
- Use the label to add the title **Courses**
- Make the label as wide as the grid
- Amend the properties of the label so that the text is:
    - **Segoe UI**
    - **28 point**
    - **Centred**
- Change the height of the label if necessary to ensure that the title is fully displayed
- Modify the depth of the Header if necessary so that the label is centred vertically



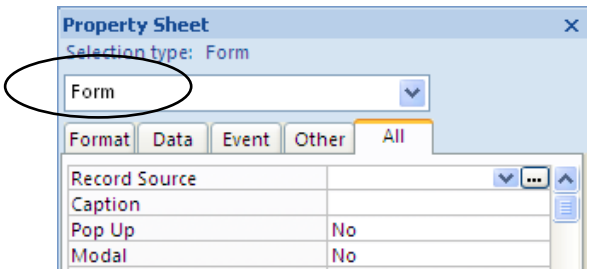
- Save the form as **frmCourses**

Adding fields

First you need to indicate where Access should look to find the data to be displayed on the form.

Earlier you used the Report Property Sheet to set the Record Source property for a report. The method is exactly the same for forms.

- Click **Property Sheet** from the **Tools** group on the **Design** tab
- Ensure that the properties for the **Form** are displayed



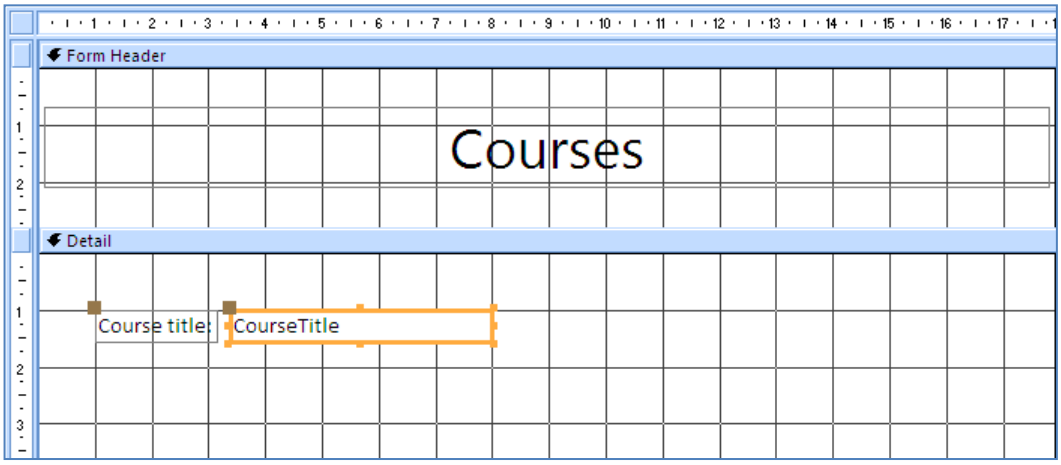
To set the courses table as the source for the form's data:

- Click the down arrow in the **Record Source** property
- Select **courses**

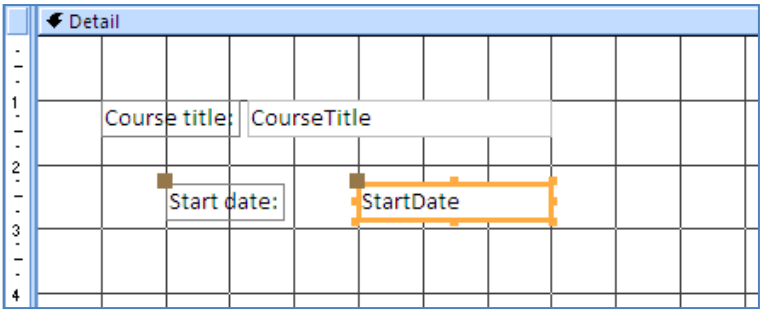
The procedure for adding and positioning fields on to the form is also the same as for reports.

To add controls bound to fields in the courses table:

- Click **Add Existing Fields** from the **Tools** group on the **Design** tab
- Drag the **CourseTitle** field on to the **Detail** section
- Resize and position it as shown below (see page 59 to review the method)



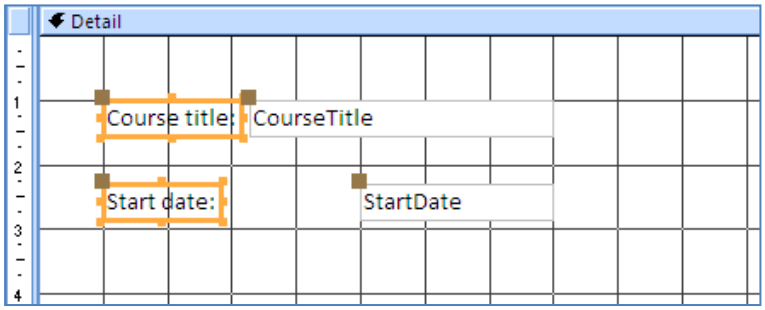
- Drag the **StartDate** field on to the **Detail** section



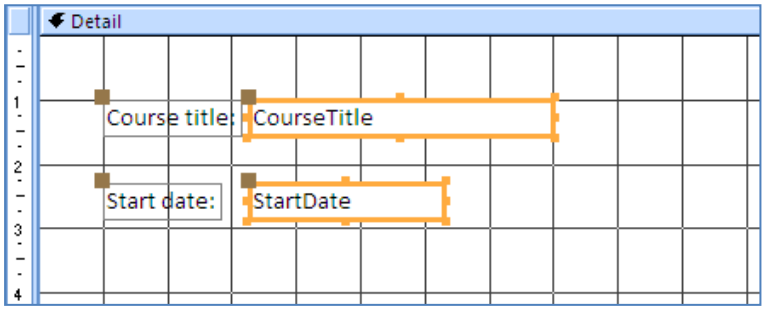
To make the design look as professional as possible, you want the two labels to be positioned the same distance from the left edge of the form.

- Select the **Course title** label, hold down the Shift key and select the **Start date** label

- With both labels selected, click the **Left** button from the **Control Alignment** group on the **Arrange** tab



- Do the same with the **CourseTitle** and **StartDate** text boxes so that their left edges are aligned

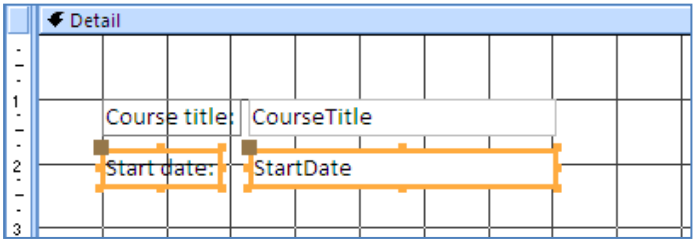


The next task is to make both text boxes the same width.

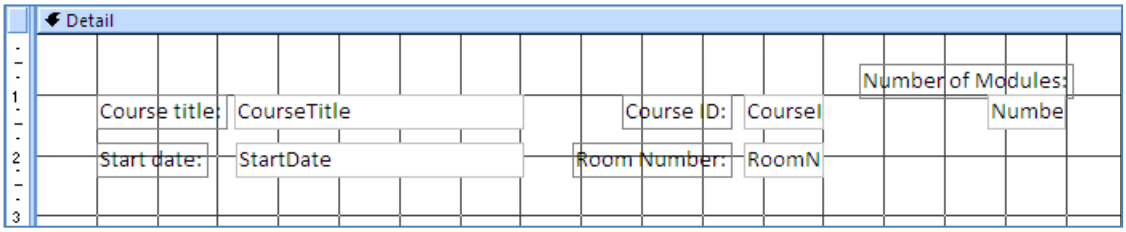
- Display the Property Sheet
- With both text boxes still selected, type **4.75cm** for the **Width** property
- Press Enter

The arrow keys are useful when you need to move controls short distances. You may need to move the Start date label and text box vertically so that they are closer to the Course title. To do this:

- Select the **Start date** label and the **StartDate** text box
- Tap the Up Arrow key repeatedly until the controls are positioned as below

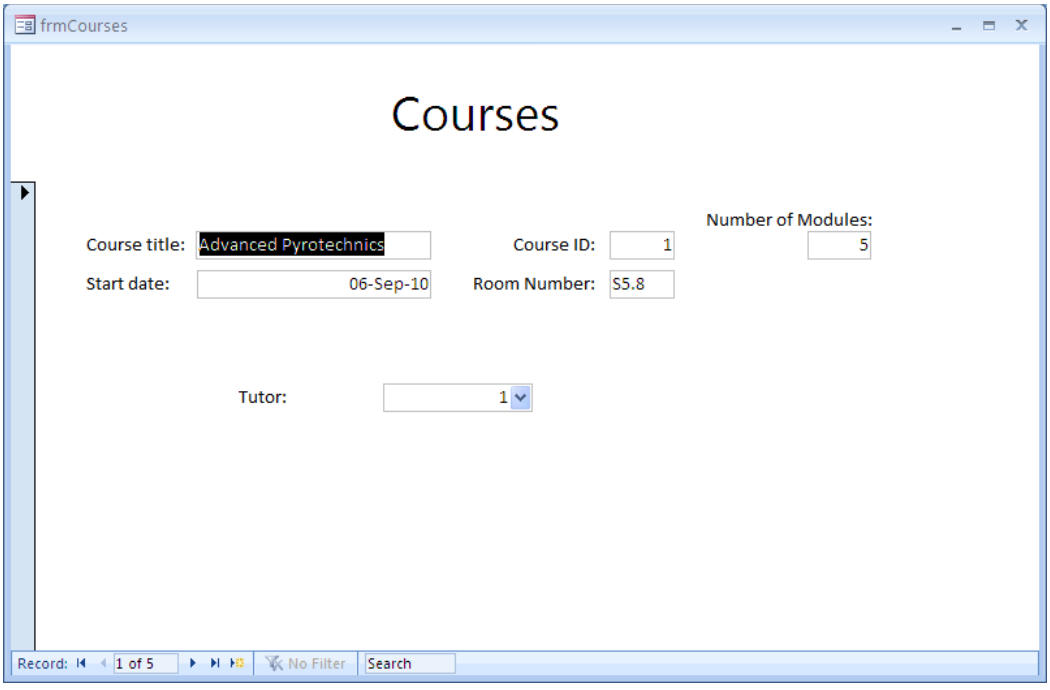


- Add the **CourseID**, **RoomNumber** and **NumberOfModules** fields to the form, resizing and repositioning the controls as shown below



- Drag the **Tutor** field on to the form
- Display the form in Form View

The lookup wizard was used to link the Tutor field in the courses table to the tutors table, so a combo box control is added when the field is dragged on to a form.



It would be more useful, however, to display the tutor's name instead of the tutor's ID number.

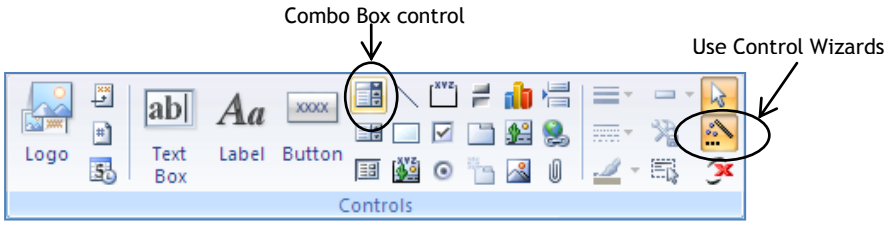
- Display the form in Design View
- Delete the **Tutor** controls from the form

Adding a combo box

You will create a combo box control bound to the Tutor field, but displaying the Tutor's name instead of the ID number.

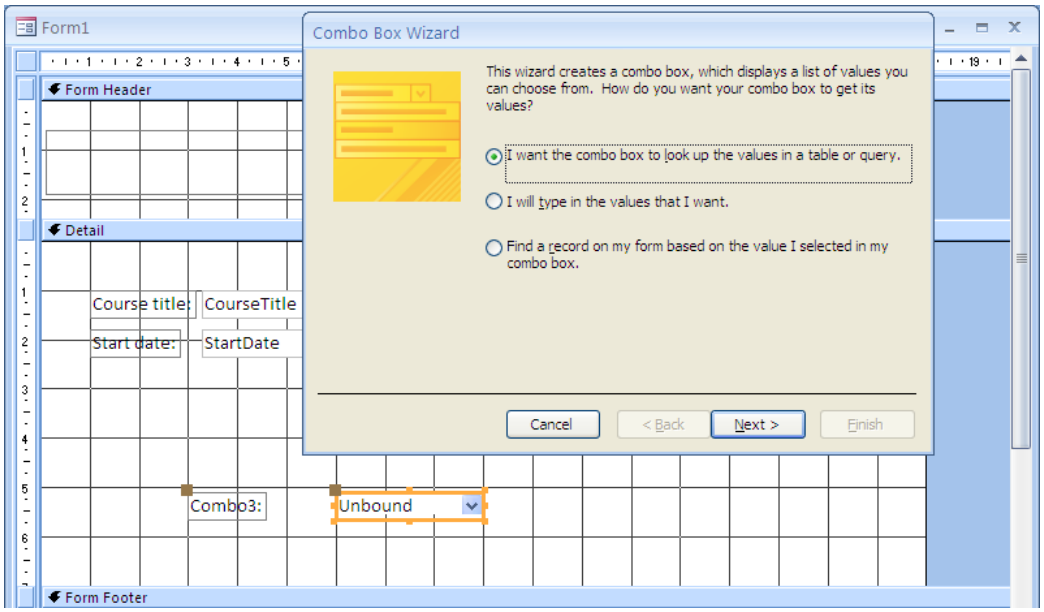
- Ensure that the **Use Control Wizards** button, in the **Controls** group on the **Design** tab, is selected/highlighted. If necessary, click the Use Control Wizards button to select it

- Click the **Combo Box control** button from the **Controls** group



- Click on a blank area within the Detail section

A Combo Box and an associated label are added to the form and the Combo Box Wizard is displayed.

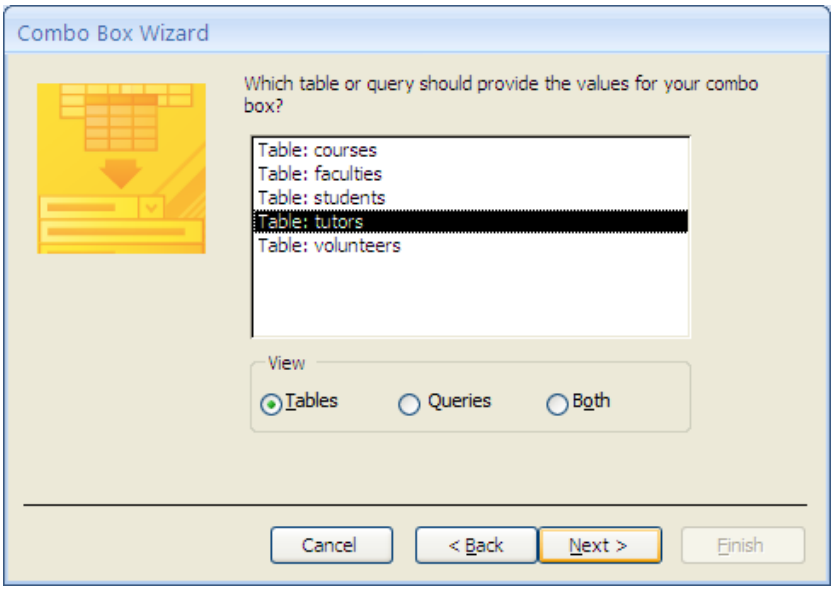


The text on the label may not be exactly the same as that shown in the diagram, but that is not important since you will be changing it in due course.

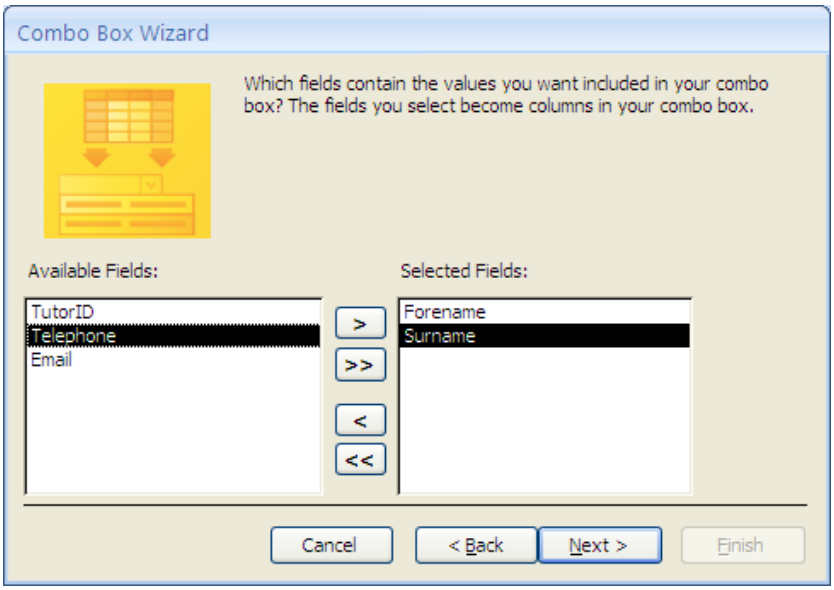
- Note the different options that you can choose from the wizard

- Ensure that **I want the combo box to look up the values in a table or query** is selected and click **Next**

- Select the **tutors** table



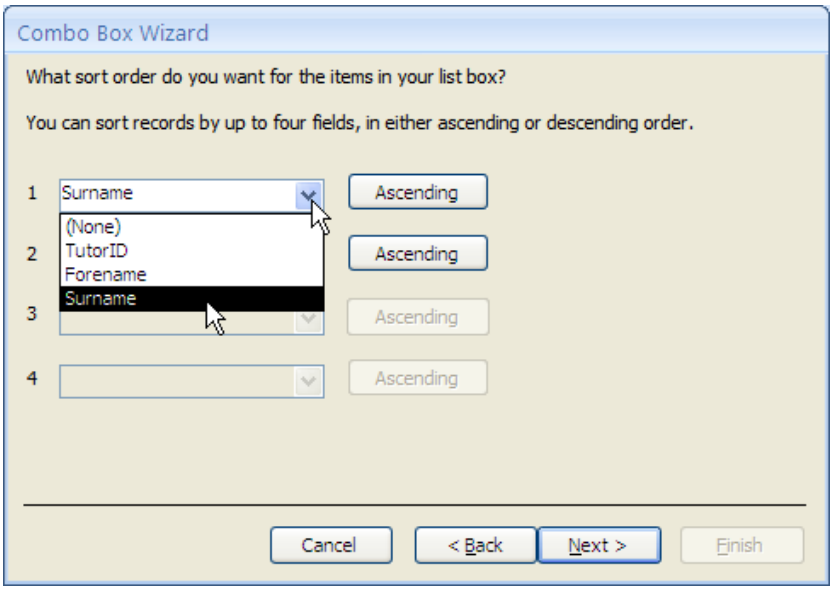
- Click **Next**
- Select **Forename** and **Surname** as the fields to be included in the combo box



- Click **Next**

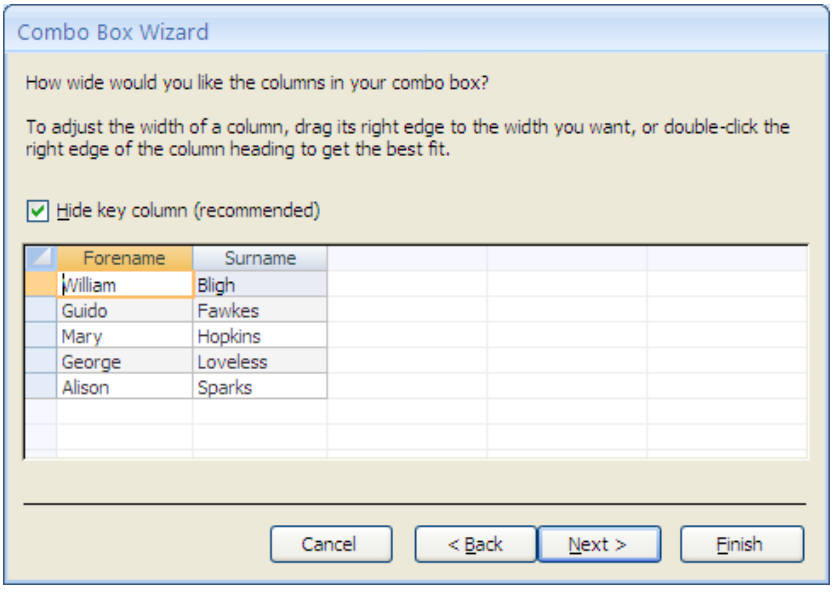
To sort the list by Surname:

- Click the down arrow and select **Surname**



- Click **Next**

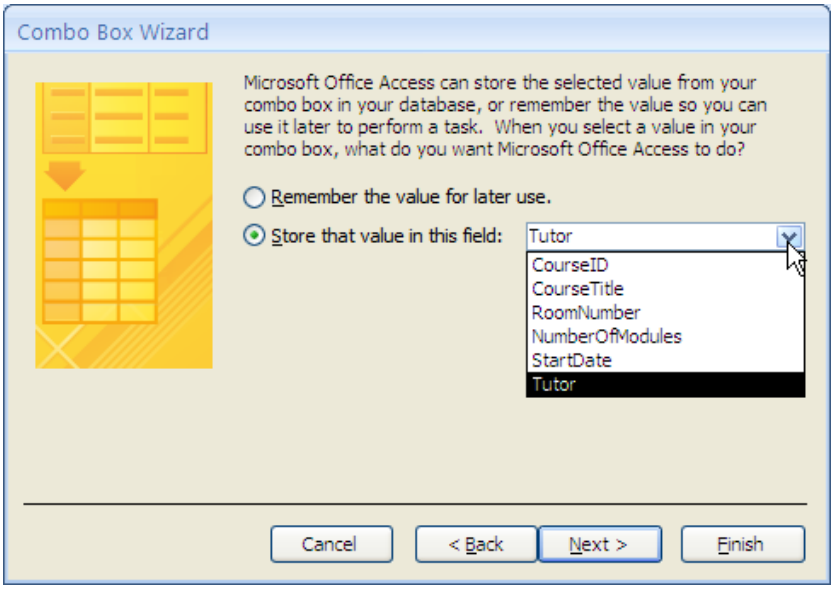
- Ensure that there is a tick in the check box to **Hide key column** and adjust the widths of the columns if necessary to ensure that the names are fully displayed



- Click **Next**

To bind the combo box value to the Tutor field:

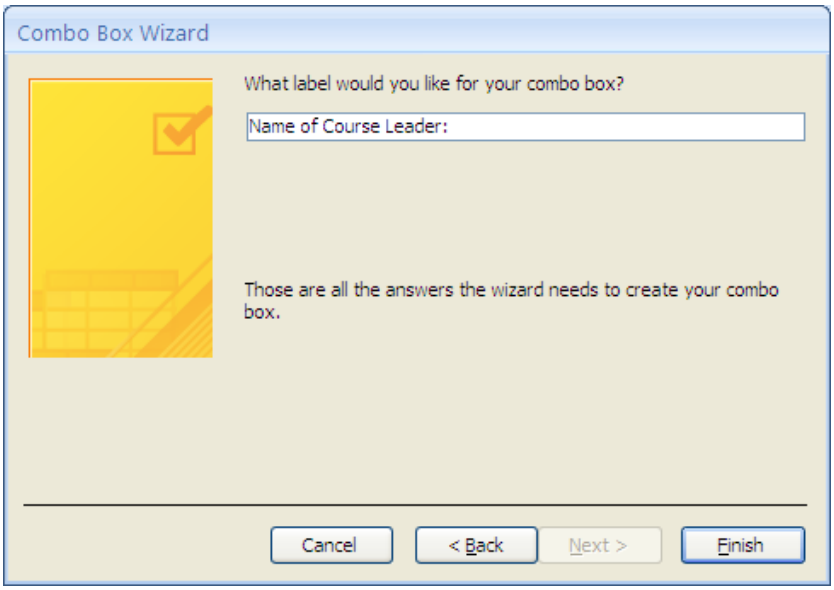
- In the **Store that value in this field** box, click the down arrow and select **Tutor**



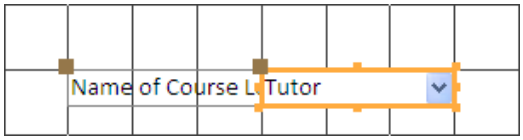
- Click **Next**

The last dialog box of the wizard gives you the option to change the text on the label.

- Change the default text to **Name of Course Leader:**

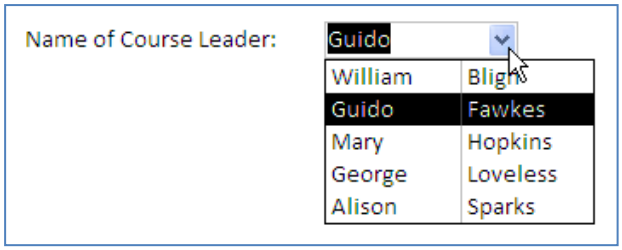


- Click **Finish**



The extra text in the label has caused it to overlap with the combo box.

