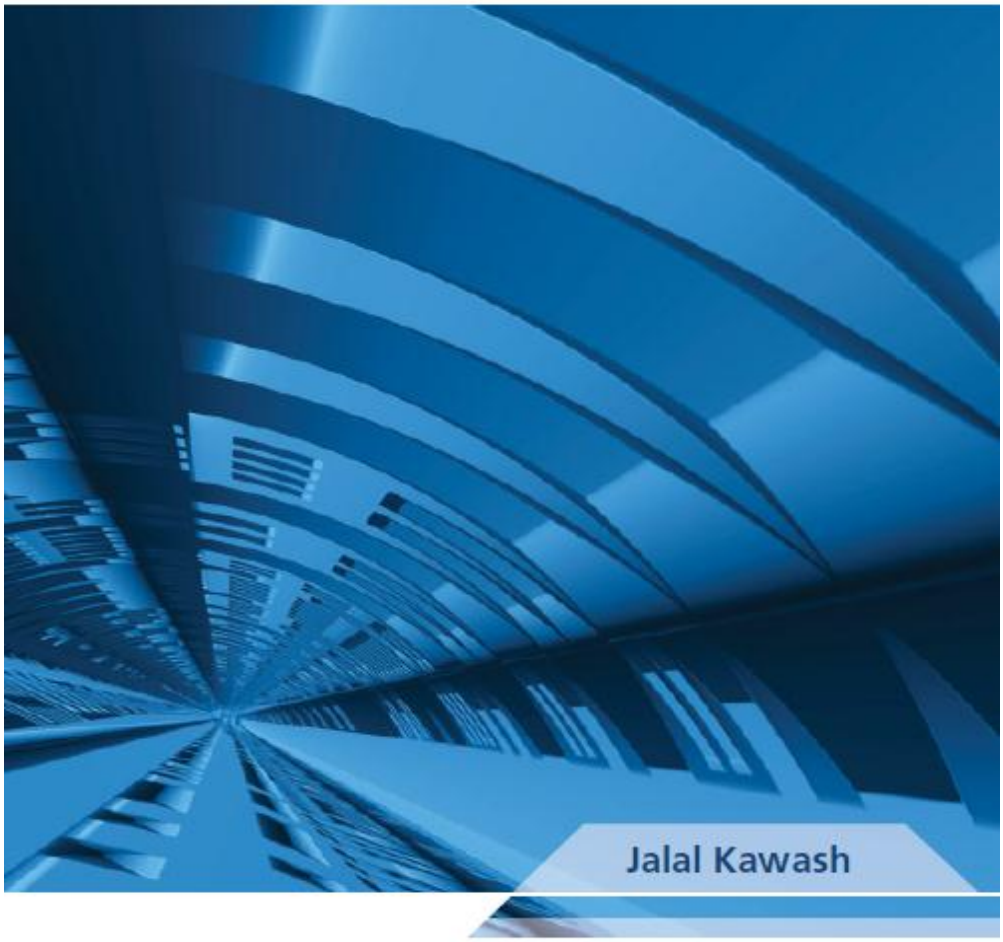
SECOND EDITION

Peeking into Computer Science



Lab Manual

DATABASES

MICROSOFT ACCESS

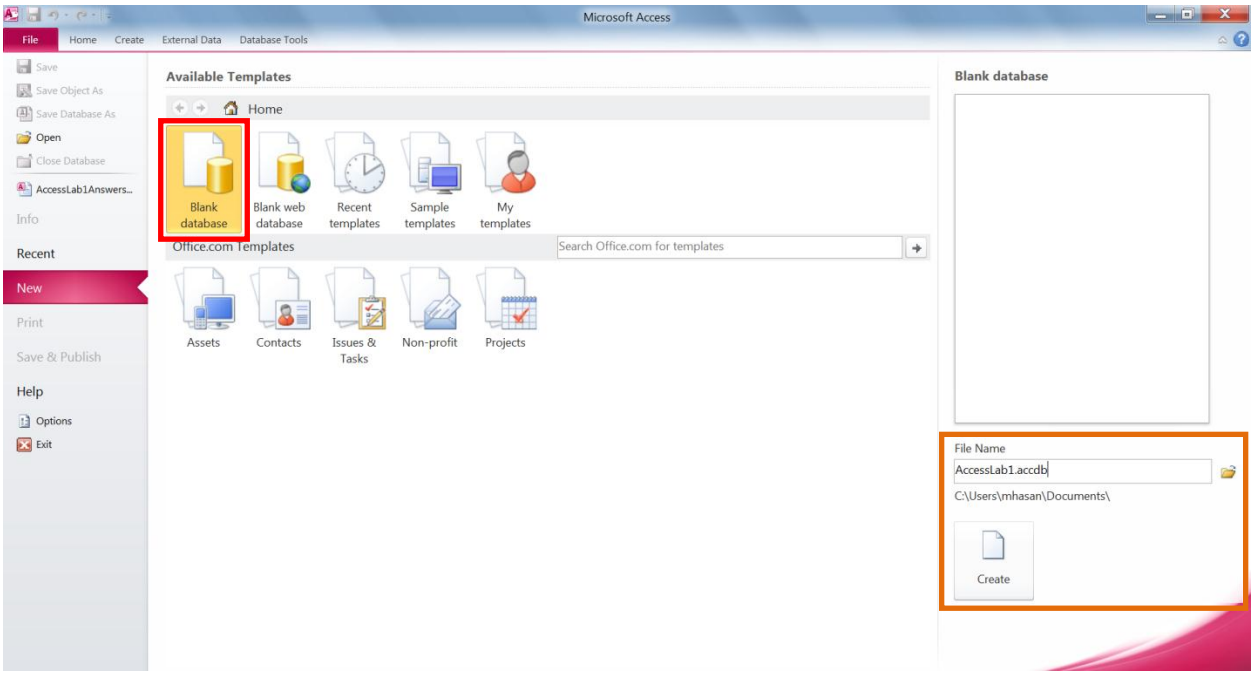
Table of Contents

Lab 1: Introduction to Microsoft Access

In the Excel module of this manual, we covered how to organize, utilize, and analyze information using spreadsheets. However, when there are large amounts of interconnected data, spreadsheets fall short of allowing us to conveniently keep track of all the linked data. This is why we need databases. Databases allow us to easily store, retrieve, manipulate, and summarize data from multiple files.

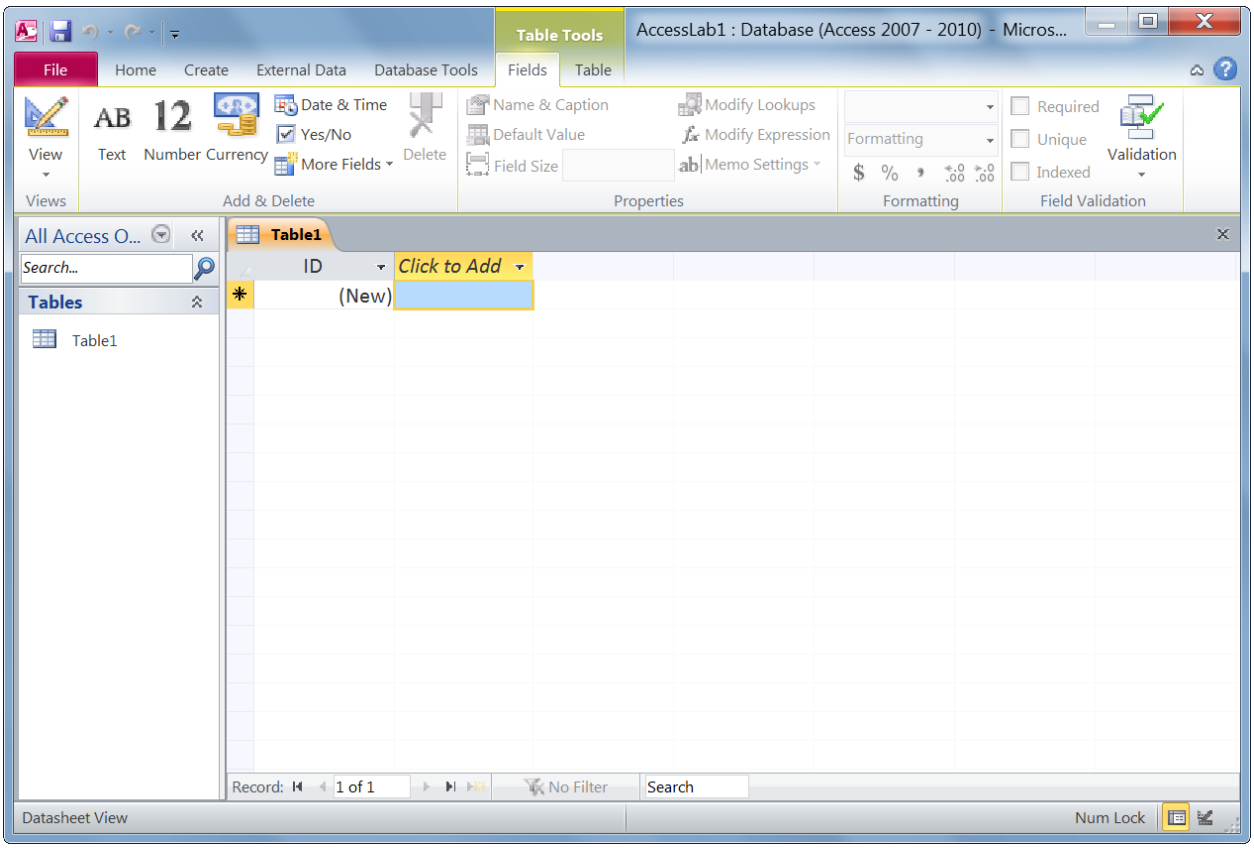
Getting started

Open Microsoft Access. Click on the *Blank Database* icon enclosed in the red rectangle below. Access asks you to enter the new database name near the bottom-right corner of the window (enclosed in an orange rectangle). Type in the database name and press the *Create* button.

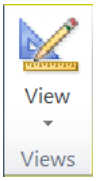


Tables

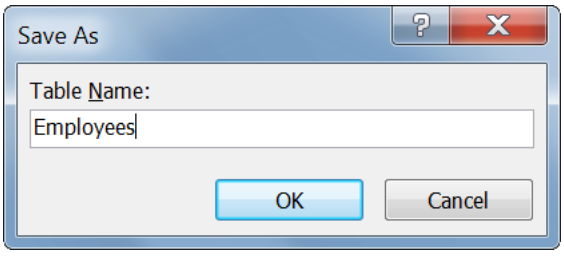
Information in a database is stored in tables. A table has fields, which are similar to column headings in Excel spreadsheets. A table also consists of records, which are similar to spreadsheet rows. A record contains information about one subject or entity. As you can see below, your new database is created with an empty table. This table comes with one predefined field named ID.



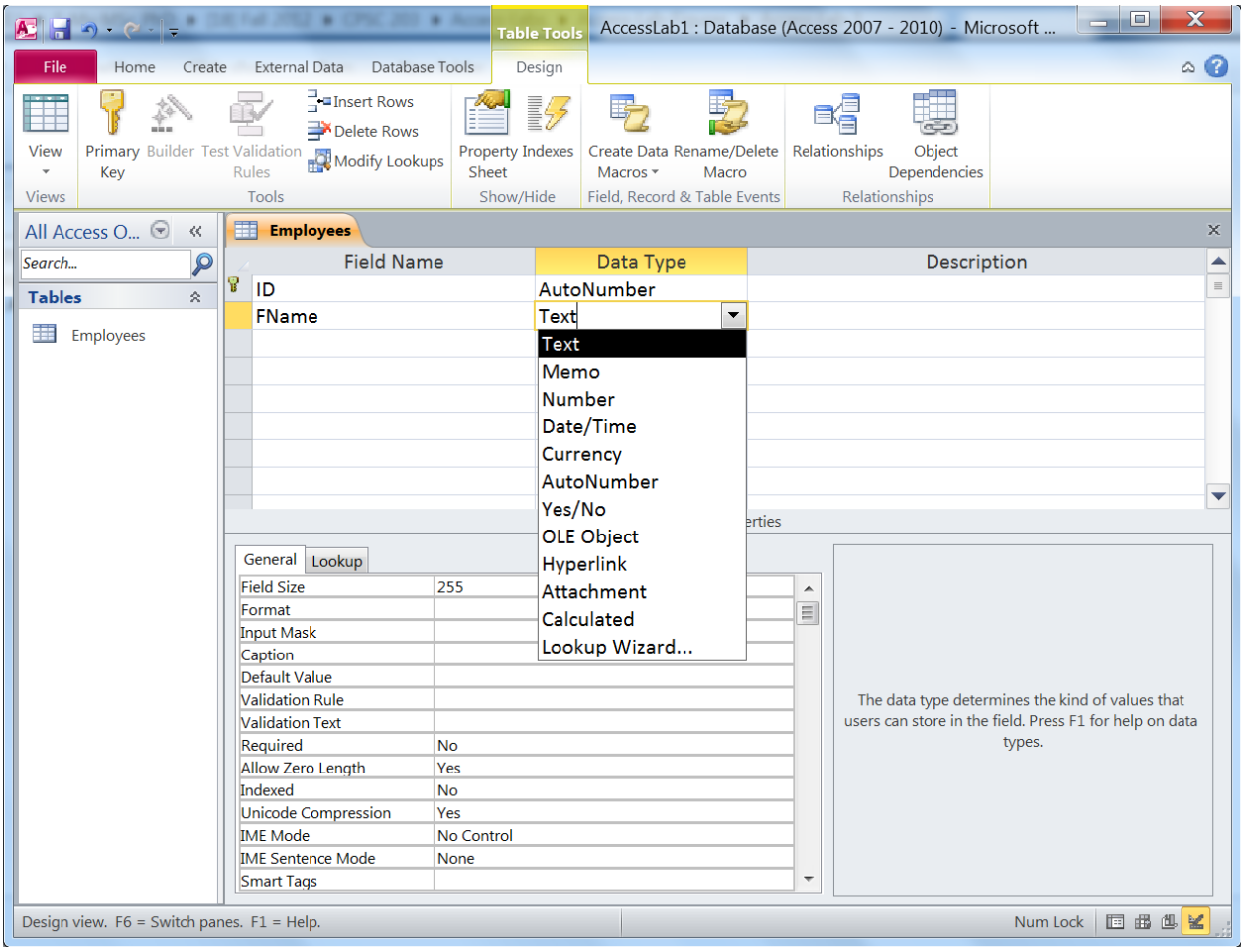
Let us create a table containing Employee information. We first want to add all the fields we need into the table. This is done by clicking on the button in the *Views* group of the *Datasheet* tab:



You are then prompted to enter a name for your table. Type in **Employees** and press the *OK* button:



In the design view, you will see the field ID, and that it is set as an AutoNumber. Let us add the field FName (first name) to the table.



Notice the different data types available. FName is obviously of type Text. The following list from Access describes what each data type is.

- **Attachment**: Files, such as digital photos. Multiple files can be attached per record. This data type is not available in earlier versions of Access.
- **AutoNumber**: Numbers that are automatically generated for each record.
- **Currency**: Monetary values.
- **Date/Time**: Dates and times.
- **Hyperlink**: Hyperlinks, such as e-mail addresses.
- **Memo**: Long blocks of text and text that use text formatting. A typical use of a Memo field would be a detailed product description.
- **Number**: Numeric values, such as distances. Note that there is a separate data type for currency.
- **OLE Object**: OLE objects, such as Word documents.
- **Text**: Short, alphanumeric values, such as a last name or a street address.
- **Yes/No**: Boolean values.

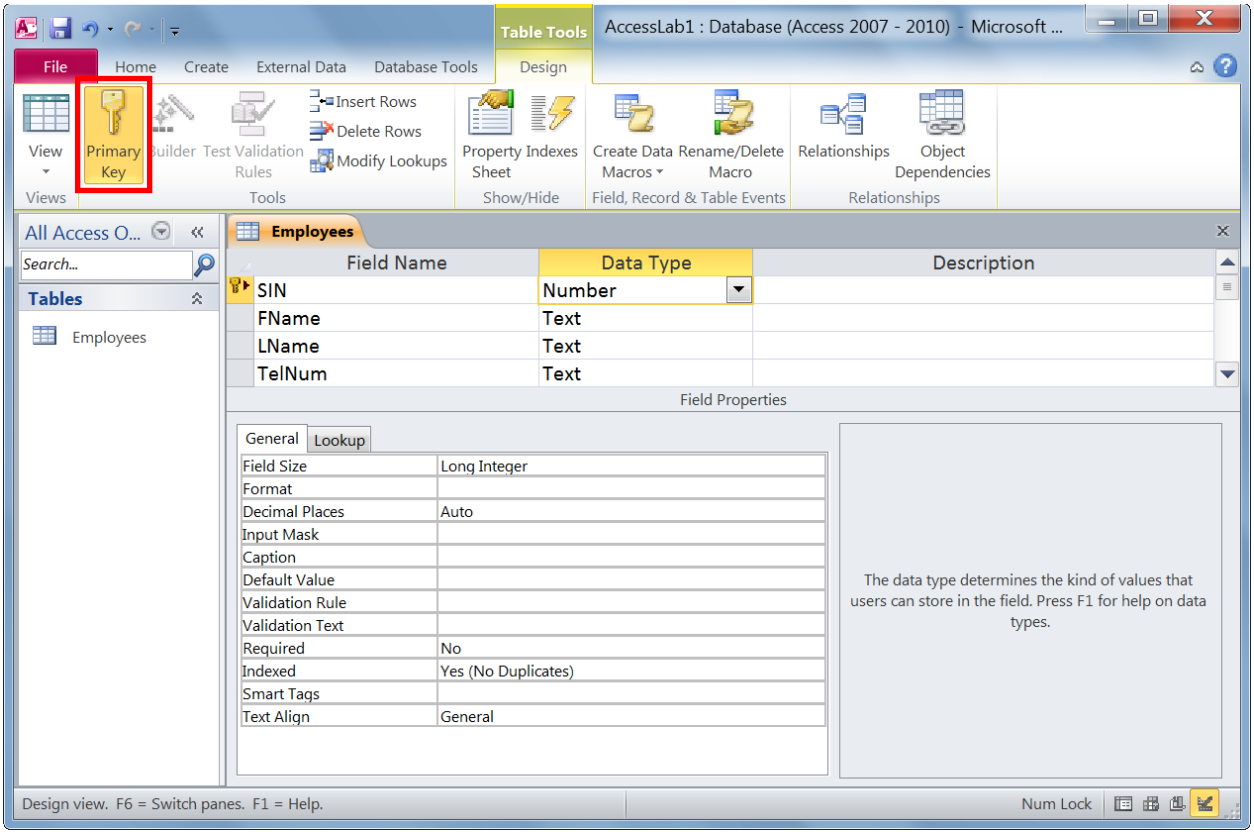
Exercise 1

Add the following fields to the table, with the appropriate data types: LName (for last name), TelNum (for telephone number), DOB (for date of birth), DateJoined, Salary, Address, City, PostalCode, OnLeave and Gender. Note that some fields may have more than one correct data type.

Primary Keys

You probably noticed that there is a key symbol next to the ID field. This means that the ID field is the primary key of the table. A primary key is a field or set of fields used to uniquely identify records in a table. A primary key value is always unique in the table. A table can only have one primary key, and each table in your database must have a primary key.

It is important to decide on what you want the primary key of the Employees table to be. This field value needs to be unique for each employee. One unique field is the SIN (Social Insurance Number). Replace the word ID with SIN and change the data type to Number. This is because we do not want the SIN to be automatically generated by Access. To set or remove a primary key, you can click on the Primary Key button shown in the red rectangle below. We do not need to do this now.



Field Properties

Field properties are used to set additional properties for your field. Following is a description of the field properties for each data type from Microsoft Office help. You do NOT need to memorize the contents of the following table. You can always look it up for reference.

DATE TYPE	SUPPORTED FIELD PROPERTY	USE
Attachment	**Caption**	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Required	Requires that each record has at least one attachment for the field.
AutoNumber	**Field Size**	Determines the amount of space that is allocated for each value. For AutoNumber fields, only two values are allowed: ▪ The Long Integer field size is used for AutoNumber fields that are not used as replication IDs. This is the default value. You should not change this value unless you are creating a replication ID field. **Note** Replication is not supported in databases that use a new file format, such as .accdb. This setting makes AutoNumber fields compatible with other Long Integer Number fields when they are used in relationships or joins. Each field value requires 4 bytes of storage. ▪ The **Replication ID** field size is used for AutoNumber fields that are used as replication IDs in a database replica. Do not use this value unless you are working in or implementing the design of a replicated database. Each field value requires 16 bytes of storage.
	New Values	Determines whether AutoNumber field increments with each new value or uses random numbers. Select one of the following: ▪ **Increment** Starts with the value 1 and incrementally increases by 1 for each new record. ▪ **Random** Starts with a random value and assigns a random value to each new record. Values are of the Long Integer field size, and range from -2,147,483,648 to 2,147,483,647.
	Format	If you are using an AutoNumber field as a primary key or as a Replication ID, you should not set this property. Otherwise, choose a number format that meets your specific needs.
	Caption	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Without a unique index, it is possible to enter duplicate values, which can break any relationships in which the key is a part. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
Currency	**Format**	Determines the way that the field appears when it is displayed or printed in datasheets or in forms or reports that are bound to the field. You can use any valid number format. In most cases, you should set the **Format** value to **Currency**.
	Decimal Places	Specifies the number of decimal places to use when displaying numbers.
	Input Mask	Displays editing characters to guide data entry. For example, an input mask might display a dollar sign (**$**) at the beginning of the field.
	Caption	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
	Required	Requires that data be entered in the field.
	Indexed	Specifies whether the field has an index. There are three available values:

		▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
Date/Time	**Caption**	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets, or in forms or reports that are bound to the field. You can use a predefined format or build your own custom format.
	IME Mode	Controls the conversion of characters in East Asian versions of Windows.
	IME Sentence Mode	Controls the conversion of sentences in East Asian versions of Windows.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Input Mask	Displays editing characters to guide data entry. For example, an input mask might display a dollar sign (**$**) at the beginning of the field.
	Required	Requires that data be entered in the field.
	Show Date Picker	Specifies whether to show the **Date Picker** control. **Note** If you use an input mask for a Date/Time field, the **Date Picker** control is unavailable regardless of how you set this property.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
Hyperlink	**Allow Zero Length**	Allows entry (by setting to **Yes**) of a zero-length string (**""**) in a Hyperlink, Text, or Memo field.
	Append Only	Determines whether to track field value changes. There are two settings: ▪ **Yes** Tracks changes. To view the field value history, right-click the field, and then click Show column history. ▪ **No** Does not track changes. **Warning** Warning Setting this property to **No** deletes any existing field value history.
	Caption	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets or in forms or reports that are bound to the field. You can define a custom format for a Hyperlink field.
	IME Mode	Controls the conversion of characters in East Asian versions of Windows.
	IME Sentence Mode	Controls the conversion of sentences in East Asian versions of Windows.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Required	Requires that data be entered in the field.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.

	Unicode Compression	Compresses text that is stored in this field when less than 4,096 characters are stored.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
Memo	**Allow Zero Length**	Allows entry (by setting to **Yes**) of a zero-length string ("") in a Hyperlink, Text, or Memo field.
	Append Only	Determines whether to track field value changes. There are two settings: ▪ **Yes** Tracks changes. To view the field value history, right-click the field, and then click **Show column history**. ▪ **No** Does not track changes. **Warning** Setting this property to **No** deletes any existing field value history.
	Caption	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets or in forms or reports that are bound to the field. You can define a custom format for a Memo field.
	IME Mode	Controls the conversion of characters in East Asian versions of Windows.
	IME Sentence Mode	Controls the conversion of sentences in East Asian versions of Windows.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Required	Requires that data be entered in the field.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
	Unicode Compression	Compresses text that is stored in this field when less than 4,096 characters are stored.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
Number	**Caption**	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Decimal Places	Specifies the number of decimal places to use when displaying numbers.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Field Size	Select one of the following: ▪ **Byte** — Use for integers that range from 0 to 255. Storage requirement is 1 byte. ▪ **Integer** — Use for integers that range from -32,768 to 32,767. Storage requirement is 2 bytes. ▪ **Long Integer** — Use for integers that range from -2,147,483,648 to 2,147,483,647. Storage requirement is 4 bytes. **Tip** Use **Long Integer** when you create a foreign key to relate to another table's AutoNumber primary key field. ▪ **Single** Use for numeric floating point values that range from -3.4×10^{38} to 3.4×10^{38} and up to seven significant digits. Storage requirement is 4 bytes. ▪ **Double** Use for numeric floating point values that range from -1.797×10^{308} to 1.797×10^{308} and up to fifteen significant digits. Storage requirement is 8 bytes. ▪ **Replication ID** Use for storing a globally unique identifier required for replication. Storage requirement is 16 bytes. Note that replication is not supported using the .accdb file format. ▪ **Decimal** Use for numeric values that range from $-9.999... \times 10^{27}$ to $9.999... \times 10^{27}$. Storage requirement is 12 bytes. **Tip** For best performance, always specify the smallest sufficient **Field Size**.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets, or in forms or reports that are bound to the field. You can use any valid number format.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key.

		Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Input Mask	Displays editing characters to guide data entry. For example, an input mask might display a dollar sign (**$**) at the beginning of the field.
	Required	Requires that data be entered in the field.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
OLE Object	**Caption**	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Required	Requires that data be entered in the field.
	Text Align	Specifies the default alignment of text within a control.
Text	**Allow Zero Length**	Allows entry (by setting to **Yes**) of a zero-length string ("") in a Hyperlink, Text, or Memo field.
	Caption	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Field Size	Enter a value from 1 to 255. Text fields can range from 1 to 255 characters. For larger text fields, use the Memo data type. **Tip** For best performance, always specify the smallest sufficient **Field Size**. For example, if you are storing postal codes of a known length, you should specify that length as the **Field Size**.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets or in forms or reports that are bound to the field. You can define a custom format for a Text field.
	IME Mode	Controls the conversion of characters in East Asian versions of Windows.
	IME Sentence Mode	Controls the conversion of sentences in East Asian versions of Windows.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field index by setting this property.
	Required	Requires that data be entered in the field.
	Smart Tags	Attaches a smart tag to the field.
	Text Align	Specifies the default alignment of text within a control.
	Unicode Compression	Compresses text that is stored in this field when less than 4,096 characters are stored.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.
Yes/No	**Caption**	The label text that is displayed for this field by default in forms, reports, and queries. If this property is empty, the name of the field is used. Any text string is allowed. **Tip** An effective caption is usually brief.
	Default Value	Automatically assigns the specified value to this field when a new record is added.
	Format	Determines the way that the field appears when it is displayed or printed in datasheets, or in forms or reports that are bound to the field. Select one of the following: ▪ **True/False** Displays the value as either True or False. ▪ **Yes/No** Displays the value as either Yes or No. ▪ **On/Off** Displays the value as either On or Off.
	Indexed	Specifies whether the field has an index. There are three available values: ▪ **Yes (No duplicates)** Creates a unique index on the field. ▪ **Yes (Duplicates OK)** Creates a non-unique index on the field. ▪ **No** Removes any index on the field. **Note** Do not change this property for a field that is used in a primary key. Although you can create an index on a single field by setting the **Indexed** field property, some kinds of indexes cannot be created in this manner. For example, you cannot create a multi-field

		index by setting this property.
	Text Align	Specifies the default alignment of text within a control.
	Validation Rule	Supplies an expression that must be true whenever you add or change the value in this field. Use in conjunction with the **Validation Text** property.
	Validation Text	Enter a message to display when a value that is entered violates the expression in the **Validation Rule** property.

Exercise 2

Modify the field properties so that the SIN, FName, and LName are required fields. FName and LName should not allow zero length entries. Set the default value for City to "Calgary". Set the default value for DateJoined to today's date (enter Date()). As the caption of DOB, type in BirthDate.

Validation Rules

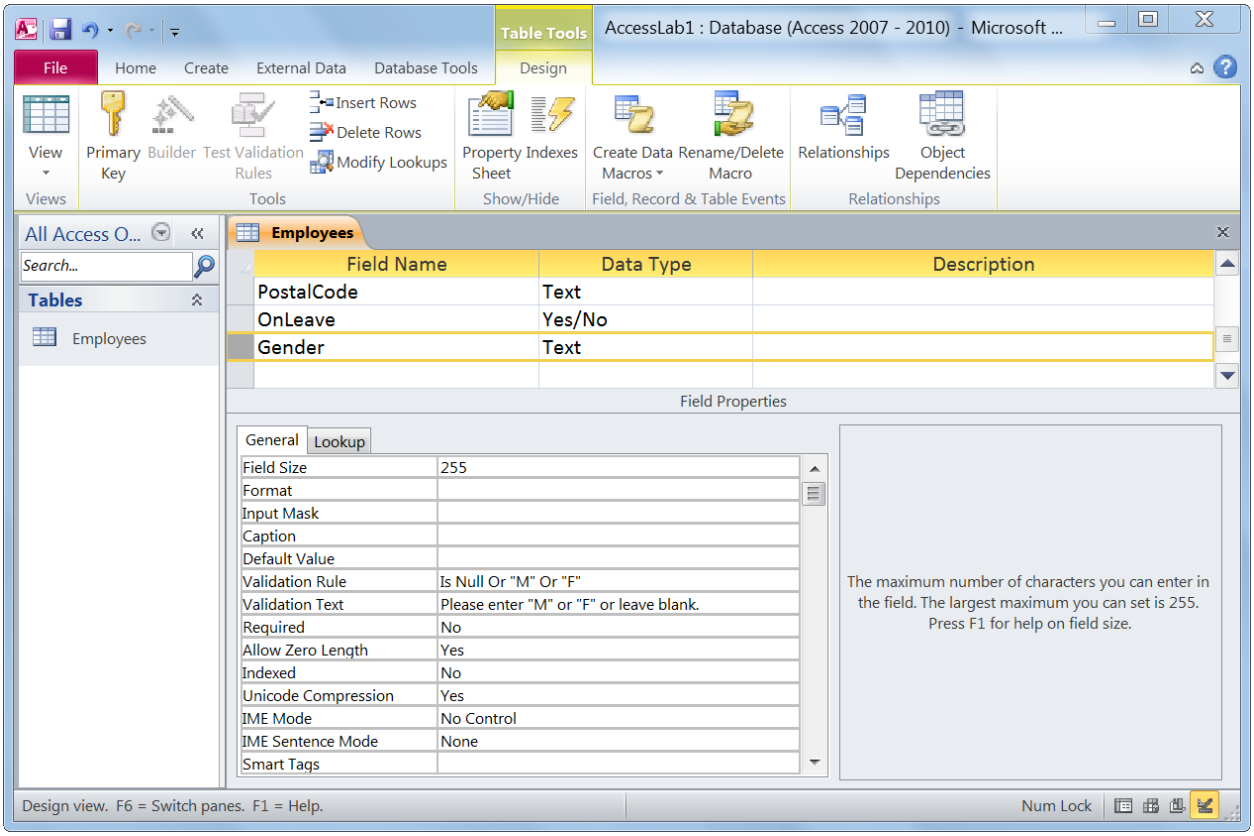
Validation rules stop users from entering incorrect data into your tables. For instance, one validation rule can make sure that the DOB is earlier than today's date. Another one can be used to ensure that the salary is not zero or it is not a negative number!

When a user enters data that is incorrect (according to the validation rule that you supply), the error message that s/he sees is set in the Validation Text property.

Let us create a validation rule so that the Gender field is limited to either "M" or "F" (Male or Female). The field can also be left blank.

Click on the Gender field so that the field properties show up. In the Validation Rule property, enter **Is Null Or "M" Or "F"**. In the Validation Text property, enter **Please enter "M" or "F" or leave blank**.

Blank or empty values in a database are called *null* values. *Is Null* signifies that a value in the Gender field is permitted to be blank. The reason we add 'Is Null' to our validation rule is that we do not want this to be a required field. Even if you set the field property to *not required* and you set a validation rule, some versions of Access will not allow you to leave the field blank.



Exercise 3

Set validation rules (and respective validation texts) for the DOB and DateJoined fields so that they are always before or equal to today's date.

Exercise 4

Set a validation rule and text for the salary field so that it is empty or greater than zero.

Input Masks

Input Masks allow you to set a particular format for your field. For instance, we can set that the postal code has to be three alphanumeric characters, followed by a space, followed by three other alphanumeric characters.

You will sometimes find that input masks and validation rules can be used interchangeably. Just do whatever seems easier (on a test however, do whatever the question asks!).

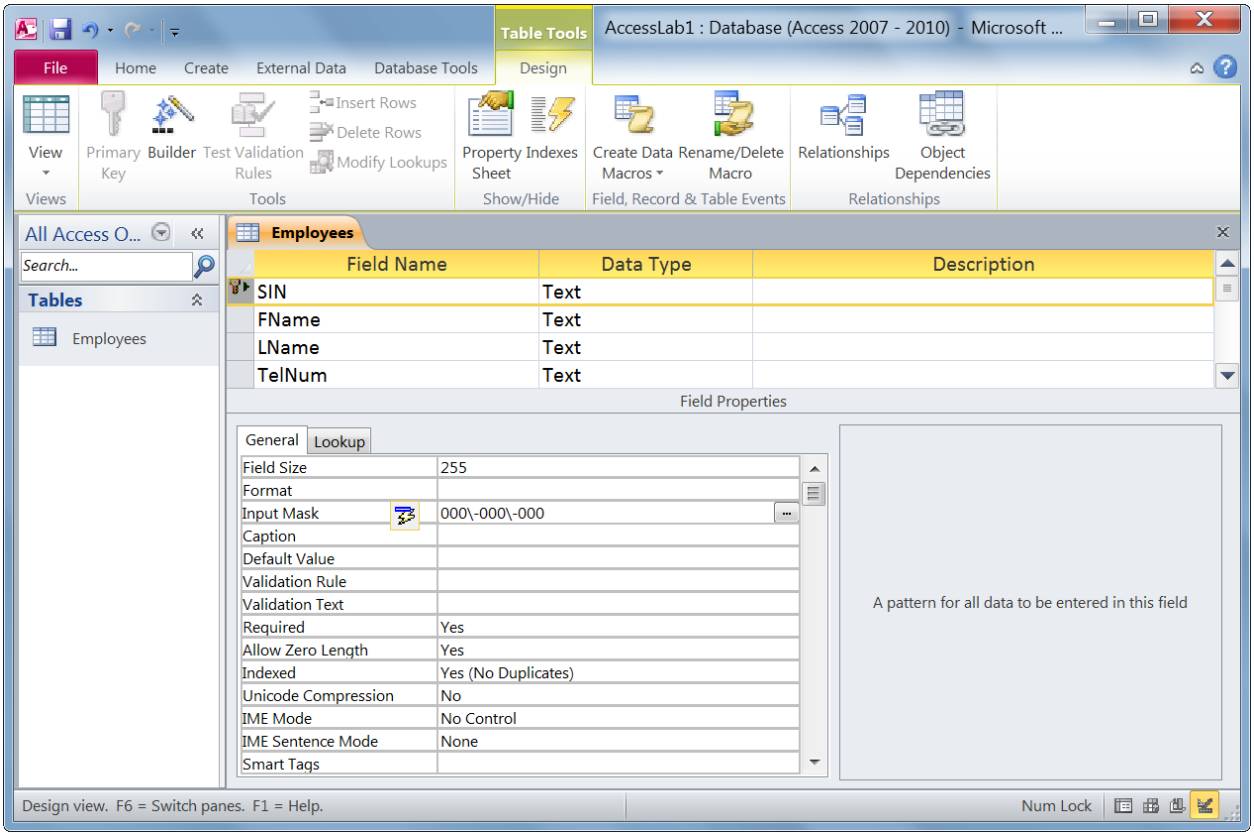
To set an input mask, you use placeholder (or literal) characters that show the format you want for your field. Following is the table of characters (from Microsoft Office help) that can be used in input masks. Again, you do not need to memorize this table. Just use it as a reference.

CHARACTER	EXPLANATION
0	User must enter a digit (0 to 9).
9	User can enter a digit (0 to 9).
#	User can enter a digit, space, plus or minus sign. If skipped, Access enters a blank space.
L	User must enter a letter.
?	User can enter a letter.
A	User must enter a letter or a digit.
a	User can enter a letter or a digit.
&	User must enter either a character or a space.
C	User can enter characters or spaces.
. , : ; - /	Decimal and thousands placeholders, date and time separators. The character you select depends on your Microsoft Windows regional settings.
>	Coverts all characters that follow to uppercase.
<	Converts all characters that follow to lowercase.
!	Causes the input mask to fill from left to right instead of from right to left.
\	Characters immediately following will be displayed literally.
""	Characters enclosed in double quotation marks will be displayed literally.

Notice the way some explanations say a user MUST enter something (like 0) while others say a user CAN enter something (like 9).

An example will make things clearer. The SIN consists of nine digits. Each group has three digits separated by the dash ('-') character. The SIN MUST have exactly nine digits. Now since we need to add dashes to the SIN, we need to change the SIN data type to text.

After you have done that, set 000-000-000 as the input mask. This input masks restricts values entered into the SIN field to be of the form: 3 required digits, followed by '-', followed by 3 required digits, followed by '-', and finally 3 required digits.



Notice that backward slashes were placed before the dashes automatically, as soon as you place the cursor away from the Input Mask field. This is because dashes do not have a predefined meaning and so it is taken by Access to be a literal character. The preceding backslash states explicitly that the '-' will be shown as is.

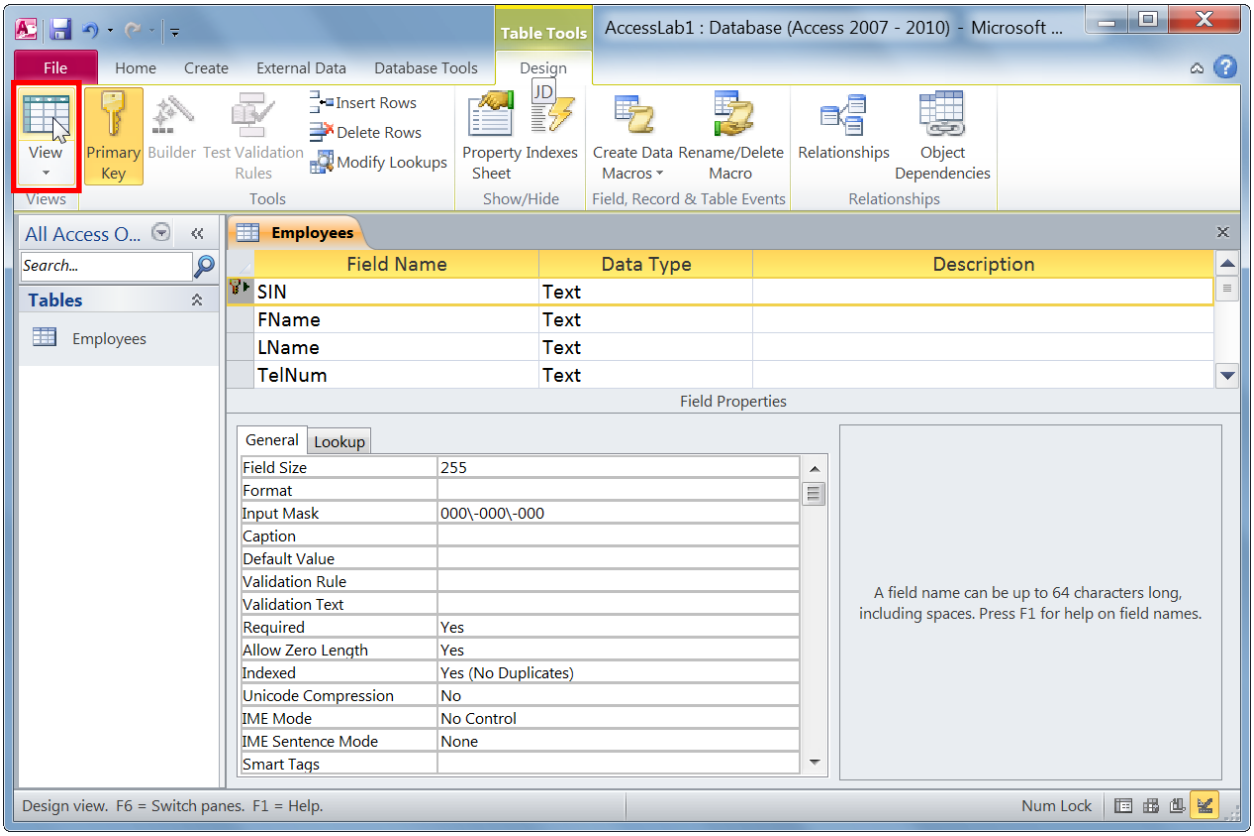
Exercise 5

Add an input mask to the TelNum field that allows entry of numbers like: (403) 789-1234. The area code should be optional. (Hint: don't forget to change the data type of TelNum first. Also, consider using #).

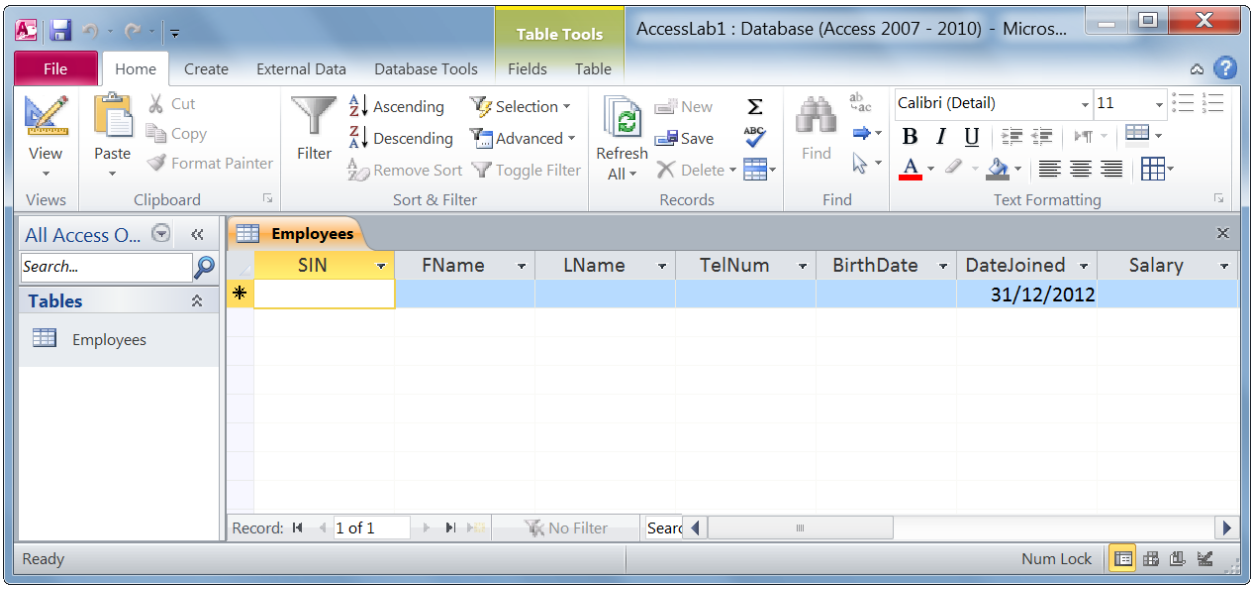
Exercise 6

Enter an input mask for the postal code so that it restricts the format to three alphanumeric characters, followed by a space, followed by three other alphanumeric characters (for instance, T2N 1N4).

Now we are finally ready to start entering data! Go to the datasheet view by clicking the button enclosed in the red rectangle. If prompted to Save, please confirm.



The table in datasheet view looks as follows:



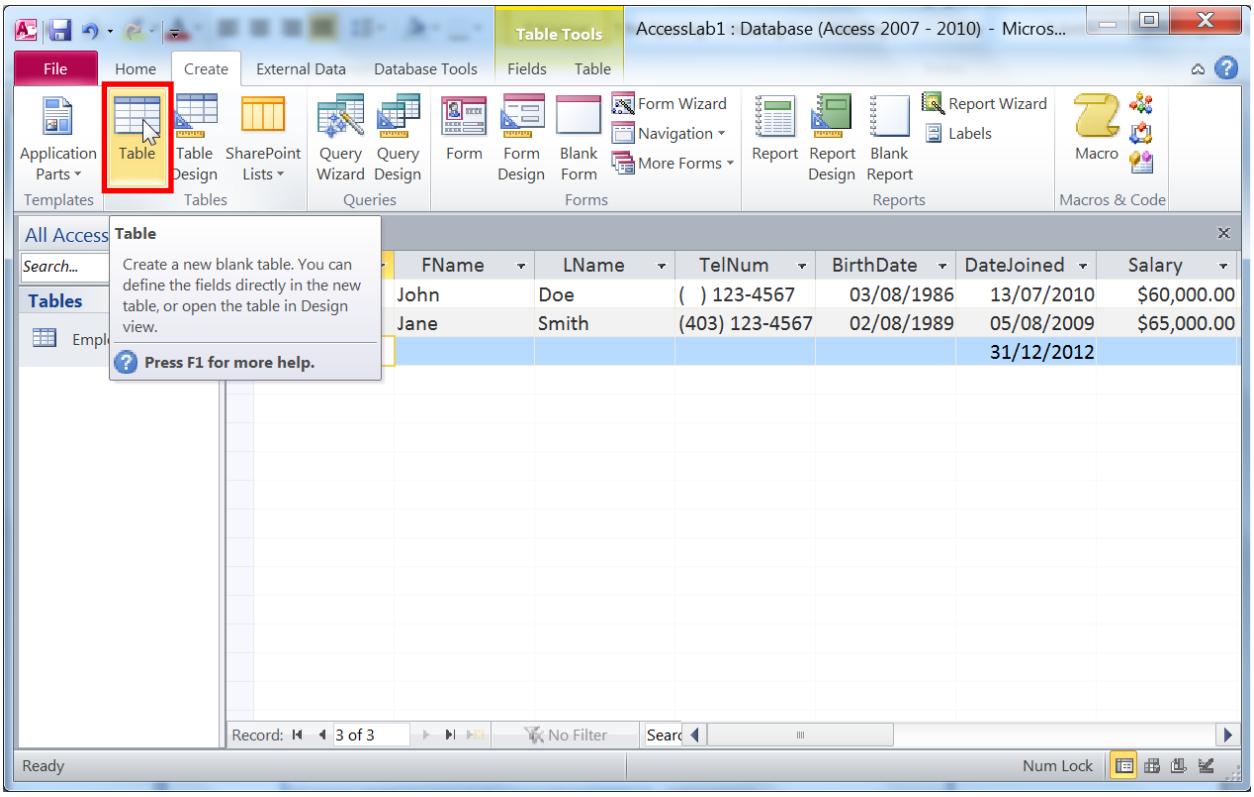
Exercise 7

Fill in at least one record in your table. Try to enter incorrect data into some fields with validation rules to see how your validation text is displayed. Also, notice that DOB is displayed as BirthDate because of the caption that was set.

Note: The following section of this lab is so simple that students are encouraged to do it on their own.

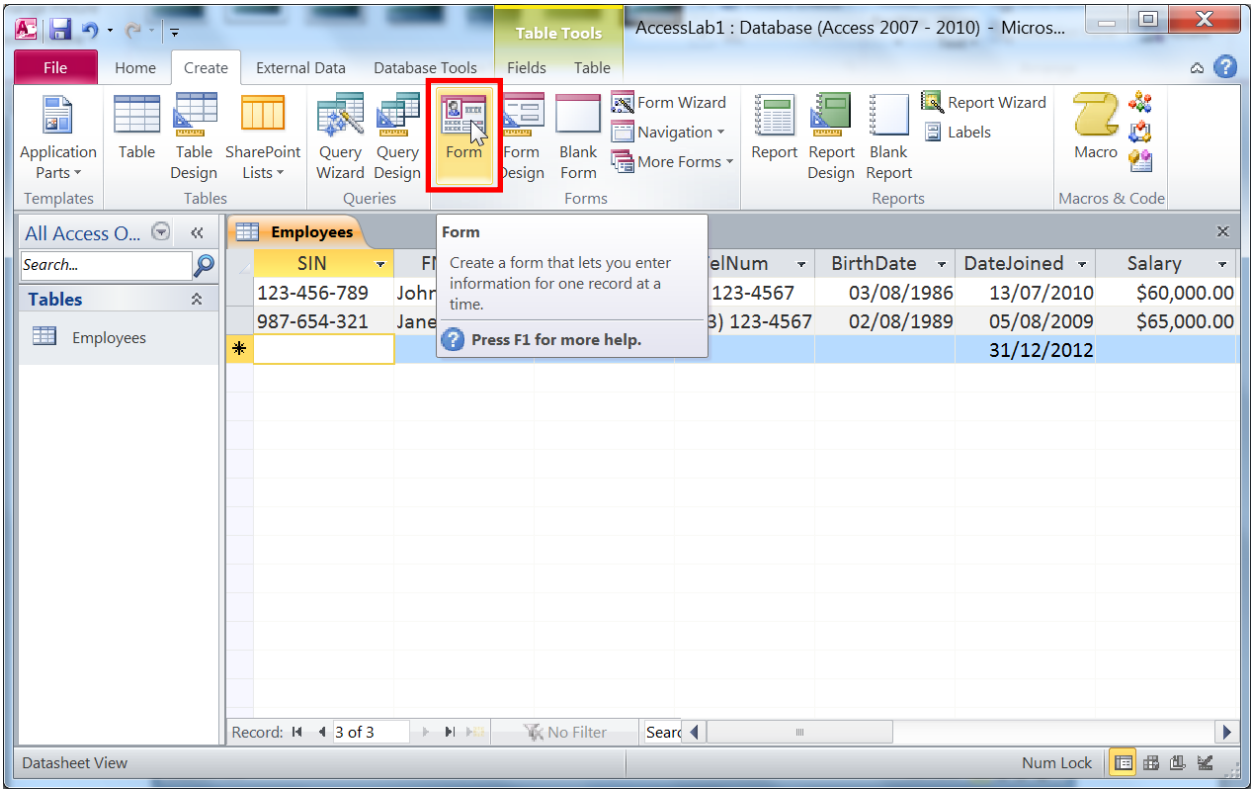
Creating new tables

A new table can be created by pressing the *Table* button in the *Tables* group under the *Create* tab on the ribbon as shown below:

