



Microsoft Access Level II

Skills for the Electronic Workplace

Copyright © 2002 IST

Permission to use this document for non-commercial purposes, in original or modified form, is granted, provided that the original source of the document is acknowledged as Skills for the Electronic Workplace, Information Systems and Technology, University of Waterloo.

Contents

1.	Concepts and Terminology.....	2
1.1	Modeling our Conference Application	2
1.2	Designing the Tables.....	2
	1 st Normal	3
	2 nd and 3 rd Normal Form.....	3
2.	Creating An Electronic Database and Describing Your Data.....	3
2.1	The Conference Database	3
2.2	More about Table Definitions.....	4
2.2	Other Tables That May Be Related to Your Primary Table.....	5
2.3	Defining Relationships Between Tables.....	6
3.	Queries	7
3.1	Joining Tables.....	7
3.2	Summarizing Data	9
3.3	Parameter Queries	11
4.	Creating Forms (layouts) to Manipulate Data.....	12
4.1	Creating Basic Forms Using the Forms Wizard	12
4.2	Using Form Design View	13
4.3	Adding Command Buttons to Your Forms.....	14
4.4	A Brief Look at SubForms	15
5.	Reports	16
5.1	Using the Report Wizard to Generate Mailing Labels	16
5.2	Detail Reports.....	16
5.4	Modifying Report Layouts	17

Introduction

What may start as a simple collection of data that you need to manage, will frequently grow in size and complexity. The data may need to be shared with other people and combined with other data to produce more complex reports.

This course is a continuation of the Access Level I SEW course. It is intended for people who have attended the Level I course and have had an opportunity to work with MS Access for simple applications.

1. Concepts and Terminology

1.1 *Modeling our Conference Application*

As the person in charge of registrations for the Cybernetics Conference, you need to keep track of all the details related to this conference.

The first thing you will need to do is identify the “things” that you will require. These “things” are called **entities** in database language and we will use **tables** to contain all of the information about them. An entity could be a real world object or an event. Real world objects might be students, buildings, universities or instructors. Events might be baseball games, lectures or fee payments. The **tables** are the basic building blocks in a relational database system. An individual table can be compared to a spreadsheet, made up of rows and columns. In database terminology, the rows of data in a table are called **records**, and the columns are called **fields**. A record is the information held about one item or person. A record contains **fields** and a **field** is a single item of data, such as a Surname, First Name or Student ID Number. One of the fields in the table is normally used to uniquely identify a record in a database so that you can find a particular record quickly. Such a field is called the **primary key** and a typical example would be a Student ID Number.

In our Cybernetics Conference, we will attempt to “model” the real objects and events of our conference, the more accurate our model, the better the application will serve our needs.

In attempting to build our database model of the conference, one popular technique is to use simple statements to describe what is going to take place.

- ① **attendees** will **register** to attend the conference and will **pay** their conference fees
- ② **speakers** will deliver papers at **seminars**

1.2 *Designing the Tables*

In our attempt to model our application it seemed clear that, at least, we need a table to describe the **Attendees**, and a likely second table to describe the **Seminars**.

However, do we need a third table to identify the seminars that each person wants to attend? What fields should be contained in each table? How are the tables linked together?

The process of deciding what tables are required and what fields are contained in each table is described as the **normalization** of a relational database. This process is described in the following texts:

Database Design with Microsoft Access. Michael J. Hernandez. Pinnacle Publishing 1994.
An Introduction to Database Systems, Vol. 1 CJ Date. Addison Wesley
The Relational Model, Version 2. EF Codd. Addison Wesley.

Seven steps are described in this process. We will look at a few of the most important steps:

1st Normal

Eliminate repeating fields. For example, including several columns in the Attendees table to keep track of the various seminars would be a violation of this rule.

Attendee	Seminar1	Seminar2	Seminar3	Seminar4
John Smith	Database Design	Web Security	etc.	etc.

Where repeating fields are found, this usually means that we need to define a new table. We will call this table **SeminarRegistrations**.

Attendee	Seminar
John Smith	Database Design
John Smith	Web Security

2nd and 3rd Normal Form

Eliminate fields that do not directly “describe” and depend on the thing (entity) you are tracking. To check for this condition, suppose you changed the value of one of the fields in a table. Would this result in a change to another field?

Consider the following example. Suppose that, for each attendee, we included the Institution’s name and address.

Attendee	Institution	Institution Address
John Smith	University of Waterloo	200 University Ave. W.

The institution address describes the institution rather than describing the attendee, a violation of the second normal form.

This might lead us to define a new table called Institutions. Whether or not we actually need this separate table will depend on our application. If most people are coming from different institutions, it is probably easier to knowingly violate this normalization rule in this case. Every table you add to your application adds a certain amount of complexity.