- Drag the label left to ensure that it is fully displayed
- Display the form in Form View
- Click the down arrow on the combo box



- Select **Alison Sparks**

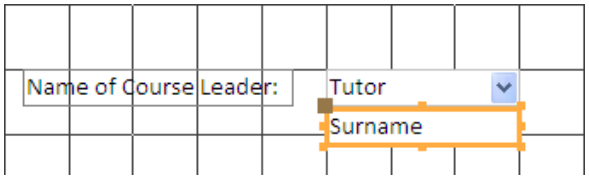
The combo box list includes the Forename and Surname values as you specified. However, only the Forename value is displayed in the combo box.

It would be useful to also display the course leader's Surname on the form.

- Display the form in Design View
- Ensure that the Field List pane is displayed (see page 59)
- If there is not a **Fields available in related tables** section, click the **Show all tables** link at the bottom of the pane
- In the **Fields available in related tables** section of the pane, click the plus sign next to the **tutors** table to display the fields

- Drag the **Surname** field on to a blank area of the form
- Select and delete the **Surname** label

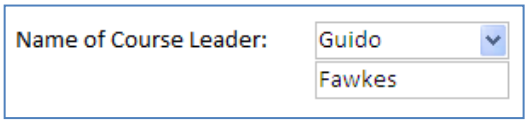
- Position the text box below the combo box as shown below (resize the text box if necessary)



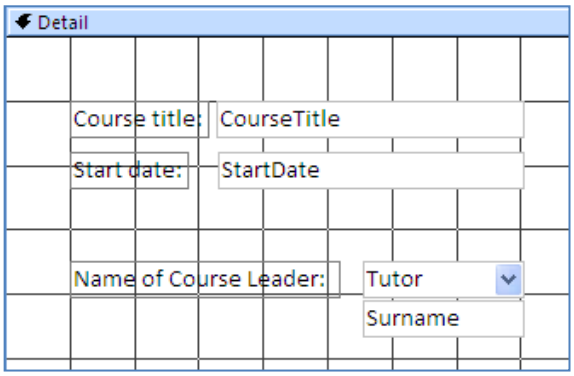
- Display the form in Form View

The course leader's forename and surname are both displayed.

- Use the combo box to select **Guido Fawkes**



- In Design View, align the Course Leader controls below the Start date controls as shown below

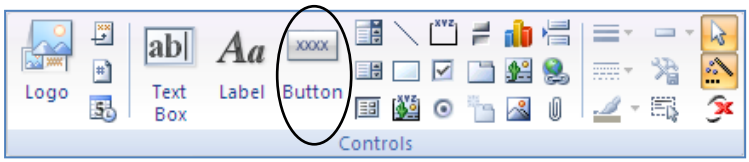


Adding a command button

Most forms would benefit from a Command Button to take the user out of the form and possibly into another form.

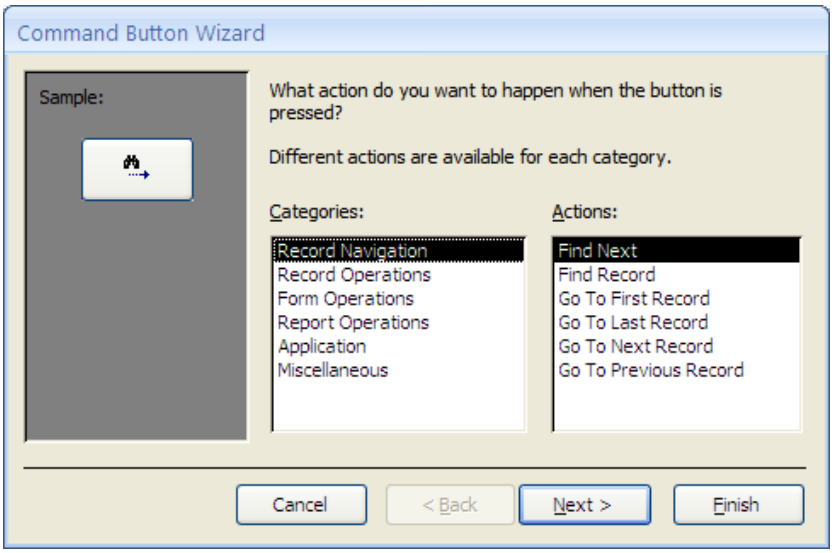
You will add a Command Button that will simply close the form.

- Ensure that the **Use Control Wizards** button is selected/highlighted in the **Controls** group
- Click the **Button** control from the **Controls** group



- Click on a blank area in the Detail section of the form

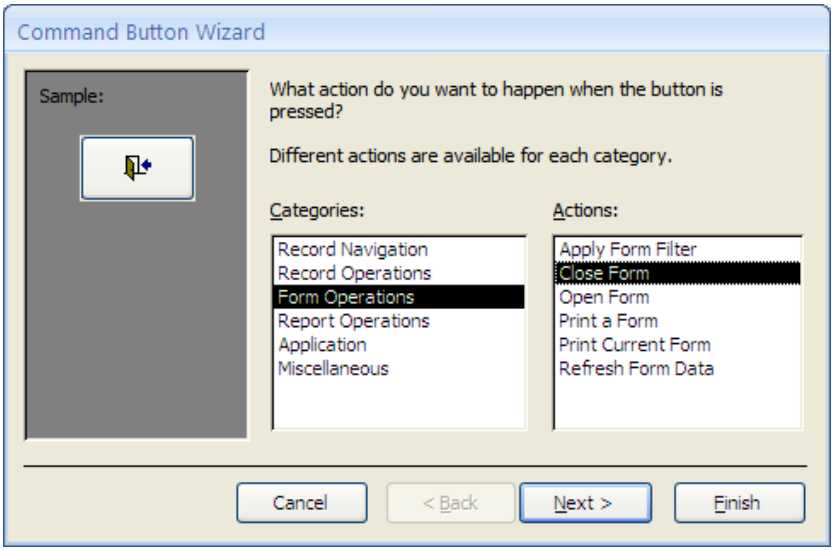
A rectangular button is added to the form and the Command Button Wizard is displayed.



- In the **Categories** section, click **Form Operations**

The Actions section displays actions associated with this category.

- Click the **Close Form** action

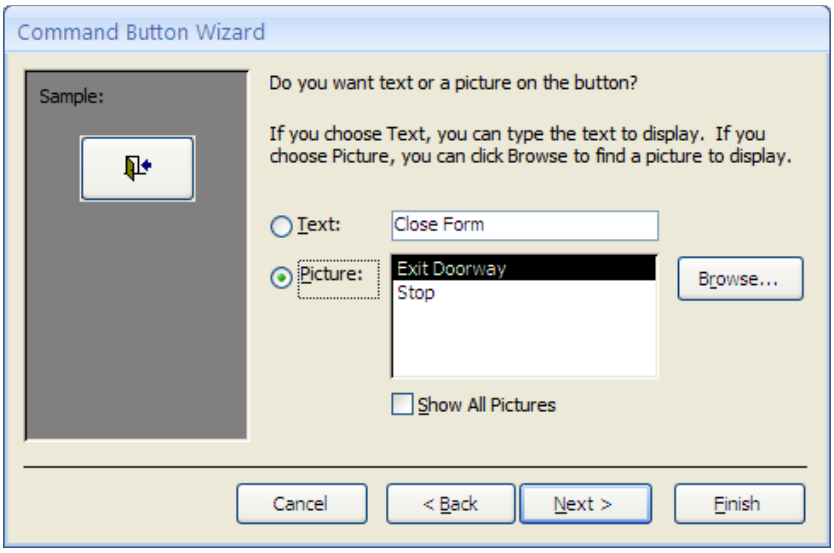


- Click **Next**

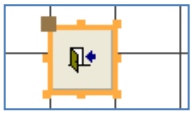
You can now decide what is displayed on the Command Button. You have the choice of two images or display text only.

For this button you will use an image and add text to it later.

- In the **Picture** section, ensure that the **Exit Doorway** option is selected



- Click **Next**
- Change the suggested name for the command button to **cmdCloseCourses**
- Click **Finish**

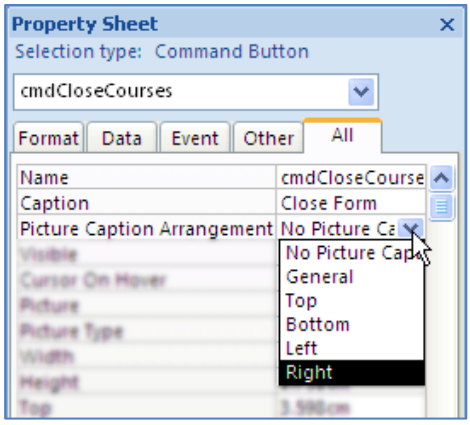


The button can be resized by dragging the resize handles, but instead you will set the size on the property sheet.

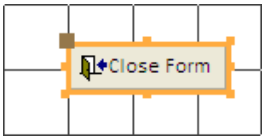
- Display the Property Sheet
- Set the **Width** to **2.5cm** and the **Height** to **0.75cm**

To display text on the button as well as the image:

- Change the **Caption** property to **Close Form**
- Click into the **Picture Caption Arrangement** property
- Click the down arrow and select **Right** to put the caption to the right of the picture



- Change the **Font Size** to **9**



- Reposition the Command Button if necessary so that the form looks similar to the image on page 94
- Save the form
- Switch to Form View and try the button

SPECIAL EFFECTS

Although the Form is functional, you are going to improve its appearance by adding colour and graphics. You will also add extra formatting to some of the controls.

Graphics

You may want to include an image in a Form or Report for varying reasons. It could be a logo that needs to be added to every Form or Report, or you may decide to insert an appropriate image to enhance the visual appeal of a Form.

- Open the form **frmCourses** in Design View

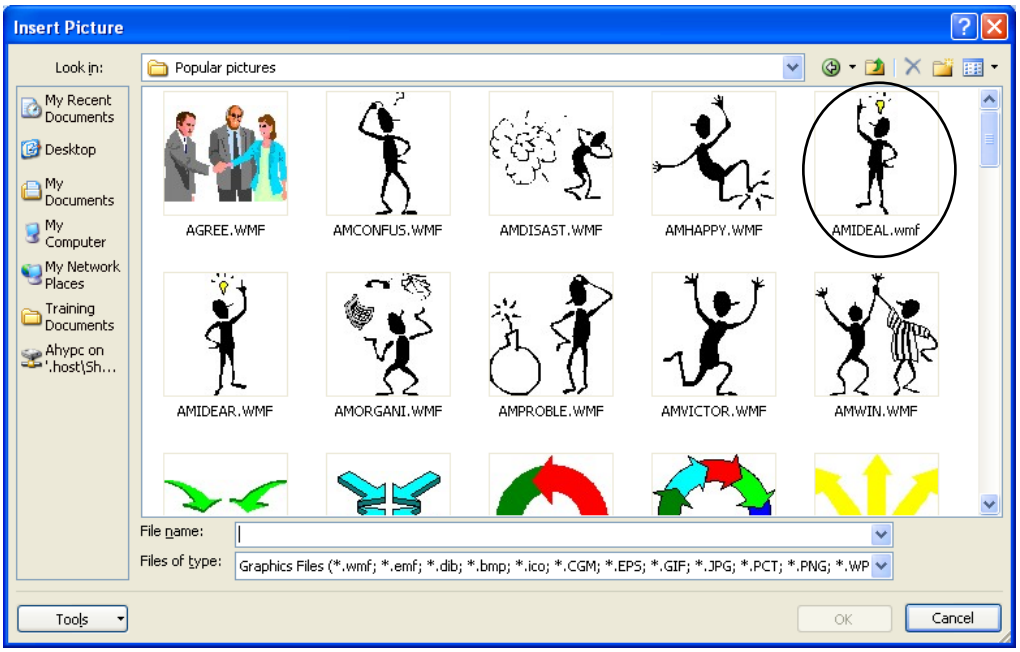
To add an image below the Number of Modules controls:

- Click the **Image** button from the **Controls** group

- Click on a blank area in the Detail section

A control is added to the form and the Insert Picture window is displayed.

- Navigate to the folder **C:\Popular pictures**



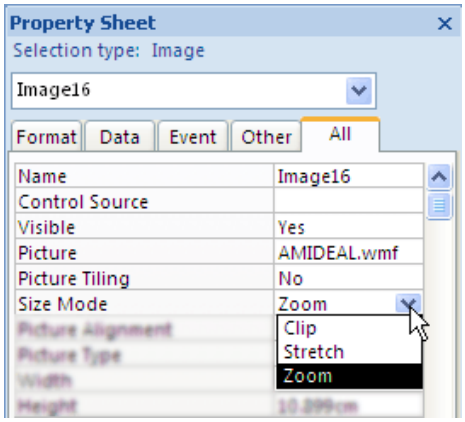
- Select the image **AMIDEAL**

- Click **OK**

The image is added to the form.

The image is large and therefore the form may have extended in width and/or height in order to accommodate the image.

- With the image selected, click in the **Size Mode** property
- Click the down arrow

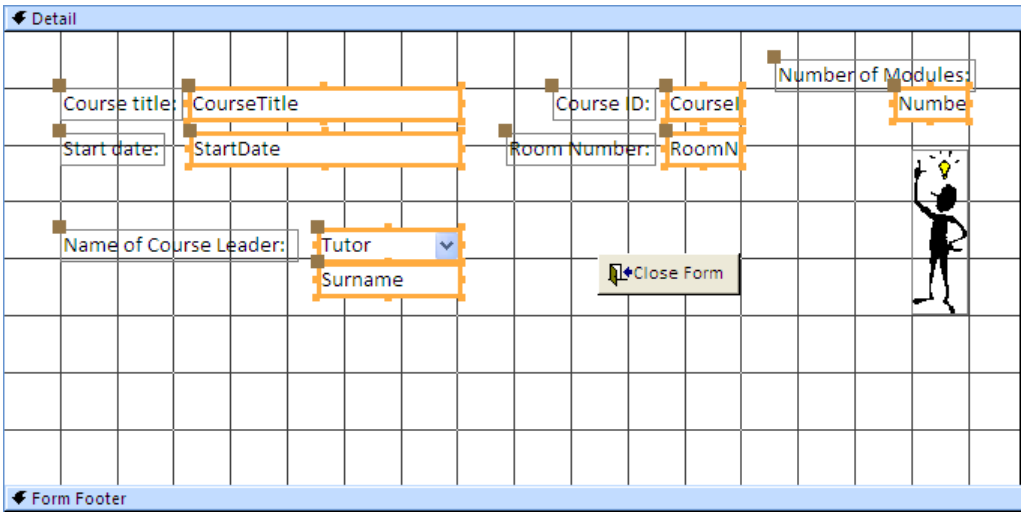


The options are Clip, Stretch, and Zoom.

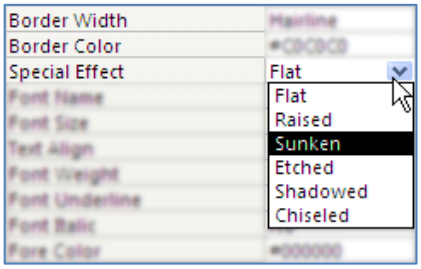
The Clip option will cause the image to be cropped as you resize the control to make it smaller. The Stretch option will stretch the image as you resize; if you resize the width, then only the width will be stretched. The Zoom option enlarges the image, retaining the original aspect ratio, as you resize the control.

- Ensure that **Zoom** is selected
- Scroll to the bottom of the image and drag one of the resize handles to make the image smaller
- Resize and reposition the image so that the form looks similar to the diagram on page 94
- If the image caused the width of the form to extend, drag the width back to the original setting (see page 95)
- Drag the Form Footer bar upward to reduce the height of the form, leaving no more than two or three blank rows below the controls
- Save the form

- Select all the text boxes and combo box



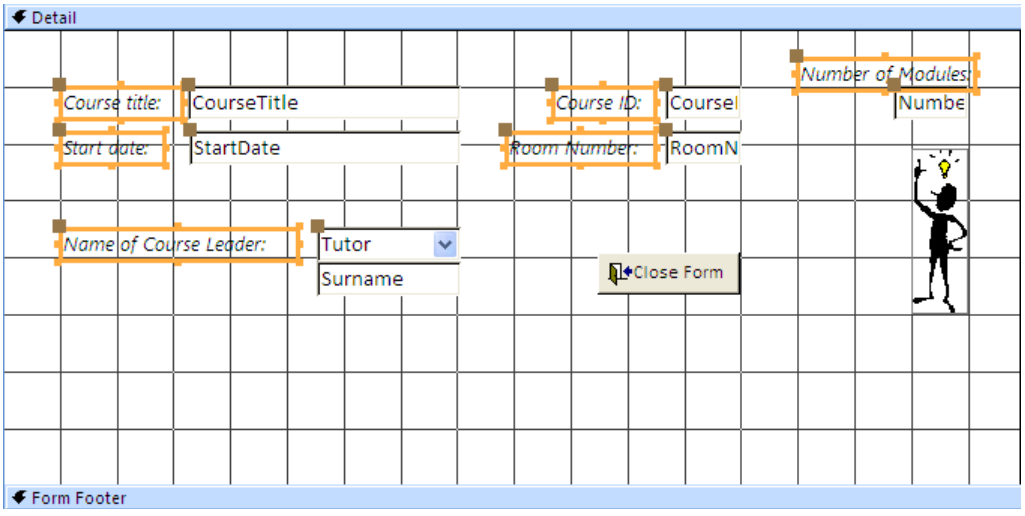
- Click in the **Special Effect** property on the Property Sheet
- Click the down arrow



- Select **Sunken**

To make the text on the labels slightly different from that in the text boxes:

- Select all the labels in the Detail section



- Set the **Font Name** property to **Segoe UI**, the **Font Size** to **10**, and **Font Italic** to **Yes**

Using colour

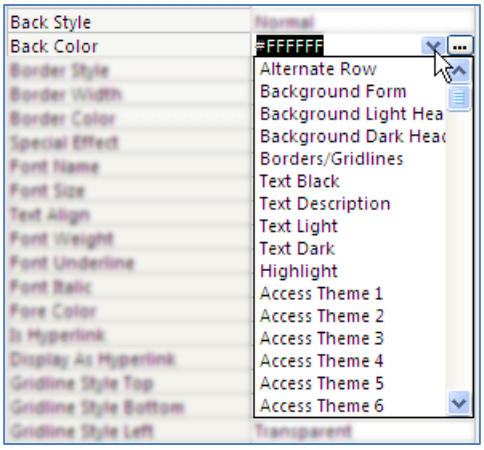
The Form comprises two sections and several controls: text boxes, labels, and a combo box. Each individual one of these components can be coloured differently.

- Select the label in the Form Header
- Click in the **Back Color** property

A down arrow and an ellipsis button are both displayed for this property.

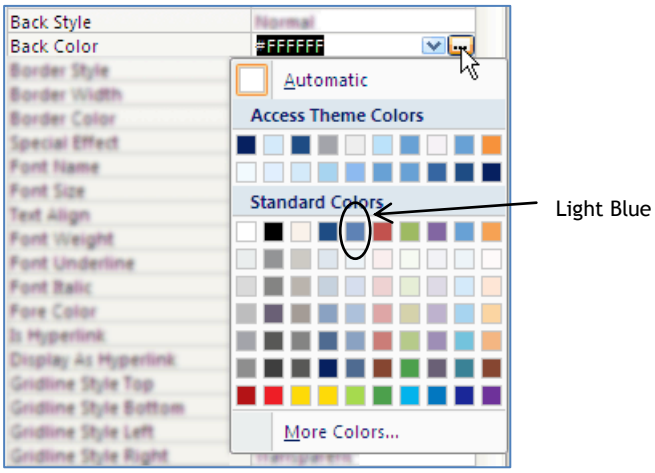
- Click the down arrow

A long list of options is displayed enabling you to choose one of the theme or system colours by name.

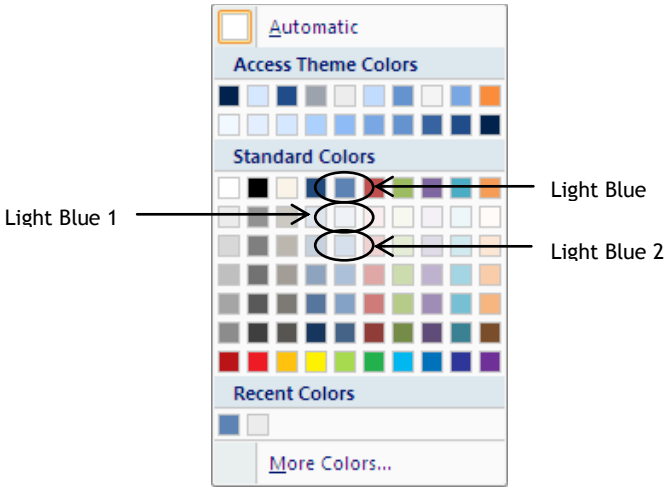


- Click the ellipsis button

A more conventional colour picker is displayed.



- Select **Light Blue** from the **Standard Colors** section
- Press Enter (or click on a different property) to see the colour change
- Click outside the label in the Form Header section
- Display the colour picker in the **Back Color** property
- Select **Light Blue 2**



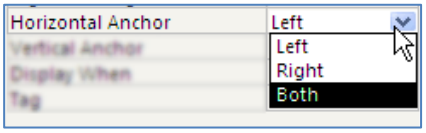
- Click on a blank area in the Detail section
- Display the colour picker in the **Back Color** property
- Select **Light Blue 1**
- Select all the text boxes and combo box in the Detail section
- Display the colour picker in the **Back Color** property
- Select the lightest grey colour from the **Standard Colors** section
- Press Enter (or click on a different property)
- In the Form Header section, draw a horizontal line at the bottom, stretching right across the form
- Save the form
- Display the form in Form View

OTHER PROPERTIES WORTH NOTING

The form is more visually appealing than before, but you may notice that the label for the title and the horizontal line do not stretch right across the form.

Horizontal anchor

- Display the form in Design View
- Select the horizontal line in the Form Header
- Click in the **Horizontal Anchor** property
- Click the down arrow



- Select **Both**

This anchors the line at both edges, ensuring that, if the form is extended in Form View, the line will also extend.

- Set the Horizontal Anchor for the label in the Form Header so that it is also anchored at both ends

Enabled and Locked

These two properties work together to secure an individual control. By default each control is Enabled and Not Locked.

- Select the **CourseID** text box

You do not want users to attempt to change the ID number. To ensure that users will not be able to change this value on the form, you can either Disable the control, or Lock it, or both Disable and Lock it.

When a control is disabled, it cannot be given the focus. In other words, users are not able to click into it. It is also greyed out, unless it is also locked.

When a control is locked, users cannot make changes to the data even if they are able to click into the control.

The following table lists the effects of the different combinations of settings:

Enabled	*Locked*	*Can select?*	*Can edit value?*	*Control greyed out?*
Yes	No	Yes	Yes	No
Yes	Yes	Yes	No	No
No	No	No	No	Yes
No	Yes	No	No	No

- By selecting from the drop down list, change the **Enabled** property to **No** and the **Locked** property to **Yes** (Hint: You may find it easier to locate these properties from the Data tab)

The Course Leader's surname is another value that you do not want users to edit or select, since this control is simply reflecting what was selected from the combo box.

- Click the Surname text box
- Change the **Enabled** property to **No** and the **Locked** property to **Yes**

Limit To List

This property and the one that follows are available for combo boxes and can both be found on the Data tab.

Limit To List enables you to decide whether or not users can type a value into the control if it is not in the list. On this form the course leaders are to be limited to the names that exist in the tutors table.

- Click into the Combo Box control
- Check that the **Limit To List** property is set to **Yes**

Auto Expand

This property will help users who decide to type into the combo box.

When Auto Expand is set to Yes, if the user starts typing into the control, Access displays the first value from the available list that begins with the characters already typed.

- Check that the **Auto Expand** property is set to **Yes**

Tab index

When all the controls have been placed on the form, you need to check that the Tab key takes the user from one control to another in a logical and helpful order. If necessary, the Tab Index property can be changed for some controls to ensure that the order is correct.

- Display the form in Form View
- Use the Tab key to cycle through the controls

The Course ID and Course Leader's Surname are disabled and locked; therefore these fields are skipped as you cycle through the record.

To practice reordering the tab sequence, you will change the order to:

Course title
Course ID
Number of Modules
Start date
Room Number
Tutor combo box
Surname
Command button

Course ID and Surname have been included in the list because every field must have a Tab Index value even if they are to be skipped when tabbing.

- Display the form in Design View
- Ensure that the Property Sheet is displayed
- Click the **CourseTitle** text box
- Select the **Other** tab in the Property Sheet
- Note that the **Tab Index** value is **0**

This was the first field to be placed on the form, so it was automatically assigned the smallest index value.

- Click the **CourseID** text box
- Change the **Tab Index** to **1**

The *Tab Stop* property enables you to decide if an individual control should be skipped when the user Tabs through the controls on the form.