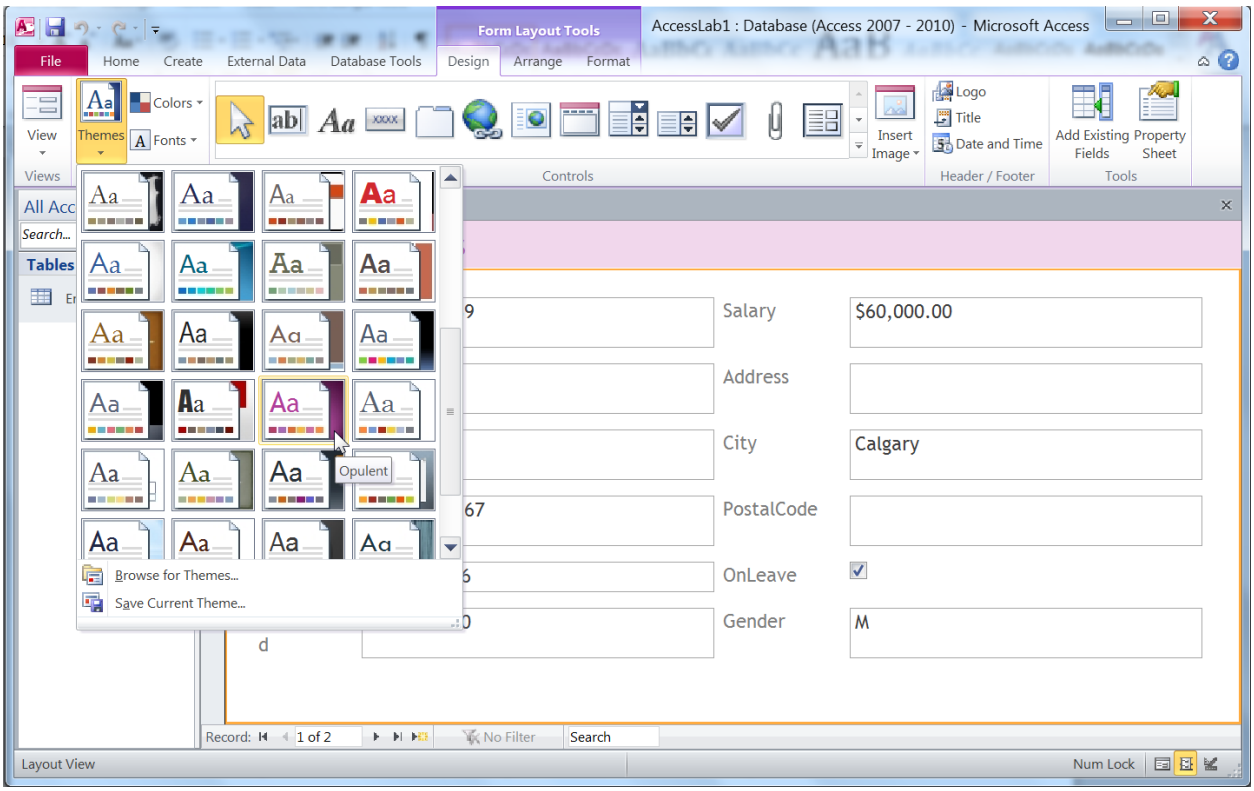


Forms

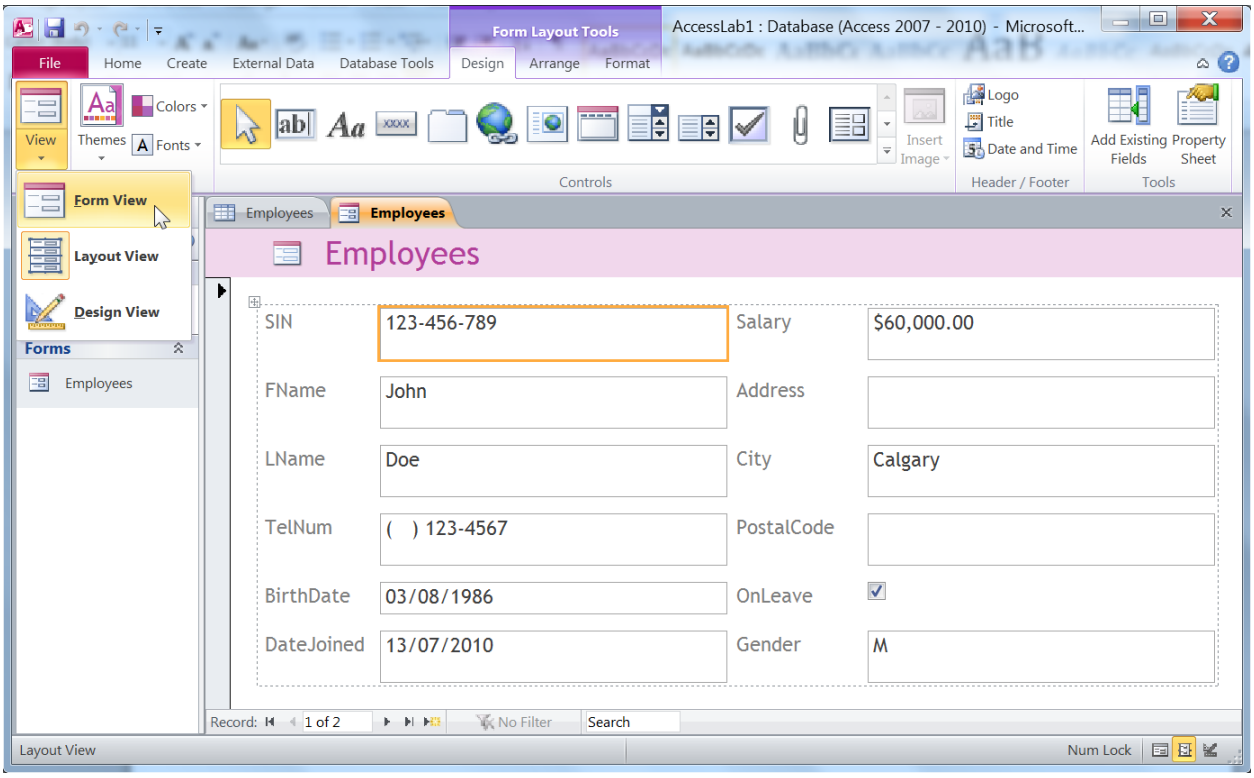
Forms provide a nice interface for entering data into tables and for viewing existing data. To create a form for the Employees table, first make sure that the table is open. Navigate to the *Create* tab, then in the *Forms* group, click on the *Form* button.



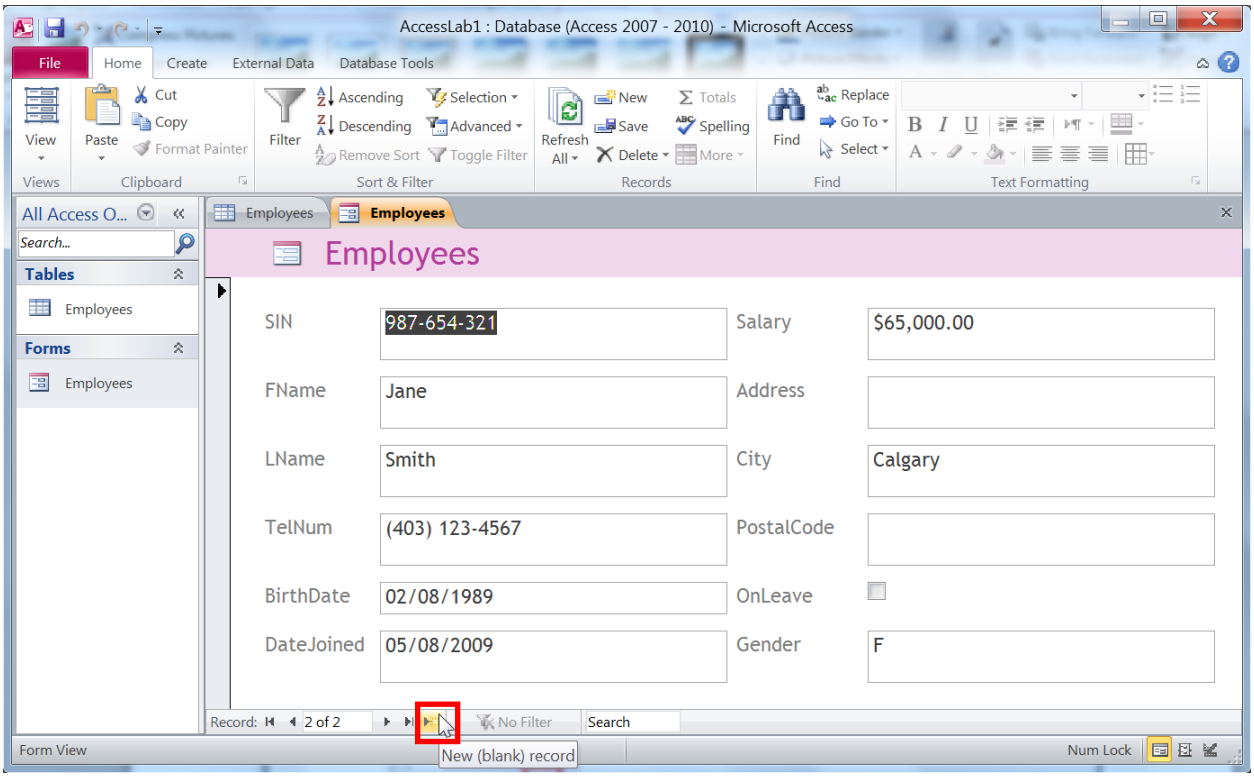
You are now in the layout view of the form that was just created. You can drag fields and labels around the form as you wish. You can also change the colors and formatting of the form. Here, we give the form a purple theme (named Opulent) by pressing the *Themes* button in the *Themes* group under the *Design* tab.



When you are done formatting your form's layout, go to the *Form View* from the *View* button under the *Design* tab as shown below.



You are now allowed to view and edit the existing records and add new records to your table. To add a new record, click on the *New* blank record button at the bottom of the screen.



You can now see a blank record.

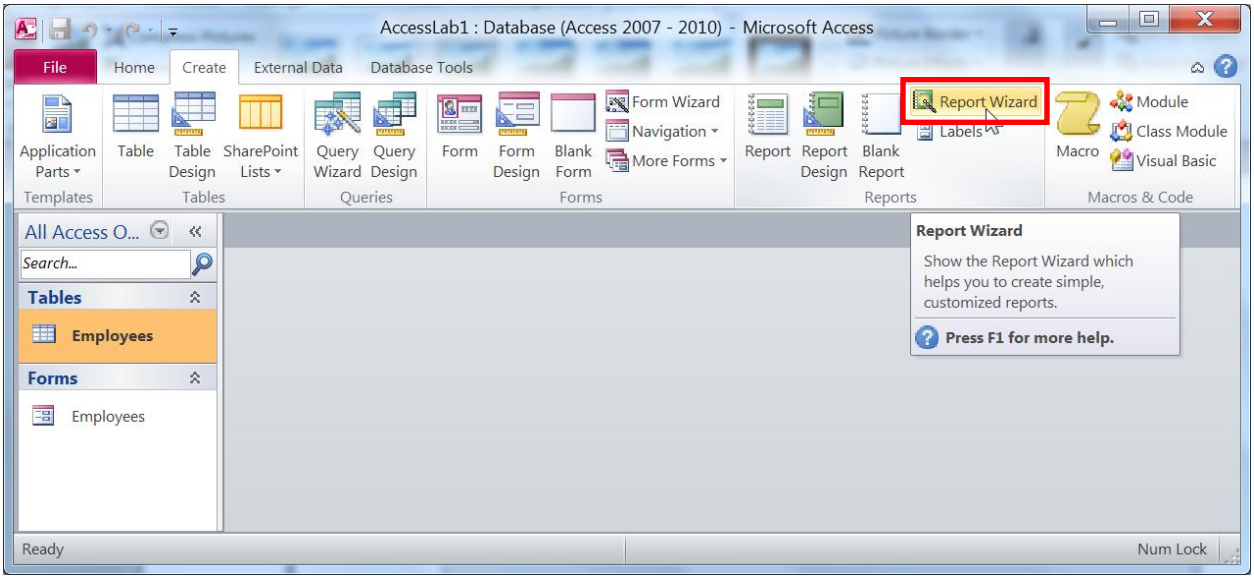
Exercise 8

Fill in the blank record then use the arrow buttons to navigate between your records. Make sure you save your form. You will also notice that the same validation rules apply in the form.

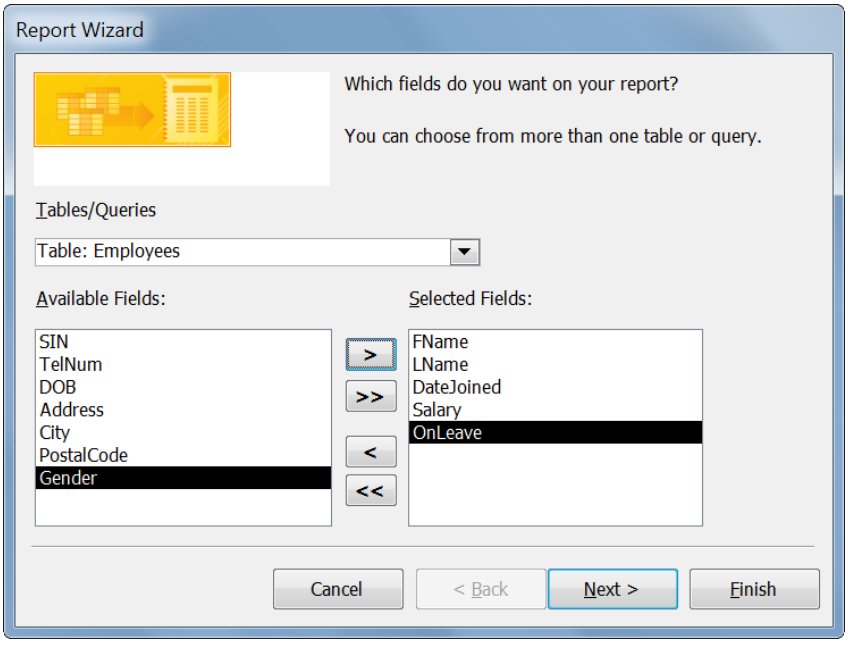
Reports

Reports provide you with a read-only, printable view of tables, queries (more on this later) or a combination of them.

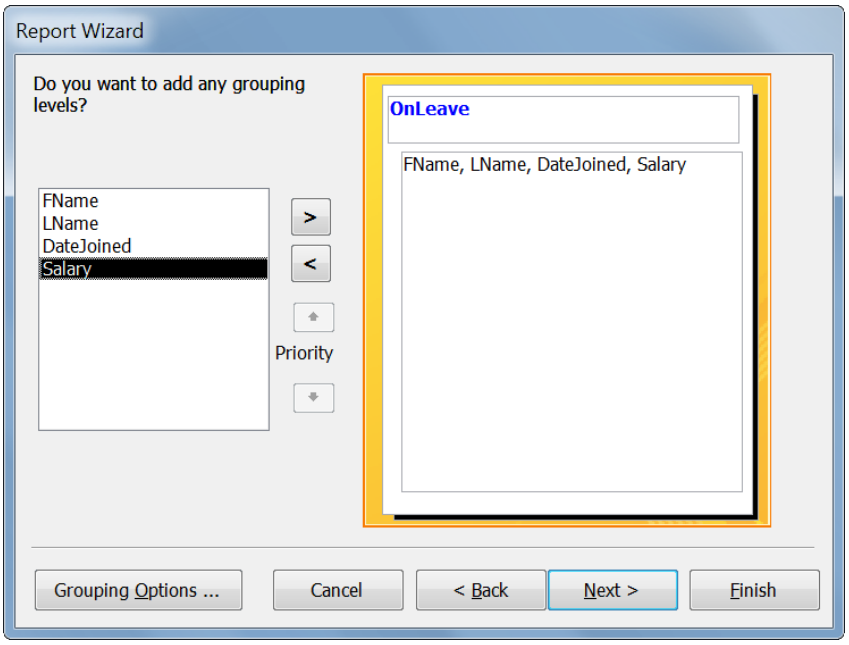
You can make a report in the same way a form is created but let us try the *Report Wizard*. Go to the *Create* tab and click on *Report Wizard* in the *Reports* Group.



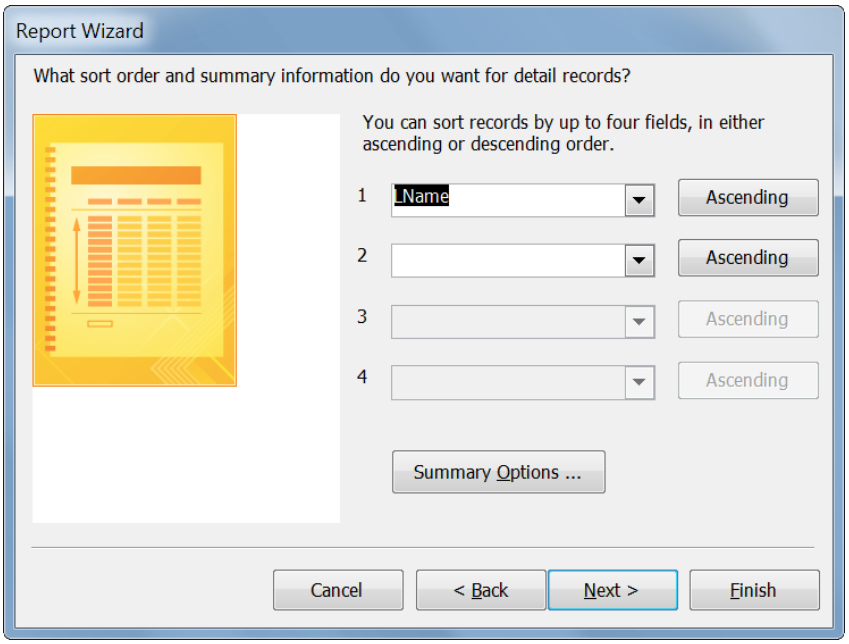
The report wizard pops up and allows you to select one or more tables and/or queries you want to include in your report. Remember that you can combine data from more than one table or query. You can select the fields by using the arrow buttons in the wizard. Select the fields FName, LName, DateJoined, Salary, and OnLeave. Once you are done, press the *Next* button.



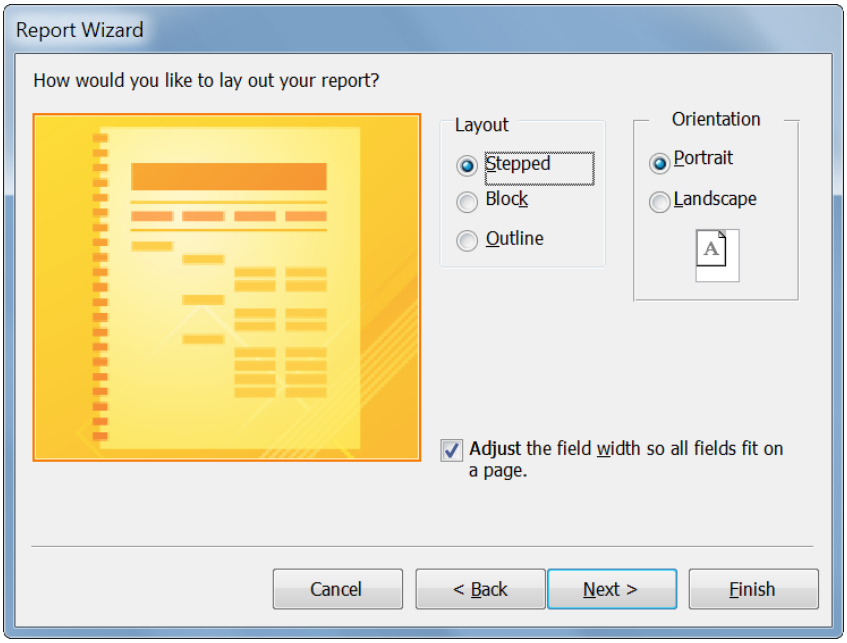
The next screen allows you to group records based on a particular field. For instance, we can have a separate group for employees who are currently on leave. Add the OnLeave field as a grouping level and press the *Next* button.



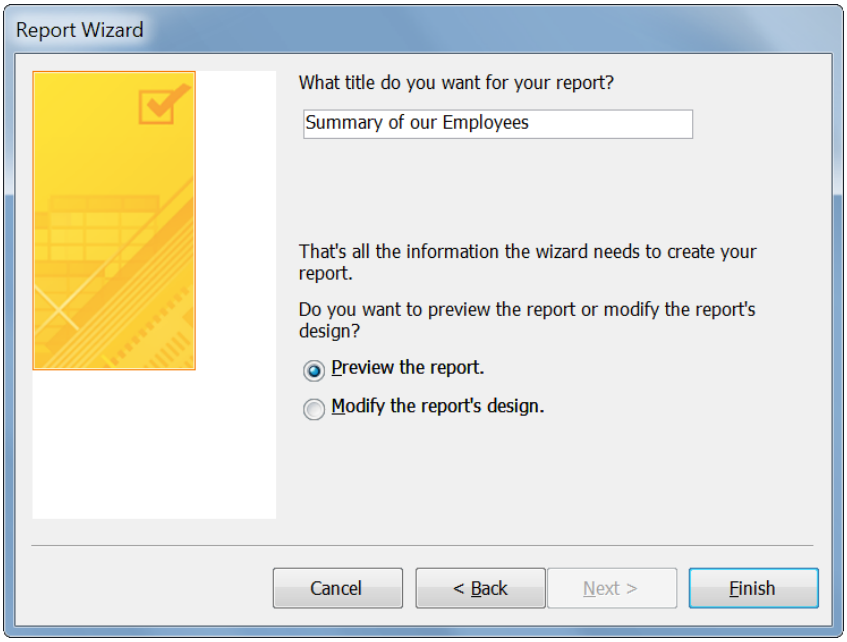
You can also sort records within a group. Let us sort ascending order of LName. Then, press *Next*.



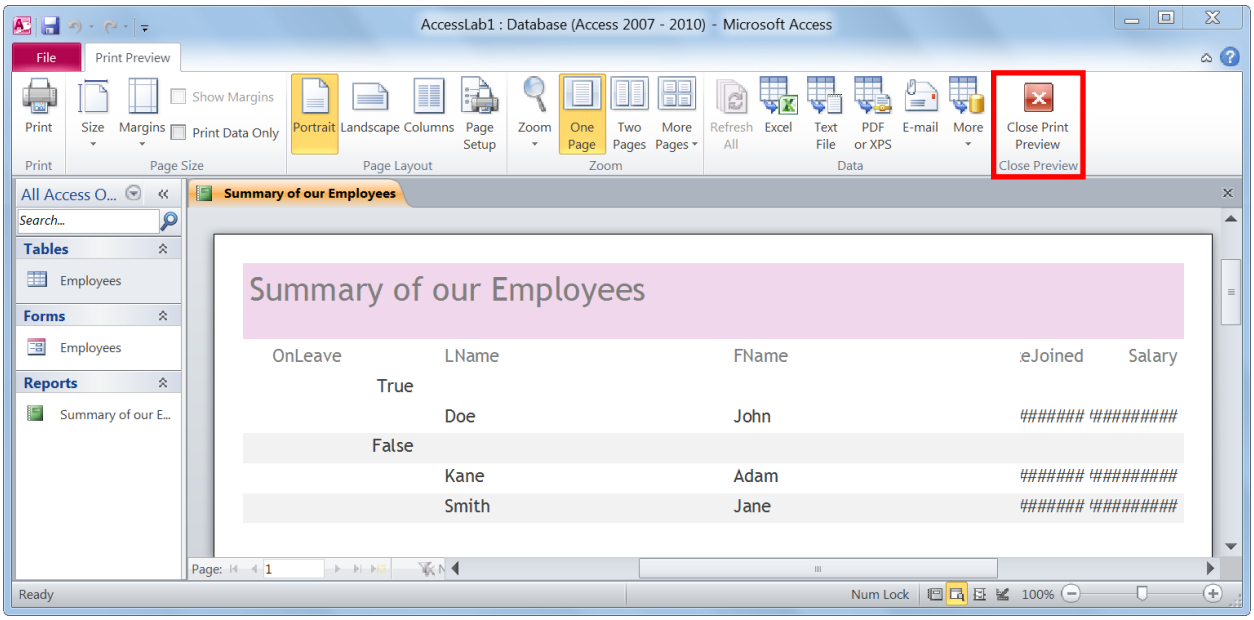
You can also select different layouts for your report. Keep it as stepped and press *Next*.



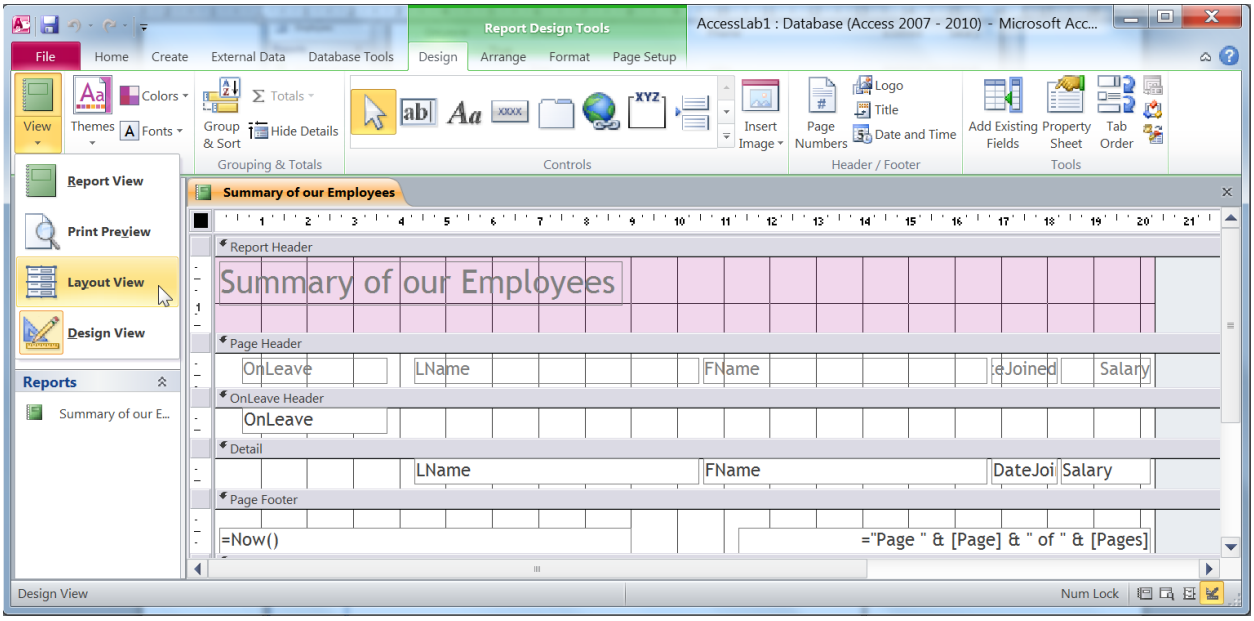
Finally, type in your report title and press *Finish*.