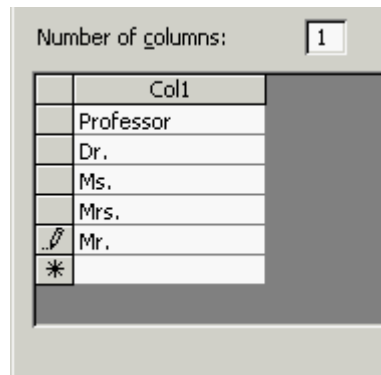
2. Creating An Electronic Database and Describing Your Data

2.1 The Conference Database

1. Launch MS Access using the Start menu.
2. Open the database called **conference.mdb** (you will find this on the M: drive under the directory for your workstation, e.g., M:|sew#p###).

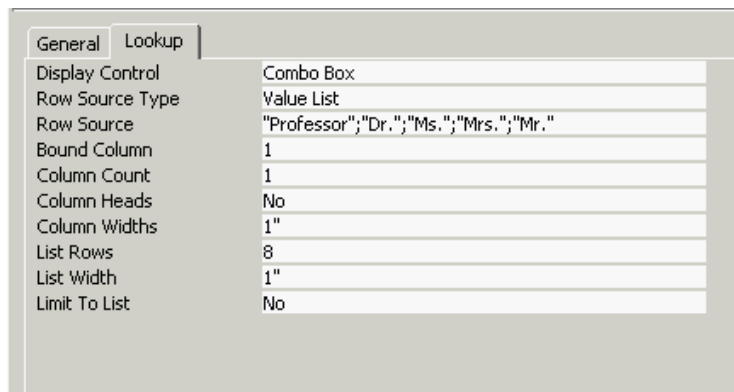
2.2 More about Table Definitions

1. With the Tables tab selected, open the Attendees table. Recall that this was the table we built during the MS Access Level I course. It contains the fields needed to describe each attendee.
2. In some cases, we may want to restrict a field to a predefined set of values. The **Title** field is an example of this. It should only contain one of the following: Professor, Dr., Ms. or Mr. Switch back to Design View and move the cursor into the Data Type portion of the **Title** field. Click once with the left mouse button for the Data Type drop down list and choose Lookup Wizard. The Lookup Wizard is used to predefine the acceptable values. Select the option **I will type in the values that I want** and press **Next**. Leave the number of columns set to 1.

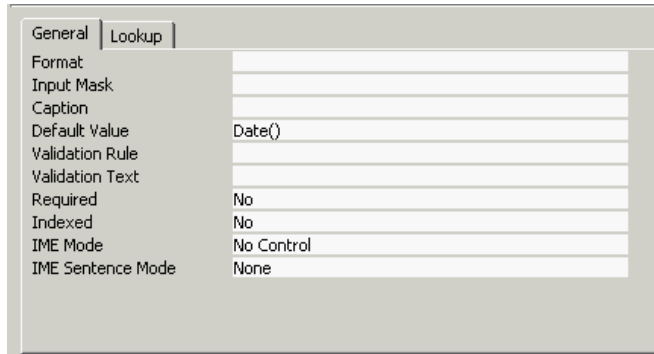


Press **Next** when you are finished, leave **Title** as the label and press **Finish**.

3. Press the Lookup tab in the bottom section of the window. Notice the values in the Row Source column that have been created by the Wizard. Notice that the **Limit to List** is set to **No**. Change this to **Yes** to ensure that only proper values are entered into the table.



4. Let's add one more field named **DateRegistered** with a data type of **Date/Time**. We will want to automatically set the DateRegistered field to today's date. To do this, press the **General** tab and enter the **Date()** function in the **Default Value** section for this field.



5. Let's make sure that our database will handle the year 2000. Minimize your MS Access application and from the **Start** menu, select **Settings** → **Control Panel** → **Regional Settings**, click the **Date** tab. Change the Short date style to **mm/dd/yyyy**. This will specify that dates are to be entered and displayed in month/day/4digitYear format.
6. All finished! From the **File** menu, select **Save** and then close the table Design View window.

2.2 Other Tables That May Be Related to Your Primary Table

1. In addition to the Attendees table, we will also need a table called **Seminars** to describe the time, location and speaker for each presentation. This table has already been created for you. Select the table and press **Design** to examine the description.

Seminars : Table		
	Field Name	Data Type
	SeminarID	AutoNumber
	Title	Text
	Abstract	Memo
	AttendeeID	Number
	Time	Date/Time
	Location	Text

Note that SeminarID is the **primary key** to the table. Note also the use of a memo field for the abstract. Use Help (F1) to find out how long a memo field may be. The AttendeeID field is identifying the Attendee (from the Attendees table) who will be giving the talk.

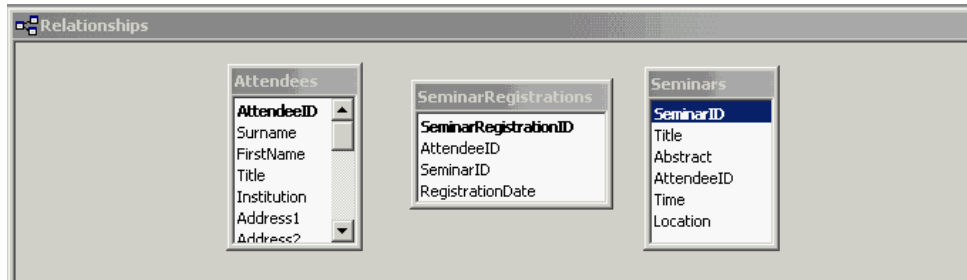
2. Check out the design of this table to ensure that it meets the 3rd normal form criteria. Suppose someone suggested adding the speaker's name to the table. Which rule would this be breaking?
3. Our last table is called **SeminarRegistrations**. It has also been defined for you. This table will contain one record for every seminar registration for each attendee. Note the links to both the Attendee and Seminars tables.