- Since users are not allowed to change the CourseID, change the **Tab Stop** property of the **CourseID** text box to **No** (this is considered to be good practice, even though it is not entirely necessary since the Enabled property is set to No and this also ensures that the control is skipped)

- Change the **Tab Index** for each of the following controls:
  NumberOfModules text box to **2**
  StartDate text box to **3**
  RoomNumber text box to **4**
  Tutor combo box to **5**
  Surname text box to **6**
  Close Form command button to **7**

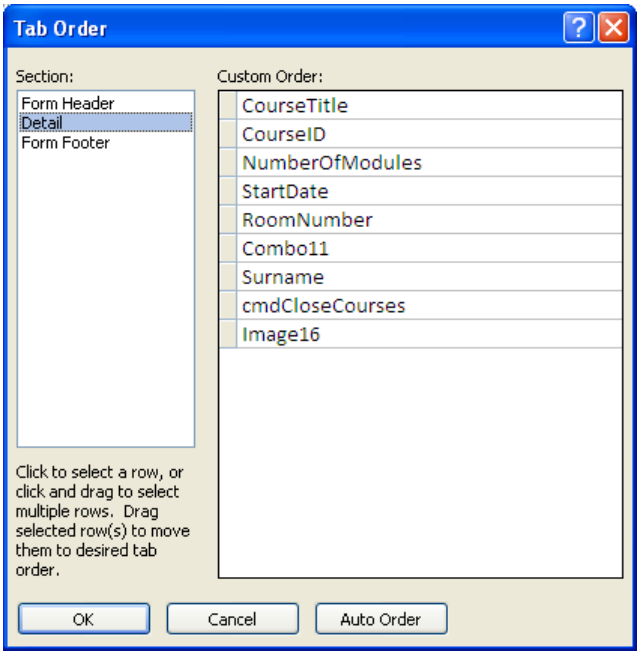
- Set the **Tab Stop** property for the **Surname** text box to **No**

- Verify in Form View that the Tab sequence is as expected, and that the Course ID and Surname boxes are skipped

Alternative method for changing the tab order

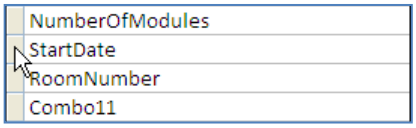
- In Design View, select the **CourseTitle** text box (any text box would be okay)

- Click **Tab Order** from the **Control Layout** group on the **Arrange** tab



To make **StartDate** second in the order:

- Click the selection box to the left of **StartDate** to select the row



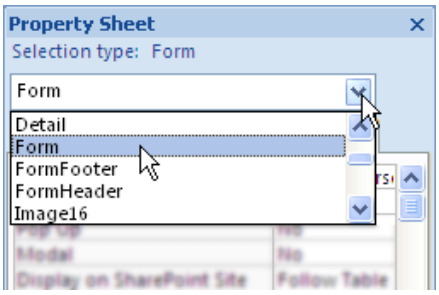
- Drag the row up above **CourseID** by dragging the selection box

- Drag **RoomNumber** above **NumberOfModules**

- Verify in Form View that the Tab sequence is as expected

The other properties discussed in this section are general Form properties.

- Display the form in Design View
- Display the properties of the Form



Caption

When a Caption property is set for the form, this is displayed as a title on the Form Title Bar when in Form View; otherwise the form name is displayed on the Title bar.

The name frmCourses is not a useful title for the form.

- Select the **All** tab on the Property Sheet
- Click in the **Caption** property and type **2010 Courses**

Pop Up and Modal

These two properties are often set in conjunction.

When the Pop Up property is set to Yes, the form opens in its own window above all other open objects. Be aware that setting to Yes also disables the Views group on the ribbon when the form is displayed in Form View, making it more difficult to change to Design View.

When the Modal property is set to Yes, the form opens in a Modal window, meaning that the user must close the form before continuing with another task on the database.

This particular form does not need to be modal.

- Ensure that both the **Pop Up** and **Modal** properties are set to **No**

Restricting available views

It is advisable to decide what view or views your form is designed to be displayed in, and disable all other views.

- Restrict the user to only Form View and Layout View by setting **Allow Datasheet View**, **Allow PivotTable View** and **Allow PivotChart View** to **No**

Allow Form View	Yes
Allow Datasheet View	No
Allow PivotTable View	No
Allow PivotChart View	No
Allow Layout View	Yes

Record Selectors and Scroll Bars

The Record Selector allows the user to select and delete or print an individual record. It is a good idea to remove the Record Selector whenever this facility is not needed because, apart from disabling this feature, it enhances the look of the form.

- Set the **Record Selectors** property to **No**

Scroll Bars are also often disabled in forms for aesthetic reasons.

- Set the **Scroll Bars** property to **Neither**

- Ensure that the **Auto Resize** property is set to **Yes**

Auto Resize automatically adjusts the display of the form to the size that you designed it. If it is set to No, the form appears in the way the user last saved it, which can easily be too small if they resized it for some reason. Since scroll bars will not be present to warn the user that the complete form is not displayed, it is important that Auto Resize is set to Yes.

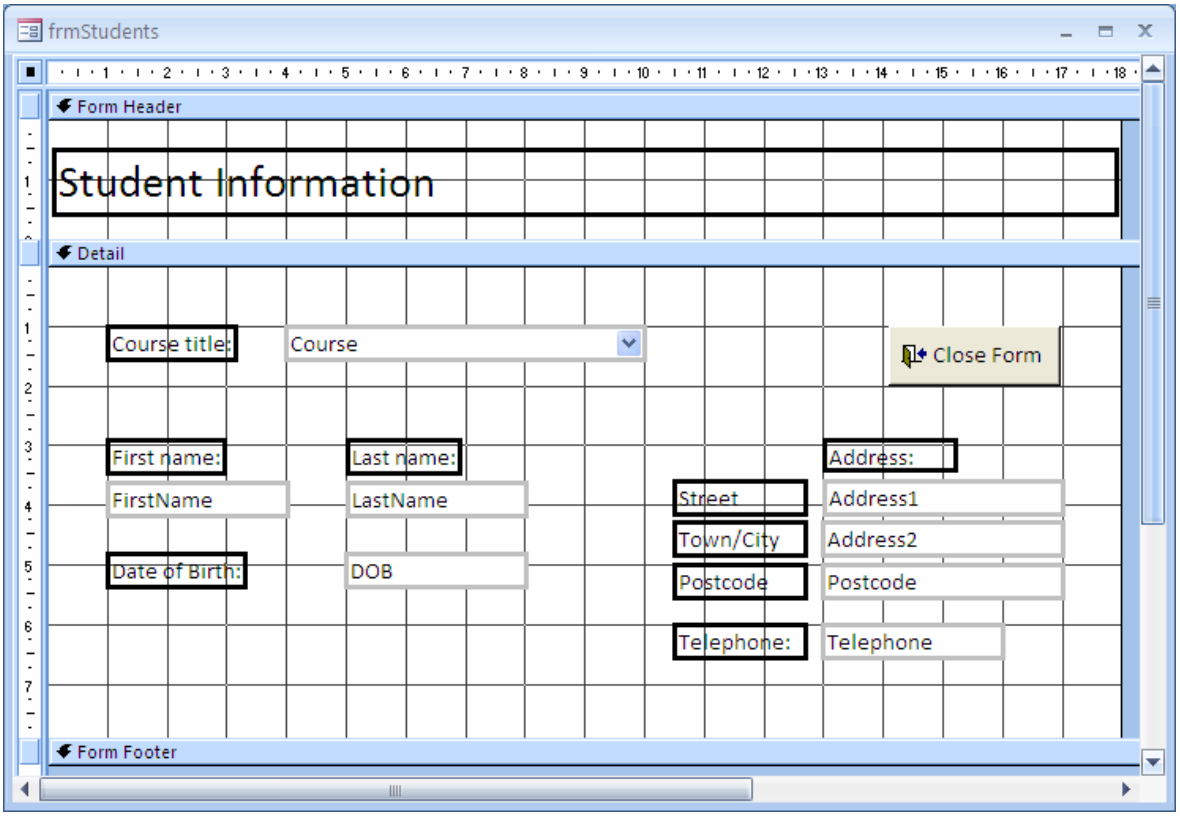
- Save the form

- Display the form in Form View to see the effect of the changes

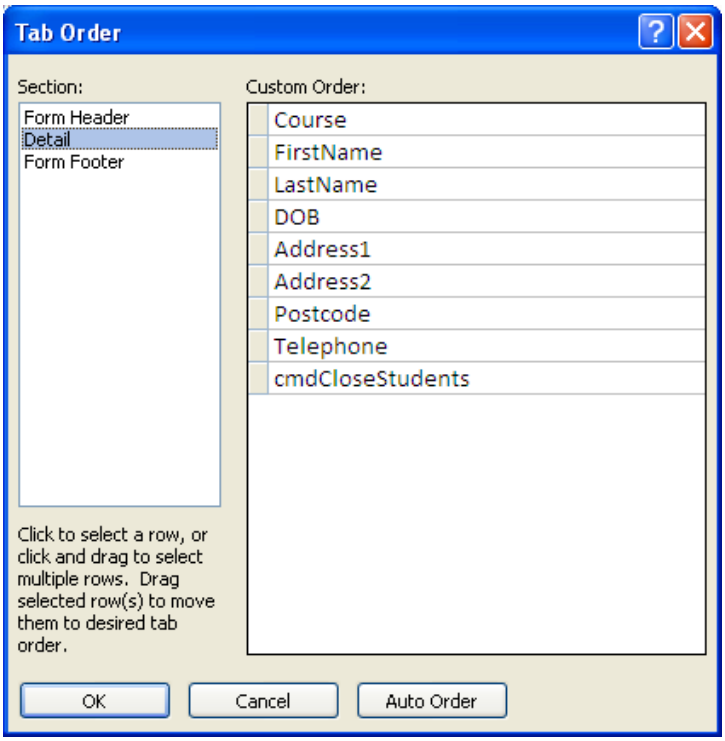
- Close the form

EXERCISE

- Create a blank form in Design View
- Display the Form Header (you do not need the Form Footer)
- Make the **width** of the form **18cm**
- Create a label in the Form Header, and use this to give the form the heading **Student Information**. Change the font size to **22**. Resize and reposition as shown in the diagram below
- Set the **Record Source** for the form to be the **students** table
- Drag all the fields from the students table on to the Detail section of the form and reposition and resize as shown in the diagram below (the darker boxes are labels)
- Add a Command Button, named **cmdCloseStudents**, to close the form



- Ensure that the Tab Order is as displayed in the diagram below



- Save the form as **frmStudents**
- Change the form's **Caption** to **2010 Students**
- Display the form in Form View and check that all fields are displayed as expected
- Use the **Tab** key to check the Tab Order
- Save the form
- Use the Command Button to close the form

APPLYING AUTOFORMATS

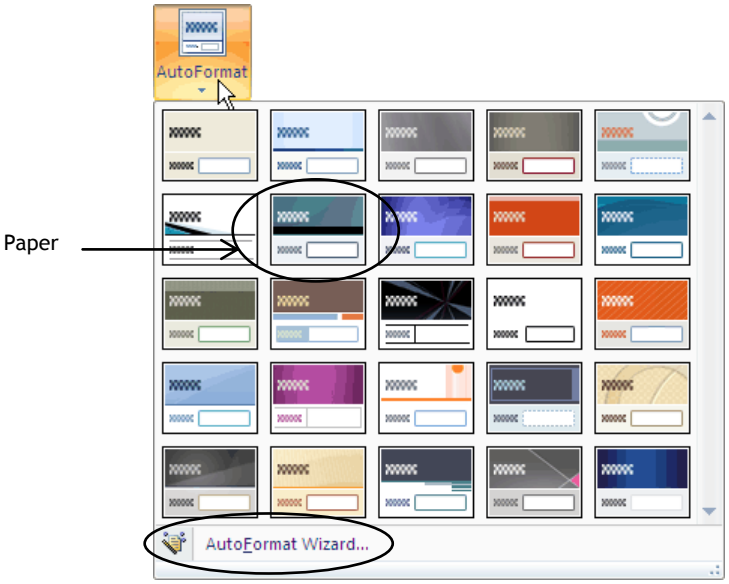
Rather than define the format of each object in the form separately, you can apply one of the AutoFormats installed with Access.

- Open the **frmStudents** form in Design View

When a form is opened the whole form is initially selected. This is important for the next step, because AutoFormats are applied to the selected object(s).

- Click **AutoFormat** on the **Arrange** tab

A gallery of AutoFormat designs is displayed.



- Hover the mouse over an AutoFormat design to see its name
- Select the **Paper** AutoFormat

The whole form is formatted using the colours and fonts from the selected AutoFormat.

- Click **Undo** to remove the formatting

If Undo is not initially available, switch to Form View and back to Design View.

AutoFormats can be applied to individual sections of a form, or to individual controls.

- Click the Form Header bar to select only the Header
- Apply the **Paper** AutoFormat

The formatting is applied to the Header section only.

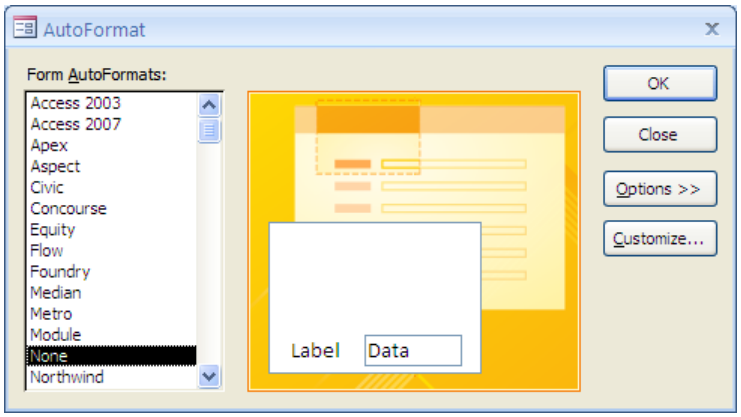
- Click **Undo**

User-defined AutoFormats

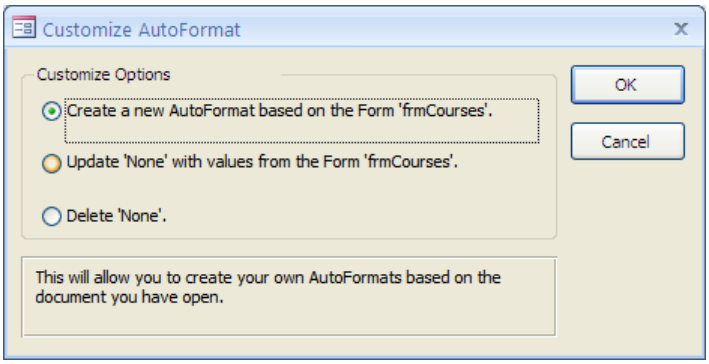
If you have a formatted form, and wish to use the same formatting on other forms, you can create a new AutoFormat from your existing form.

You will save the formatting from the frmCourses form and apply it to frmStudents.

- Open the form **frmCourses** in Design View
- Click **AutoFormat**
- Click **AutoFormat Wizard**
- Select **None** from the AutoFormats list (this is not strictly necessary, but is good practice in case the AutoFormat gets accidentally overwritten)

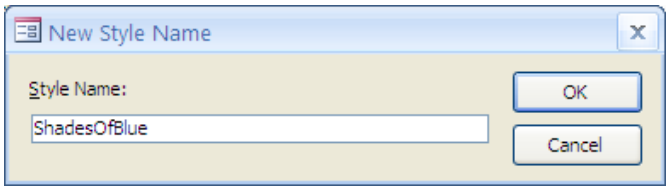


- Click **Customize**
- Select **Create a new AutoFormat based on the Form 'frmCourses'**



- Click **OK**

- Change the suggested Style Name to **ShadesOfBlue**



- Click **OK**
- Click **Close** on the AutoFormat dialog box
- Close **frmCourses**

To apply the ShadesOfBlue AutoFormat to frmStudents:

- Ensure that **frmStudents** is open in Design View and that the whole form is selected
- Click **AutoFormat**
- Click **AutoFormat Wizard**
- Select **ShadesOfBlue** from the AutoFormat list
- Click **OK**

One current problem with AutoFormat is that it moves labels to the right if the contents are either centre or right aligned. This can be overcome by creating your user-defined styles without special text alignment, leaving any such alignment tasks to be done after the AutoFormat is applied.

The ShadesOfBlue AutoFormat includes an aligned label in the Form Header. Some adjustment is therefore needed.

- Drag the **Student Information** label to the left edge of the form without resizing it
- Check the right edge of the form and resize the **width** to **18cm**
- Save the form

To prepare for the next example:

- Apply the **None** AutoFormat to the form

AutoFormat options

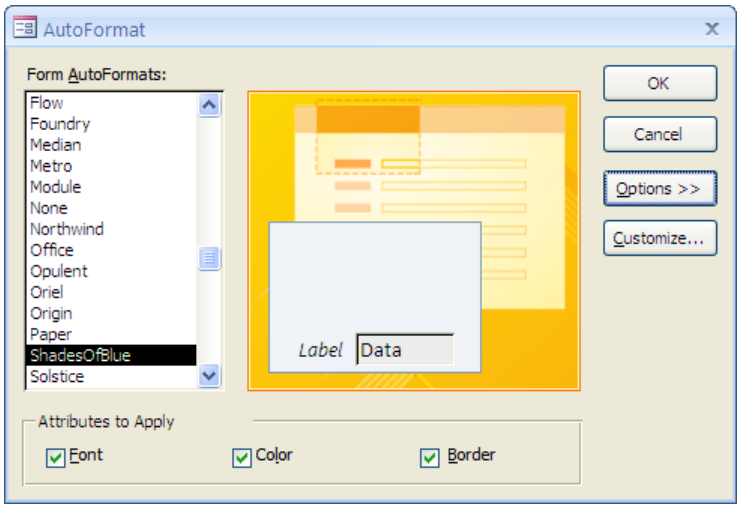
When applying User-Defined AutoFormat styles, it is possible to omit certain aspects of the style.

- Ensure that the whole form is selected in Design View
- Click **AutoFormat**
- Click **AutoFormat Wizard**

The AutoFormat dialog box is displayed.

- Select the **ShadesOfBlue** AutoFormat
- Click the **Options** button

Three check boxes are displayed at the bottom of the dialog box.



These attributes refer solely to the data boxes (text boxes, combo boxes, etc) and labels. They have no effect on other controls, or the background of the form itself.

To apply the AutoFormat without changing the background colour or border effect of the data boxes and labels:

- Remove the ticks from the **Color** and **Border** check boxes
- Click **OK**

At present, these options do not work smoothly for the built-in AutoFormats, and are therefore only recommended for use with User-Defined AutoFormat styles.

- Close the form without saving

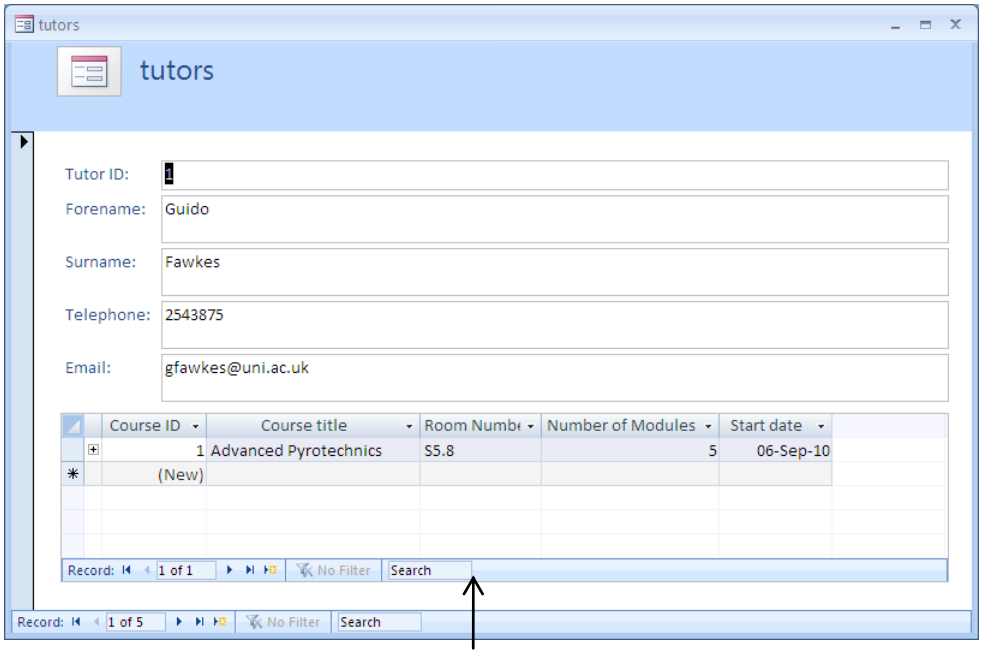
SUBFORMS

A sub form is a form within another form. Sub forms are useful when displaying information from tables where a one-to-many relationship exists.

There is a one-to-many relationship between the tutors table and the courses table. Each tutor can be assigned to many courses. The example that follows, uses this relationship to illustrate the concept of a sub form.

Automatically creating a form with a sub form

- Select the **tutors** table in the Navigation pane (do not open the table)
- Click **Form** from the **Forms** group on the **Create** tab



Sub form showing the course that the tutor is linked to

There are two forms displayed on the screen.

The main form has a Columnar layout and displays the fields from the tutors table, while the embedded sub form (showing the course information) has a Datasheet layout.

Sub forms usually have a Datasheet, Tabular, or similar layout that displays each record in a separate row.

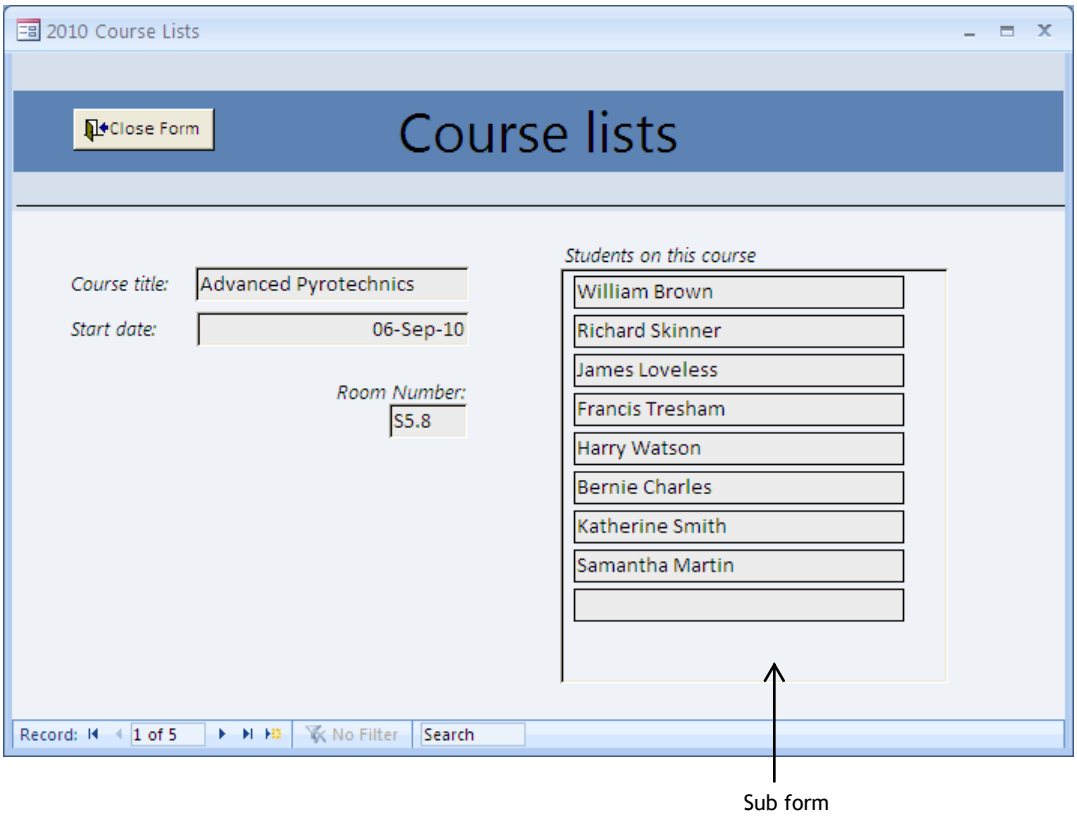
- Use the Navigation buttons at the bottom of the main form to scroll through the different tutors

Courses can be added to a tutor's list by using the sub form.

- Close the form without saving it

Creating and adding a sub form manually

In this next exercise, you will create the form shown below.



Here the main form (also referred to as the Parent form) displays data from the courses table, while the sub form (also referred to as the Child form) displays information from the students table.

This arrangement is possible because there is a one-to-many relationship between the courses and students tables, such that each course can contain many students.

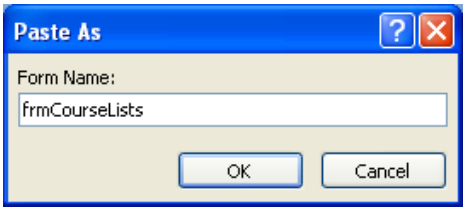
In the diagram above, the Parent form shows details of the Advanced Pyrotechnics course and the Child form displays a list of students enrolled on this course.

When a different course is selected, a different set of students is displayed.