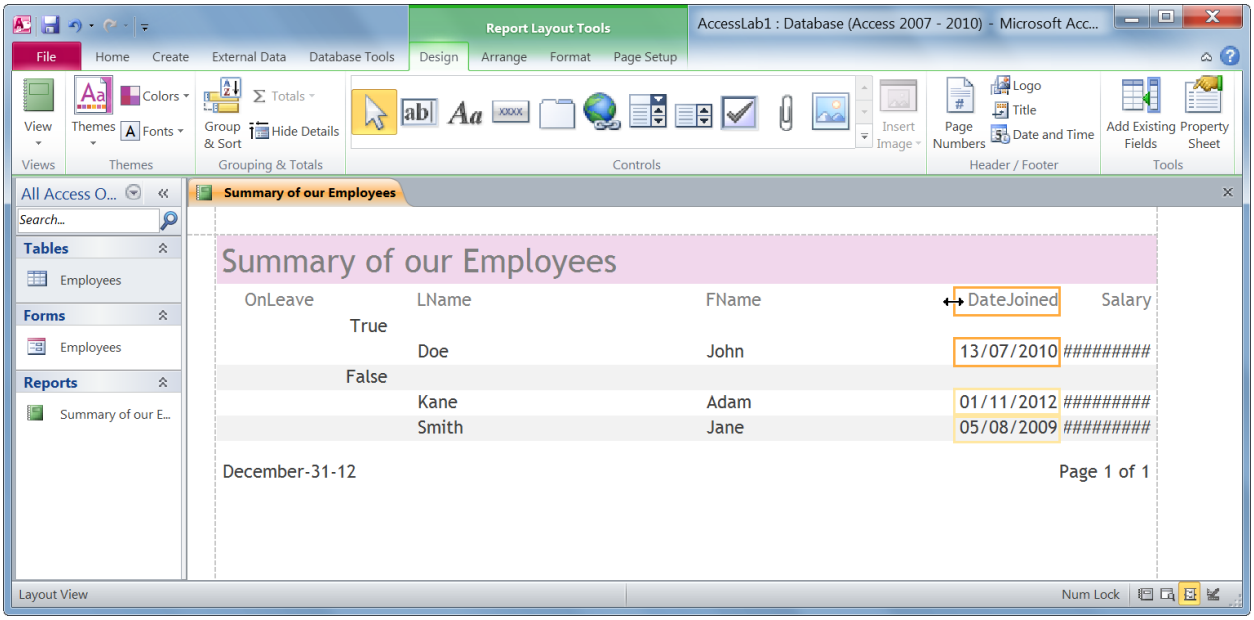
You can now see the print preview of your report. You will notice that there are a few problems with the report – there are some '#' signs in DateJoined and Salary columns and the title of the DateJoined is not entirely visible. Close the print preview pressing the *Close Print Preview* button so that we can fix these problems.



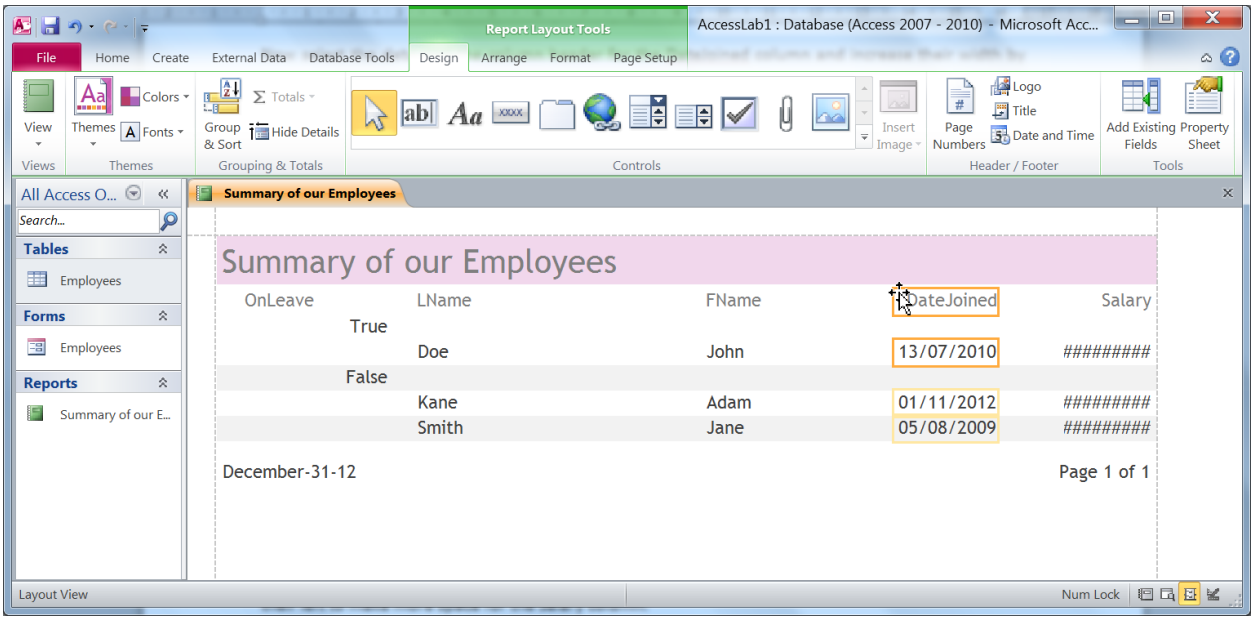
Go to the *Layout View* of the report from the *View* button under the *Design* tab as shown below.



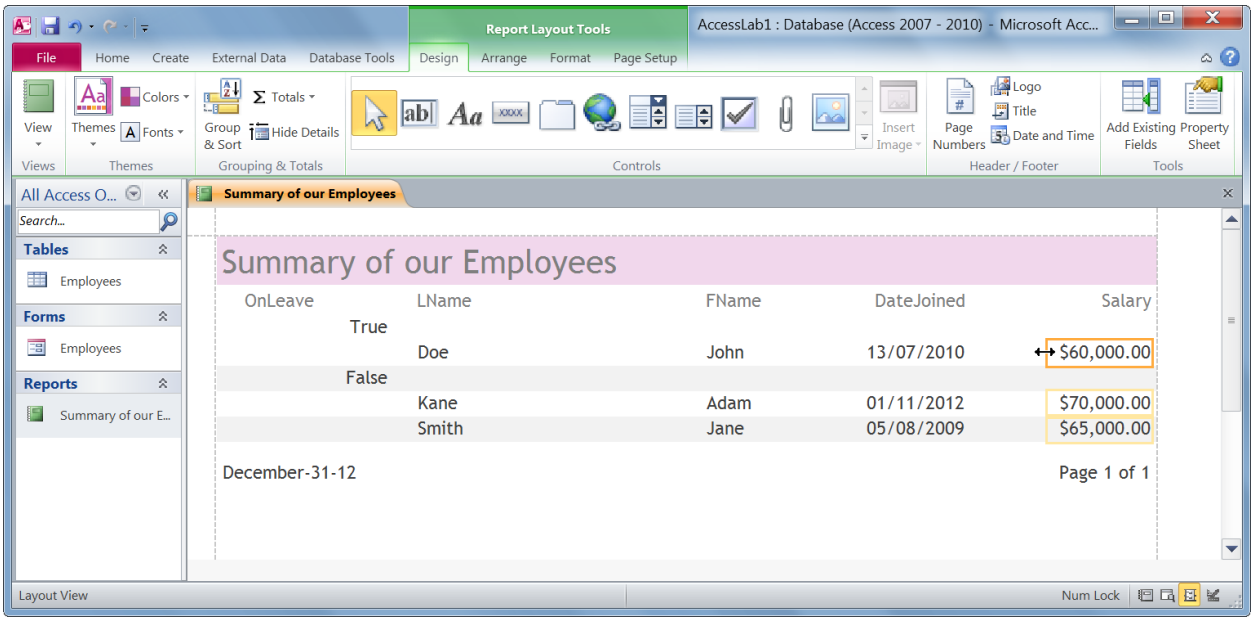
Now select the data and the column header (by clicking on them keeping the Ctrl button pressed) for the DateJoined column and increase their width by dragging their left borders with the mouse as shown below.



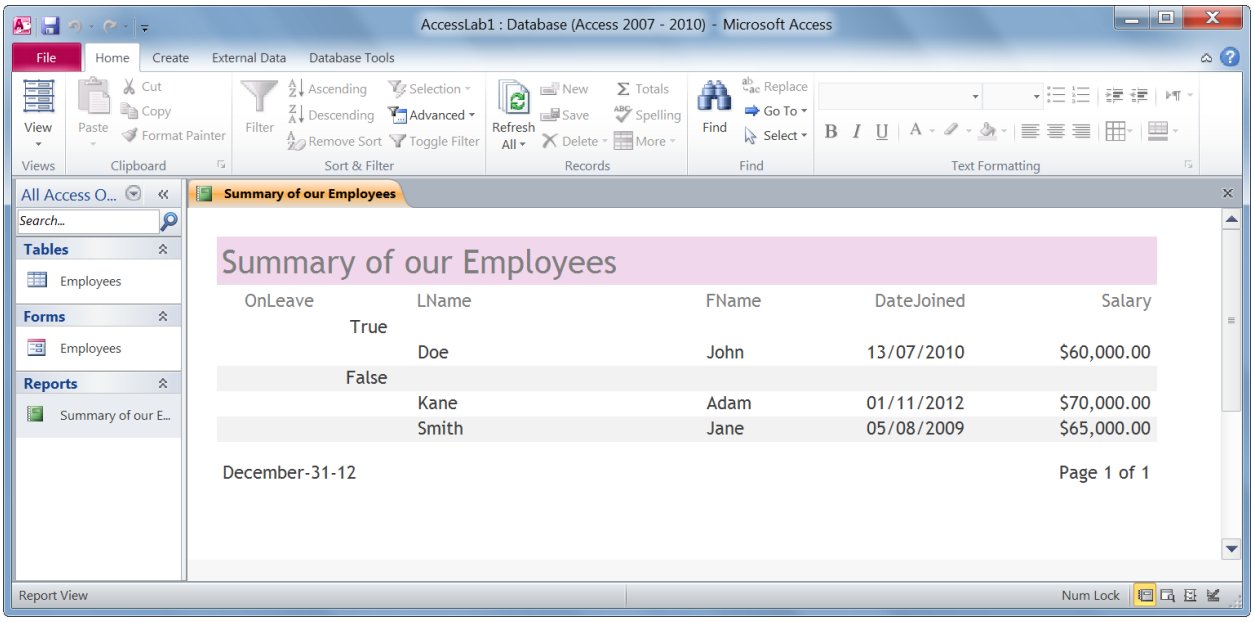
While having the data and the column header for the DateJoined column selected, move them towards their left by dragging with the mouse as shown below.



This will make more room for the Salary column. Now select the data in the Salary column (by clicking on them) and increase their width by dragging their left borders with the mouse as shown below.



Next, check out your report by going to the *Report View* from the *View* button under the *Design* tab.



Lab 2: Relationships and Queries

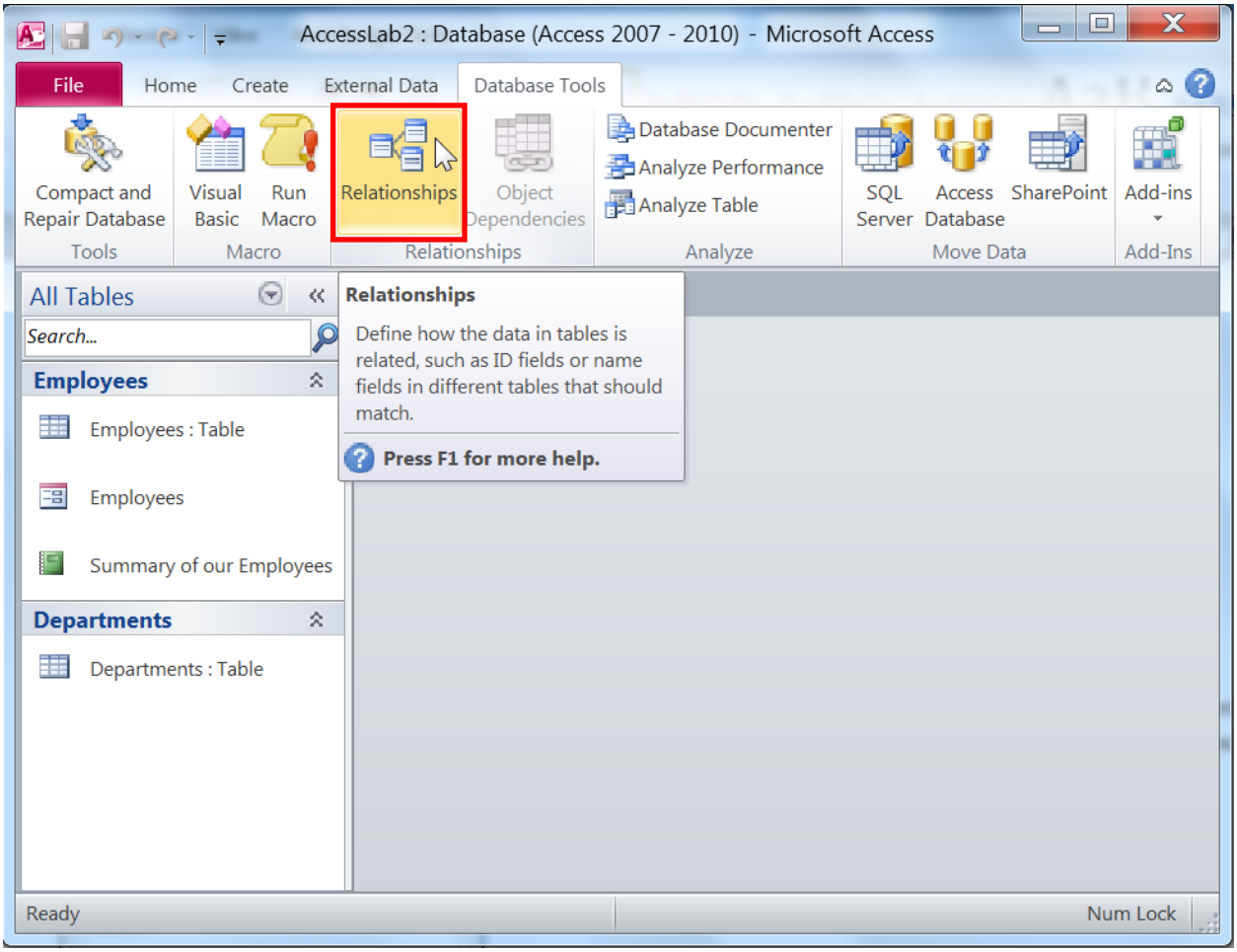
Relationships

Tables are related by *relationships* in order to minimize data redundancy. Such redundancy is both wasteful and can lead to anomalies as explained in your text.

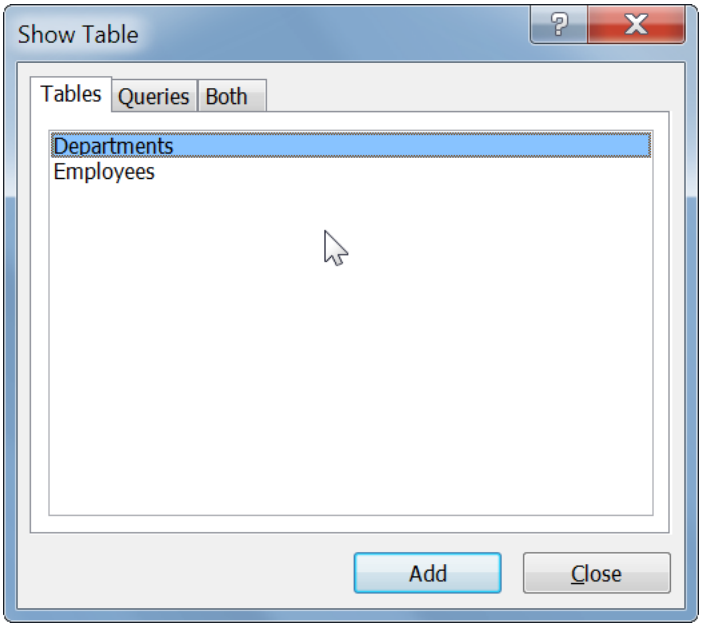
Open AccessLab2.accdb to see how a relationship is useful. The database shown is an extension of the database that was created during the last lab. A new table named Departments has been added to the database. In addition, a Department field has been added to the Employees table.

Since employees work for departments, we need to link or relate these tables together. This allows us to determine which department an employee works for. This is established by linking the Department field in the Employees table and the ID field in the Departments table. In addition, this will ensure that no data is duplicated between the tables. For instance, we do not need to duplicate the department details for each employee. We only need to enter the ID of the department and it will always link us to the the rest of the department information in the Departments table.

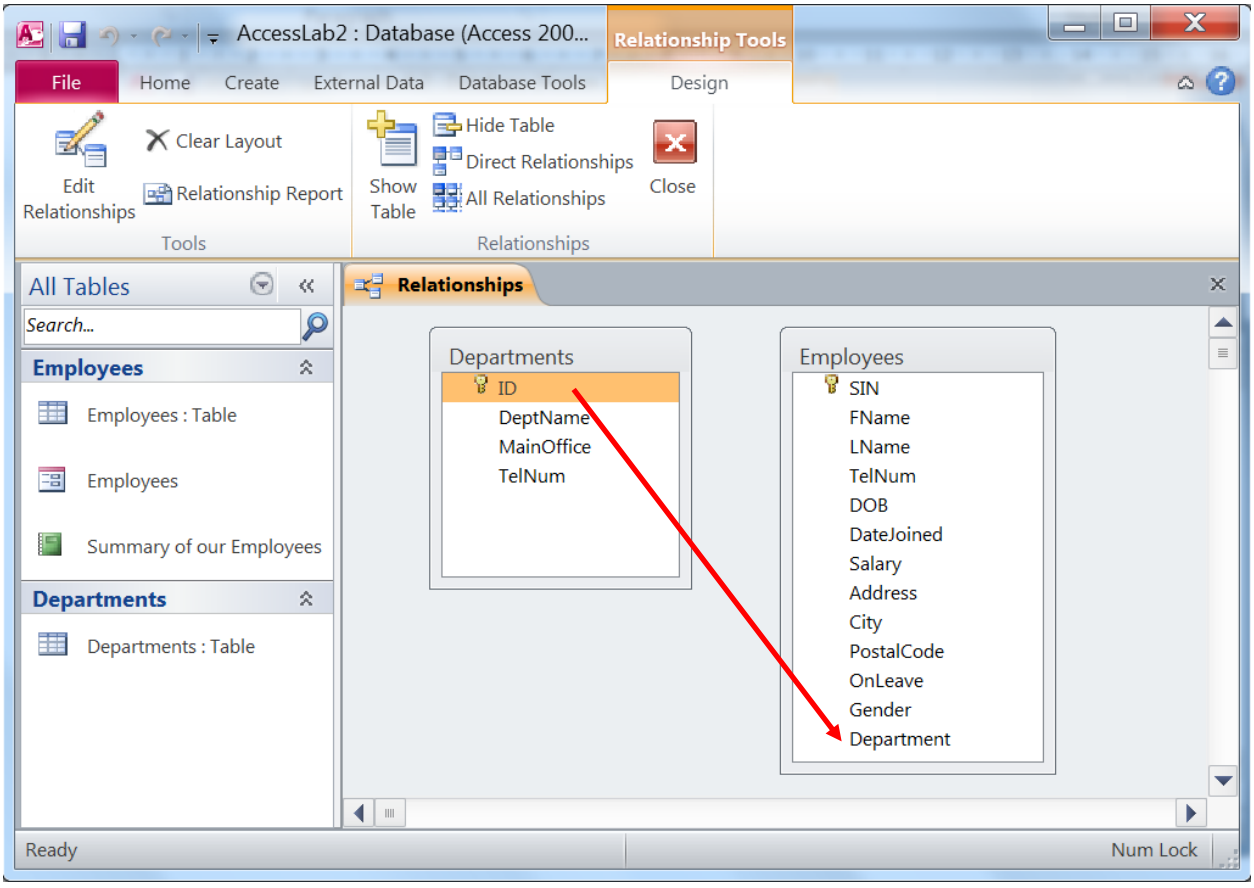
Click on the *Relationships* button in the *Relationships* group under the *Database Tools* tab.



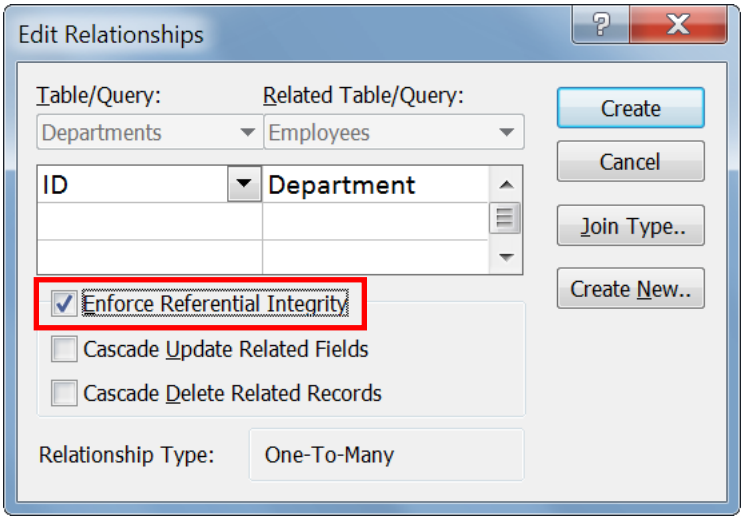
The *Show Table* dialog box pops up, requesting you to select the tables you wish to create relationships for. Add both tables and press *Close*. You can always bring this dialog box up back by pressing the *Show Table* button in the *Relationships* group under the *Design* tab.



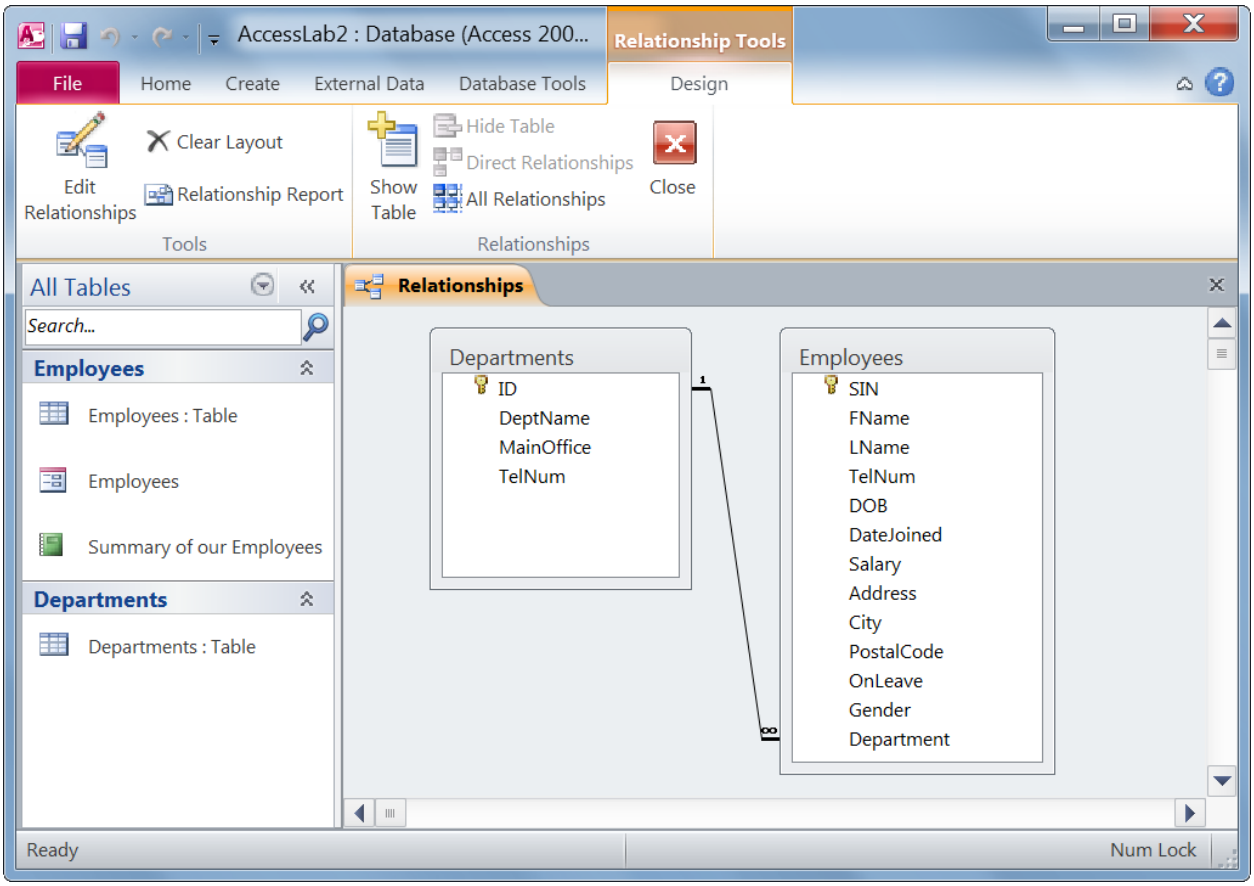
To create a relationship between ID in Departments and Department in Employees, drag the ID field to the Department field. You should always drag from the primary key to the foreign key. In other words, drag the *original* field to the related field.



The *Edit Relationships* dialog box pops up. Make sure the *Enforce Referential Integrity* box is checked and then press *Create*. Enforcing referential integrity ensures that each Department ID entered in the Employees table has a corresponding record in the Departments table. For instance, you will not be allowed to delete a certain department from the Departments table as long as there are some employees who are working for this department. (Note: You need to make sure your tables are closed before creating these relationships).



The relationship has been created!



Save your changes.

Exercise 1

Try entering a value for Department in the Employees table that does not correspond to one in the Departments table. What do you notice?