SeminarRegistrations : Table		
	Field Name	Data Type
	SeminarRegistrationID	AutoNumber
	AttendeeID	Number
	SeminarID	Number
	RegistrationDate	Date/Time

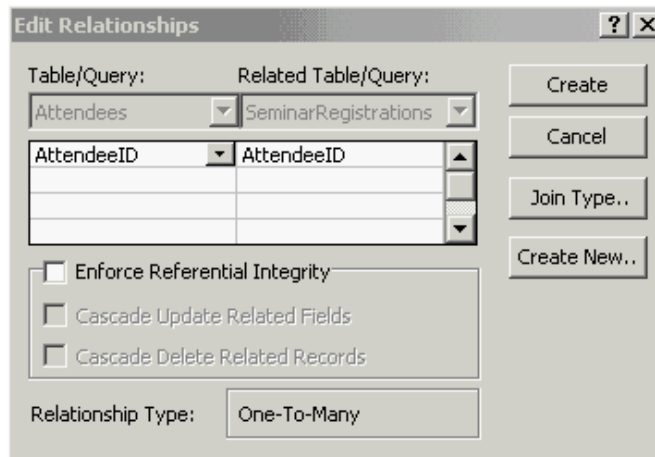
2.3 Defining Relationships Between Tables

One of the powerful features of a relational database is the ability to join information that is contained in more than one table.

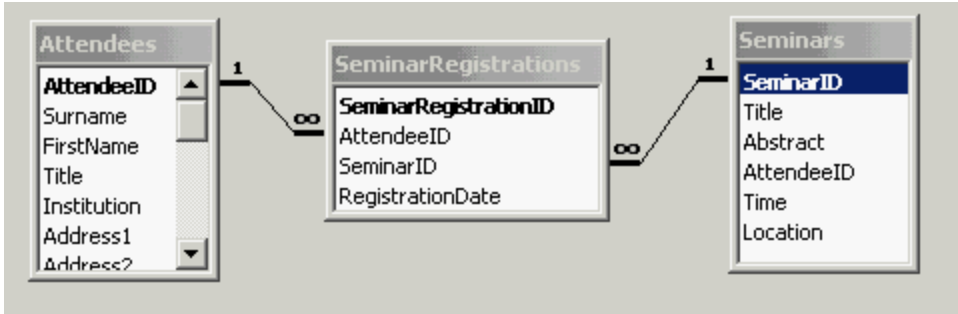
1. We need to define how the three tables are linked. From the **Tools** menu, select **Relationships**. Press the right mouse button and select the **Show Table** entry. Add the **Attendees**, **Seminars** and **SeminarRegistrations** tables and close the Show Table window. Position the three tables as illustrated below:



2. Now, select **AttendeeID** from **Attendees** and drag it to the **AttendeeID** in the **SeminarRegistrations** table. A Relationships Window will appear confirming that the AttendeeID field will be used to link the records of the two tables. Click on **Enforce Referential Integrity**. This will ensure that the SeminarRegistrations table contains valid AttendeeIDs as defined by the Attendees table.



3. Notice the **One-To-Many** relationship between the two tables. This is stating that any one record in the Attendees table may match many records in the SeminarRegistrations table (i.e., one person may attend many talks). However, a specific record in the SeminarRegistrations table may only match one record in the Attendees table.
4. Click **Create** to finish the creation of the relationship.
5. Now drag **SeminarID** from the **Seminars** table into **SeminarID** of the **SeminarRegistrations** table. As above, enforce Reference Integrity to ensure that only valid SeminarIDs are entered into the SeminarRegistrations table.



Close the Relationships Window saving your definitions.


3. Queries

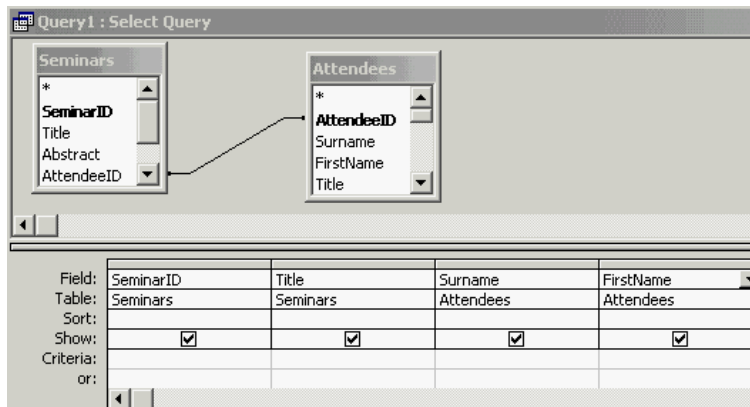
While tables provide the basic means of storing data, queries allow you to easily retrieve data, join information contained in different tables, update and even create new table definitions.