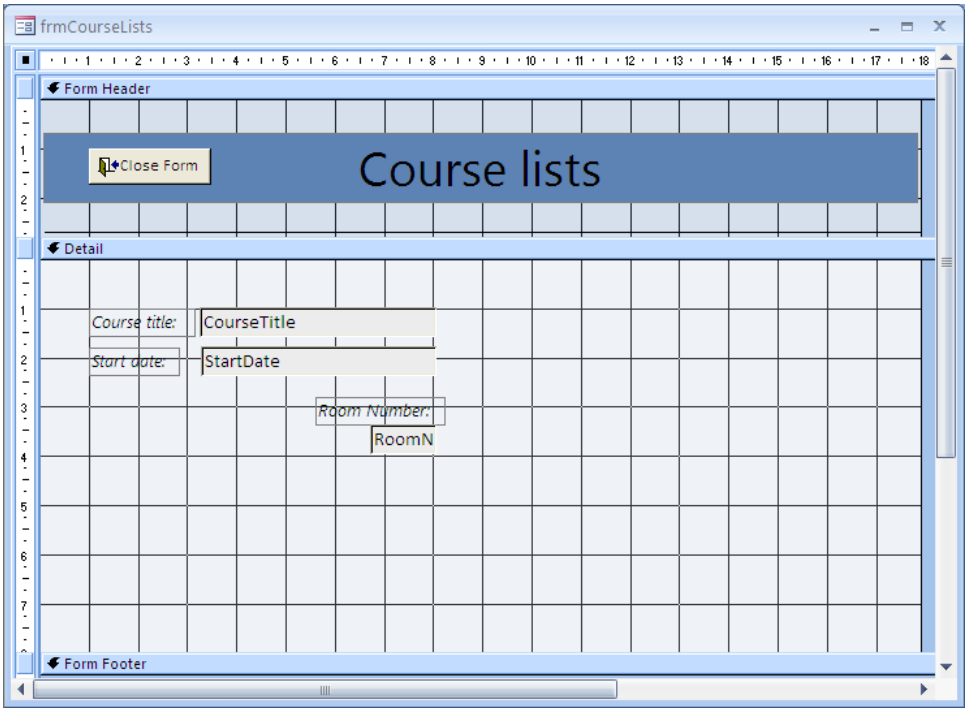
You will edit a copy of the form frmCourses to create the parent form, create a separate form to contain the student names as the child, and then put the two forms together.

To create the parent form:

- Select **frmCourses** in the Navigation pane (do not open the form)
- Click **Copy** from the **Clipboard** group on the **Home** tab
- Click **Paste**
- Change the suggested name to **frmCourseLists**



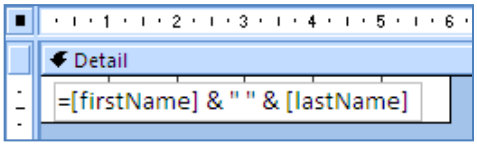
- Click **OK**
- Open **frmCourseLists** in Design View
- Change the title in the Form Header to **Course lists**
- Drag the **Close Form** button to the left of the title in the Form Header
- Delete the **image** and the **Course Leader**, **Course ID**, and **Number of Modules** controls from the Detail section
- Arrange the remaining controls as shown below



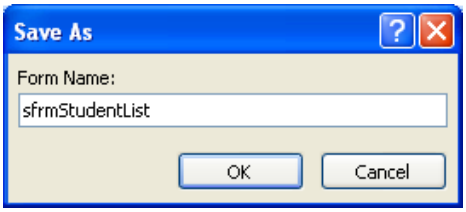
- Change the **Caption** property of the form to **2010 Course Lists**
- Change the **Record Source** property of the form to **courses**
- Save and close the form

To create the Child form:

- Click **Form Design** from the **Forms** group on the **Create** tab
- Set the **Record Source** property to **students**
- Place a text box on the form and create a **Calculated Control** to combine the **firstName** and **lastName** fields with a space between the two values
- Resize the Calculated Control so that it is just wide enough to display the text in full
- Delete the associated label
- Reposition the Calculated Control near the left side of the form and as close as possible to the top
- Drag the right edge of the form towards the Control so that the space on the right is approximately the same as that on the left
- Drag the bottom of the form up to the bottom border of the Control



- Save the form as **sfrmStudentList**



- Display the form in Form View

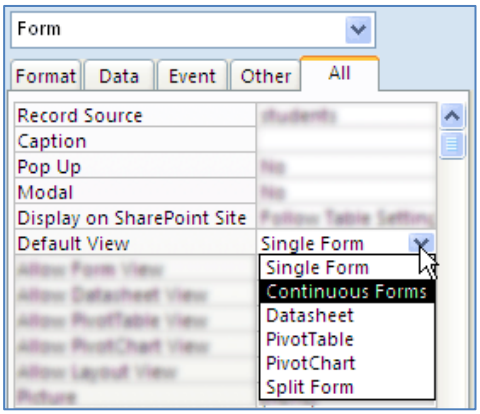
The name of one student is displayed on the form. To see the other students you need to use the navigation buttons to scroll through the individual records.

Sub forms are usually configured to show the records as a continuous list.

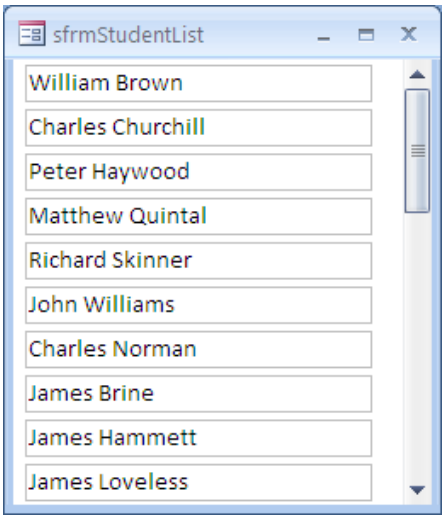
- Display the form in Design View

To display the names as a list:

- Ensure that the Form's Property Sheet is displayed
- Click the **Default View** property
- Click the down arrow



- Select **Continuous Forms**
- Display the form in Form View



- Display the form in Design View

The other properties that you will set are mainly for cosmetic purposes.

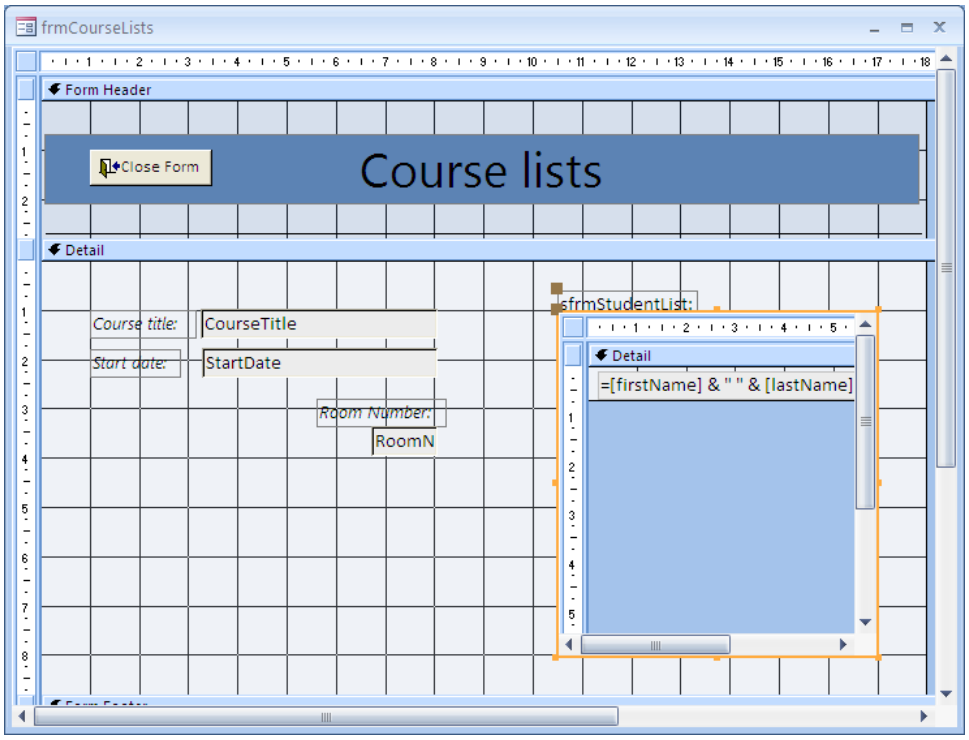
- Set the properties shown below:

Object	*Property*	*Setting*
Form	**Record Selectors** **Navigation Buttons** **Scroll Bars**	**No** **No** **Vertical Only**
Detail	**Back Color**	**Light Blue 1** (#EFF2F7)
Calculated Control	**Back Color**	**Light Gray 1** (#ECECEC)

- Save and close the form

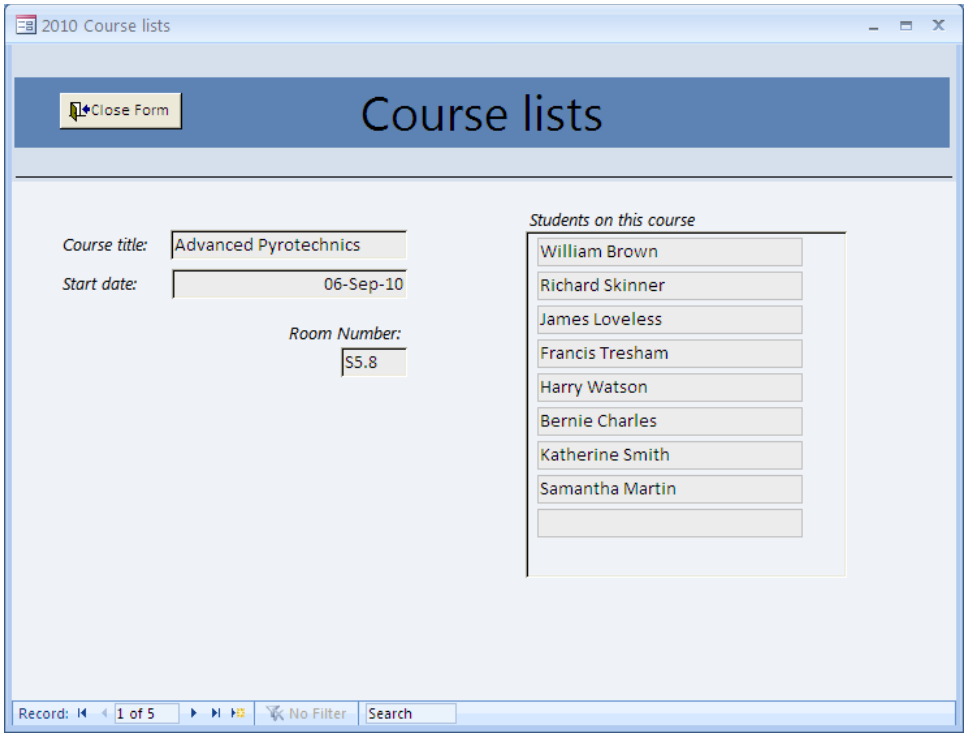
To combine the two forms:

- Open **frmCourseLists** in Design View
- Drag **sfrmStudentList** on to the Detail section from the Navigation pane
- Position and resize the sub form control as shown below



- Change the text on the sub form label to: **Students on this course**
- Use the Format Painter to copy the formatting of the Course title label to the sub form label

- If necessary, drag the label up to fully display the text
- Display the Property Sheet for the sub form control
- Change the **Special Effect** property to **Sunken**
- Save the form
- Display the form in Form View



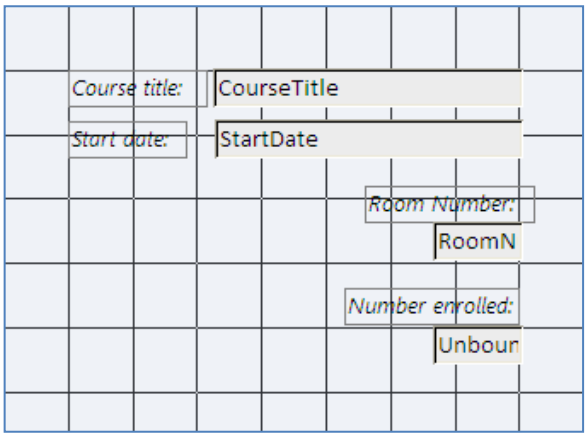
- Use the navigation buttons to scroll through the different courses
- Display the form in Design View

Adding a count on the parent form

You will create a calculated control on the parent form that displays the number of students enrolled on each course.

- Add a text box to the form in the space below the Room Number
- Set the **Name** property of the text box to **txtNumStudents**
- Copy the formatting from the Room Number text box to the new unbound text box
- Change the text on the associated label to **Number enrolled**

- Reposition and resize the new controls as shown below



Access does not allow you to count the records on the sub form directly from the parent form. You need to create a control on the sub form to calculate a count of the records, and reference this from the parent form.

- Save and close **frmCourseLists**

- Open **sfrmStudentList** in Design View

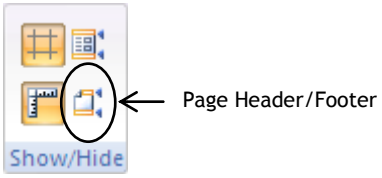
You now need to decide where on the form to place the control to count the records.

The function of the control that you are about to create on the sub form is to perform a calculation and hold the result, but not to display it.

There are two possible solutions. It can be positioned anywhere, including overlapping other controls, with its **Visible** property set to **No**, or it can be positioned where it cannot be seen when the form is displayed in Form View.

Page Headers and Page Footers are not displayed in Form View. You will therefore create the calculated control in the form's Page Header.

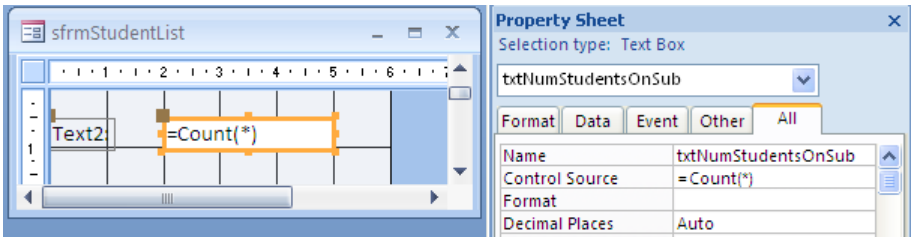
- Click the **Page Header/Footer** button from the **Show/Hide** group on the **Arrange** tab



A Page Header and Page Footer are added to the form.

There is no need to resize any of these sections because they will not be displayed in Form View.

- Add a text box to the Page Header
- Set the **Name** property to **txtNumStudentsOnSub**
- Set the **Control Source** property to **=Count(*)**
  (this means count all records)



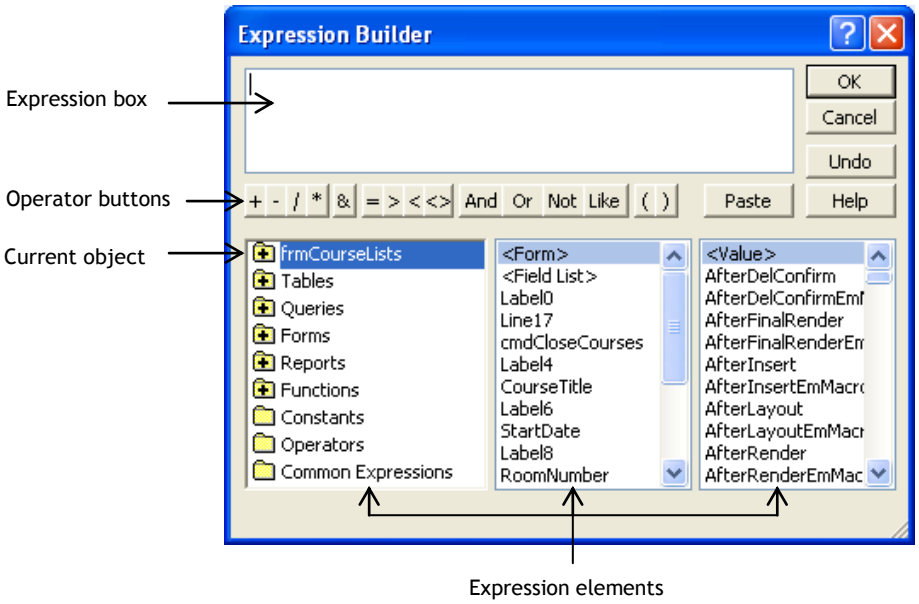
- Save and close the form
- Open **frmCourseLists** in Design View
- Select the Unbound text box

You now need to set the Control Source of this control to equal the value of txtNumStudentsOnSub.

The Expression Builder

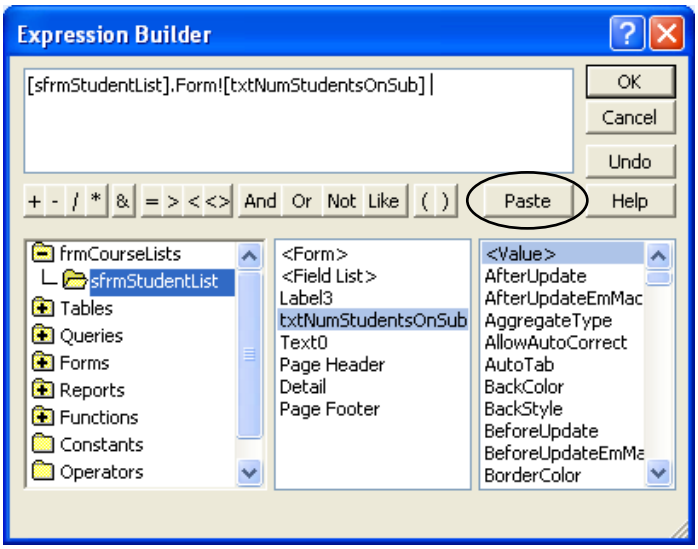
The naming conventions of forms and sub forms can be tricky, so you will use the **Expression Builder** to build the name for you. The expression builder organises all the elements of the database into a hierarchical structure.

- Click in the **Control Source** property
- Click the ellipsis button to launch the Expression Builder

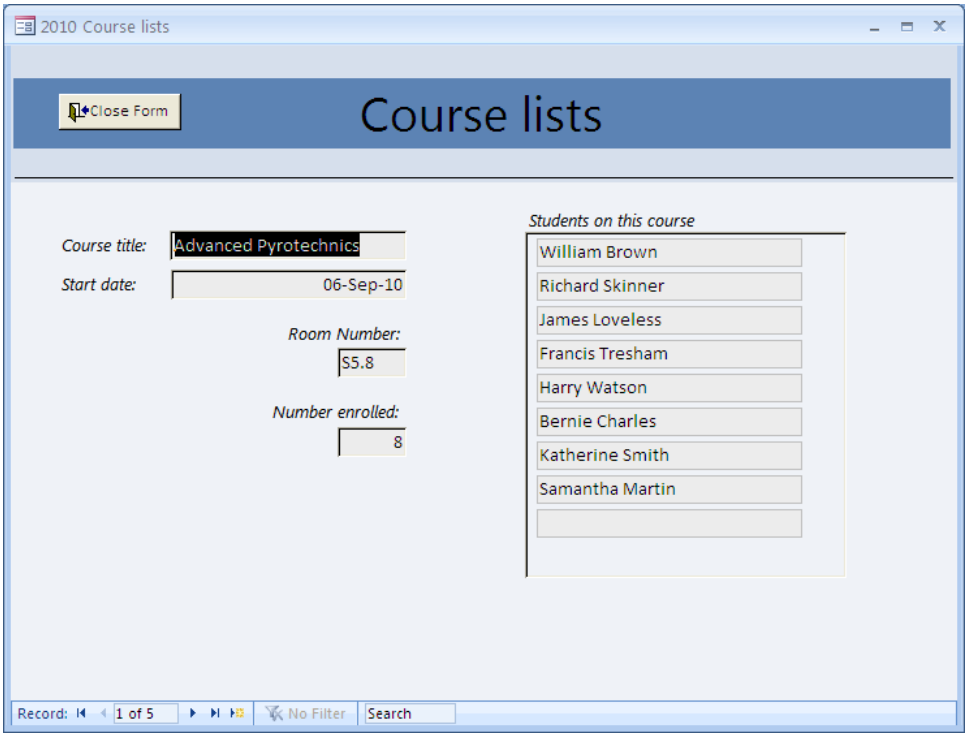


The upper section of the Expression Builder contains an expression box where you construct your expression. To build the expression, you can type directly into the expression box, or locate elements using the three columns in the lower section of the Expression Builder and paste them into the box.

- Double-click the plus sign by **frmCourseLists** in the left column of the builder
- Click the sub form **sfrmStudentList**
- Click the element **txtNumStudentsOnSub** in the middle column
- Ensure that **<Value>** is selected in the column on the right
- Click **Paste**



- Click **OK**
- Save the form
- Display the form in Form View
- Scroll through the courses to verify that the count is working as expected

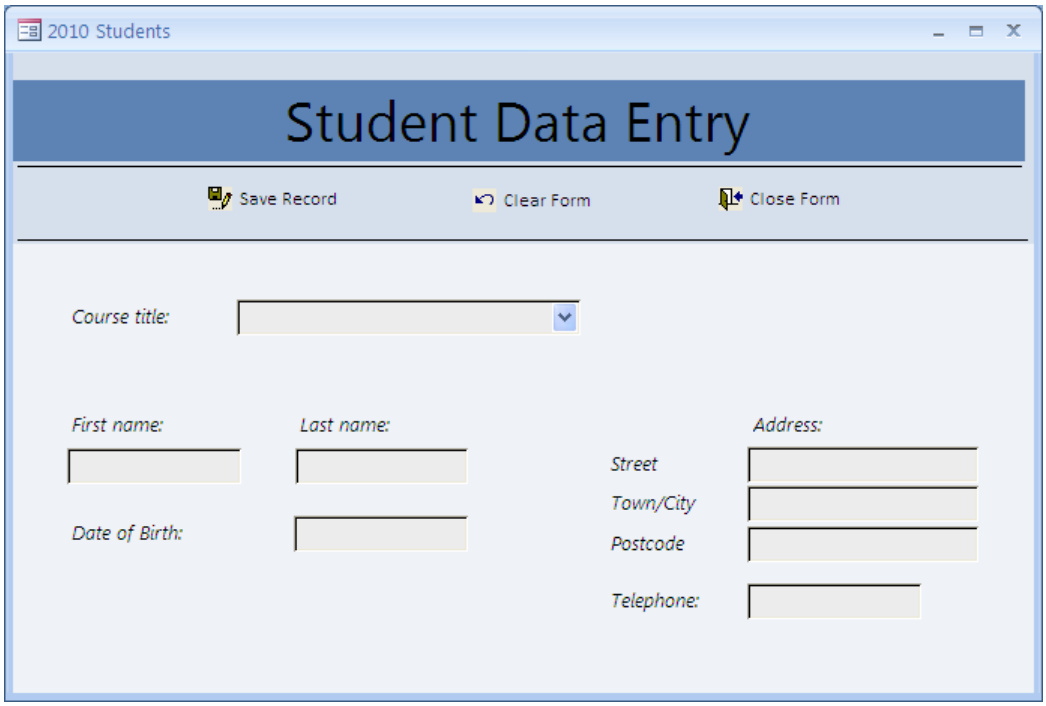


- Close the form

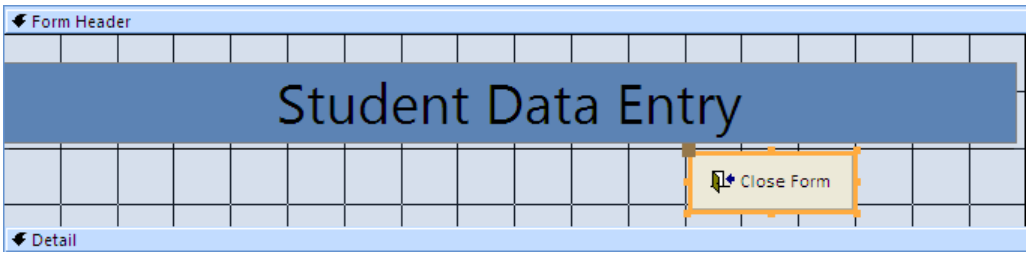
DATA ENTRY FORMS

The forms that you have already created can all be used to enter data into the tables on which they are based. However, many database designers prefer to create different forms for different purposes, and would present the user with a blank form if it is to be used for data entry.

You will create the following form for the purpose of entering new students into the database.