Now fill in the Department field as follows:

SIN	Department
123-456-789	1
444-333-555	1
657-656-756	2
877-334-556	2
987-654-321	2

Exercise 2

Adam Kane is the manager of the IT department. Cyris Amjad is the manager of the marketing department. Create a field in the Departments table named MGR_SIN that contains the SIN of the

manager. Use the same Input Mask we used in the Employees table. Make sure you create the appropriate relationship.

Queries

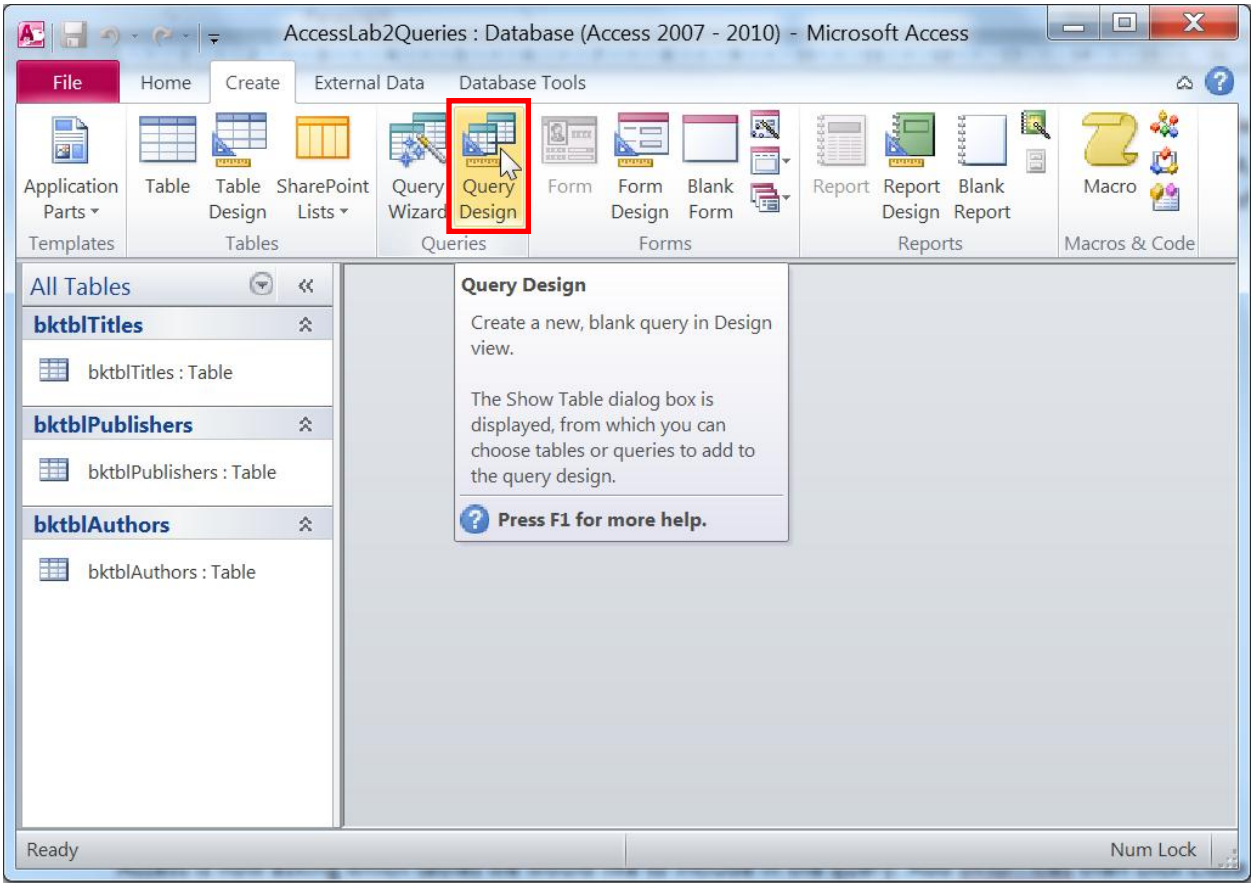
Queries are questions submitted to the database. They can be used to filter, perform calculations on, and summarize data.

The simplest type of query is one which shows some or all of the data from one table. Open the file AccessLab2Queries.accdb to get started. Take a moment to familiarize yourself with the database. Also, take a look at the *Relationships* in the database. This should give you a good overview of the type of data we are dealing with here.

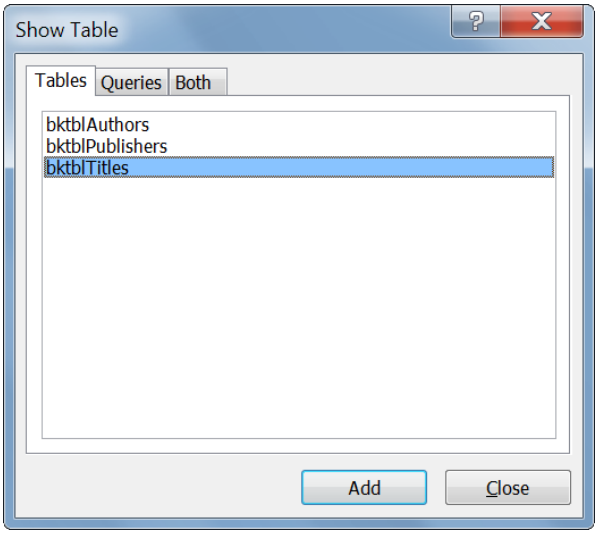
The database keeps tracks of books, along with their publishers and authors.

Let us start by creating a query showing all the information in bktblTitles.

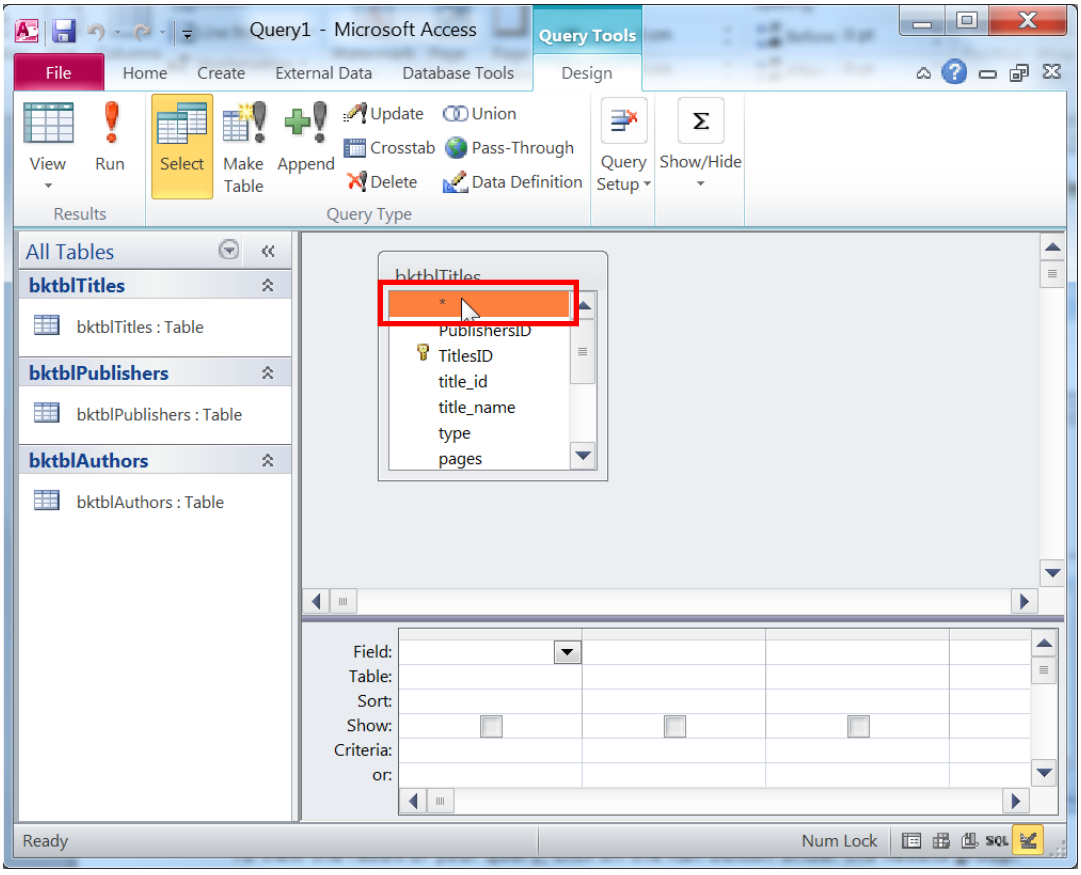
Under the *Create* tab, click on the *Query Design* button from the *Queries* group.



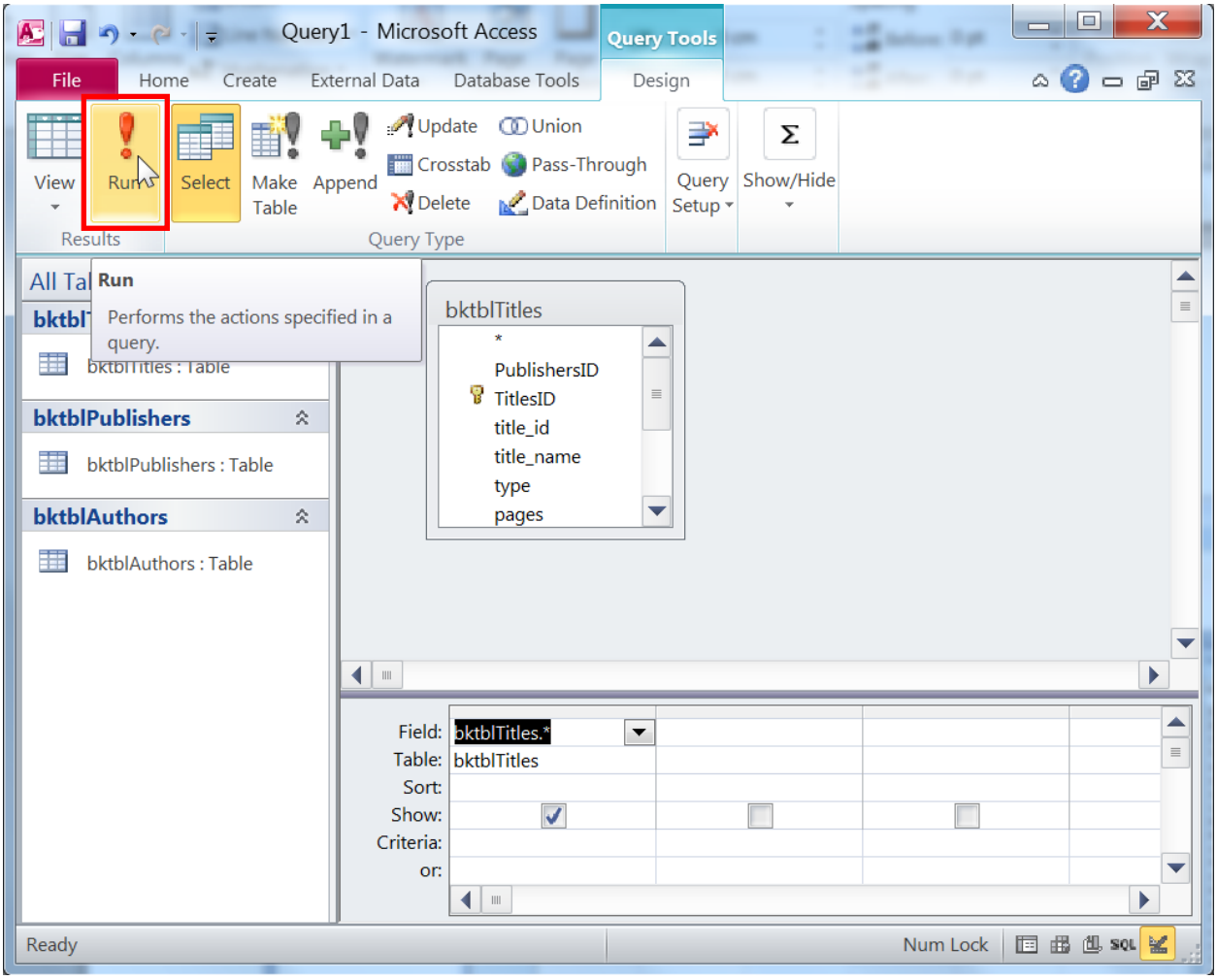
Access is now asking which tables we would like to include in the query. Add bktblTitles and press *Close*. Remember, you can always bring up the *Show Table* dialog box either by right-clicking, or by selecting the *Show Table* button from the *Query Setup* group.



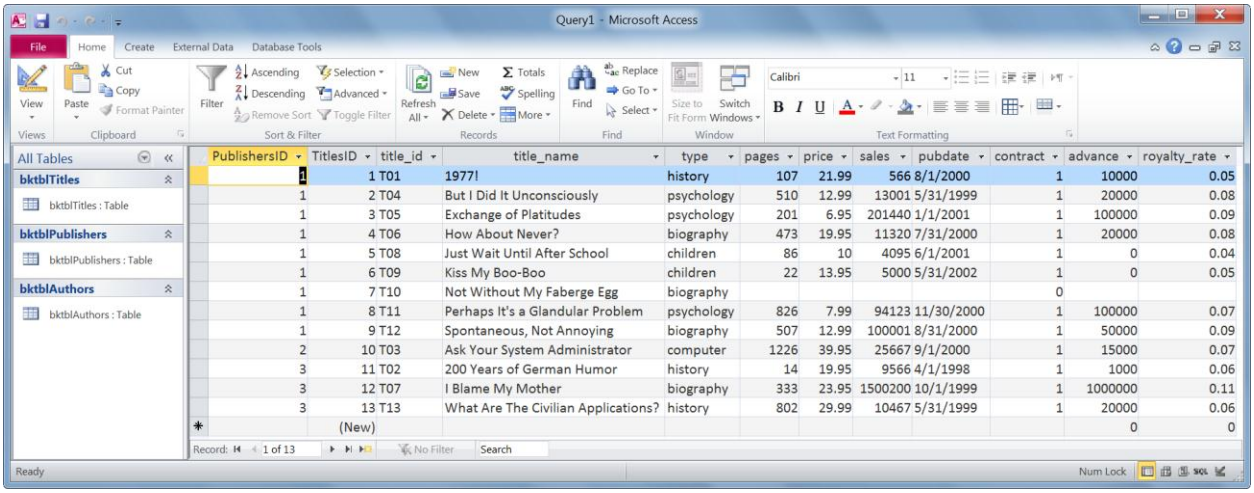
Access now shows you the design view of your query. This form of queries is called *query by example* or *QBE*. To add all the fields of the table into the query, double-click the * under bktblTitles.



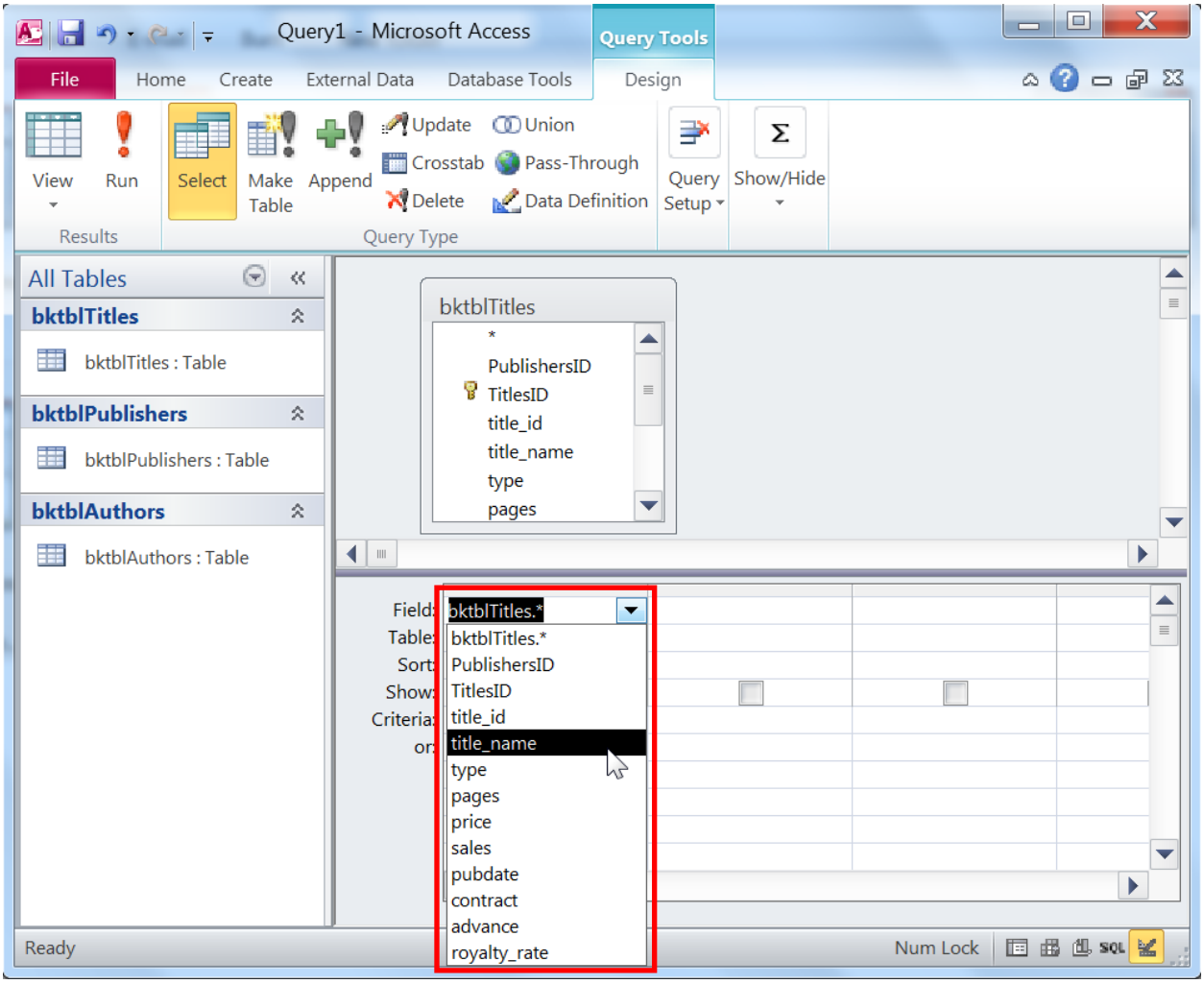
Now, to see the result of your query, click on the *Run* button under the *Results* group.



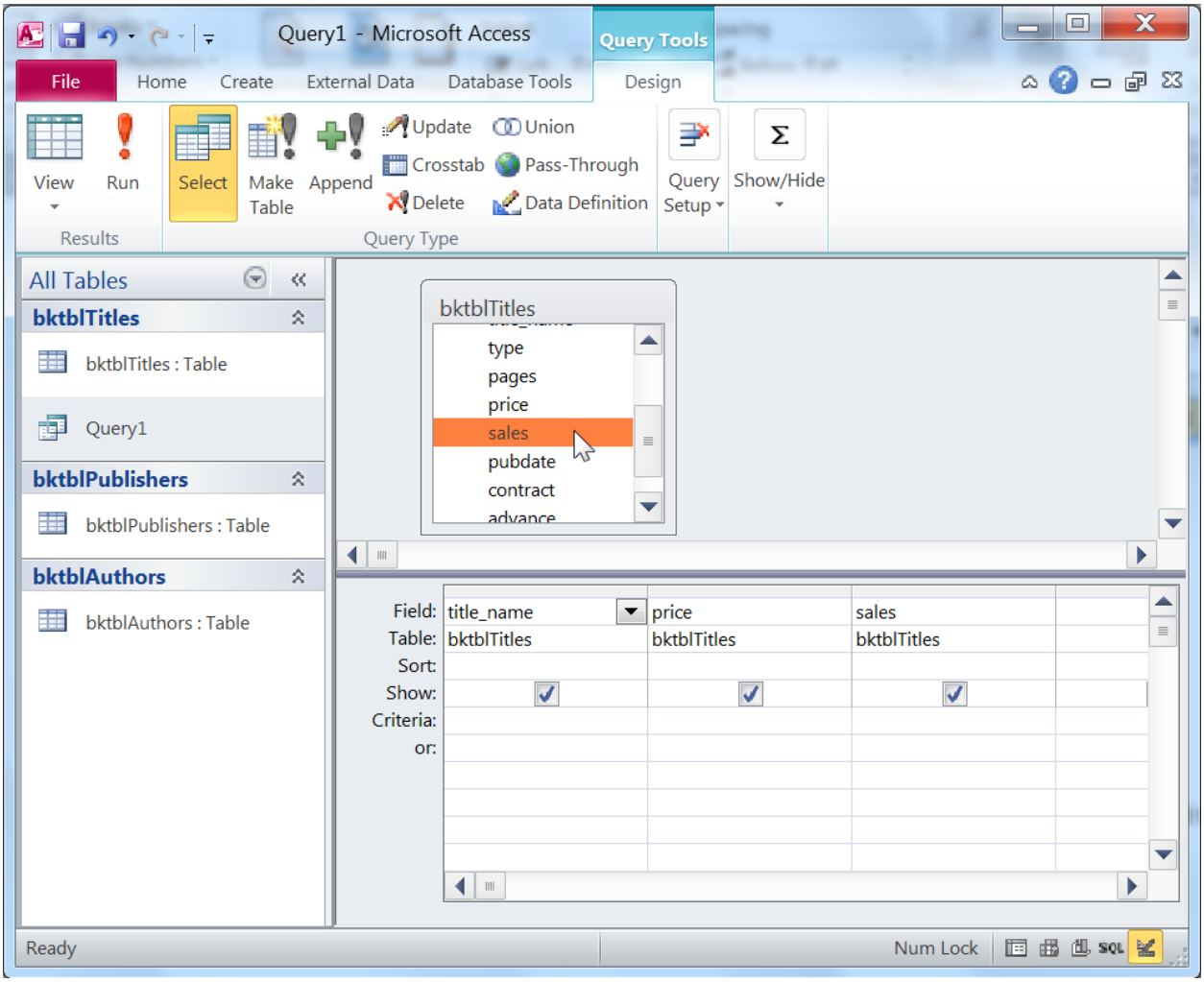
The query result now shows exactly what you asked for: the entire bktblTitles table.



Go back to the design view of the query. Now let us try to be more selective of what fields we wish to include in our query result. Let us only include the title_name, price, and sales. Click on the arrow next to bktlbTitles (shown in the red rectangle below) and select title_name instead of the * field.



To add the two other fields, double-click on price and then on sales.



Try running the query again to see the result.

Exercise 3

Create a new query that shows all the information in the bktblAuthors and bktblTitles tables.

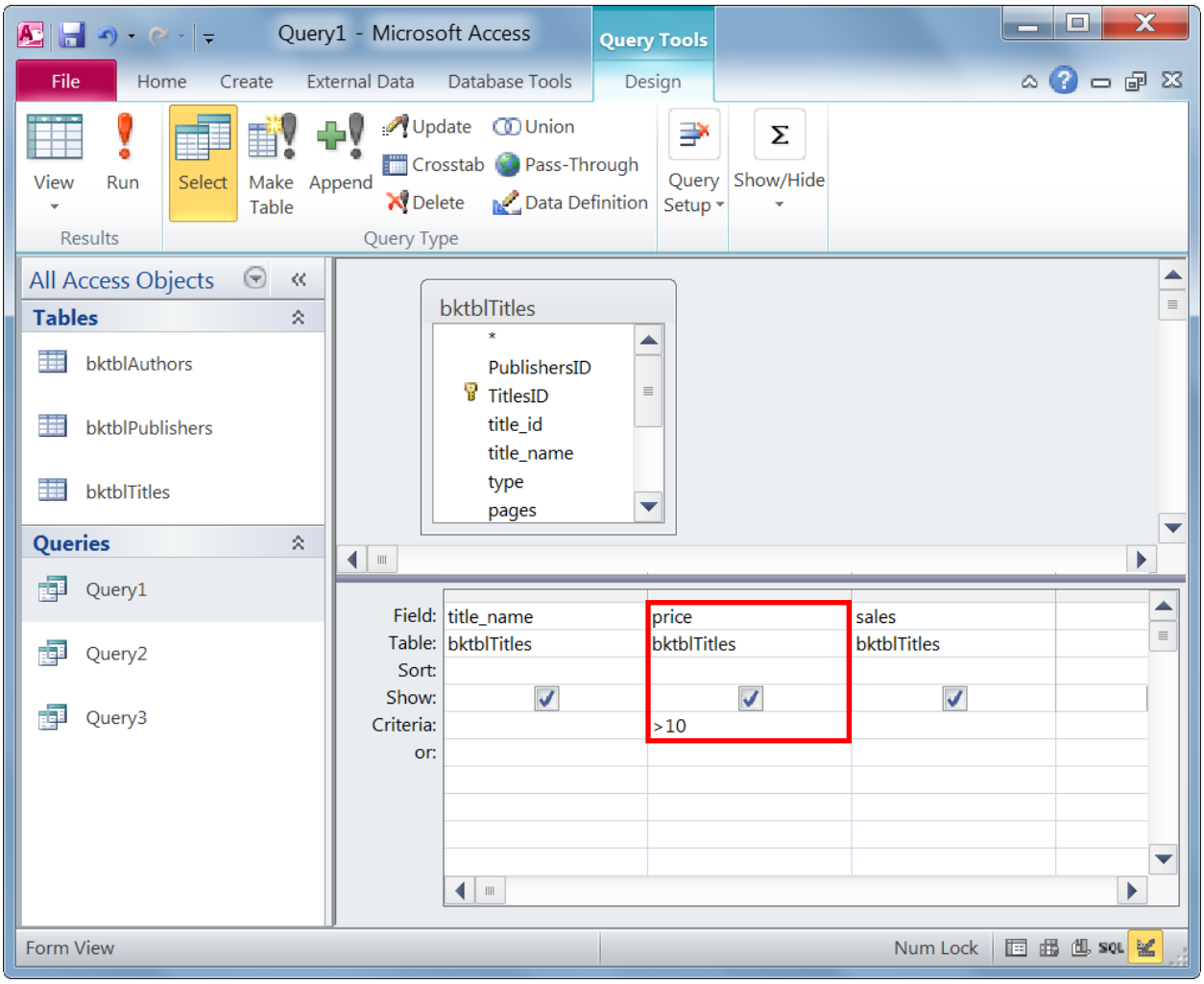
Exercise 4

Create a new query that displays title ID, title name, the publisher's name, and the author's first and last names.