3.1 Joining Tables

One of the powerful features of queries is the ability to join information that is contained in more than one table. The Seminars table contains a field called AttendeeID. This field identifies the Attendee who is giving the talk. All of the information about the speaker is contained in the Attendee table.

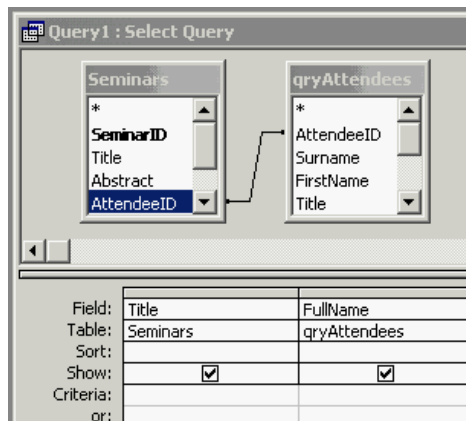
1. Supposed we wanted to produce a list of Seminars. Create a new query based on the Seminars tables. (Click the Queries button, click New button, Design View, Seminars table). Include the **SeminarID**, **Title** and **AttendeeID**. Remember that AttendeeID identifies the person who will be giving the talk.
2. We would like to include the name of the speaker rather than the AttendeeID number.

From Design View, press the show table button  and include the Attendees table as well. Drag the **Surname** and **FirstName** fields from the **Attendees** table into your query.

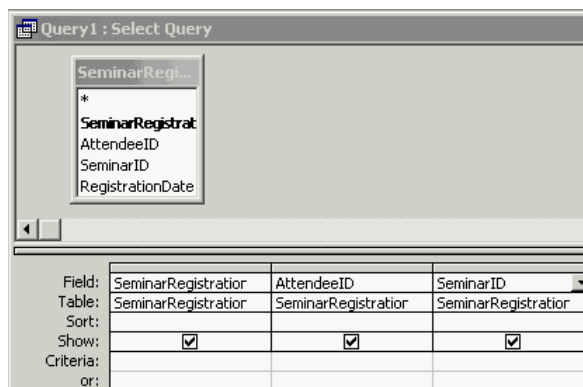


3. Execute the query and examine the results. Notice the connecting line between the two tables. The query matches up records from the two tables using the AttendeeID field to allow us to associate the name of each speaker with each session.

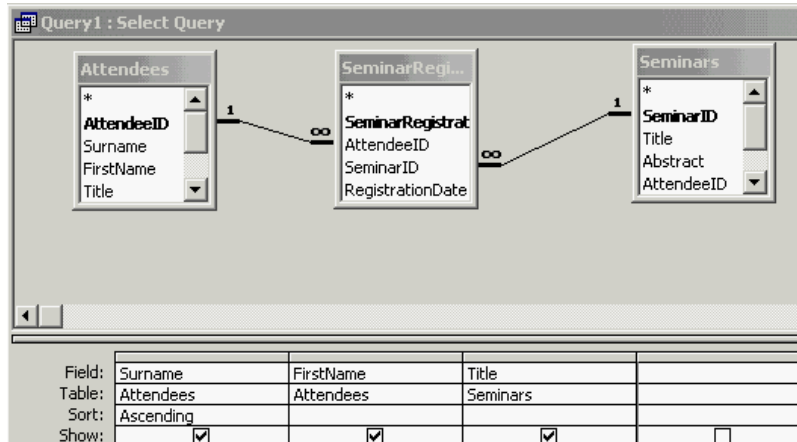
4. Take a closer look at the results of your query. **Why are Seminars 6 and 12 missing?** Notice (from above) that speakers have not been identified for these two seminars.
5. Go back to Design view of your query and **double-click the line joining the two tables**. Notice that we are only including rows “Where the joined fields from both tables are equal”. Select the option specifying that **ALL records of the Seminars table** are included, even if the AttendeeID field is null. Run this query again.
6. In a small sample of data, it’s possible to visually check for missing data. But suppose you were asked to produce a report showing only the seminars that do NOT have a speaker. Do this by including the AttendeeID field in your query and specifying “null” in the “Criteria” section of this column. Close your query, no need to save.
7. Queries may be based on other queries as well as tables. Create a new query based on the Seminars table and the query called qryAttendee (click the Queries tab in the “Show Tables” window to display the list of available queries).
 - Join **Seminars** to **qryAttendees** by dragging AttendeeID from Seminars to AttendeeID of qryAttendees
 - Select the fields Title from Seminars and FullName from qryAttendees to obtain a list of speakers for each seminar. Experiment with your results. Not need to save this query.