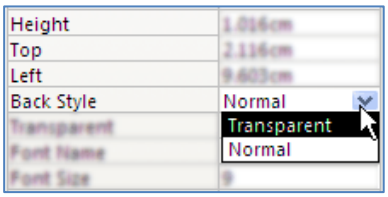


- Create a copy of the form **frmStudents** and save it as **frmStudentEntry**
- Open **frmStudentEntry** in Design View
- Change the title in the Form Header to **Student Data Entry**
- Drag the Detail bar downward to create space in the Form Header for the command buttons
- Drag the **Close Form** button into position in the Form Header

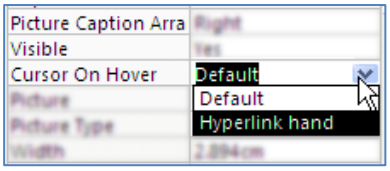


- Display the properties for the **Close Form** button

- Click the **Back Style** property
- Click the down arrow



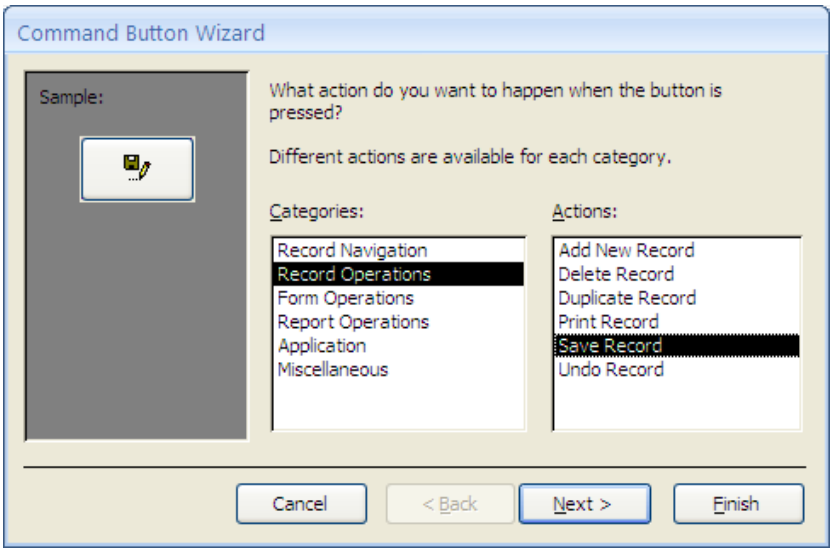
- Select **Transparent**
- Click the down arrow for the **Cursor On Hover** property



- Select **Hyperlink hand** (to help users identify this as a button)

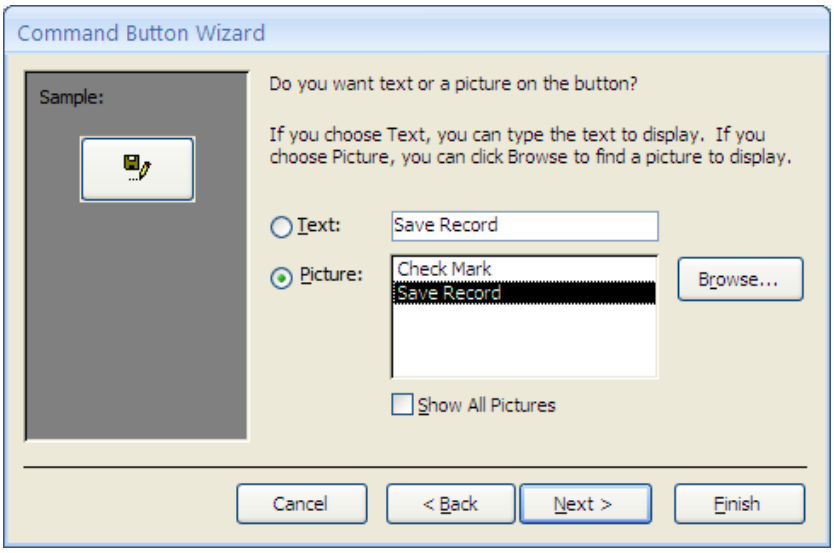
To create a Command Button to Save the record:

- Create a new Button Control to the left of the Close Form button
- In the Command Button Wizard, select **Record Operations** from the **Categories** column and select **Save Record** from the **Actions** column

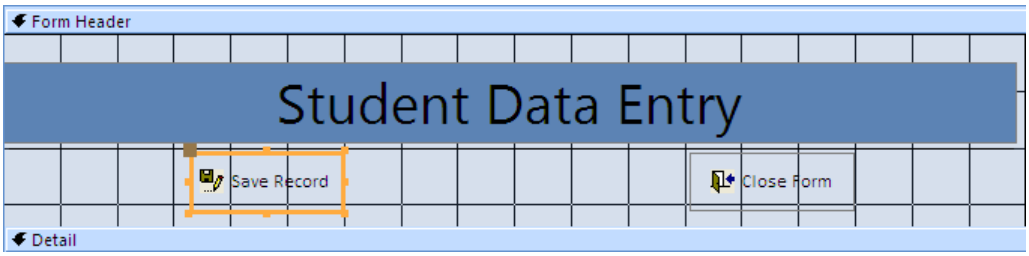


- Click **Next**

- Select the **Save Record** picture

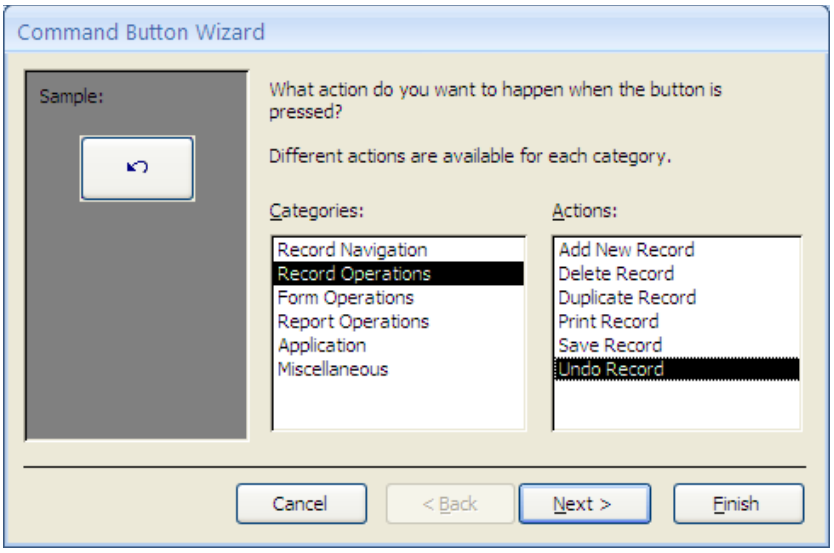


- Click **Next**
- Change the suggested name to **cmdSaveStudent**
- Click **Finish**
- Change the **Caption** property to **Save Record**
- Display the caption to the right of the picture (see page 108)
- Set the **Back Style** property to **Transparent**
- Set the **Cursor On Hover** property to **Hyperlink hand**
- Resize and reposition the button as shown below

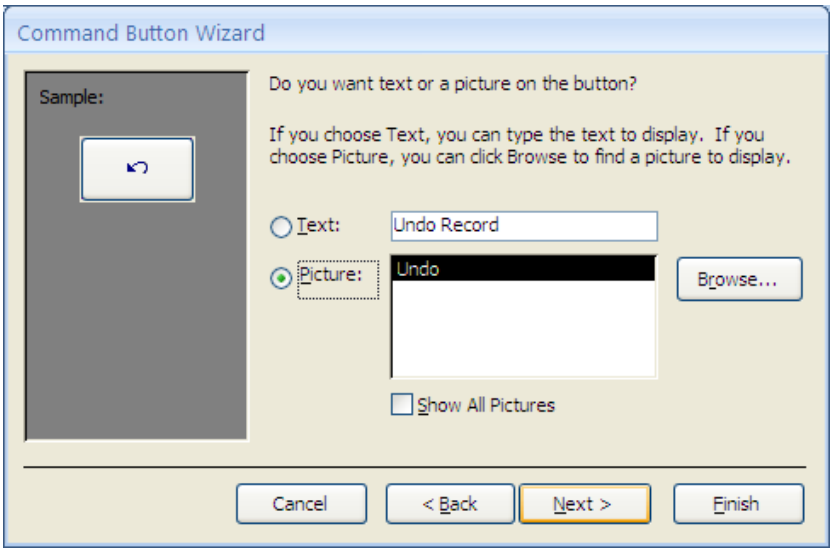


To create a Command Button to Clear the form:

- Create a new Button Control between the Save and Close buttons
- In the Command Button Wizard, select **Record Operations** from the **Categories** column and select **Undo Record** from the **Actions** column

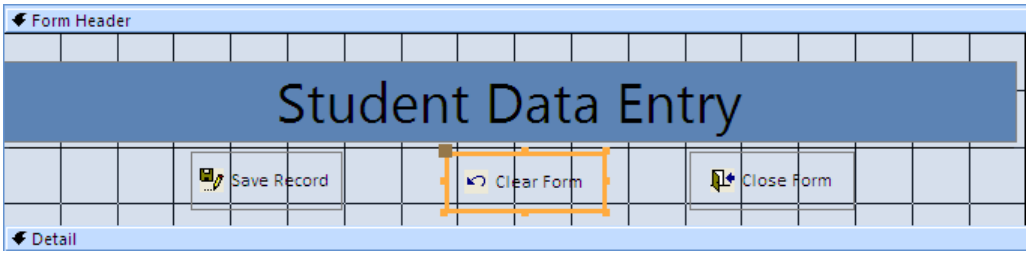


- Click **Next**
- Select the **Undo** picture



- Click **Next**
- Change the suggested name to **cmdClearForm** and click **Finish**

- Change the **Caption** property to **Clear Form**
- Set the Caption to display on the right of the picture
- Set the **Back Style** and **Cursor On Hover** properties as the other buttons
- Resize and reposition the button as shown below



- Insert horizontal lines above and below the buttons

To convert the form to a Data Entry form:

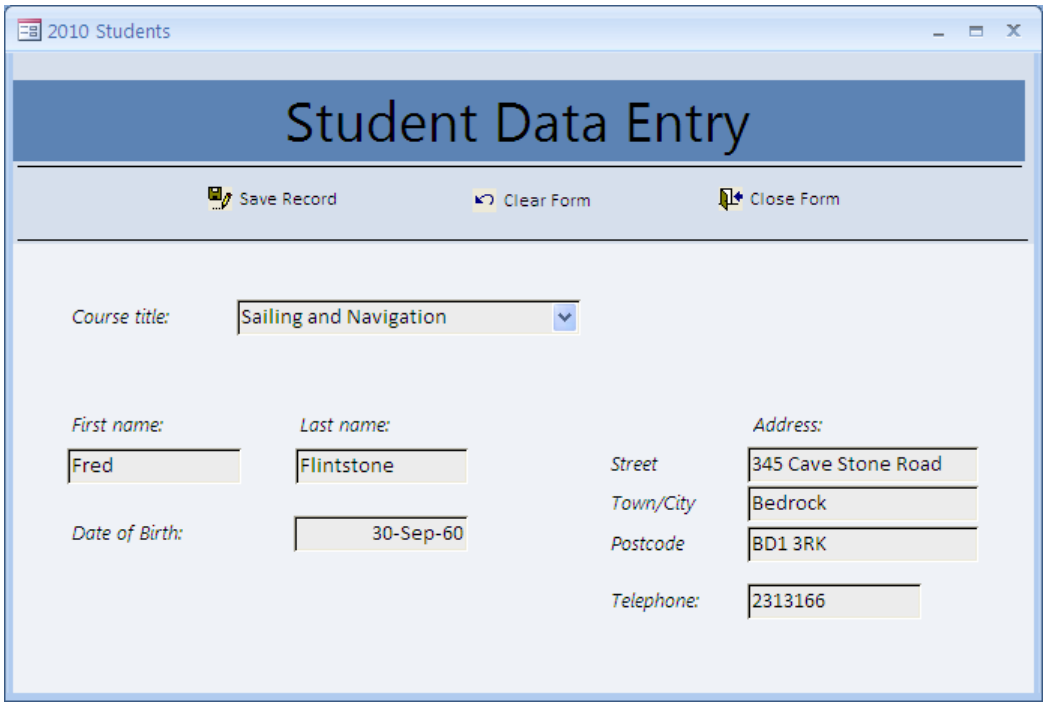
- Set the form properties listed below:

Form Property	*Setting*
Pop Up	**Yes**
Modal	**Yes**
Record Selectors	**No**
Navigation Buttons	**No**
Scroll Bars	**Neither**
Data Entry	**Yes**

The most important property in the above list is the Data Entry property. When the Data Entry property is set to Yes, the form no longer displays records from the underlying table or query. The user is therefore presented with a blank form.

- Close the Property Sheet
- If necessary, drag the bottom edge of the form down to ensure that all the controls are visible
- Save the form
- Display the form in Form View
- Enter any text in some of the fields
- Click the **Clear Form** button to undo the entry

- Enter the following data



- Click the **Save Record** button

Notice that there is no indication that the action was successful.

You will rectify this within the next topic by presenting the user with a blank form once the record is saved.

- Close the form using the **Close Form** button

MACROS

A macro is a tool that enables you to automate tasks and add functionality to your forms and reports.

In Access, macros can be regarded as a simplified programming language that you write by building a list of actions to perform.

Macros may be created as named objects, in which case they will be displayed as objects in the Navigation pane, or they can be attached to a form, report or control, in which case the macro will be embedded within the form or report, and will not be displayed as a separate object.

Each of the examples used here will be attached to a control on a form and will be linked to what is referred to as a 'form event'.

A form event is an action that is associated with the form as a whole, or with a control on the form. Often events are associated with user action. Form events include: opening a form, selecting a control, editing the value of a control, clicking a command button, closing a form.

Linking a macro to an event will mean that, when the particular event occurs, the macro will run automatically.

Without you realising it, some macros have already been added to your forms.

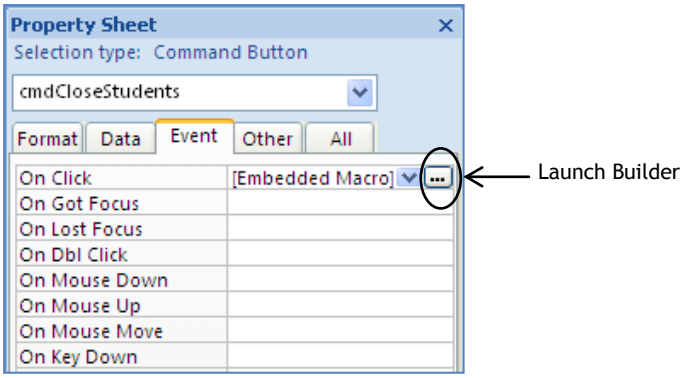
You will view the macro that is embedded in the cmdCloseStudents command button control.

If a macro is embedded in a form it will be listed as one of the property values of the form.

- Open **frmStudentEntry** in Design View
- Select the **Close Form** command button
- Ensure that the Property Sheet is displayed
- Select the **Event** tab on the Property Sheet

The possible events associated with this control are listed.

Notice that the **On Click** event has an **Embedded Macro** as the value of its property.

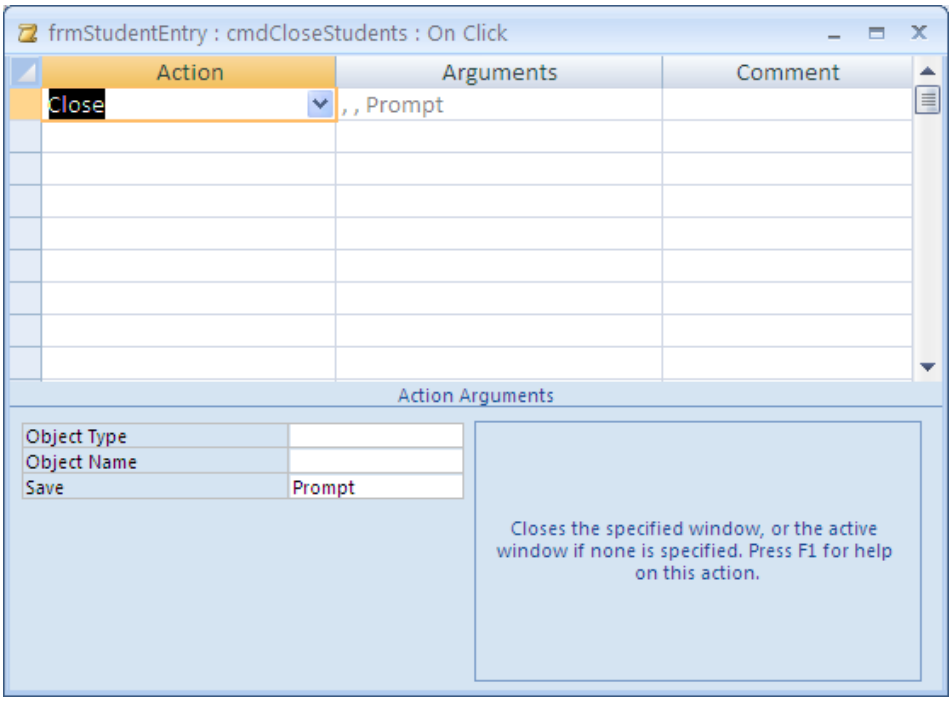


The Macro Builder

The Macro Builder is the tool that is used to create, display, and edit macros.

To display the Macro Builder:

- Click the **Launch Builder** button in the **On Click** property

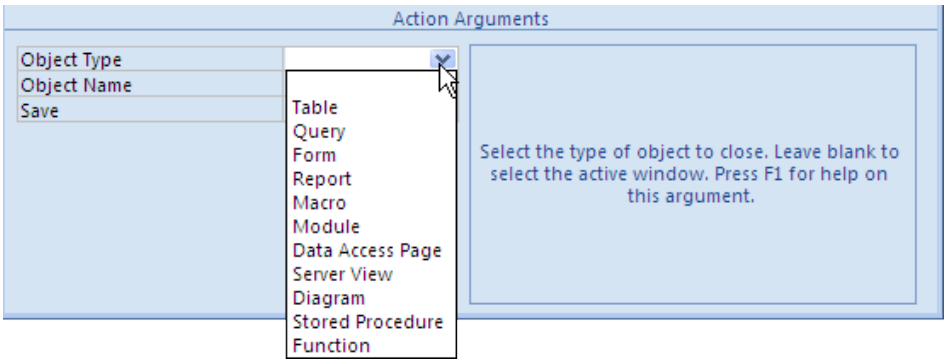


The top section of the Macro Builder lists the Actions that the macro will perform. This macro has just one Action: **Close**.

The bottom section of the Builder shows the **Arguments** for the selected Action and also displays a description of any item selected.

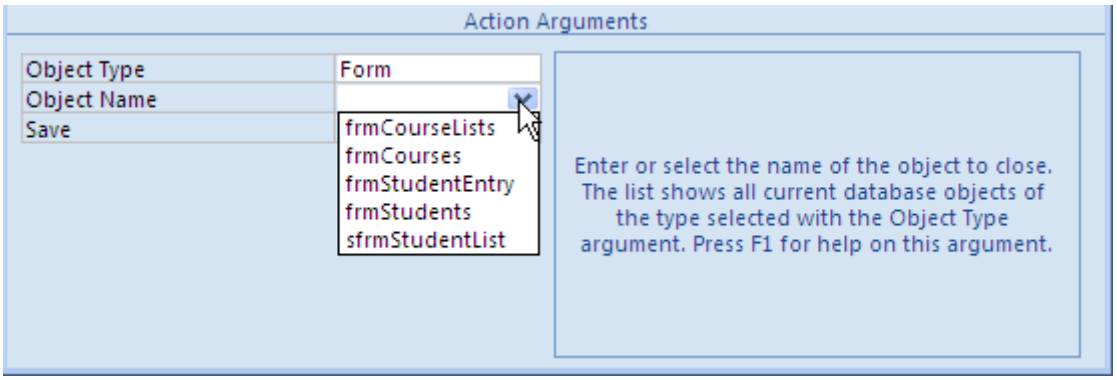
The **Close** Action will close the active window if no object is specified in the Action Arguments, therefore it correctly closes the form on which the button sits. However, to illustrate the process, you will specify the form **frmStudentEntry** as the window to be closed.

- Click in the **Object Type** argument and click the down arrow that appears



- Select **Form**

- Click the down arrow for the **Object Name** argument



- Select **frmStudentEntry**

- Click the **Save** button on the Quick Access Toolbar to save your changes

Notice that the selected arguments are displayed in the Arguments column in the top section of the window.

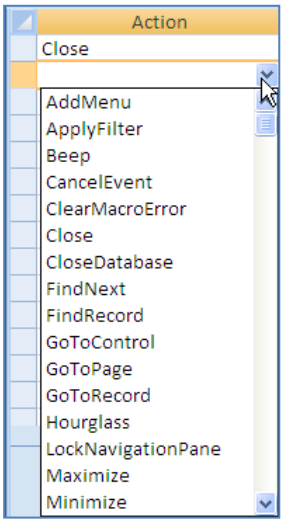
You cannot use the Arguments column to set or change argument values. The purpose of the column is to help you to read the macro when it consists of a list of different Actions.

When a macro contains more than one action, each action is put in a new row.

Action names can be selected from a list or typed into the Action column, but only the listed actions will be acknowledged and accepted.

To see a list of Actions:

- Click in any cell in the Action column and click the down arrow
- Scroll through the list



To close the Macro Builder:

- Click **Close** on the Macro Tools **Design** tab

You have already saved the desired changes, so if prompted to save further changes:

- Click **No**

Saving changes to the macro is not sufficient. You also need to save changes to the form.

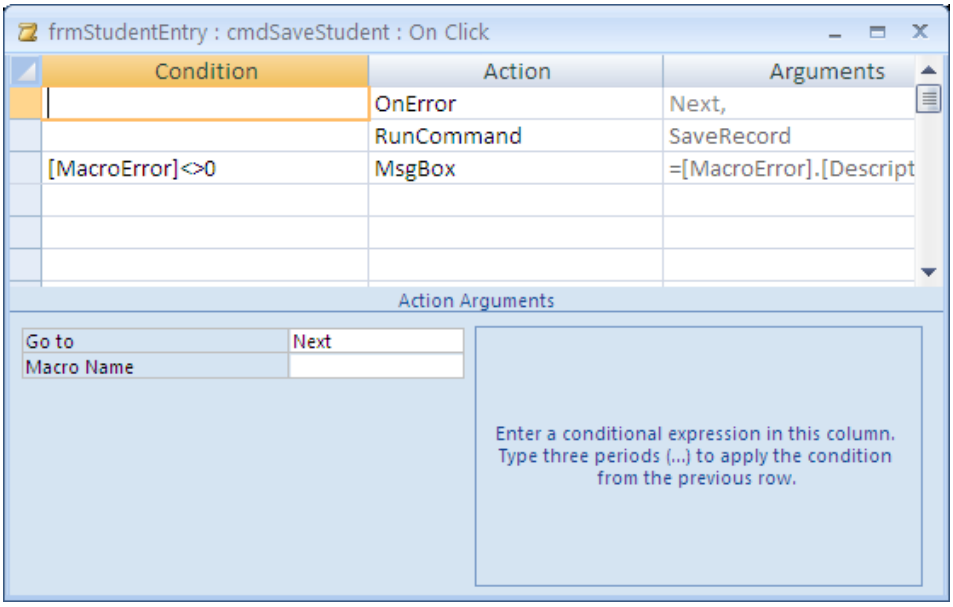
- Click **Save** on the Quick Access Toolbar
- View the form in Form View
- Check that the **Close Form** button works as expected

Adding an action to a macro

In the previous section, you used the Student Data Entry form (frmStudentEntry) to enter data about a new student.

The **Save Record** button saved the record but gave no indication that the action completed successfully (see page 142). You will edit the macro in the **On Click** event property to present the user with a new blank record after the data is saved.

- Open **frmStudentEntry** in Design View
- Select the **Save Record** command button
- Display the **Macro Builder** for the **On Click** event property



This macro currently has three actions: **OnError**, **RunCommand**, and **MsgBox**.

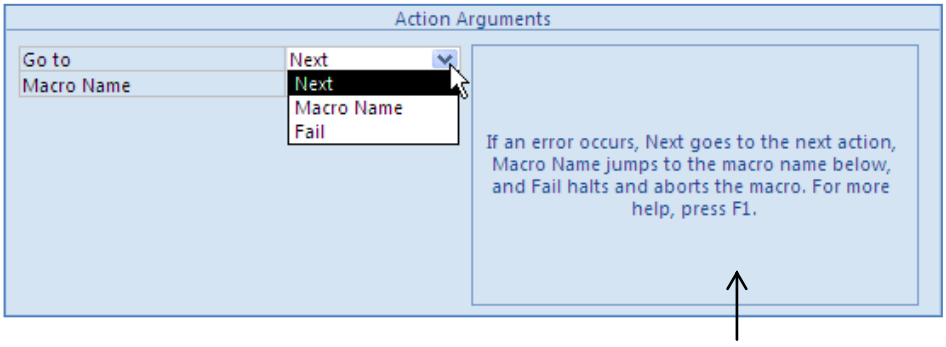
Before you add the new action to the macro, we will discuss briefly what each of the existing actions mean.

The **OnError** action is used to specify how the programme should respond if it encounters an error when executing this macro.

- Ensure that one of the cells in the first row is selected

The bottom pane displays the possible arguments for the OnError action.

- Click the **Go to** argument and click the down arrow



There are three options to choose from. Read the description box for an indication of what each of the options mean.

- Ensure that **Next** is selected

- Click the **RunCommand** action

This action is used when you want the database to run one of the built-in commands.

The Command to be executed is the only argument offered. Like most of the other arguments, the value is set by choosing from a list of possible commands.

- Click the down arrow for the **Command** argument
- Scroll through the list
- Ensure that **SaveRecord** is selected
- Click the **MsgBox** action

This action displays a message box.

The arguments for this action enable you to specify the message, decide whether to accompany the message display with an audible beep, choose the type of icon, and enter a title for the message.

Message	=[MacroError].[Description]
Beep	Yes
Type	None
Title	

You can type the text for the message directly into the Message argument, or use an expression to generate the desired message.

The message argument in this macro uses an expression to display a description of the macro error.

This message box should appear only if an error occurs, that is what the entry in the Condition column is designed to do. The action will be executed if the condition is satisfied, in other words if there is an error.

The Condition column is not displayed in the Macro Builder by default. The Conditions button in the Show/Hide group on the Design tab will toggle this column on and off.