Criteria in queries

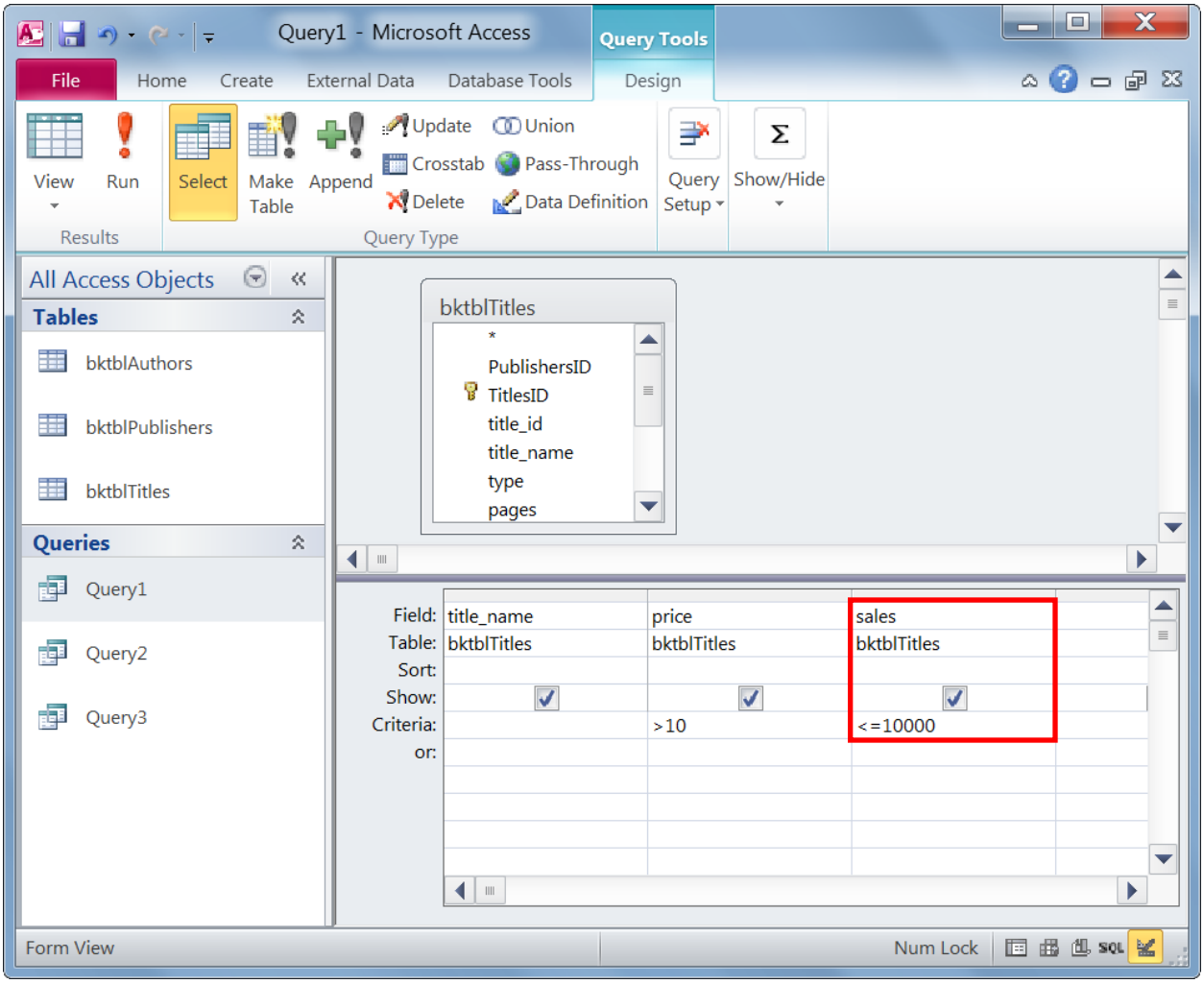
We can specify criteria for fields to limit query results based on their values. Open the first query we created. Make sure you are in the design view.

Let us display only the book entries with a price that is more than 10. Type in >10 in the *Criteria* field of price.



Run the query to see the result.

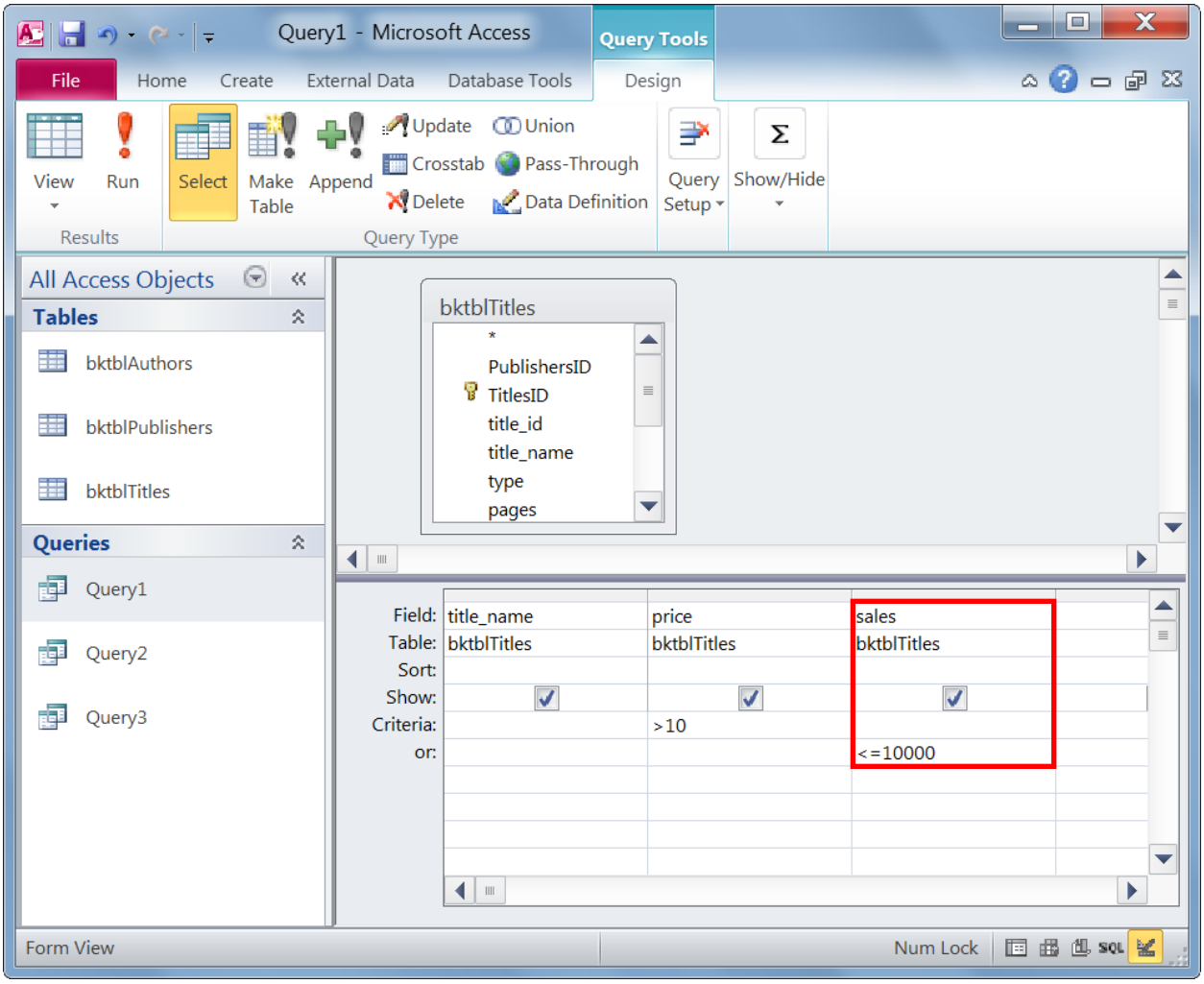
Now, what if we wanted to view all book titles whose price was greater than 10 AND at least 10000 copies? Enter <=10000 as the sales criterion.



Run your query and view its result.

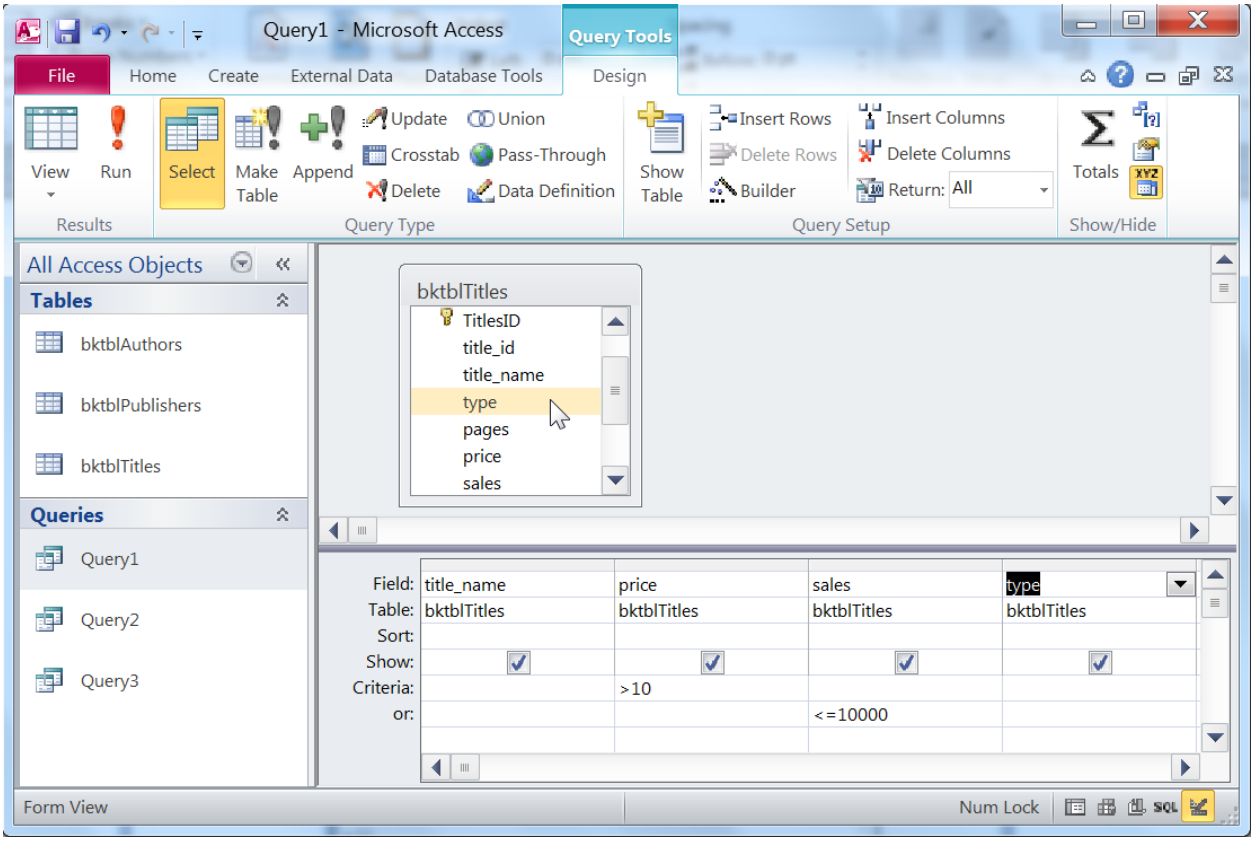
The previous query allows us to view any book title which has a price greater than 10 **and** a sales of at least 10000 copies. What if we want to display any book title with a price >10 **or** one with a sales amount <=10000?

This is where the *or* field comes in handy. Move the <=10000 criterion from the criteria field to the *or* field.

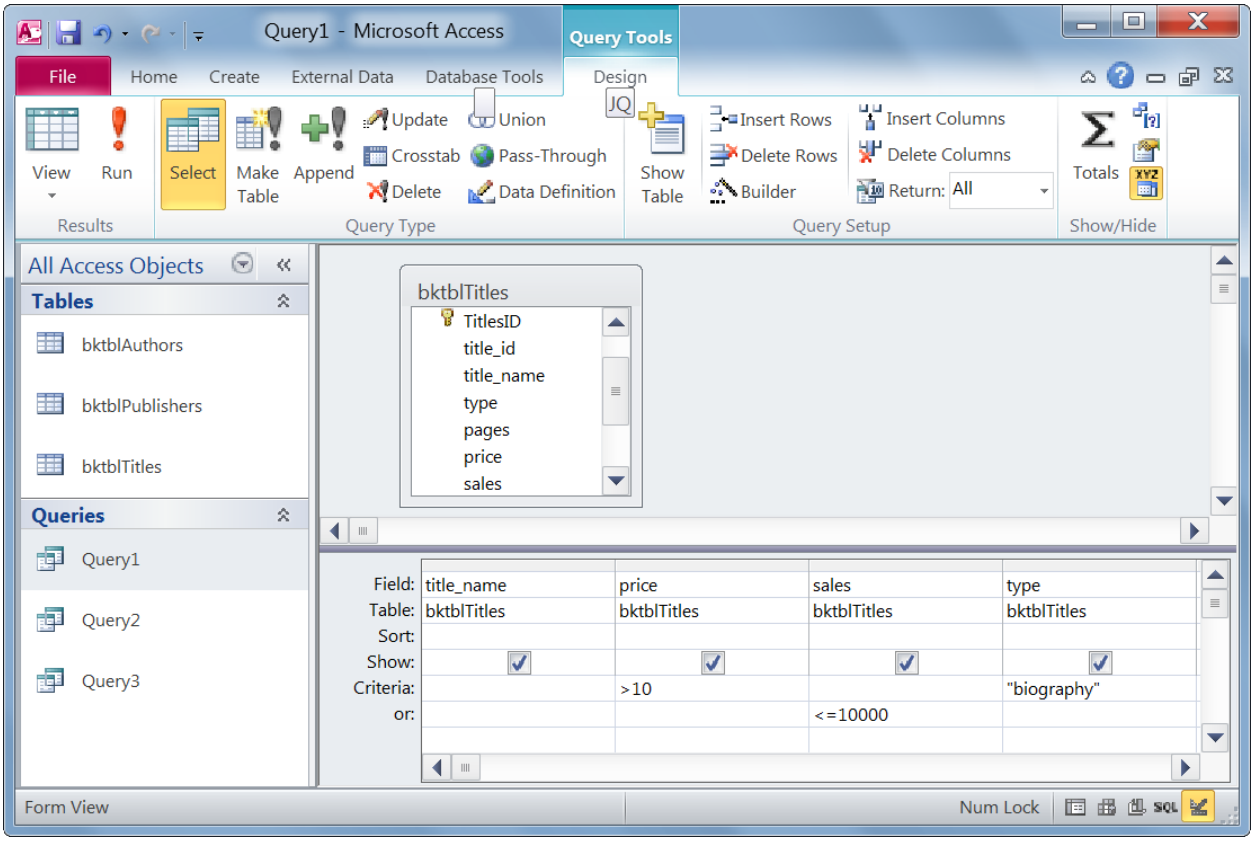


Now the query results show any book title with a price >10 or a sales amount <=10000.

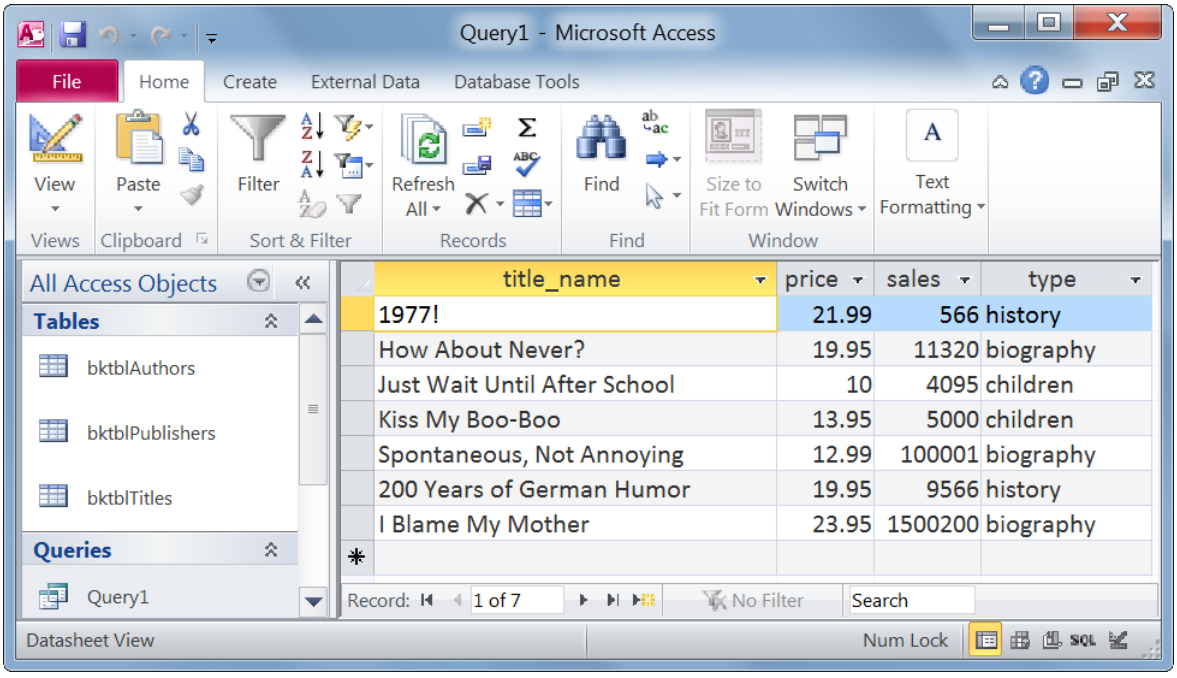
We can also add query criteria to text fields. Add the field **type** from bktblTitles to the query.



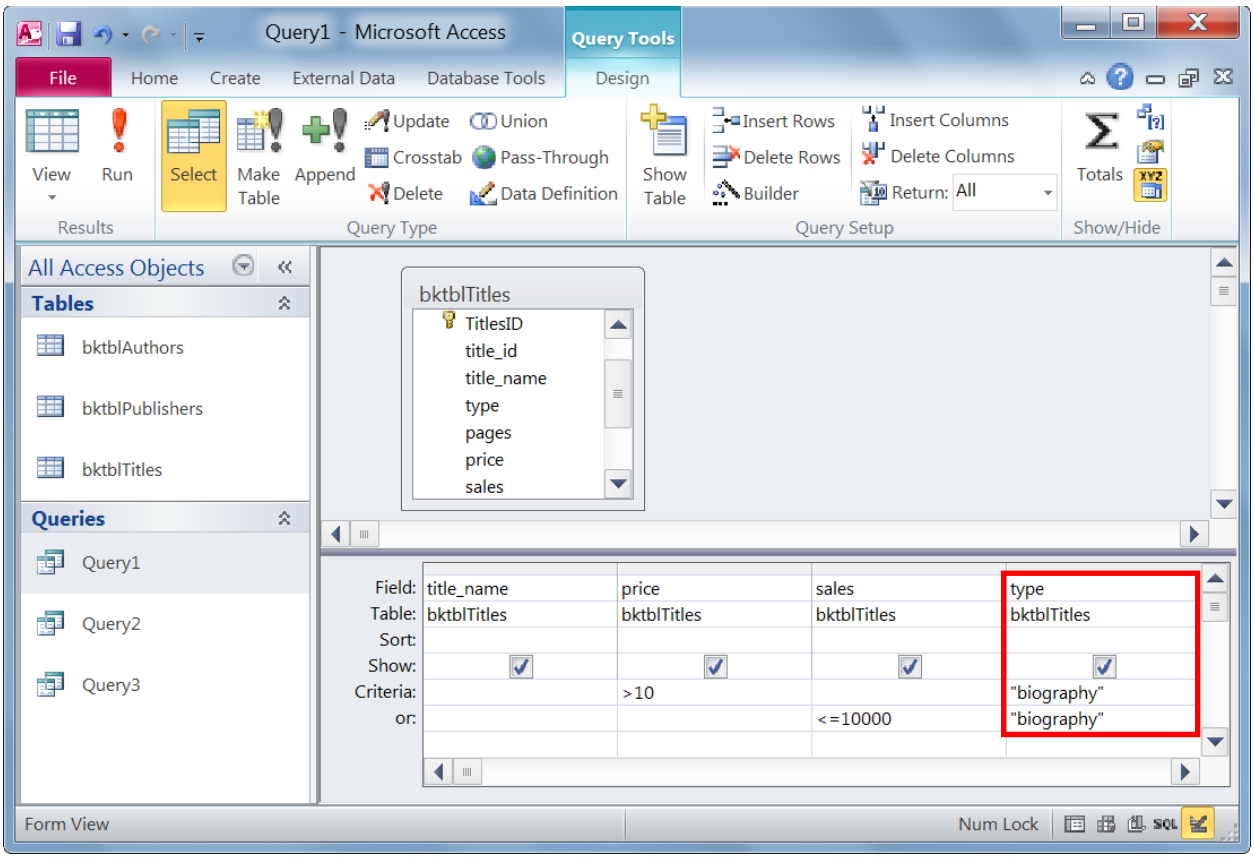
Let us only show book titles of type “biography”.



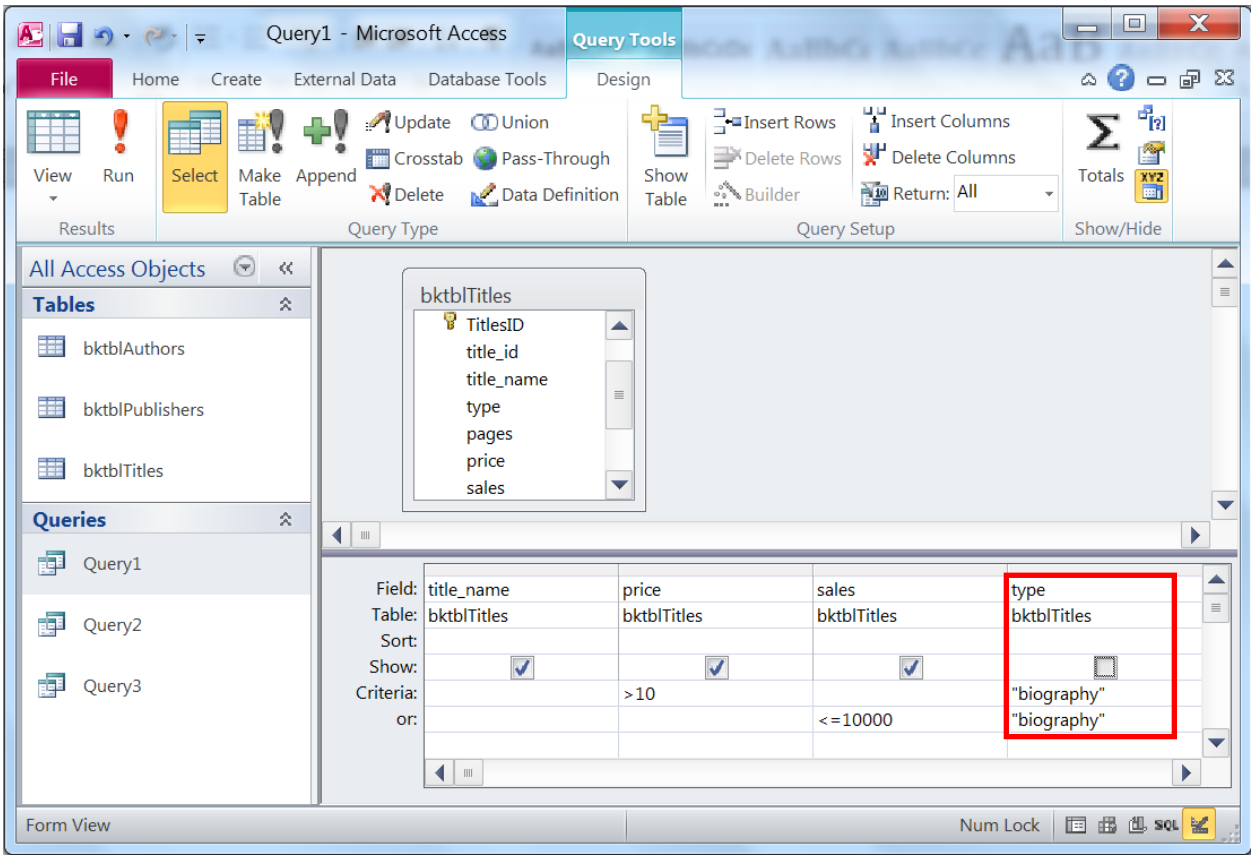
Now take a look at the results.



Notice how the **or** clause was applied in this query. The results show book titles with a price >10 and of type biography, or ones with sales <=10000. If we would like to view only biography books with price>10 or biography books with sales amount <=10000, we need to repeat the biography criterion in the *or* field.



We can also use a field to limit our query results without displaying that field in the query. We do this by unchecking the Show box. Do this for the type field.



Now the type is still included in the query criteria but does not show up as a field in the query result. To include in the result, check the *Show* box under type.

Exercise 5

Modify the query that you created in Exercise 4 so that it only shows records corresponding to Abatis Publishers. Only show Abatis Publishers records with royalty rates less than 0.08 or with advances that are less than 30000. Do not show the advance and royalty rates fields in the query.