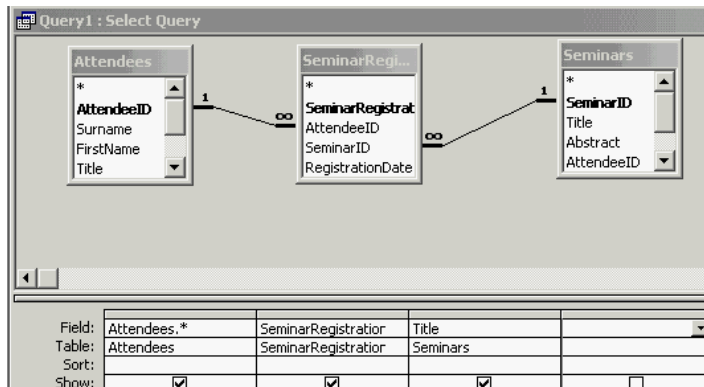
8. Suppose we wanted to identify the seminars that each person is attending. This information is contained in the SeminarRegistrations table. Create a new query based on the SeminarRegistrations table and include the following fields.



9. This is telling us that AttendeeID 1 (James Smith) will be attending seminars 1-6. Not very useful! Let's go back to design view and add the Attendees table (to pick up the person's name) and the Seminars table (to pick up the title of the seminar).
10. Position the three tables as illustrated below. Remove the link between the AttendeeID of the Attendees table and the AttendeeID of the Seminars table. We don't need or want this relationship for this query. **Sort** the output in ascending sequence by **Surname**.




11. Modify the query to include **all** of the fields from the **Attendees** and **SeminarRegistrations** tables and the **Title** from the **Seminars** table.

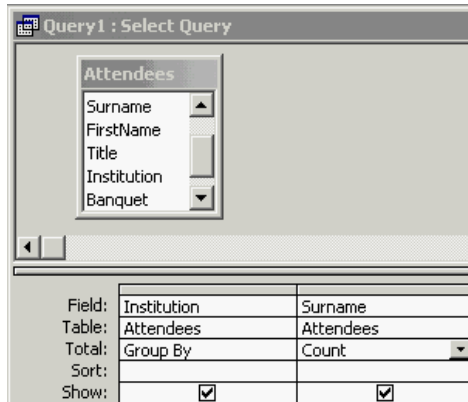


Save the query as **qrySeminarRegistrations**. This query will form the basis for a form.

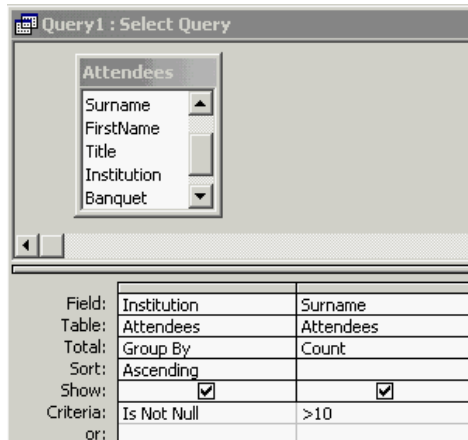
3.2 Summarizing Data

Frequently, you will need to produce summaries of data. Summaries may include counts of the number of records, totals, averages, the maximum and minimum, or other statistical functions.

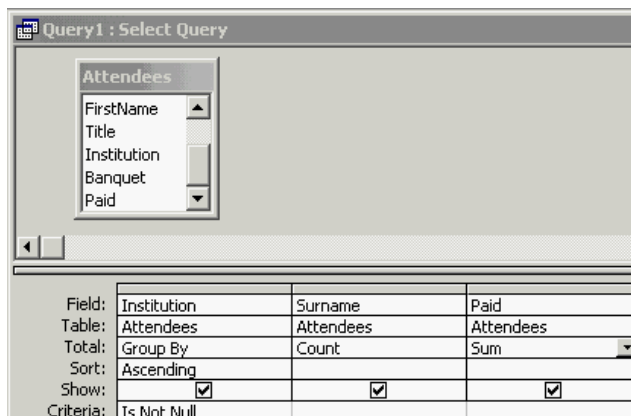
1. Create a query based on the **Attendees** table. Press the Totals button  to request a summary query. First, let's show the number of registrations per institution. This would require a summary query with the records Grouped By Institution. Add **Institution** and **Surname** to your query by dragging the fields from the Attendees table onto the query design grid. Move to the Total Row under Surname and click once with the left mouse button to access the drop down menu. Select **Count**. Run the query.



- Notice that some of the entries do not have institution filled in. Exclude these entries by adding a criteria of **Is Not Null** to the Institution column. Change the query to sort the output by Institution.
- Change the query to only show the institutions who are sending more than 10 people (**>10** in the Criteria row of **Surname**). Note that in a summary query, the criterion is being applied to the count of surnames, not to the surname itself. Test your changes and then remove this criteria before doing the next step.