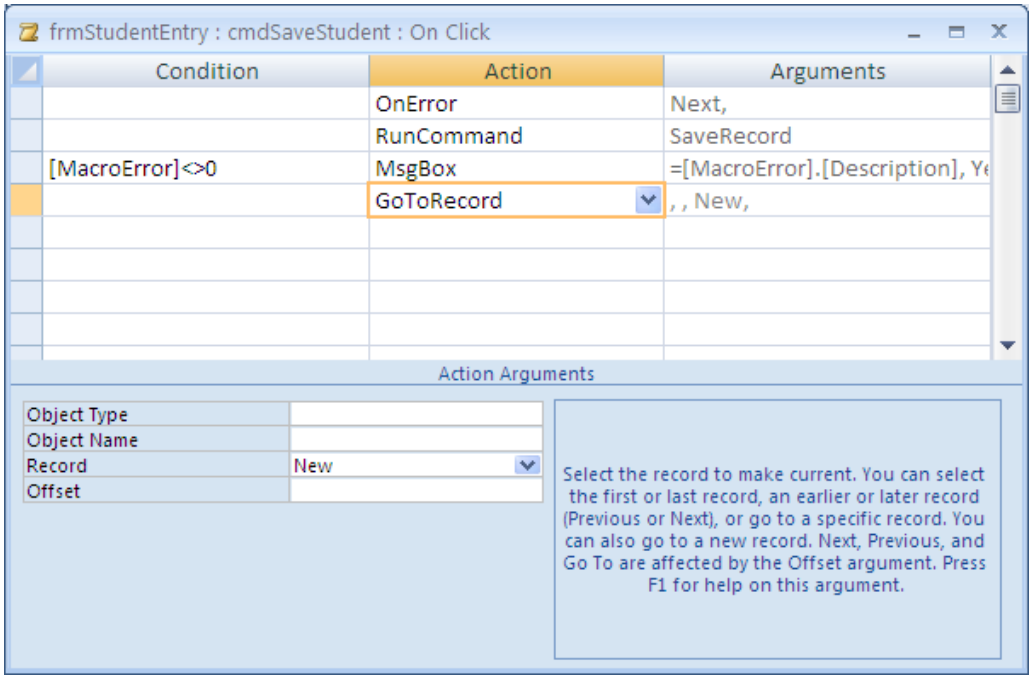
- Click the **Conditions** button in the **Show/Hide** group

The Condition column is hidden. To show the Conditions column once more:

- Click the **Conditions** button

You will add an action at the end of the existing list that will empty all the controls on the form.

- In the row below MsgBox, click in the **Action** box
- Click the down arrow and select **GoToRecord**
- Click in the **Record** argument and select **New**



This action is equivalent to clicking the New Record navigation button on the form.

- Save and close the Macro
- Save the form
- Display the form in Form View
- Enter data for a new student on the **Advanced Pyrotechnics** course and verify that the **Save Record** button works as expected
- Close the form

Synchronising Combo boxes

The idea here is that when users select a value from one combo box it alters the list of values presented in the second combo box, so that users are presented with appropriate values from which to select.

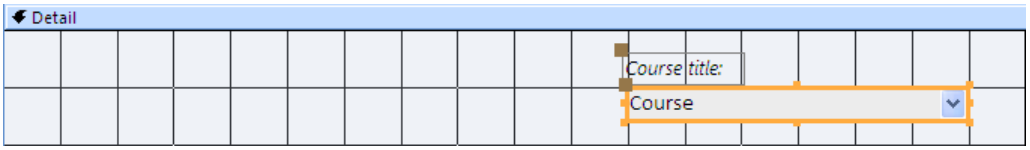
One situation in which this is useful is when a long list of options is divided into categories. Selecting a category from the first combo box causes the second combo box to list the options for the selected category.

- Create a copy of **frmStudentEntry** and save it as **frmComboSync**
- Open **frmComboSync** in Design View

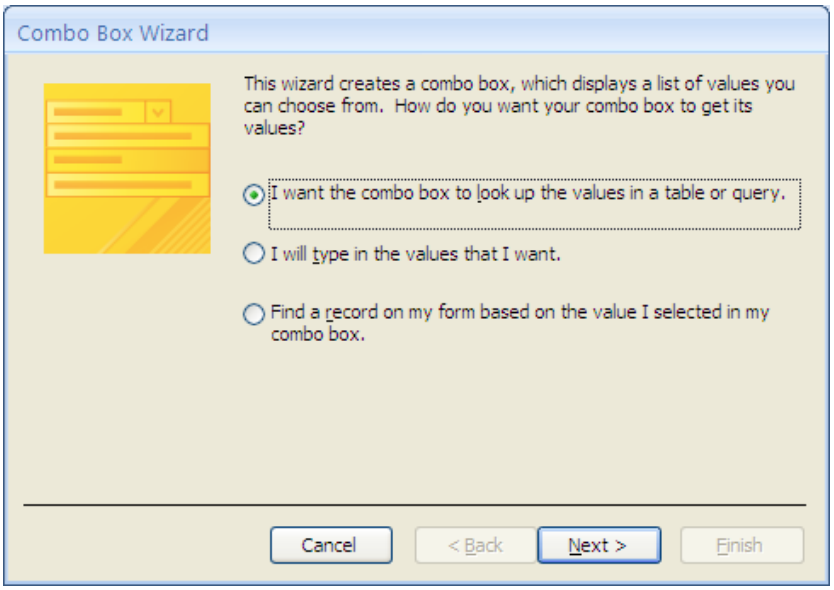
The intention is to replace the existing Course combo box with two combo boxes. The first combo box will be used to choose the faculty and the second will be used to choose a course within the selected faculty.

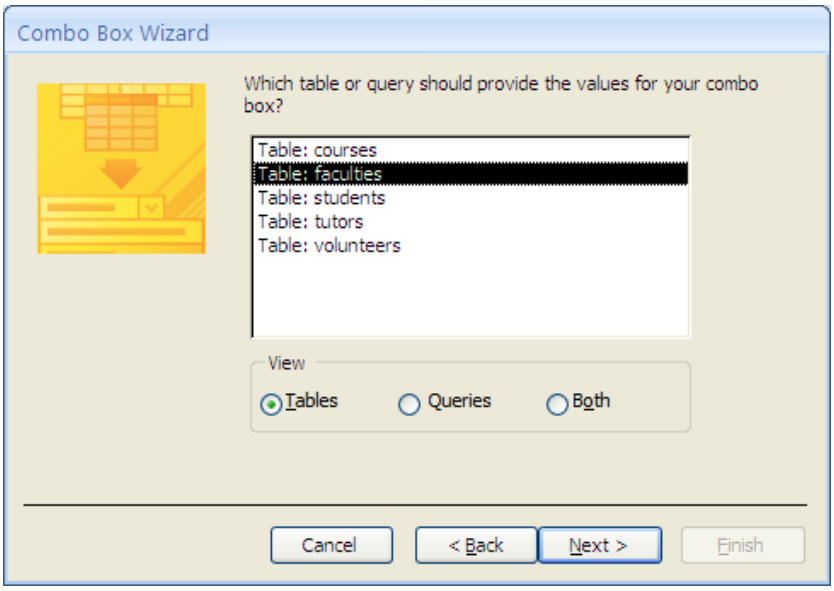
You will eventually delete the existing Course combo box, but not until you have verified that the new boxes are working correctly.

- Drag the **Course** combo box and its label to the right to make room for the new controls

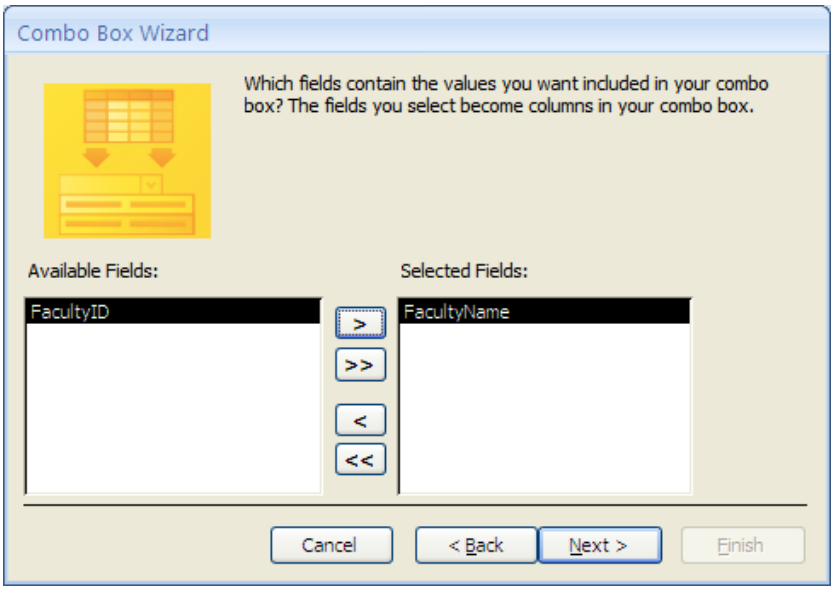


- Add a new Combo Box control to the form
- Choose to look up values from the **faculties** table

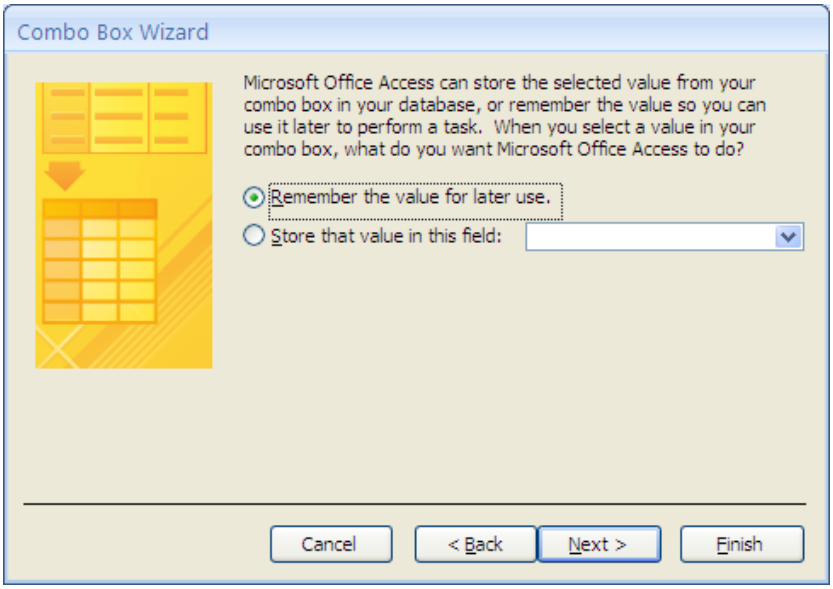




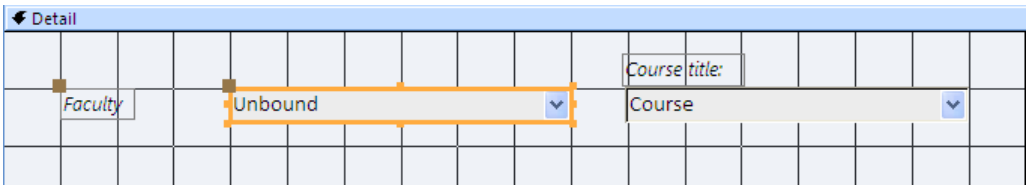
- Select **FacultyName** as the field that contains the values to be displayed



- It is not necessary to Sort the data in this example
- Ensure that the **Hide Key column** check box is selected as you work through the Wizard
- The value selected in this control will not be stored in any of fields on the students table, therefore select the radio button to **Remember the value for later use**



- Use **Faculty** as the text on the label
- Reposition and resize the Combo Box as shown below



- Set the **Name** property for this control to **cboFaculties**
- Save the form and keep it open in Design View

For the second Combo Box to display the courses in the selected faculty, you need to create a query that will display the appropriate courses and use this as the source for the list.

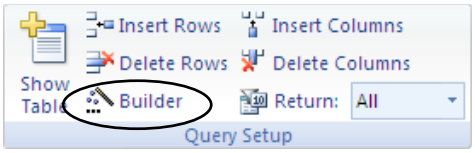
- Create a query in Design View
- Add the **faculties** and **courses** tables to the Design
- Add the **CourseID** and **CourseTitle** fields from the **courses** table, and the **FacultyID** field from the faculties table

You want this query to display the courses for the faculty selected in the cboFaculties control. An appropriate criterion must therefore be written in the FacultyID field.

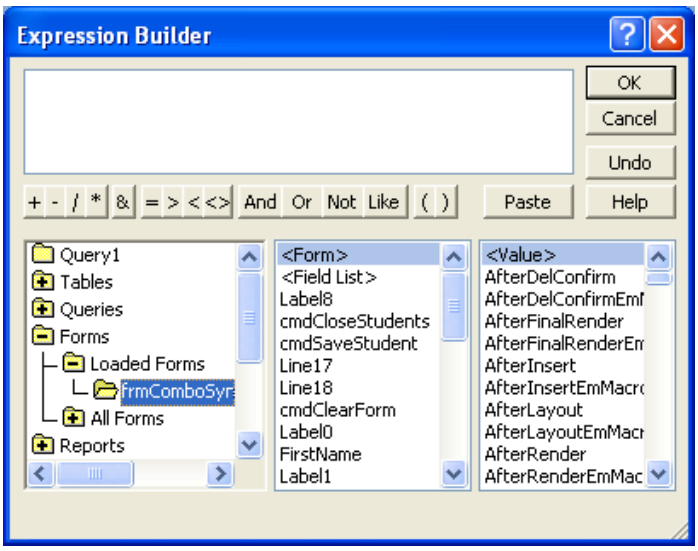
- Click in the **Criteria** row for the **FacultyID** field in the query grid

You need to instruct the application to pick up the value from the cboFaculties control on the frmComboSync form. This can be typed in if you know the syntax for the expression, but for this example you will use the Expression Builder.

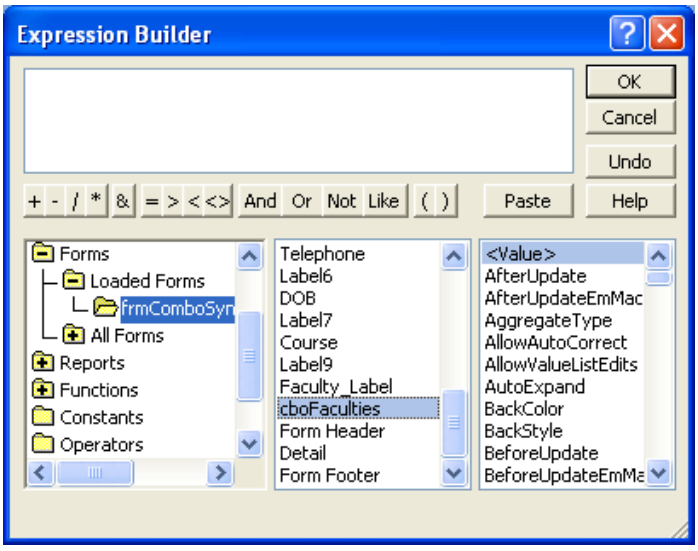
- Click **Builder** from the **Query Setup** group on the **Design** tab



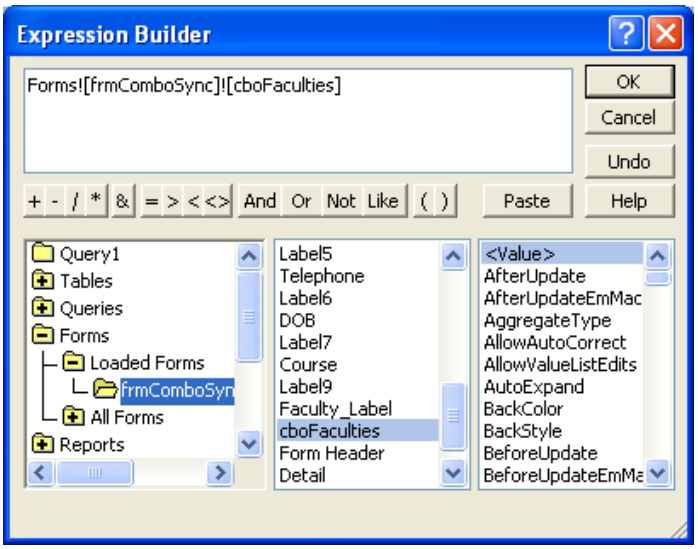
- Expand the **Forms** and **Loaded Forms** category and sub-category and select **frmComboSync**



- Scroll down the middle column if necessary and select **cboFaculties**



- Ensure that **Value** is selected in the column on the right
- Click **Paste**



- Click **OK**

The appropriate expression is inserted as the FacultyID criterion.

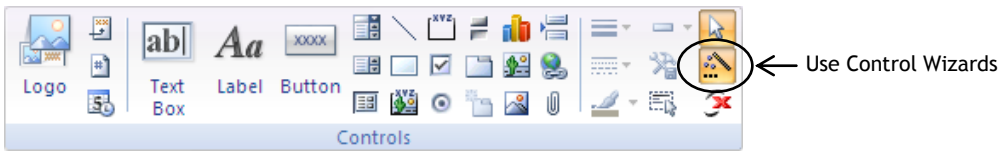
Do not run the query at this stage because the combo box cboFaculties does not currently have a value.

- The **FacultyID** field is not to be displayed when the query is Run. Remove the tick from the **Show** check box
- Save the query as **qryFacultyCourses**
- Close the query

The Combo Box Wizard will not be as helpful in configuring the second combo box control, so you will switch off the Wizard and manually set all the necessary properties.

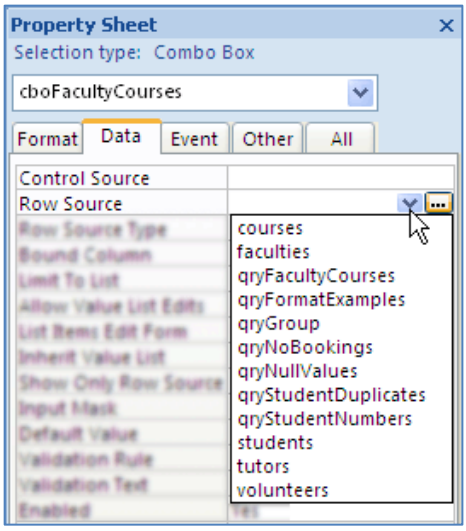
To turn off the Wizard:

- Click the **Use Control Wizards** button from the **Controls** group on the **Design** tab



- On form **frmComboSync**, insert a new **Combo Box** below the Faculty Unbound control

- Set the **Name** property to **cboFacultyCourses**
- Select the **Data** tab on the Property Sheet
- Click the **Row Source** property and click the down arrow



- Select **qryFacultyCourses**

This sets the query as the source of the list of values for the Combo Box.

- Ensure that the **Row Source Type** property is **Table/Query**
- Ensure that the **Bound Column** property is **1**

The query will display two fields (columns): CourseID and CourseTitle. The Bound Column property specifies which of these two values will be taken to be the value of the control.

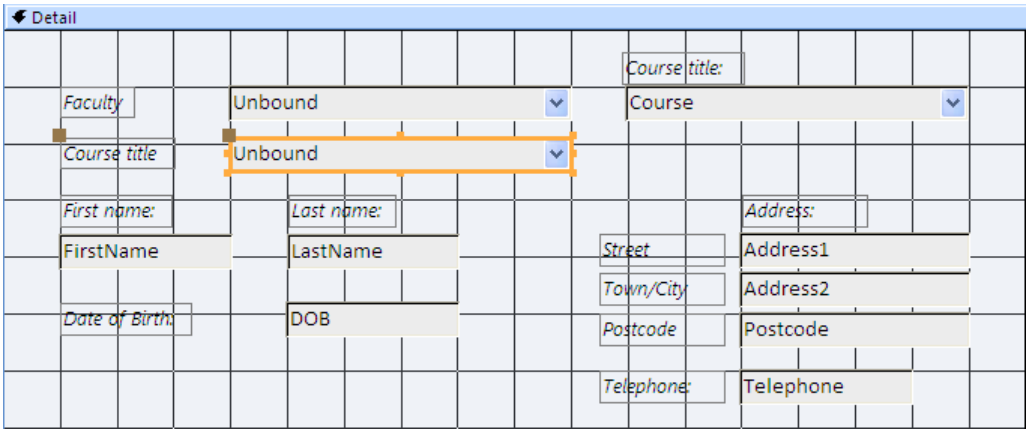
This is not to be confused with the value to be displayed. You will set this next.

- Select the **Format** tab on the Property Sheet
- Set the **Column Count** property to **2**, since there will be two fields in the query result
- Set the **Column Widths** property to **0cm;4cm**

The 0cm for the first column means that the CourseID value is not to be displayed. The 4cm value for the second column should be sufficient to display each course title fully. If this value is not quite sufficient then it can be adjusted later.

- Change the text on the associated label to **Course title**

- Reposition and resize the control as shown below



- Save the form
- Display the form in Form View
- Try choosing a faculty using the first combo box, then use the second control to choose a course

Notice that there is a slight problem with the second control. It works as expected when there are no values in the boxes to begin with, but once you make a selection, if you then change the faculty it has no effect on the second combo box.

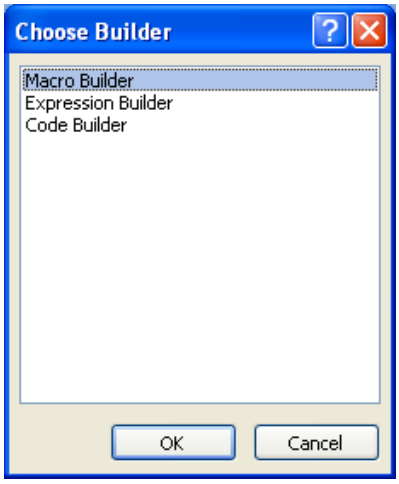
To correct this you need to use a macro to instruct the application to update the courses combo box whenever the faculty combo box is updated.

- Close the form and reopen it in Design View
- Select **cboFaculties** (the Faculty Unbound Combo Box)
- Select the **Event** tab on the Property Sheet
- Click the **After Update** event property

Any macro embedded here will be executed whenever the value in the cboFaculties control changes.

- Click the **Launch Builder** button

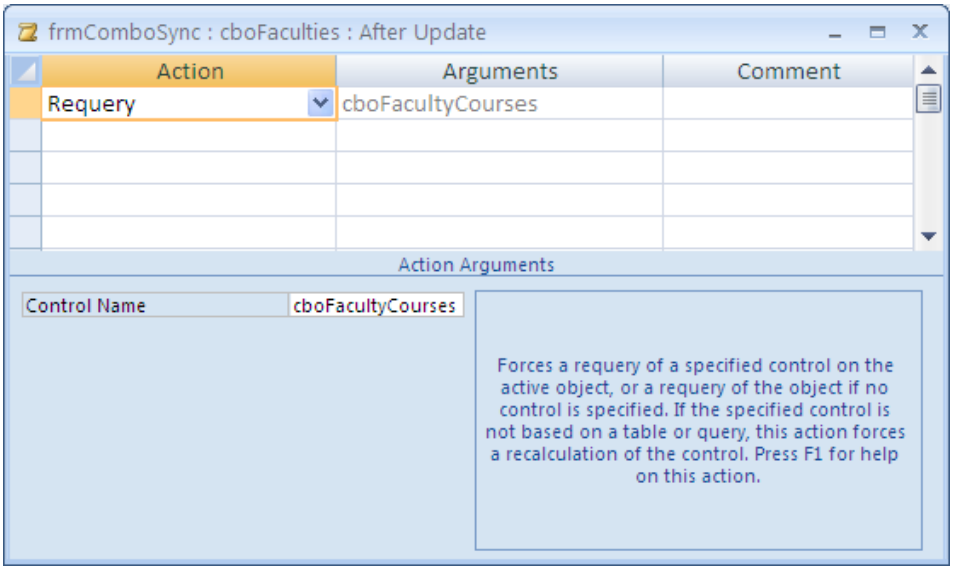
No Macro, Visual Basic Code, or Expression is linked to this event, so the Choose Builder dialog box is displayed.



- Select **Macro Builder** and click **OK**
- Click the down arrow in the Action box of the first row
- Scroll through the list of Actions and select **Requery**

The Requery Action is used to update the data in a specified control by requerying the source of the control.

- In the **Control Name** Argument type **cboFacultyCourses**



- Save and Close the Macro Builder
- Save the form
- Display the form in Form View
- Verify that the two combo boxes are now synchronised correctly

The two controls work correctly, but they have absolutely no effect on the student record displayed on the form. To ensure that the course selected is accepted as the course for the current record, you need to create one further macro.

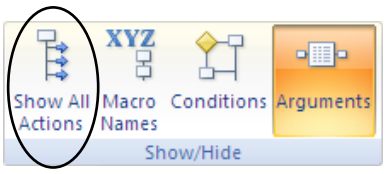
- Close the form and reopen it in Design View
- Select **cboFacultyCourses** (the new Course title combo box)
- Ensure that the **Event** tab on the Property Sheet is selected
- Launch the **Macro Builder** for the **After Update** event

The Action that you need to use for this macro is called **SetValue**. This Action is used to set the value of a field, control or property on a form or report.

You will find, however, that the SetValue Action is not available in the list by default. Access regards some Actions, including SetValue, as having a higher security risk and therefore does not display them in the default list.