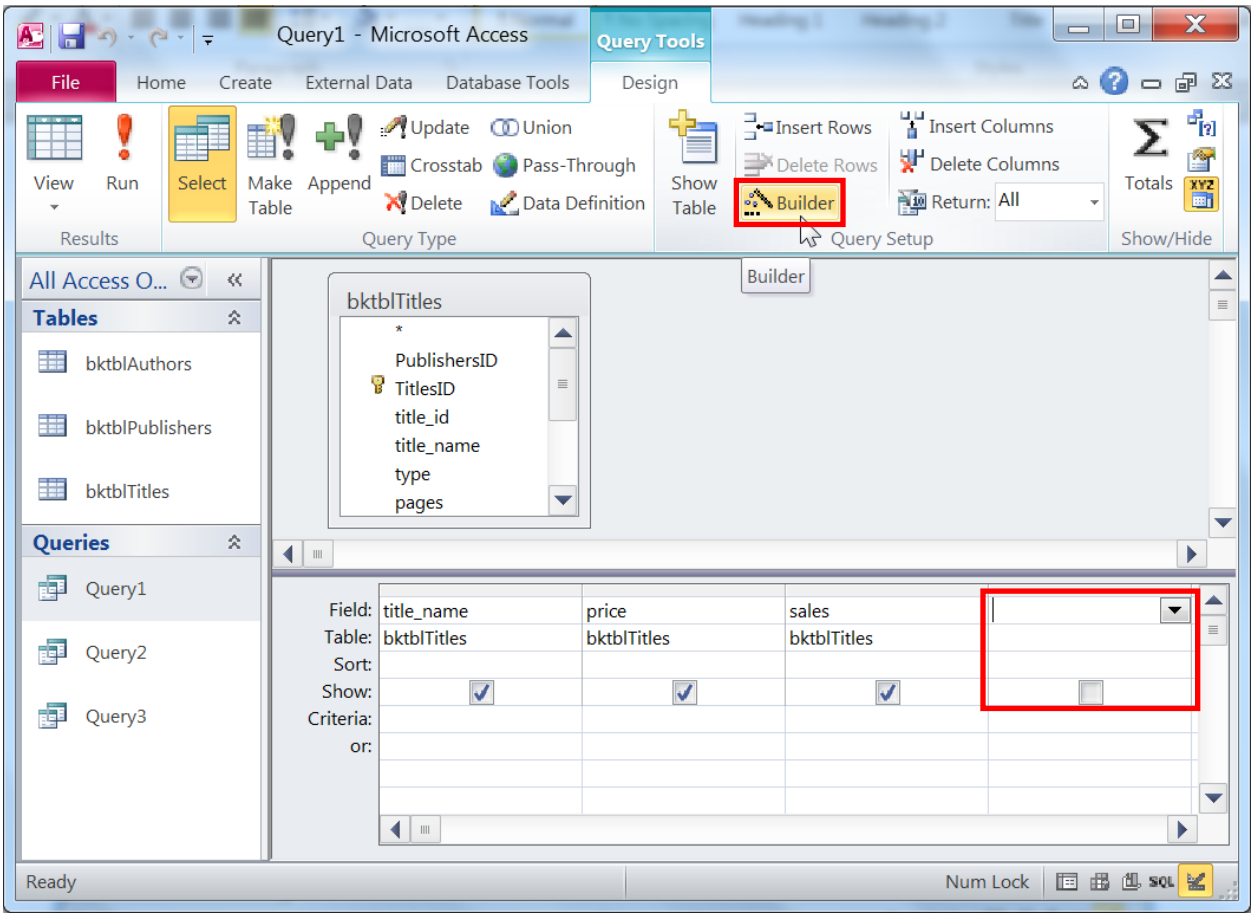
Lab 3: Advanced Queries

Custom calculations in queries

Sometimes, you want to specify certain calculated variables that would show up in your query result. Access allows you to create custom expressions in your query, similar to the formulas used in Excel.

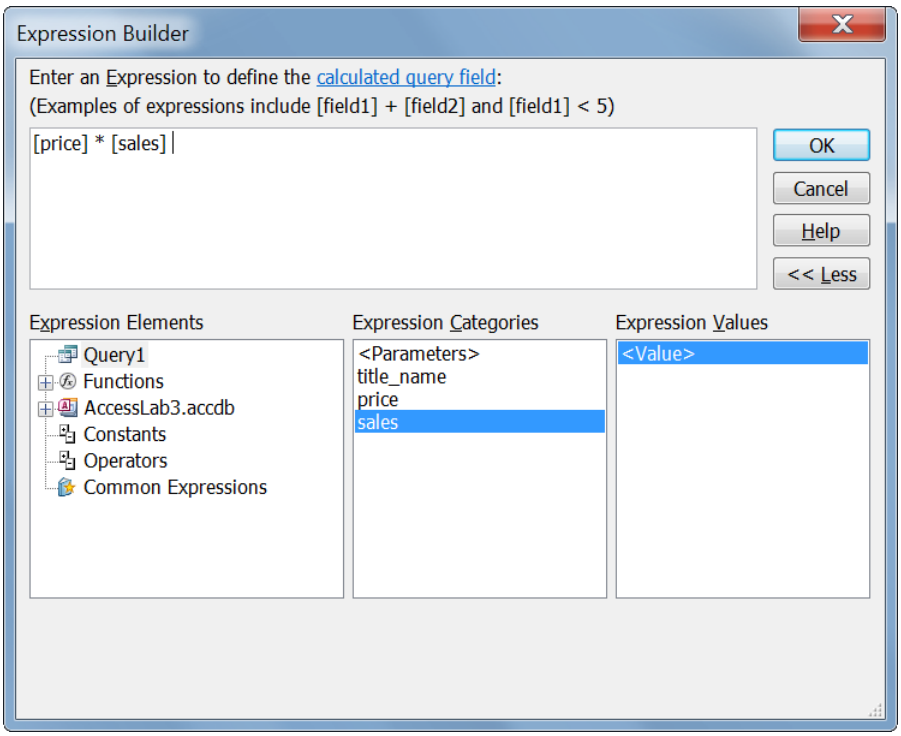
Open the file AccessLab3.accdb and go to the design view of Query1.

We need to calculate the profit made from book sales, which has a formula of price*sales. Place the cursor in a new field, then click the *Builder* button under the Query Setup group.

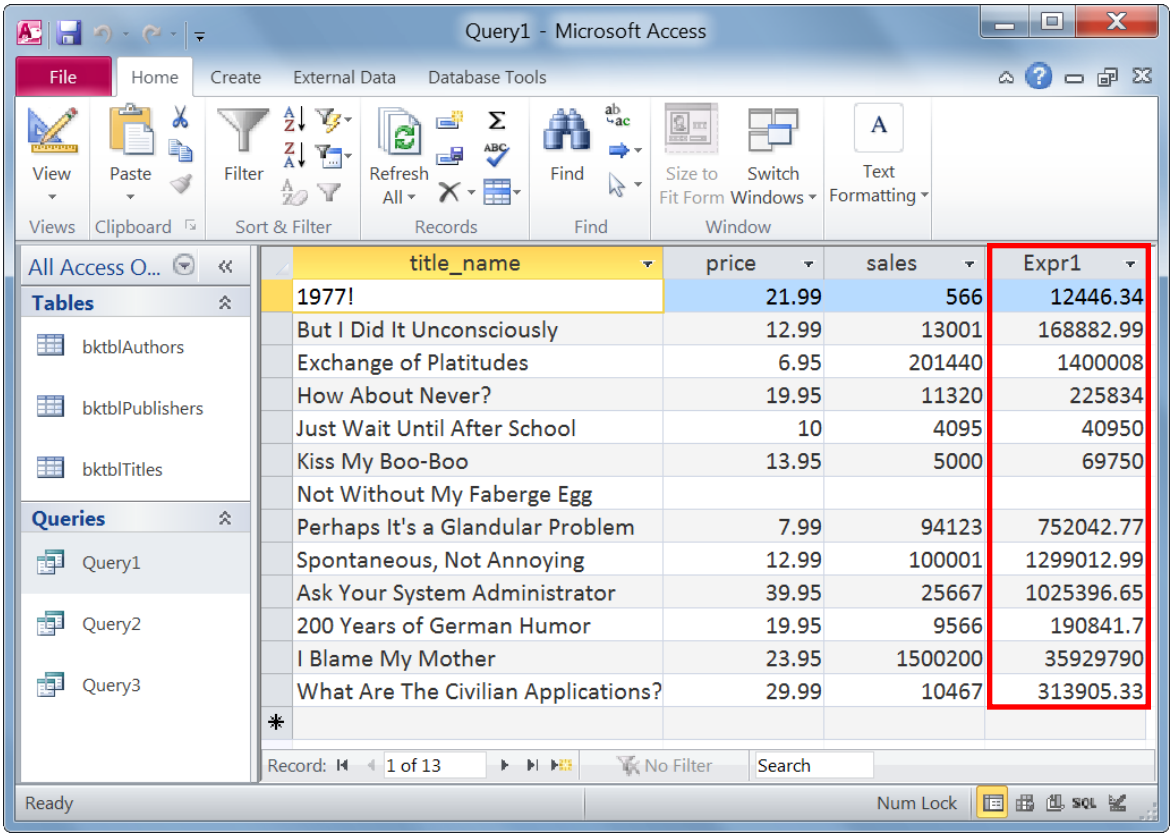


The Expression Builder that pops up allows you to create a formula using built-in functions and fields from queries and tables in your database. Since both fields that are needed for our expression are in the query, we do not need to navigate further.

Double-click price and then type in * (the asterisk is the multiplication sign). Next, double-click on the sales field. Then, press the OK button.



Run your query and notice how the new field is named Expr1 by default.



To give it a more meaningful name, return to the design view and change your expression from *Expr1: [price]*[sales]* to *Profit: [price]*[sales]*.

The name before the colon is the heading of the field.

Exercise 1

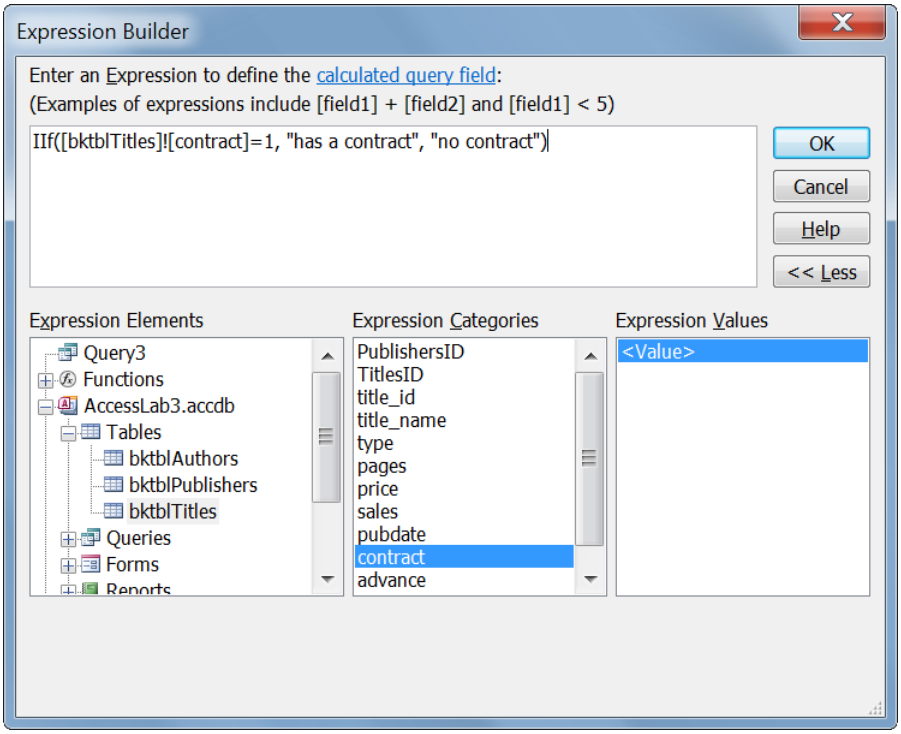
The expression we created for profit is too simplified. Modify it so that the profit is calculated as: ((sales*price)-advance) * (1-royalty rate).

Exercise 2

In Query 3, remove the au_fname and au_lname fields from the query. Create a new field called Full Name that combines both names (with a space between them). Use the & operator.

We can also use IF statements in our expressions. IF statements in Access are almost the same as in Excel, except they use the keyword IIF instead of IF.

Go to the design view of Query3. In an empty field, open up the expression builder. We would like our new field to contain the value "has a contract" if the value of the contract field in bktblTitles is 1, and "no contract" if that value is 0. Write the expression as shown in the figure below.



Press OK when you are done. Make sure you modify the expression name so that it is called “Contract?” Run the query to view the results.

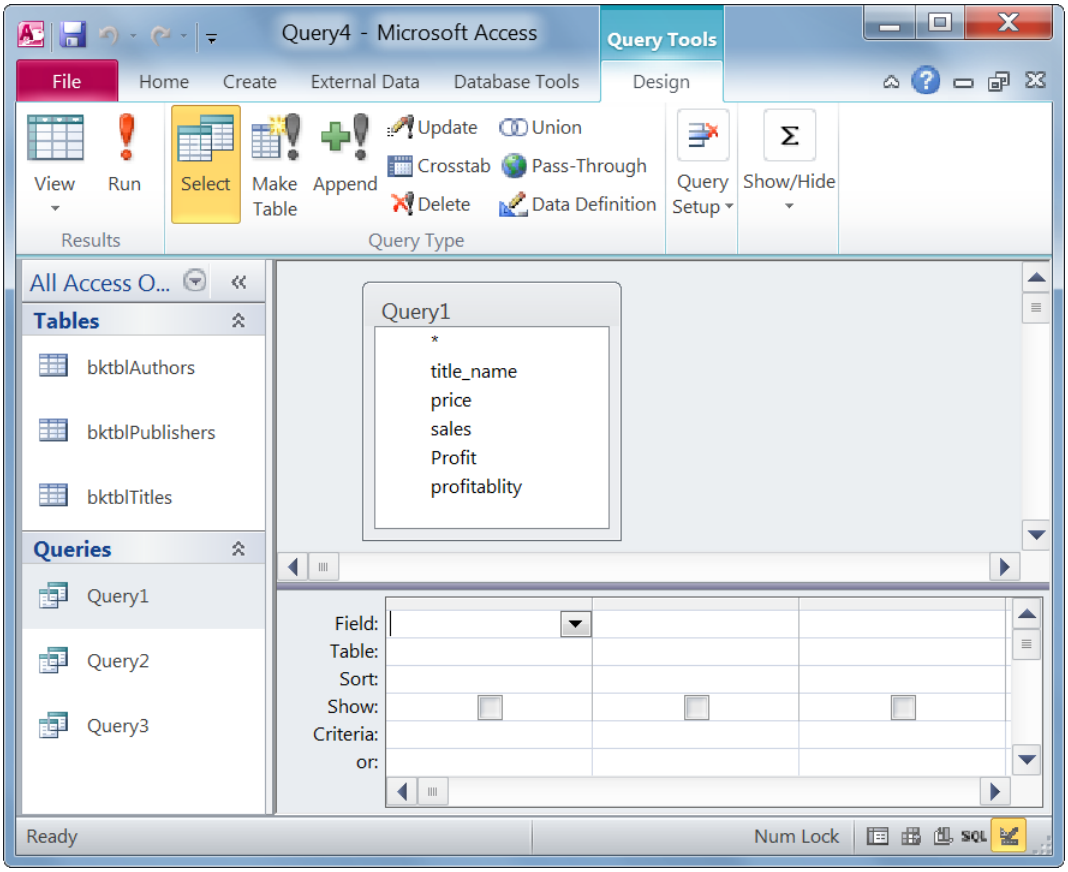
Exercise 3

Remember nesting if statements in Excel? You would better refresh your memory now. In Query1, create a new field called profitability. This field should display “very profitable” if the calculated profit was greater than 1,000,000; “profitable” if the profit was less than or equal to 1,000,000 and greater than 100,000; “OK” if the profit was less than or equal to 100,000 and greater than 10,000; and “not enough” otherwise.

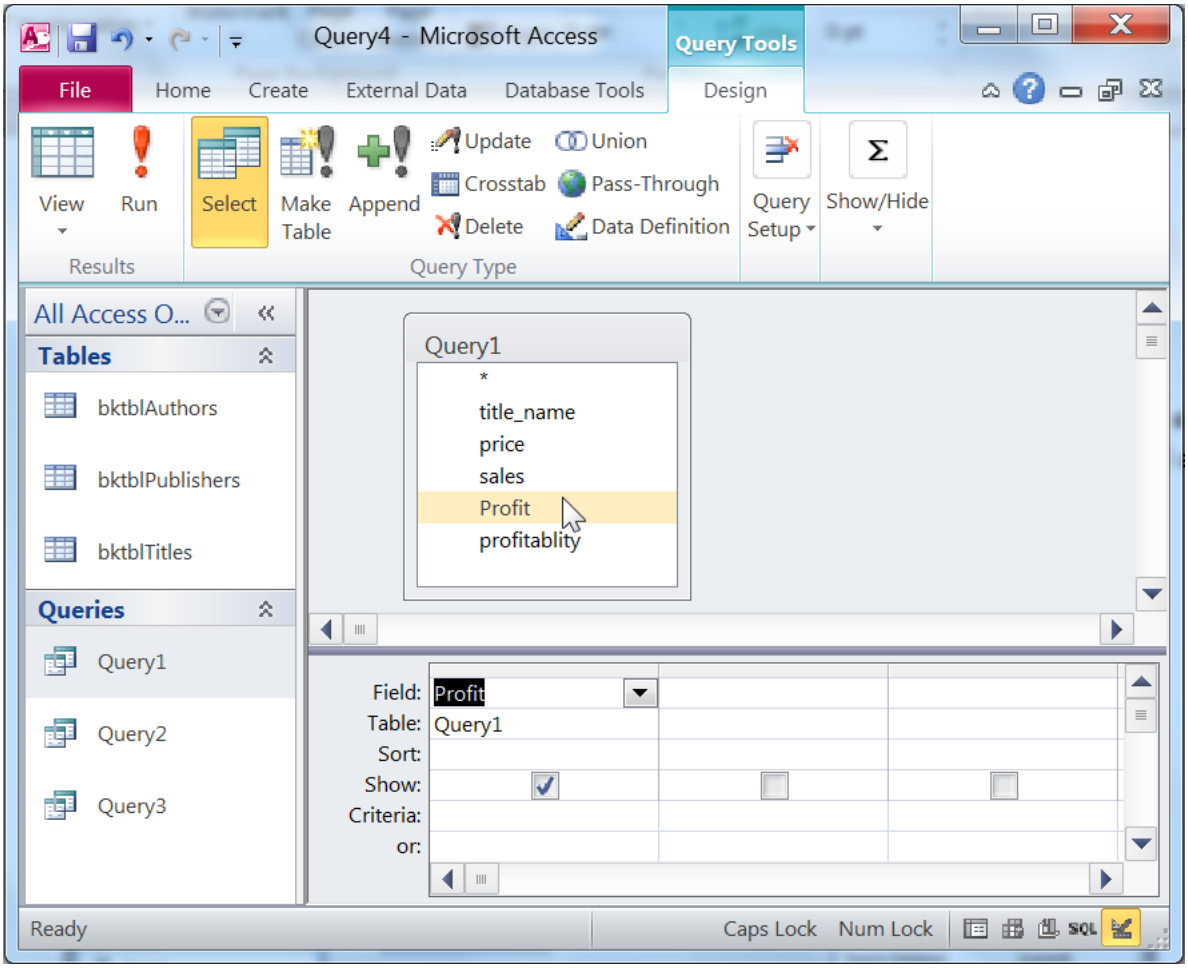
Aggregate functions

Aggregate functions are used when you need to perform a calculation on the data in one of your columns. For instance, the *sum* aggregate function can be used to add up all of the profits from the books.

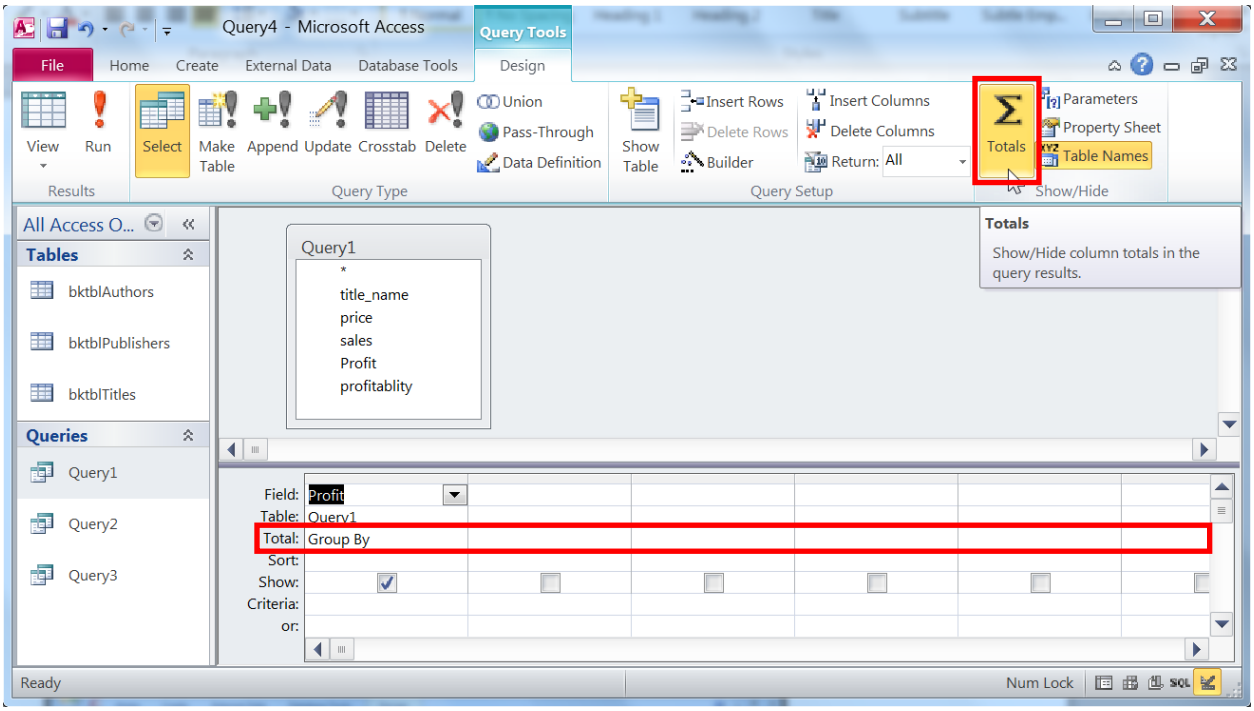
Create a new query (Query4) and add Query1 to it (the same way you would add a table).



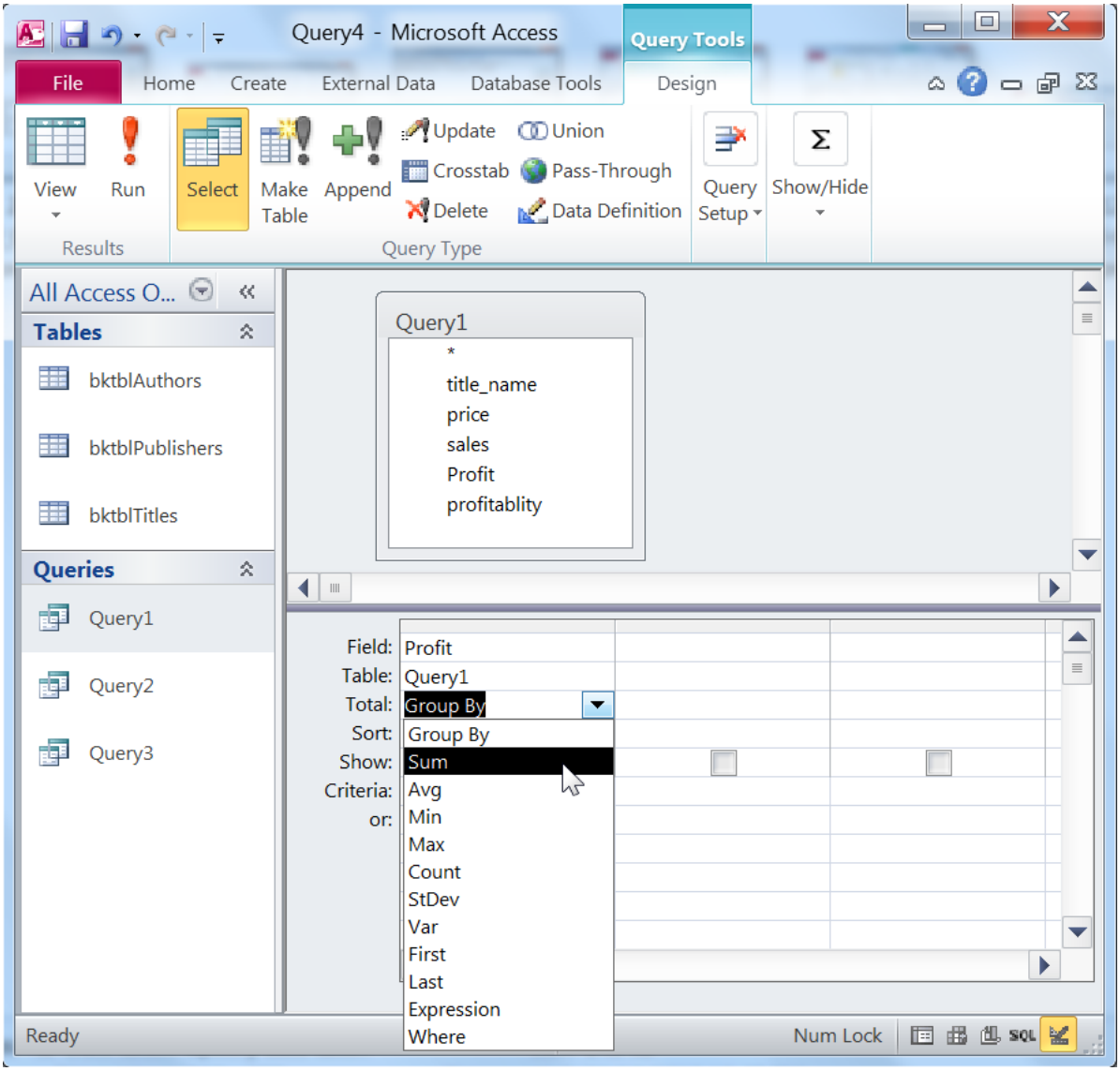
Add the profit field to the query.