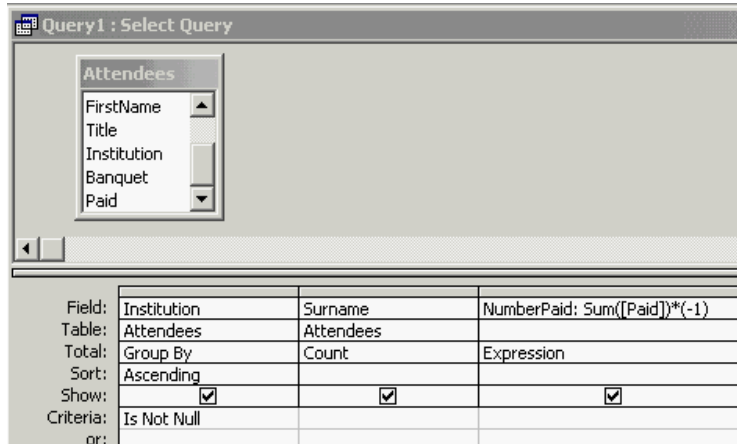
- Change the query to also show the number of people who have paid. Remember that the Paid column was defined as a Yes/No field. We will make use of the fact that a Yes value is stored as a -1.



- Rather than displaying negative values, enter in the expression in a new column in the Field row:

NumberPaid: Sum([Paid])*(-1)

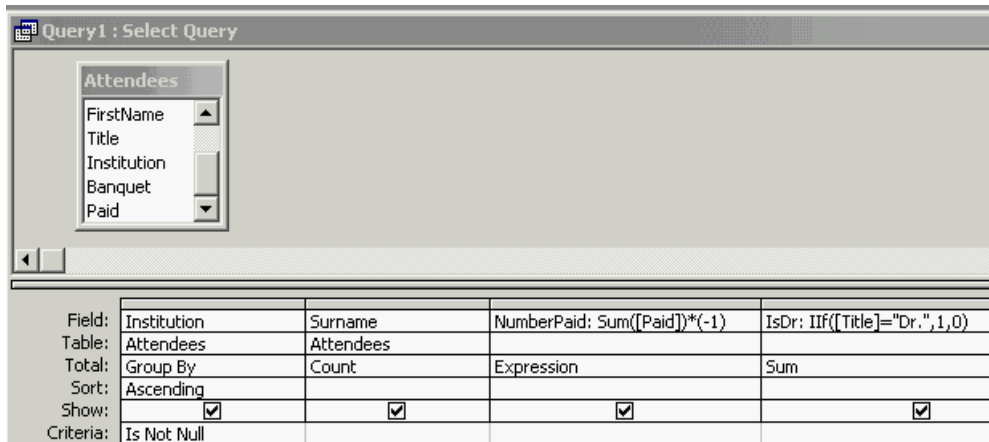
Change Group By to **Expression** in the Total row.



- Change the query to also show the number of Drs. attending from each institution. Do this by using the **iif** function to return a value of 1 if the Title = Dr and zero otherwise.

IsDR: iif([Title] = "Dr.", 1, 0)

Then use the **Sum** function in the **Total** row.



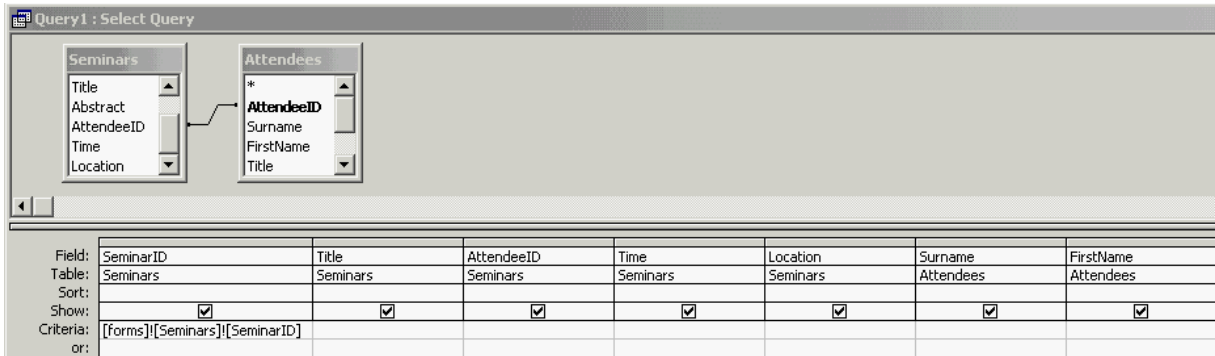
Close the query **without** saving your changes from the previous step.

3.3 Parameter Queries

Parameter queries allow you to tailor the way a query is run without needing to change the design of the query. You will find this an essential building block of forms.

- Create a new query based on Seminars table joined with Attendees. Include the fields from the **Seminars** table: **SeminarID**, **Title**, **AttendeeID**, **Time** and **Location**. From the **Attendees** table add **Surname** and **FirstName**.

Enter a criteria of **[Forms]![Seminars]![SeminarID]** and test your query. When you are prompted for a value, enter a seminar number, for example, the number 3.







2. Close the query giving it the name **SelectedSeminar**.

4. Creating Forms (layouts) to Manipulate Data

4.1 Creating Basic Forms Using the Forms Wizard



While you can input data directly into a table by using Datasheet View, Forms provide a number of features to enhance data entry. The simplest way to create a form is to use the built-in wizards.