To see all the available Actions:

- Click **Show All Actions** from the **Show/Hide** group on the **Design** tab



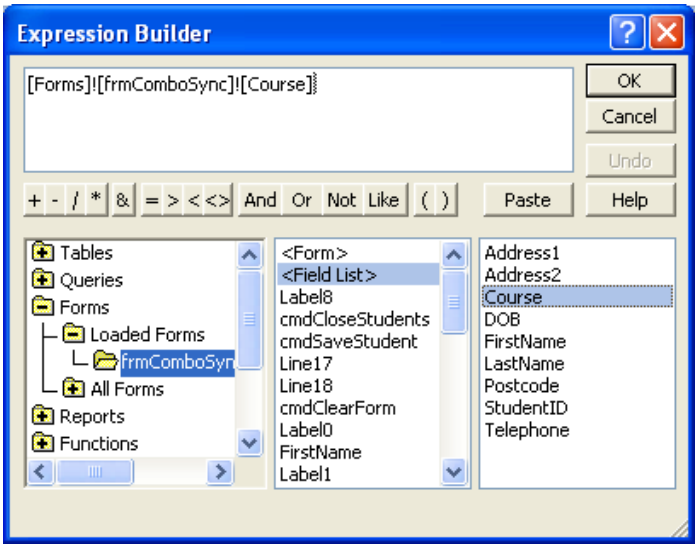
- In the first row of the Macro Builder, display the list of Actions and select **SetValue**

An exclamation mark appears on the left of the row. This is to warn you that the selected Action will only be allowed to run in 'Trusted Applications'.

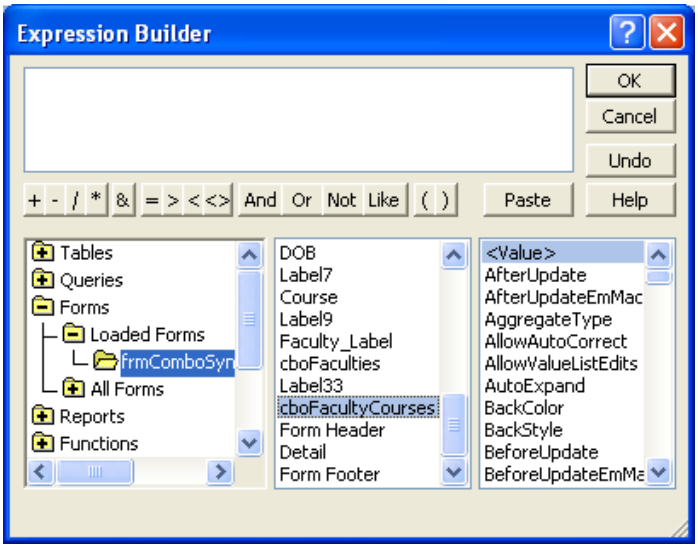
Two Arguments are available for the SetValue Action: Item and Expression. Item refers to the object whose value is to be changed, and Expression refers to the value to be set. You want to set the Course field to be equal to the value in cboFacultyCourses.

- Click in the **Item** Argument
- Click the **Launch Builder** button
- Expand the **Forms** and **Loaded Forms** category and sub-category and select **frmComboSync**
- Select **<Field List>** from the options in the second column
- Select **Course** from the list of fields in the third column

- Click **Paste**



- Click **OK** to accept this expression as the Item whose value is to be set
- Click in the **Expression** Argument
- Launch the **Expression Builder**
- Use the Builder to select the **Value** of the Combo Box **cboFacultycourses** on the open form **frmComboSync**



- Paste the expression in the Expression Box
- Click **OK**
- Save and close the Macro Builder
- Save the form

- Display the form in Form View
- Select a faculty and a course using the two new combo boxes

You should now find that the selected course is now also displayed in the control that is bound to the Course field (the original combo box). This verifies that the SetValue Action is working correctly. The two new combo boxes can now completely replace the original control.

- Close the form and open it in Design View
- Delete the original **Course title** Combo Box bound to the **Course** field
- Edit the macro embedded in the **On Click** event of the **Close Form** command button so that the current form will close when the button is clicked
- Save the form
- Display the form in Form View
- Use the Close Form button to close the form

COMPLETING THE USER INTERFACE

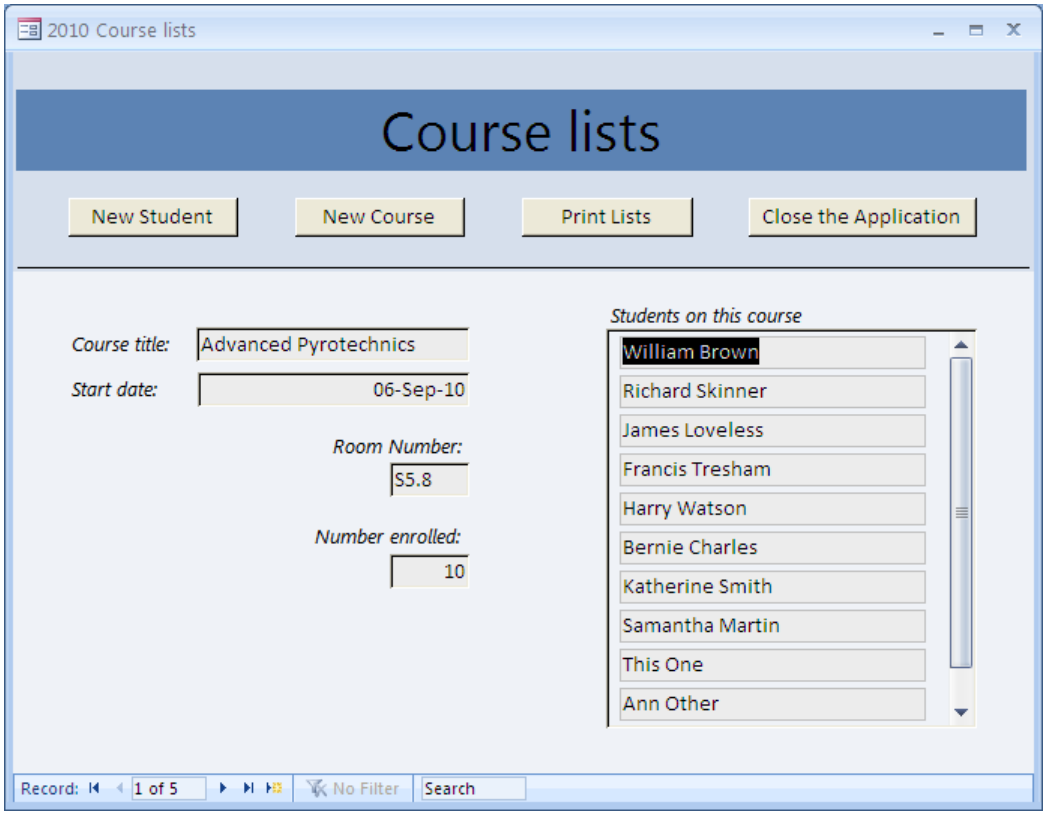
Usually a database application is organised in such a way that users never need to use the Navigation Pane or the commands on the ribbon in order to accomplish the standard tasks for which the database was designed.

The application will usually have a main form which opens automatically when the database is launched. If the database is complex, the main form may not display any data but act solely as a switchboard, with Command Buttons which when clicked will open other forms or reports.

For this database you will use the form **frmCourseLists** as the main form. You will first add four buttons to the form to enable users to input new students and courses, print a list of students in course groups, and close the application.

- Open **frmCourseLists** in Design View
- Delete the **Close Form** button
- Drag the Detail bar downward to create enough space in the header for the Command Buttons to be positioned below the title
- Drag the horizontal line down to the bottom of the header

Next you are going to create the Command Buttons as shown in the diagram below.



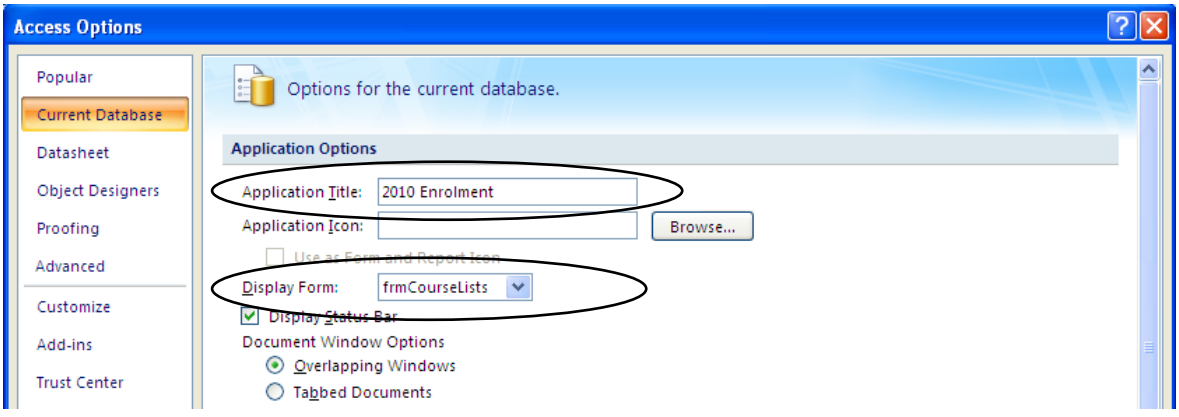
- Ensure that the **Use Control Wizards** button is selected
- Using the Wizard to select the appropriate object to be opened etc, as listed in the table below, create the following Command Buttons

Command Button	*Action when clicked*
New Student	Open the form **frmComboSync**
New Course	Open the form **frmCourses**
Print Lists	Print the report **rptAttendance**
Close the Application	Close the database application

- Minimise the Navigation Pane
- Minimise the Ribbon
- Save and close the form

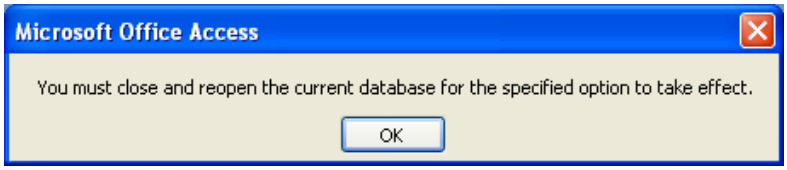
To give a name to the application and configure the database to launch frmCourseLists on startup:

- Click the **Office Button**
- Click **Access Options**
- Select the **Current Database** category
- In the **Application Title** box, type **2010 Enrolment**
- In the **Display Form** box, select **frmCourseLists**



- Click **OK**

The following message is displayed.



- Click **OK**
- Save the database on the desktop so that the file can be accessed easily

To save the database on the desktop:

- Click the Office Button
- Point to **Save As**
- Select **Access 2007 Database**
- Select **Desktop** as the place to '**Save in**'
- Click Save

- Exit Access
- Double click the **2010 Students** file on the Desktop

The application is launched with the title **2010 Enrolment** on the name bar, and the form **Course Lists** opens automatically.

- Verify that each button works as expected and finish by using the **Close the Application** button to close Access

IMPORTING FROM OTHER APPLICATIONS

If you have data already saved in a suitable format in a database, a spreadsheet, or a text file, you can import it directly into a table. The table you import into may be a new or existing one.

If you import into a new table, Access will create fields to match the data you are importing, and also allow you to edit the Field Names and Data Types.

If you choose to import into an existing table, you must make sure that the field sizes are adequate to contain the data. If there is a conflict between the field size and number of characters in the data, the offending records will not be imported.

The usual method for importing data is through the use of the Import group on the External Data tab. However, in some situations, Copy and Paste is a viable alternative.

COPYING DATA FROM EXCEL

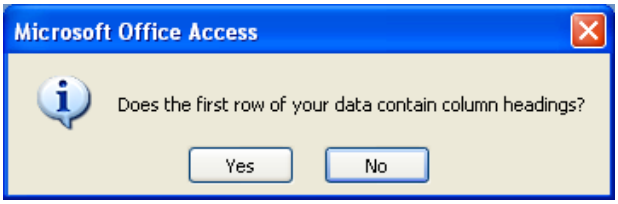
If you have data saved in a workbook you can use Copy and Paste (the Office Clipboard) to copy it directly into Access. A new table will be created and the data copied into place.

- Open the **enrolment** database
- Launch **Excel**
- Open the **PayRates** workbook that is saved in **C:\AccessIntermediate**

	A	B	C	D
1	**Course Leader**	**Module**	**Rate of Pay**	**Hours**
2	Guido Fawkes	Dangerous Occupations	£ 12.00	36
3	William Bligh	Tourism	£ 12.50	30
4	George Loveless	Politics	£ 15.00	24

- Select cells **A1** to **D4**
- Click the **Copy** button on the ribbon
- Switch to the **enrolment** database
- Click the **Paste** button on the ribbon

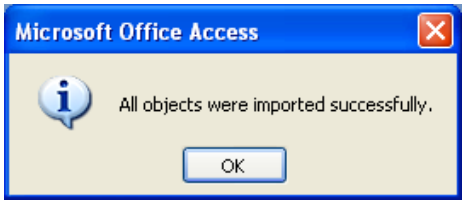
The following message is displayed:



As the first row does contain headings:

- Click **Yes**

The following message is displayed:



- Click **OK**

A new table named **Sheet1** is added in the Navigation pane.

- Rename the new table **payRates**
- Open the table and check that the information has been imported correctly

payRates

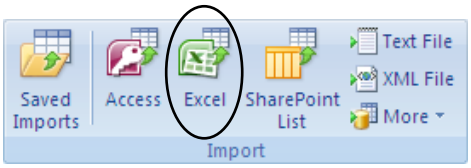
Course Leader	Module	Rate of Pay	Hours
Guido Fawkes	Dangerous Occupations	£12.00	36
William Bligh	Tourism	£12.50	30
George Loveless	Politics	£15.00	24

- Close the table
- Close Excel

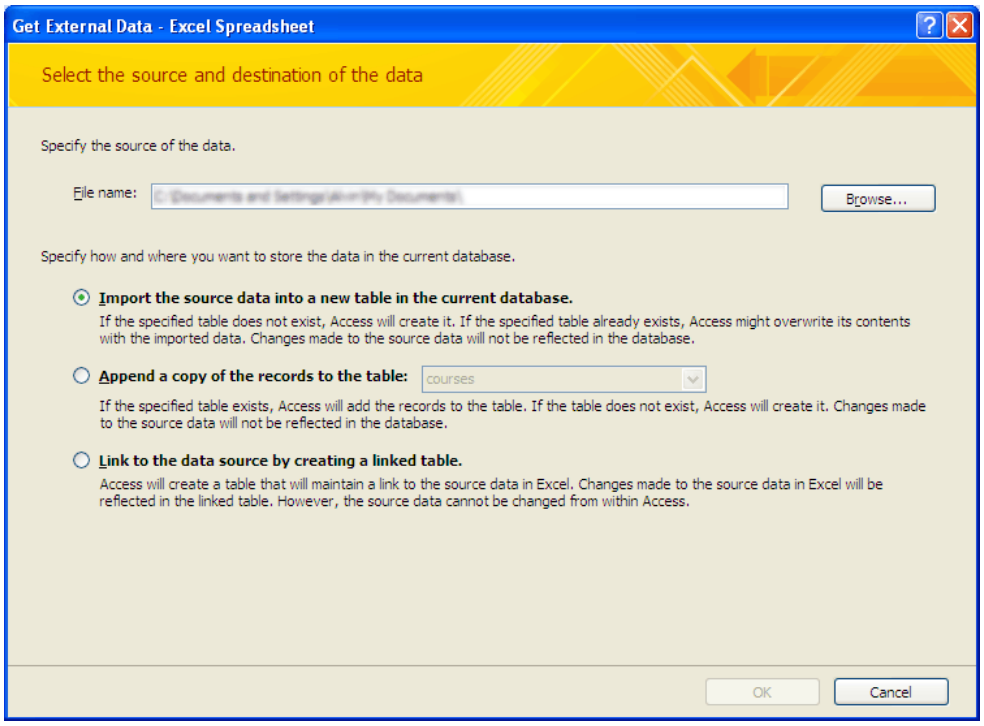
IMPORTING DATA FROM EXCEL

You are now going to use the import method to bring data from a spreadsheet into a new table.

- Ensure that the **enrolment** database is open
- Click **Excel** from the **Import** group on the **External Data** tab

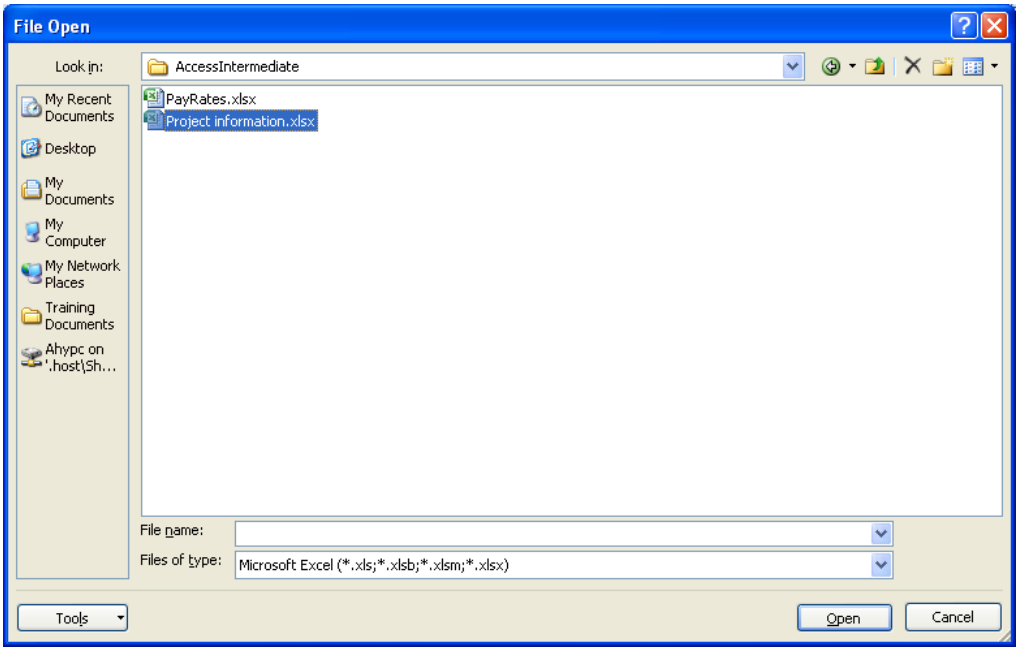


The Get External Data dialog box is displayed. From here you choose both the source and the destination of the data.



To select the Excel file **C:\AccessIntermediate\Project information** as the source:

- Click the **Browse** button
- Navigate to the **AccessIntermediate** folder on drive C
- Select the **Project information** spreadsheet



- Click **Open**

You now have three options regarding where and how the imported data should be stored: in a new standard database table, appended to an existing table, or in a new linked table. If you choose to create a linked table, changes in the spreadsheet will be reflected in the table.

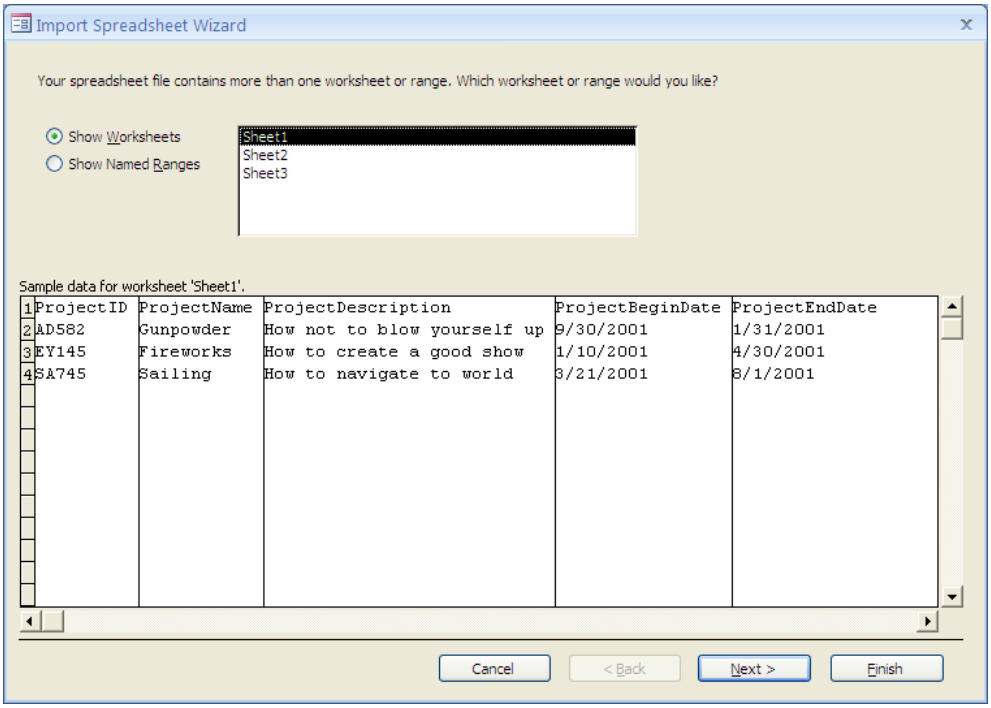
- Select **Import the source data into a new table in the current database**
- Click **OK**

The wizard begins.

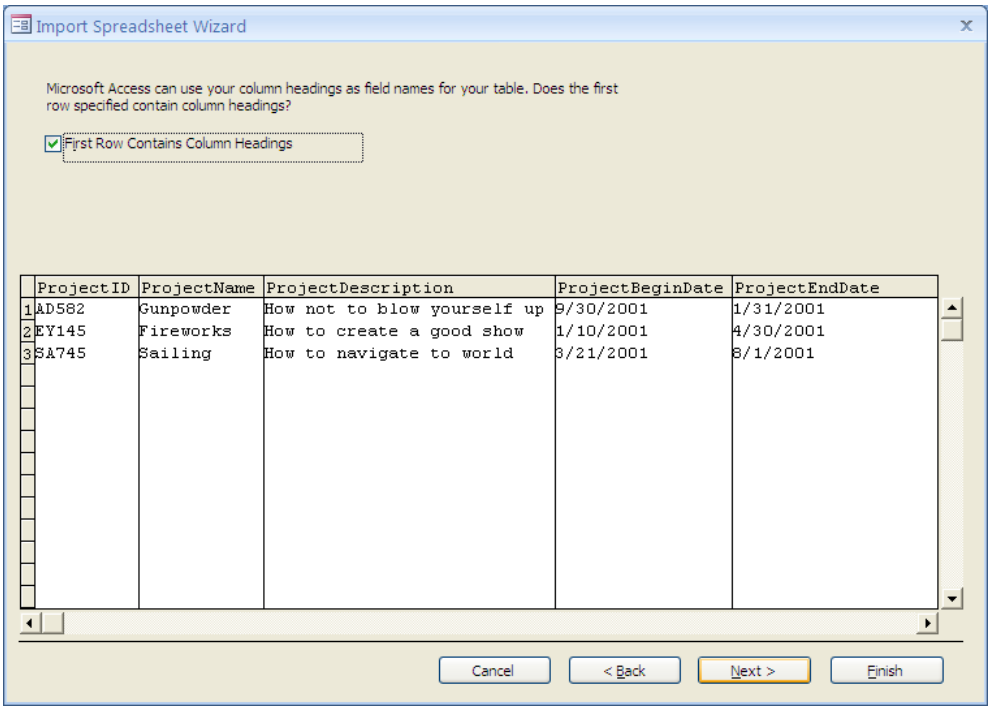
- Ensure that the **Show Worksheets** option is selected

Sheet1 is the only worksheet containing data.