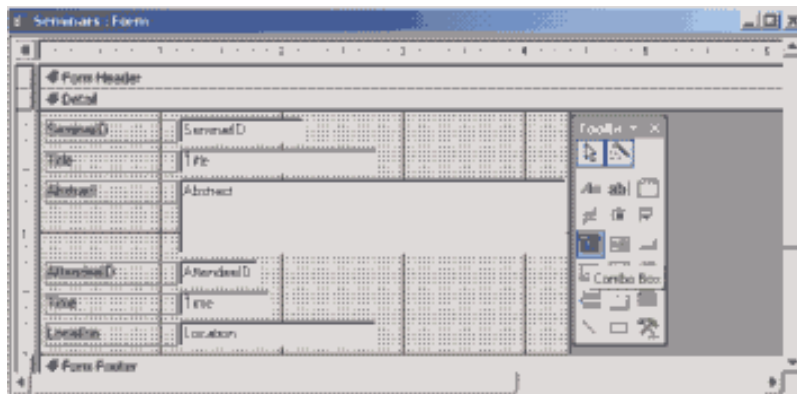
1. Make sure the **Forms** button is selected and then click on **New**. Choose **Form Wizard** and then choose **Seminars** from the pull-down list of queries and tables. Click on **OK**.
2. We will want to transfer all of the fields in the **Available Fields** list, so click on the  and then select **Next**.
3. Select **Columnar** for the layout type and click on **Next**.
4. Now choose a style for your form. Click on each selection to preview each of the styles. Make a choice and click **Next**. Click on **Finish**. Your form should look similar to the one below:

- Open the form and experiment with the various records in our collection. Try rearranging the sequence of records using the sort key . Also try the Filter By Form  and Filter by Selection  features that were introduced in Level 1.

4.2 Using Form Design View

The basic form developed by the Wizard in 5.1 is a good first step at providing an input form for our application. Let's use the Forms Design View to make a few formatting changes.

- Notice the vertical bar to the left of the form. This is called a "Record Selector" which is unnecessary in our form. Open the form in Design View and, with the cursor over the form selector button , press the right mouse button and select **Properties**. Under the **Format** tab, set the **Record Selectors** field to **No**.
- We will be using some tools that require the Control Wizard  from the Toolbox. If it is not selected, click once with the left mouse button to do so.
- We have the number of the presenter, but we really want the name as well. From the Toolbox, select the **Combo Box** button and draw a rectangle on the form so that it is large enough to hold the speakers name.



- Select **I want the combo box to look up the values in a table or query**
- Select the **Attendees** table as the source for these values
- Select **Surname** and **FirstName** to form the columns of our combo box
- Make sure **Hide Key Column** is checked. Click on **Next**.
- To the question **Store that value in this field**, enter **AttendeeID**

Notice the Combo## label. This was generated by the combo box wizard. Go into design mode and remove this label. (Click once and press the delete key.)

- In design view, experiment with making a few other cosmetic changes to the form.


- Use the Label button on the Toolbox to add a title to the form
- Try changing the position of some of the fields. To move a field, use the left button to select the field; a hand should appear on the field. Drag the field to where you would like it to appear.
- To move the label independently of the field, click on the upper left corner of the label and a hand with an extended index finger should appear. Drag the label to where you would like it to appear.

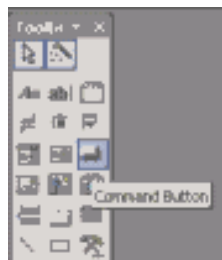
Close the form and save your changes before proceeding to the next section.

4.3 Adding Command Buttons to Your Forms

The forms that we have created contain labels, fields and navigational controls. We can add additional functions to the forms using Command buttons. We are going to add a button to our form that will execute a query.

1. Firstly, remember that we created a query called **SelectedSeminar**. This query uses a parameter to identify which seminar is to be selected by the query. We specified a parameter **[forms]![Seminars]![SeminarID]** to identify the form (Seminars) and field (SeminarID) that will contain the parameter value.
2. Now, let's add a command button to the form which will automatically run the SelectedSeminar query for the current seminar. Open the Seminars form in Design View.

- Unselect the **Control Wizards**  from the toolbox. We will create this button without the aid of the Wizard.
- Select the command button from the Toolbox and draw the shape of the button on your form.

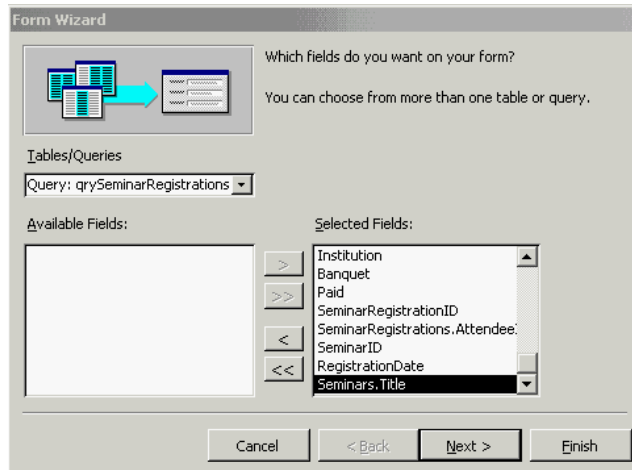


- Click with the right mouse button on the rectangle you just created and select **Properties**. Click on the **Event** tab.
- On the row labeled **On Click**, press “...” and select the Macro builder.
- Specify a macro name of **OpenQuery**.
- In the **Action Column**, specify **OpenQuery** and, in the bottom section, specify a Query Name of **SelectedSeminar**.
- Close and save the query and test your command button.

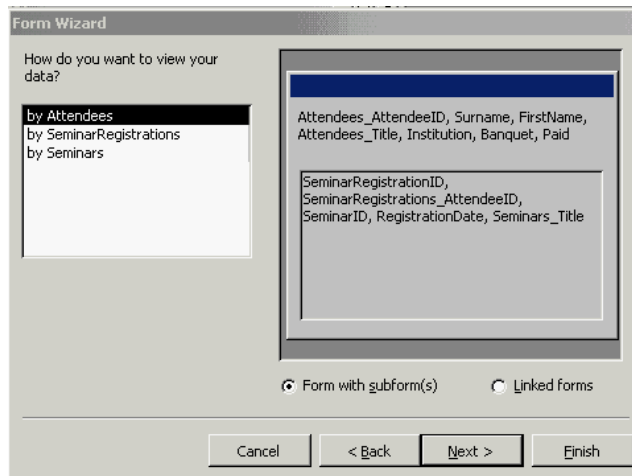
4.4 A Brief Look at SubForms

Subforms provide a powerful (although complex) means of displaying data that has a one-to-many relationship. This is a brief introduction to subforms. Other sessions will focus on this topic.

1. With the Forms button selected, create new form using the Forms Wizard, based on **qrySeminarRegistrations**. Select **all** of the fields to be included on your form.



2. View the data by **Attendees** using a Form with subforms.



3. Specify **Tabular Data** for the subform.
4. Select a **Standard** style.
5. Accept the default values for title for the form and the subform.

Attendees

AttendeeID: 1 Banquet:

Surname: Smith Paid:

FirstName: James

Title: Professor

Institution: University of Guelph

SeminarRegistrations

SeminarRegistrationID	AttendeeID	SeminarID	RegistrationDate	Title
1	1	1		Will we survive the Year 2000
2	1	2		How the Web has affected Tea
3	1	3		Media Cybernetics Image Analy:



Record: 1 of 125

Record: 1 of 8

Note that the form consists of two sections. An outer form that focuses on Attendee information. The record navigator appears at the bottom of the form (Record 1 of 125). The subform focuses on the SeminarRegistration table. Its record navigator shows the number of records associated with the particular attendee being displayed (record 1 of 9).

5. Reports

5.1 Using the Report Wizard to Generate Mailing Labels

1. Make sure the **Reports** button is selected from the database window. Click on **New**.
2. Select **Label Wizard** and choose **qryAttendees** from the drop-down list of tables and queries. Click on **OK**.
3. You will be asked what label size you would like. For this exercise, let's make name badges, so choose **Avery 5584**. Change the font size to **14**. Click on **Next**.
4. The formatting for the label is done at the same time as you select the fields. Click on **Title** then  and press the spacebar. Click on **FirstName**, press the  again. Repeat for **Surname**.
5. Press **Enter** to move to the next line of the name badge. Select **Institution** and click on **Next**.
6. Sort by **Surname** and click on **Next**.
7. Give this report the name **AttendeeTags**. Click on **Finish**.

5.2 Detail Reports

Let's use the Report Wizard to produce a detailed fee statement.

1. Click on the **New** button and select **Report Wizard** and **Attendees**. Click on **OK**.
2. Select **Surname**, **FirstName**, **Title**, **Institution**, **RegistratonFee** and **Paid** from the list of fields. Click on **Next**.

- Group by **Institution**. Click on **Next**. Within each of the Institutions, we will want to sort the attendees by **Surname**. Select Surname from the drop-down menu in the first field.
- Click on **Summary Options**. Click **Sum** on the row **RegistrationFee** and select Detail and Summary. Click on **Next**.

Summary Options

What summary values would you like calculated?

Field	Sum	Avg	Min	Max
Paid	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RegistrationFee	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK
Cancel

Show

Detail and Summary
 Summary Only

Calculate percent of total for sums

- Choose Layout **Stepped** and **Corporate** style.
- Call this report **Attendees Registration Fee Summary**.

5.4 Modifying Report Layouts

The Report Wizard provides an easy way of generating an initial report. However, as with the forms design, you may want to make changes to the report style or content.

- Click on the **View** button on the Toolbar and select **Design View**. In the **Report Header** section, select and change the title to “**Fee Statement by Institution**”.
- We will remove the “*Summary for*” line in the report, by clicking on it once with the left mouse button and pressing the delete key.
- Let’s put a line between each new company. Select the **Line** tool from the Toolbox and draw a line directly below the word Sum. (Holding down the SHIFT key while dragging the mouse will ensure that the line is straight.) Confirm how this looks by flipping to Layout Preview and back to Design View.
- To have each company on a separate page, click the **Page Break** button on the Toolbox. Place it, with a single click, under the word Sum and just above the Page Footer band. Close the summary report and save the changes.;