- Ensure that **Sheet1** is selected

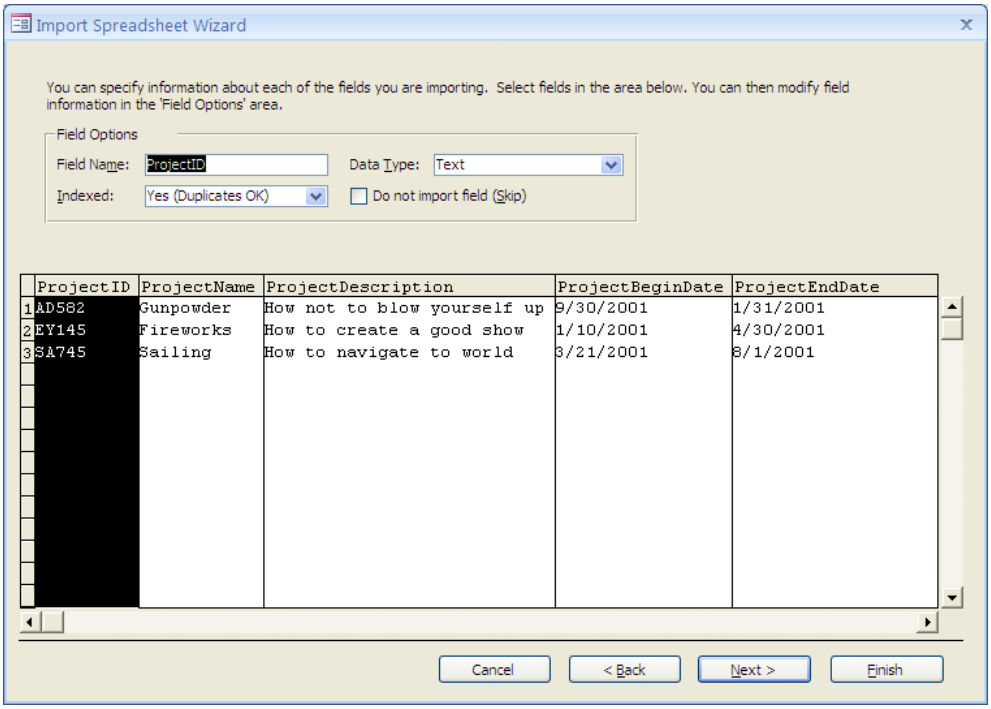


- Click **Next**
- Confirm that the first row of the data contains the column headings (these will be used as field names)



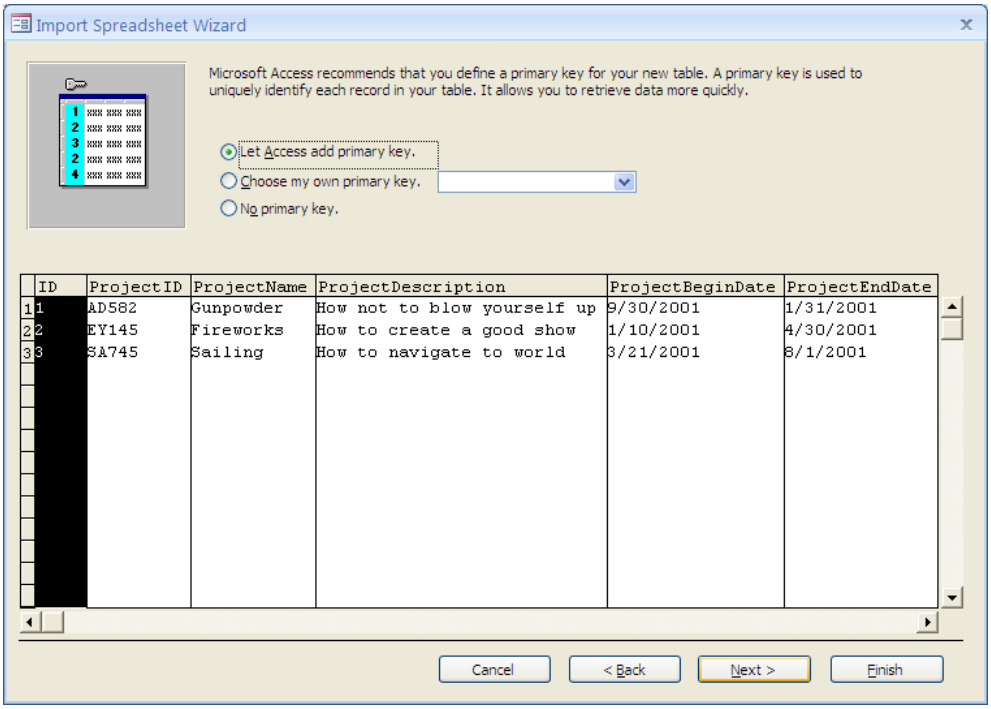
- Click **Next**

This step in the wizard enables you to select individual columns of data and edit the Field Name, Data Type and Indexed property. You are also able to stipulate that a particular field should not be imported.

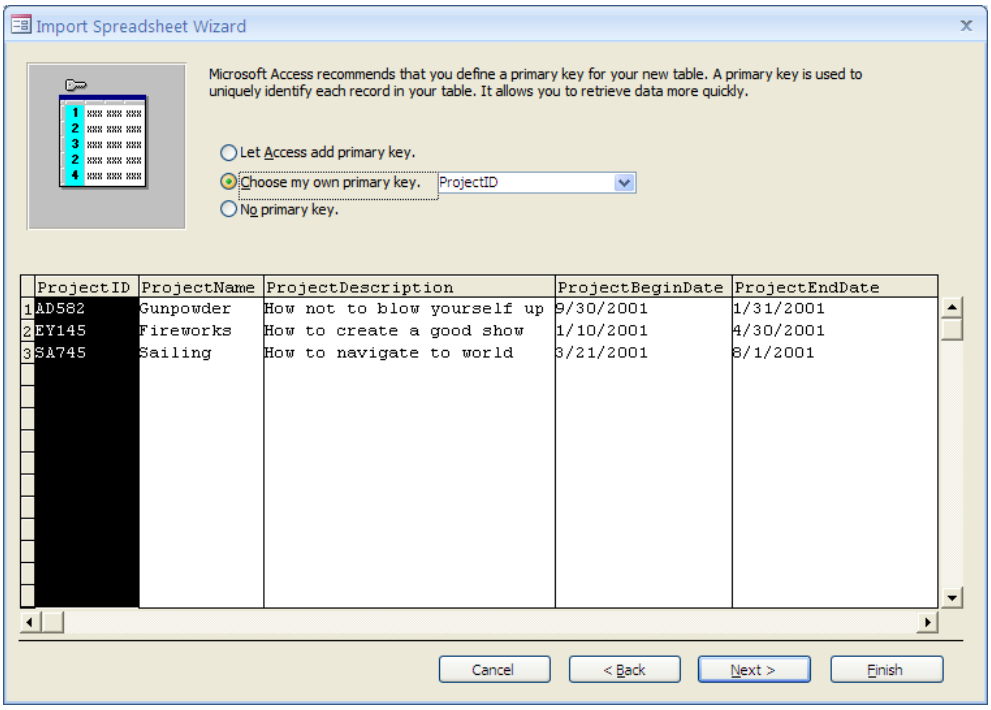


➢ Click **Next**

You need to decide if a primary key is required, and whether to let Access add the primary key.

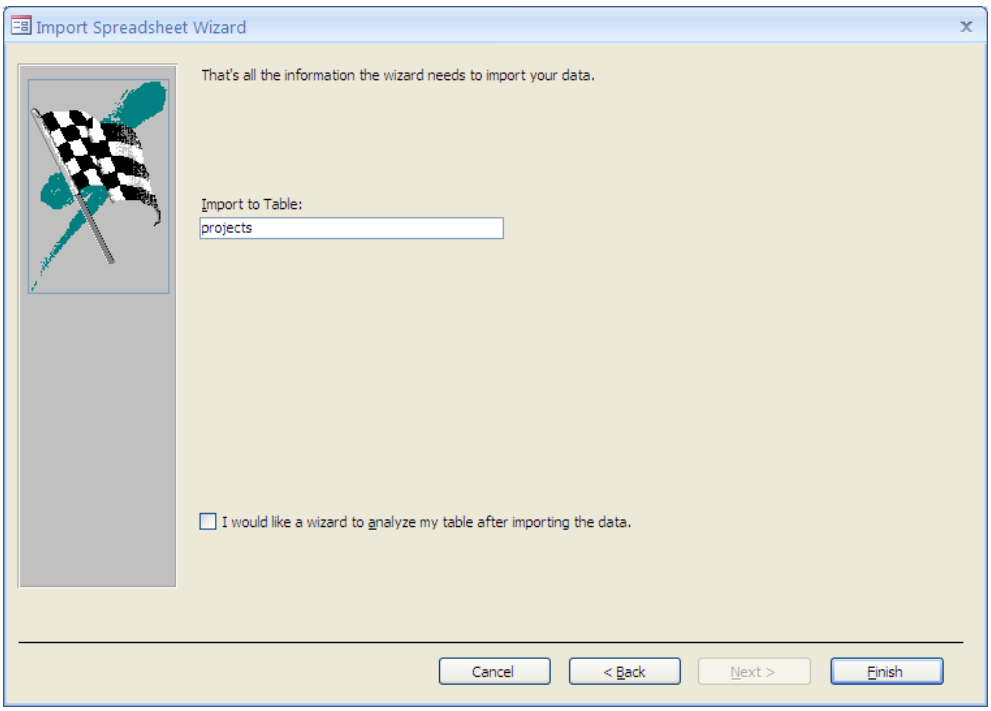


- Select **Choose my own primary key** and use **ProjectID** as the key



- Click **Next**

- Change the name of the new table to **projects**



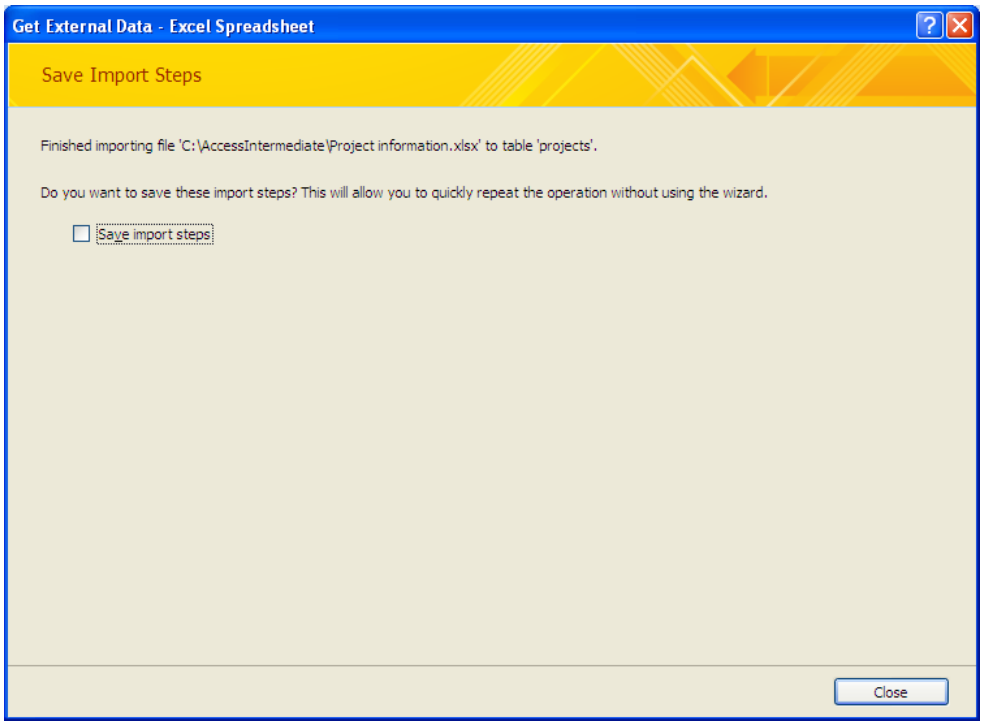
A check box option invites you to have the table analysed. If you choose to have it analysed, a wizard checks the table to determine if it needs to be split into two or more tables, to avoid repeating the same field value in different records.

It is not necessary to analyse this table.

- Click **Finish**

If you need to perform this same import on other occasions, then you would choose to **Save import steps**. You would then run the import, in future, by selecting it from the **Saved Imports** option.

For this exercise you will not save the import steps.



- Click **Close**

The projects table is added to the Navigation pane.

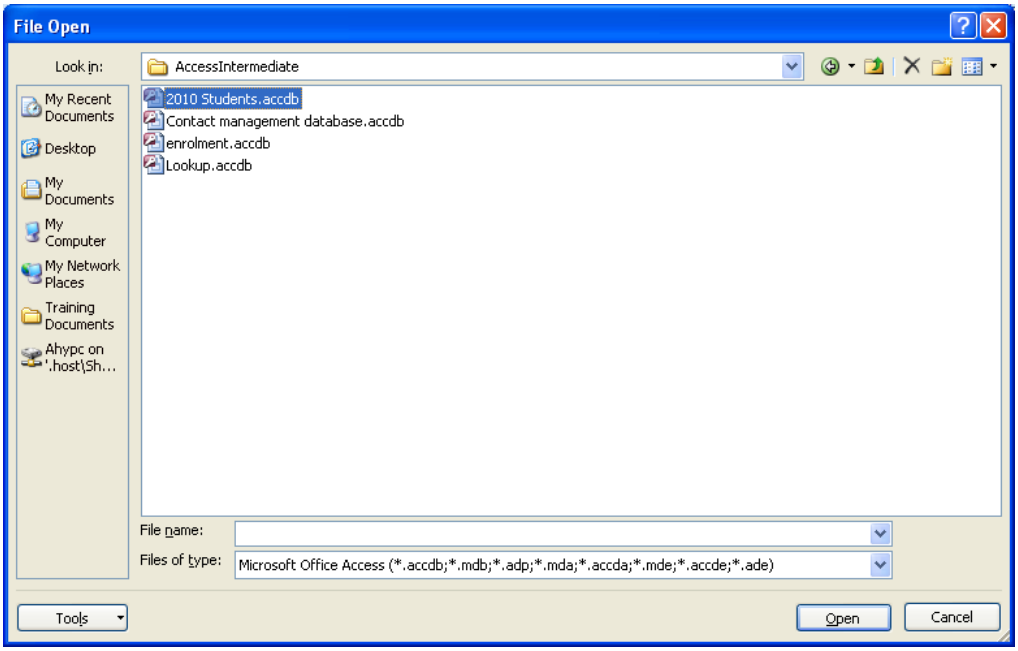
- Open the table to check that the structure and data are correct
- Close the table

IMPORTING A DATABASE OBJECT

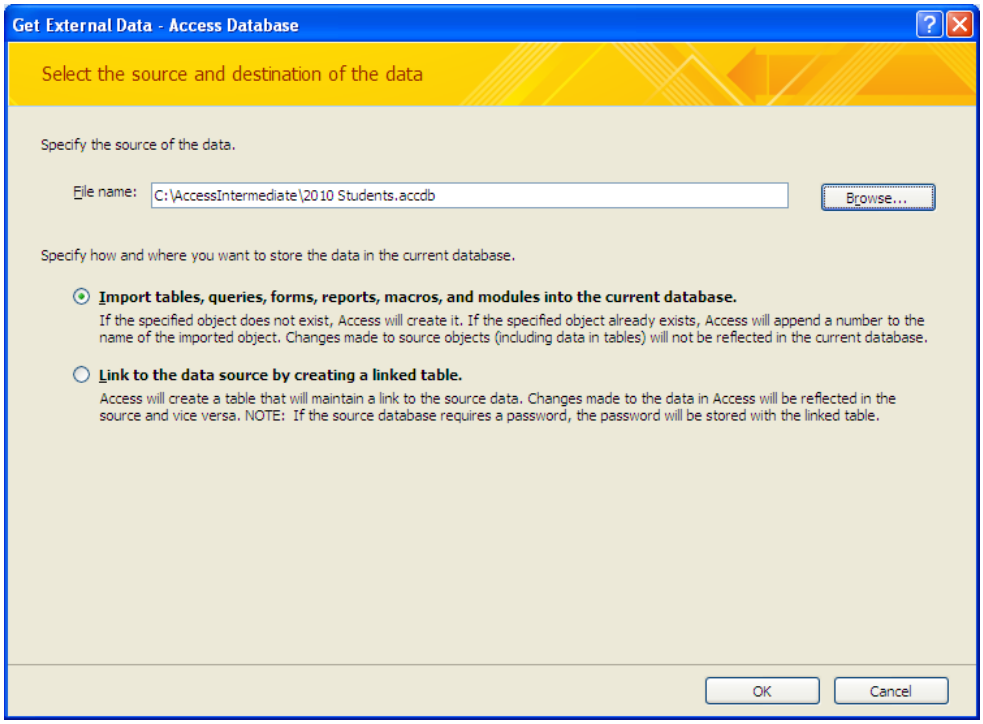
Tables, queries, forms, reports and macros can be imported from other databases, but if there are differences in field names, etc, imported objects may need to be modified before they can be used successfully in the current database.

You will import the form frmStudentEntry from the 2010 Students database into the enrolment database.

- Ensure that the **enrolment** database is open
- Click **Access** from the **Import** group on the **External Data** tab
- In the Get External Data dialog box, use the Browse button to navigate to the folder **C:\AccessIntermediate**



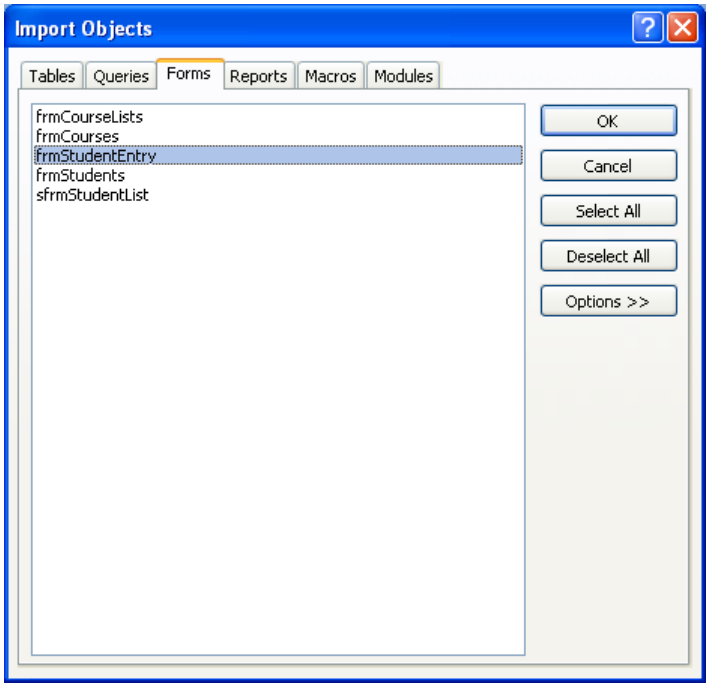
- Select **2010 Students** and click **Open**
- Ensure that **Import tables, queries, forms, reports, macros, and modules into the current database** is selected



- Click **OK**

The Import Objects dialog box is displayed.

- Select the **Forms** tab
- Click **frmStudentEntry** to select it



- Click **OK**

There is no need to save the import steps.

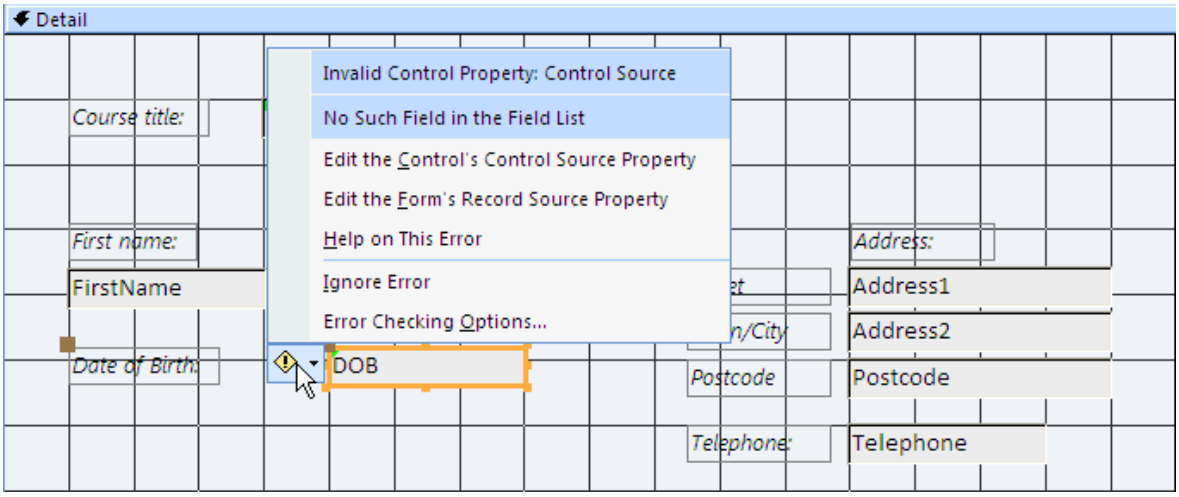
- Close the Get External Data dialog box
- Open **frmStudentEntry** in Design View

You will notice a green triangle in the top left corner of two controls: the **Course** combo box, and the **DOB** text box. This suggests that there is a problem with these controls.

- Click in the **DOB** text box

An Exclamation Mark icon is displayed on the left.

- Click on the Exclamation Mark to see information about the error and how it may be corrected



The text indicates that the control is trying to reference a non-existent field.

Sometimes it may be that a corresponding field exists in the current database but under a different name, and therefore this can be fixed by editing the Control Source property.

On this occasion, no date of birth field exists in the underlying table so, to correct the problem, either a new field will need to be created, or the control must be removed. However, you will not make any corrections to this form.

- Close the form
- Exit Access

APPENDIX A

FIELD PROPERTIES

Field properties are used to determine how Access handles and stores data, for example a date. When you create a field, a list of properties is displayed. The list of properties available is dependant on the data type chosen.

Field size

When a field has been specified as a text field, the field size can be set to any value from 1 to 255. This specifies the maximum number of characters that will be accepted in the field.

For number fields, the field size options are: Byte, Integer, Long Integer, Single, Double, Replication ID and Decimal.

Field size options for numbers	**Range**	**Decimal Places**	**Storage Size**
Byte	0 to 255	No decimal places	1 byte
Integer	-32,768 to 32,767	No decimal places	2 byte
Long Integer	-2,147,483,648 to 2,147,483,647	No decimal places	4 bytes
Single	-3.4E38 to 3.4E38	Up to 7	4 bytes
Double	-1.797E308 to 1.797E308	Up to 15	8 bytes
Replication ID	Unique ID required for replication	n/a	16 bytes
Decimal	-9.99E27 to 9.99E27	Up to 28	12 bytes

Format

The format chosen determines how data is displayed.

Numerical Formats		
	How data is entered	*How data is displayed*
General number	1234.56	1234.56
Currency	1234.56	£1,234.56
Fixed (2 places default)	1234.56	1234.56
Percent	0.1234	12.34%
Scientific	0.001234	1.234E-3

Date Formats		
	How you might enter data	*How data is displayed*
General Date	18/4/98	18/04/98 23:39:00
Long Date	18/4/98	18 April 1998
Medium Date	18/4/98	18-Apr-98
Short Date	18/4/98	18/04/98
Long Time	10.15	10:15:00 PM
Medium Time	10.15	10:15 PM
Short Time	10.15	10:15

Text Formats		
	How data is entered	*How data is displayed*
>	>uppercase	UPPERCASE
<	<LOWERCASE	lowercase

Decimal places

With numeric fields you can set the number of decimal places to between 0 and 28.

If you choose Auto, the number of decimal places will be allocated according to the field size.

Input Mask

Use an input mask to control what characters can be entered into a field. The input mask is made up of special codes and placeholders.

For example, an input mask of LL0 indicates that the field must contain two letters and one digit.

Input Mask codes	What it means
0	Enter a digit (compulsory)
9	Enter a digit (optional)
#	Digit, + or - sign or space
L	Letter (compulsory)
?	Letter (optional)
A	Letter or digit (compulsory)
a	Letter or digit (optional)
&	Any character or space (compulsory)
C	Any character or space (optional)
.,:;-/	Decimal point, date & time separators
<	Convert character(s) to lower case
>	Convert character(s) to upper case
!	Fill from right to left
\	Characters shown as literal

Caption

The caption property can be regarded as an alias for the field name. If a caption is defined, this will be used as the field heading when the table is displayed in Datasheet View. The caption will also be used to set the text displayed by default in labels for forms, reports and queries.

Default value

A default value is a value that is automatically entered into the field when a new record is created. This is useful if it is possible to predict what the value in the field is likely to be.

Validation rule

This property limits the values entered into the field.

For example, a validation rule of >1000 AND <2000 would ensure that only numbers between 1000 and 2000 would be accepted in the field.

Validation text

Used in conjunction with the validation rule. If the user attempts to input a value that violates the validation rule, then the validation text is displayed as a message to the user.

Required

Used on any field to indicate that a value must be entered.

Allow zero length

Used with a text or memo field. Access uses two different types of empty fields: the null field and the zero length field.

A field, for example 'fax number', might be empty because the person does not have a fax number, or because you do not know their fax number. Access allows you to distinguish between these two situations. You can use a null value to indicate that the information is not known, and a zero length value to indicate that the information does not exist.

To enter a zero length value in a field, type two inverted commas "" into the empty field.

Indexed

Used on any field to speed up data retrieval. Similar to an index in a book, if an indexed field is used in a search or query, Access will search the index, which is much quicker than searching the whole database.

However, indexing many fields will slow down data entry, because both the table and the indexes are updated. Use indexing wisely to get the best performance from your database.

Index setting	Meaning
NO	No index
YES (Duplicate OK)	Create an index on this field – different records are allowed to have the same value in this field
YES (No duplicates)	Create a unique index on this field - no duplicates allowed (useful for information such as a National Insurance number where there should be one only)

Unicode compression

This can be used in non-numeric fields to compress the data if fewer than 4,096 characters are stored in the field. It has no effect if the field contains more characters.

IME mode and IME sentence mode

Used to control the conversion of characters and sentences in East Asian versions of Windows.

Smart tags

Enables you to choose smart tags to attach to the field.

Append only

This is only available for memo and hyperlink data types. You can keep track of the history of changes to the value in a field by setting this property to Yes.

If the property is set to No, then any previously saved field value history will be deleted.

Text format

This property is only available for memo fields and enables you to choose between Rich Text (which allows a degree of formatting to the text) or Plain Text.

Text align

Enables you to choose the default alignment for the value within the field. The options are: General, Left, Centre, Right and Distribute.

Show date picker

This property enables you to choose whether or not to display a calendar control when users edit a date/time field value.

Note, however, that if an input mask is set for the date/time field, then the Show Date Picker property will be overridden and a calendar control will not be available.