Now press the *Totals* button in the *Show/Hide* group. When this button is pressed, we are able to use aggregate functions. Notice that once you press the *Totals* button, a new row named *Total* appears in the query design.



Click on the Total property of the Profit field and change the value from Group By to Sum.

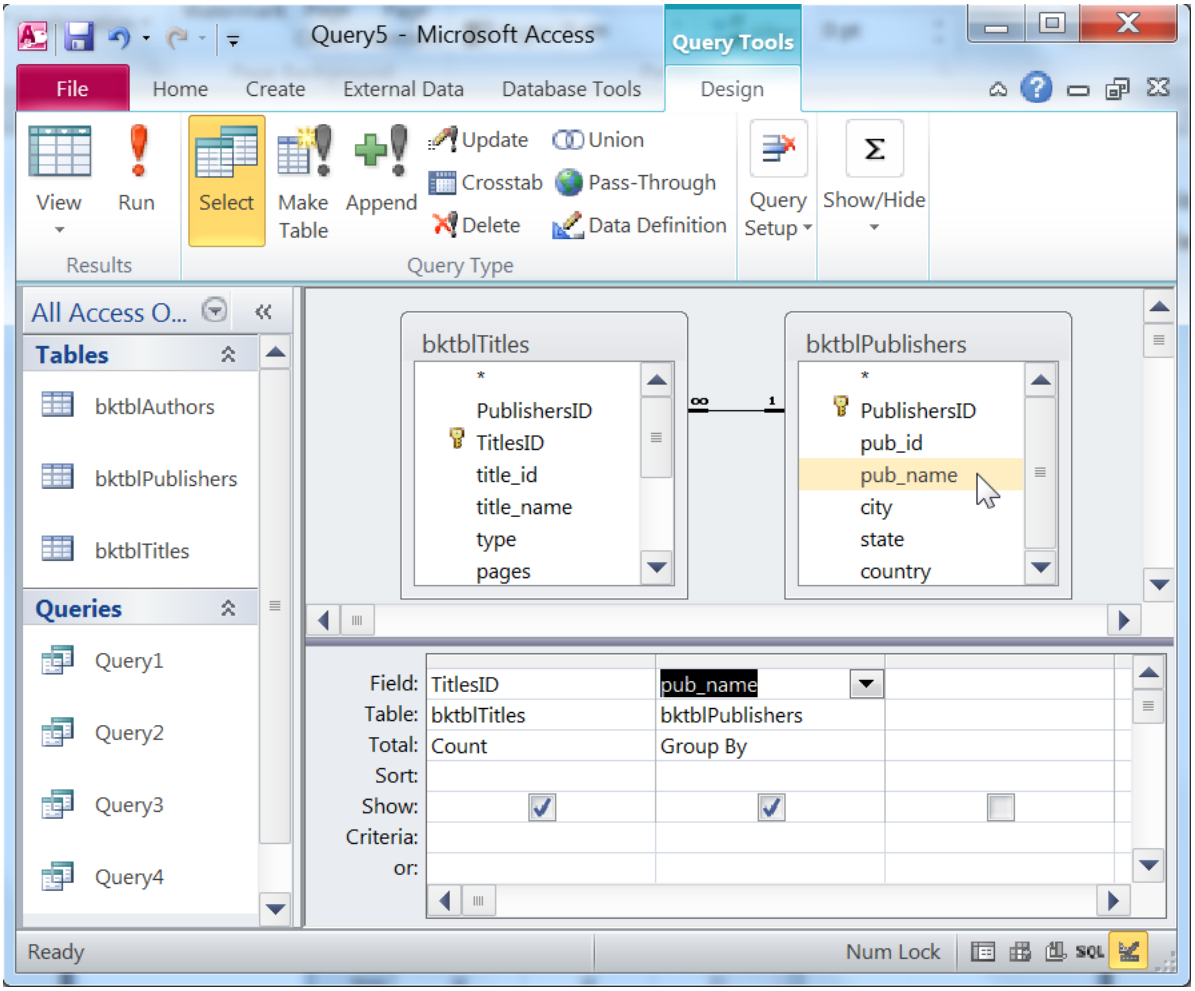


Run the query to view the result. The query adds up all of the values in the profit column and returns the result.

Exercise 4

Create a new query (Query5) and *count* the total number of books that have been published.

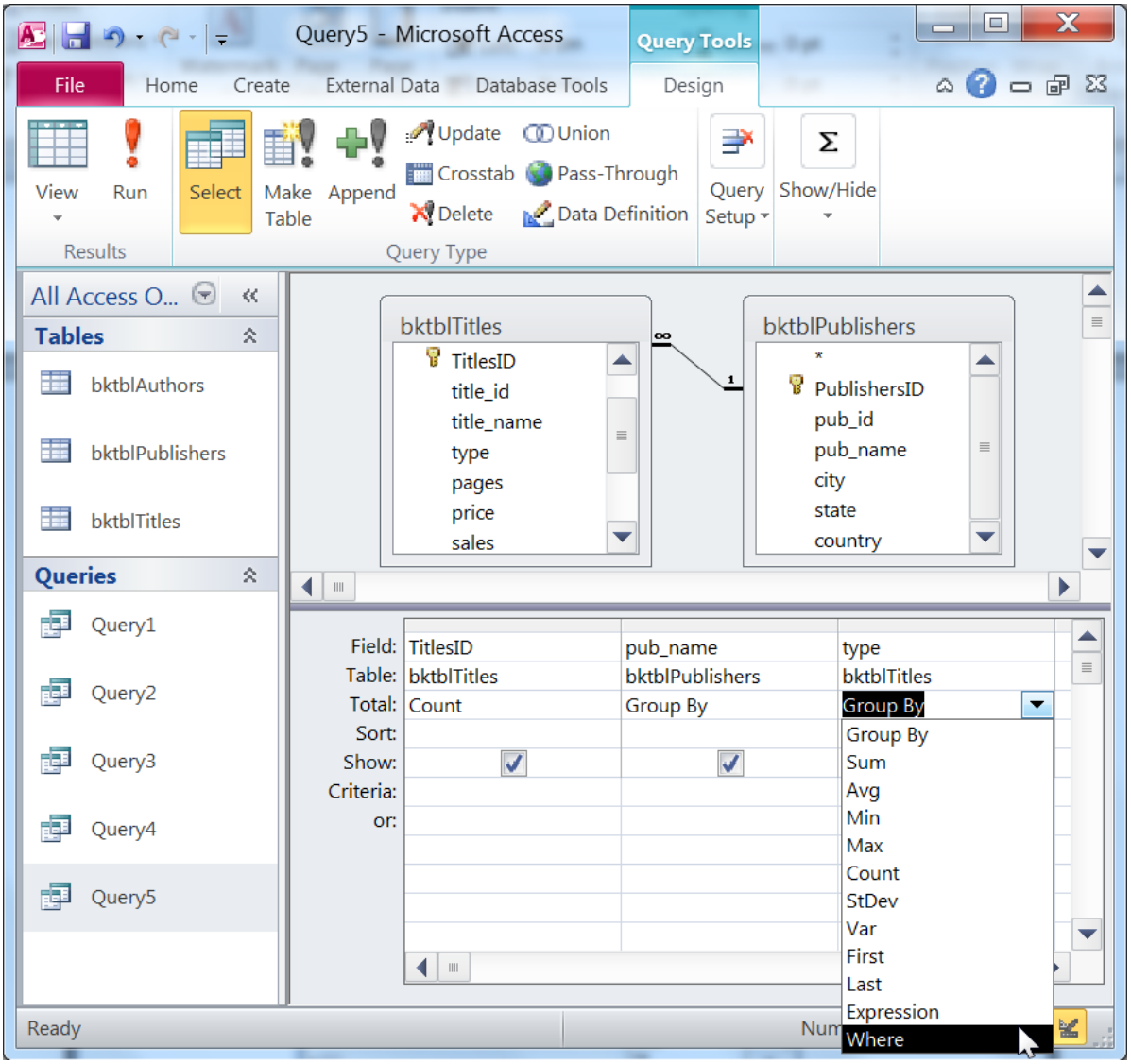
Now, modify Query5 so that it shows the number of books each publisher has published. This is where we use the *Group By* function. Add the bkTblPublishers to the query and then double-click on pub_name.



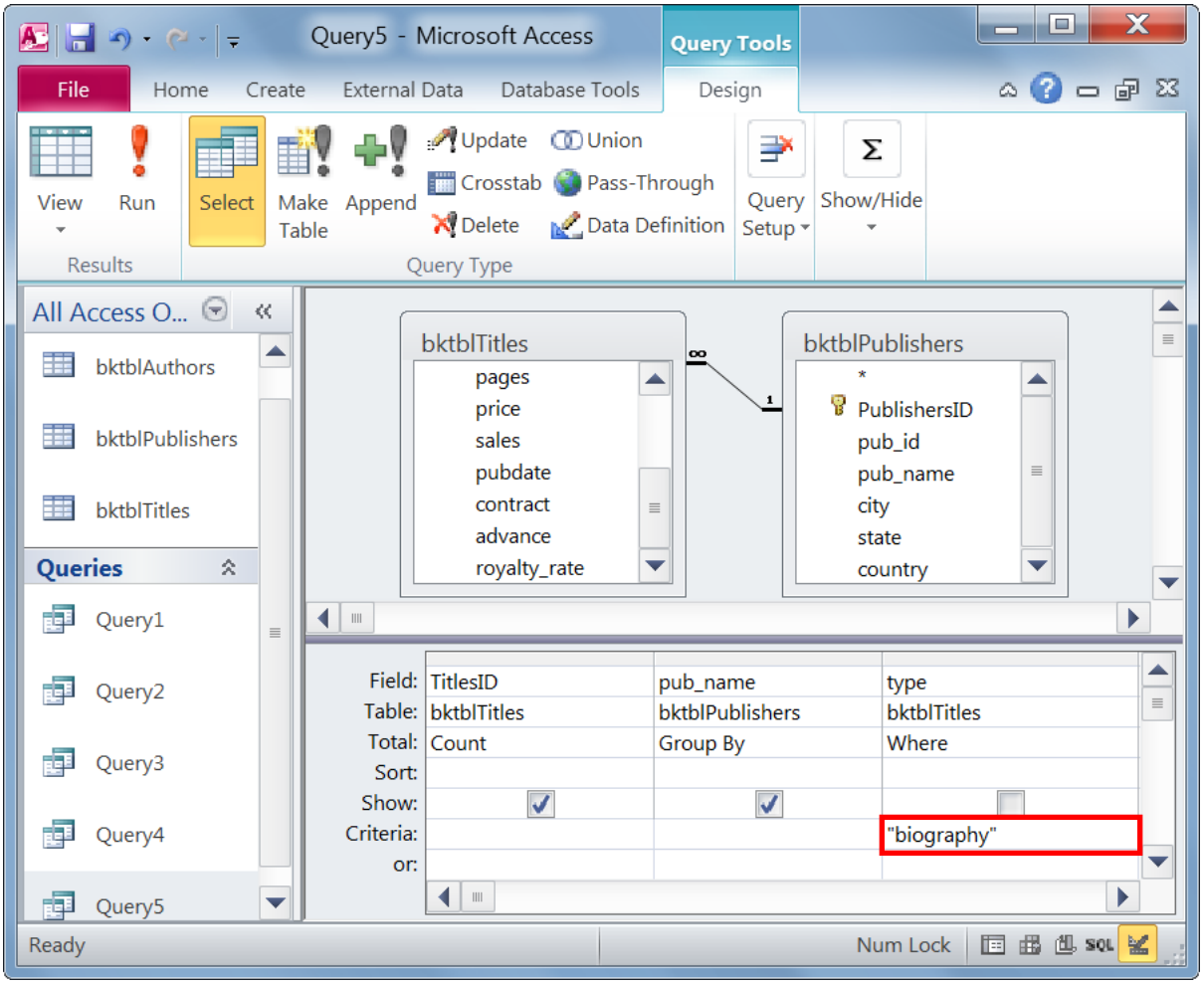
Run the query.

We can also add further conditions to aggregate queries. Let us assume that we only want to count how many books of type biography were published by each publisher.

Start by adding the *type* field to the query. Select *Where* from the totals menu.



The *Where* function allows us to set criteria in aggregate queries. Notice that the *Show* checkbox of the type field was immediately deselected after we chose *Where* from the menu. Type in "biography" in the criteria field of *type*.



Then, run the query.

Exercise 5

Modify Query 5 so that it shows the count of books of type "biography" OR "history" for each publisher. Also, the count should only include books with a royalty rate greater than 0.06.

Exercise 6

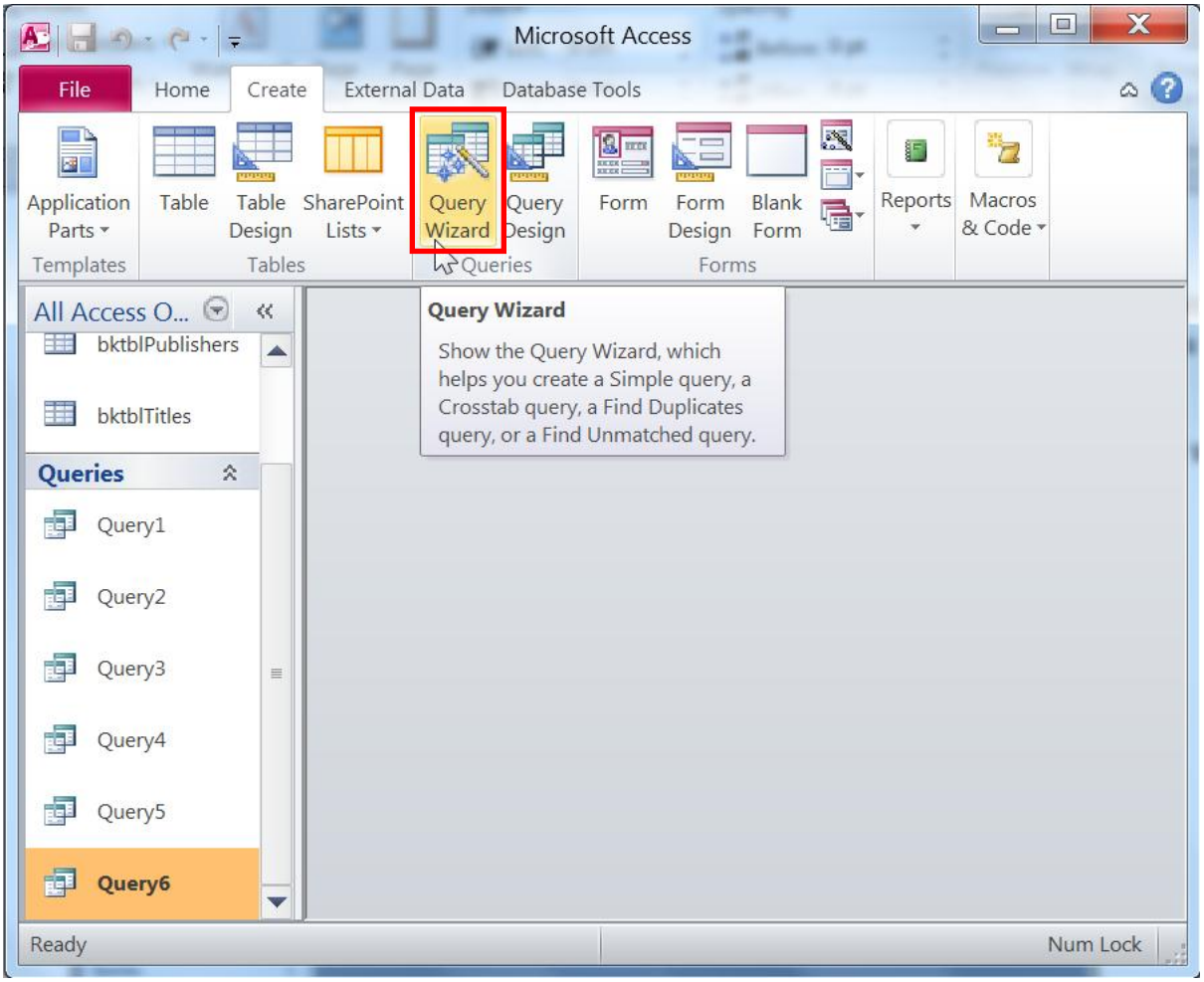
Write a query (Query6) that shows the number of authors each book has.

Crosstab queries

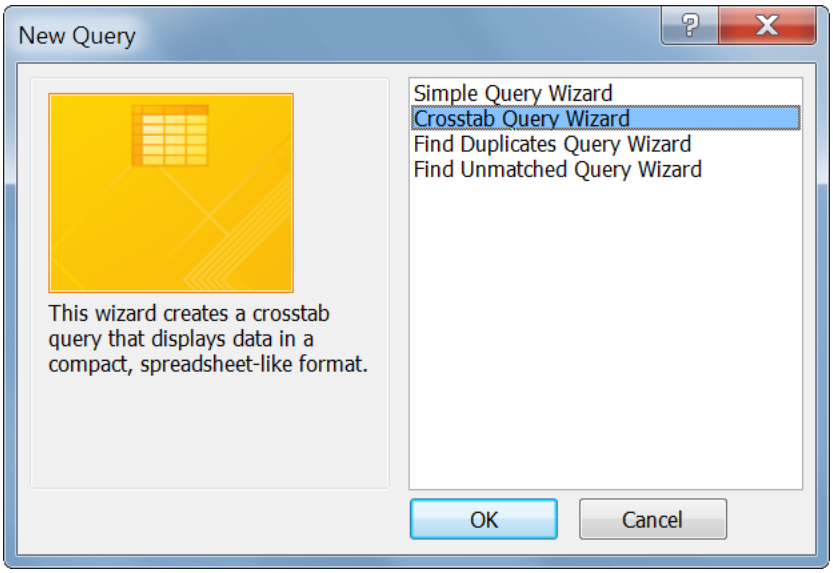
A crosstab query organizes data in a different manner than other queries do. Data in a crosstab query is grouped into horizontal and vertical columns, with aggregate functions being applied. This is better explained through an example.

We shall use the Query Wizard to create a crosstab query. Note that you can use the Query Wizard to create simple queries as well.

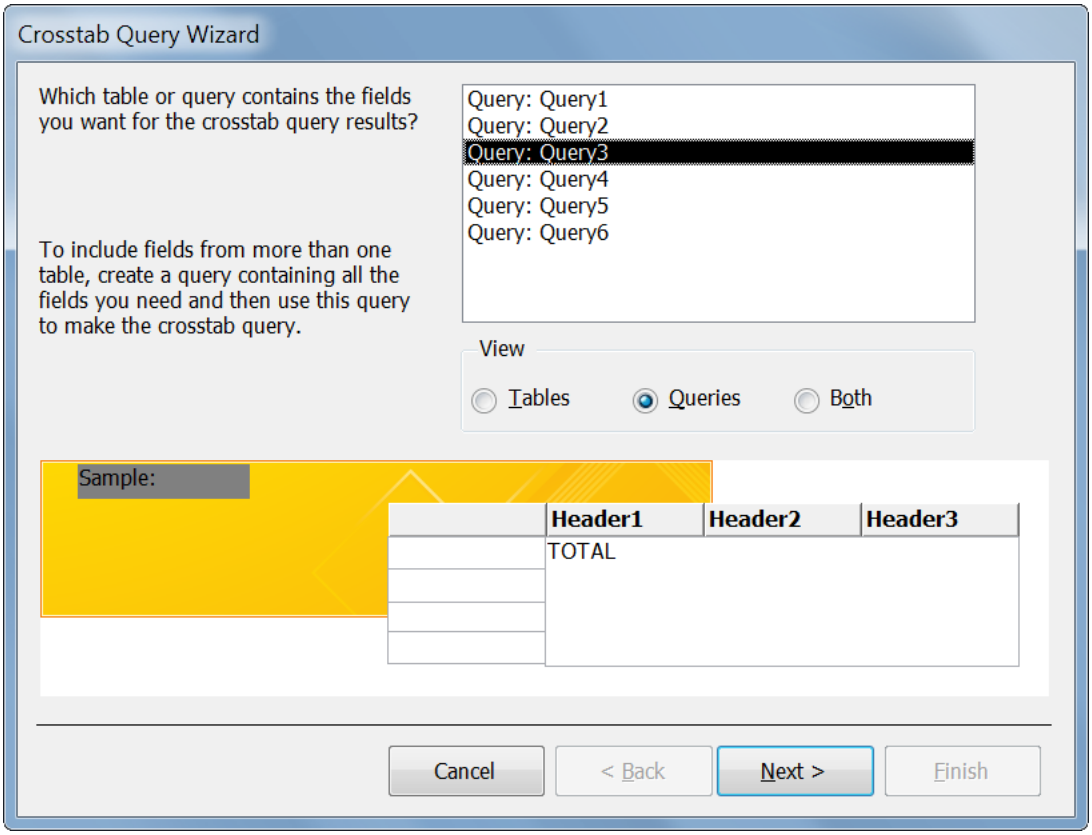
In the *Create* tab and under the *Other* group, click on the *Query Wizard* button.



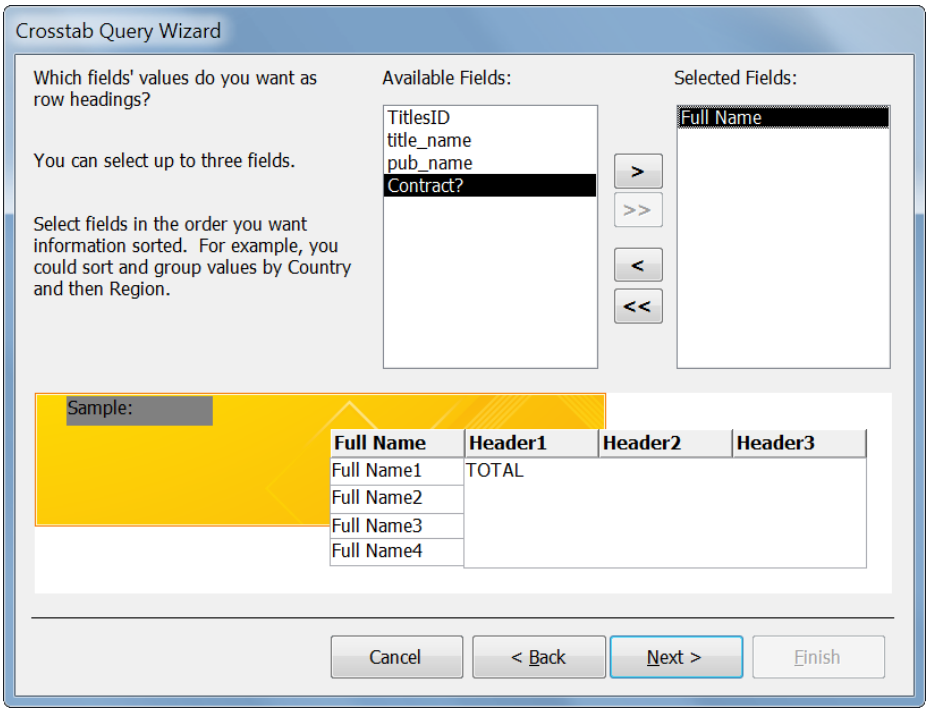
Select Crosstab Query Wizard from the list and press OK.



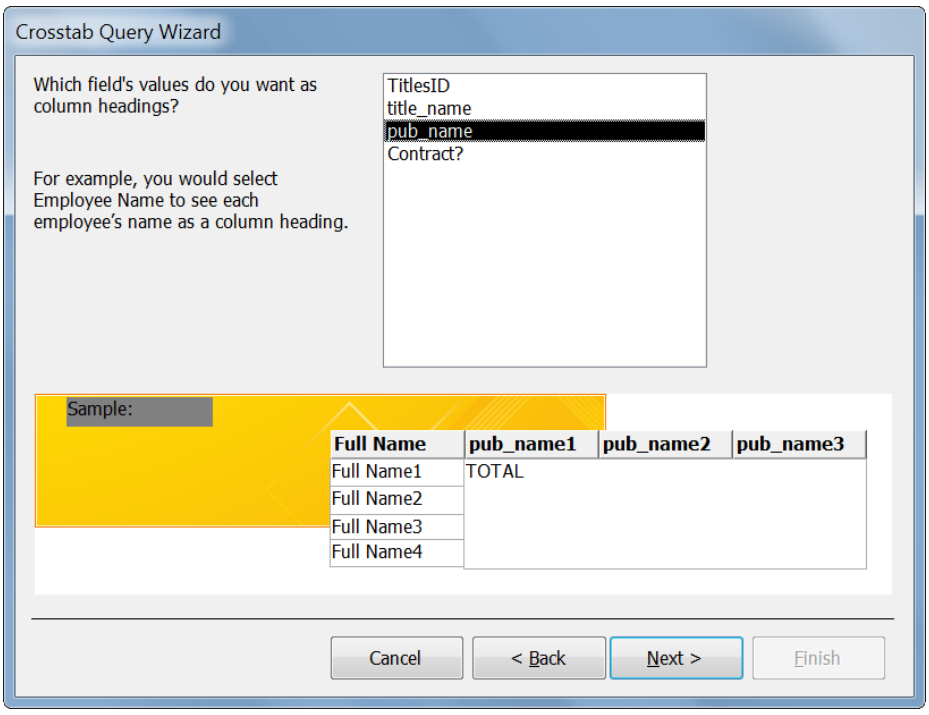
All the data for your crosstab query needs to be in either one table or one query. Select the Queries radio box and then select Query3 from the menu. Press the Next button.



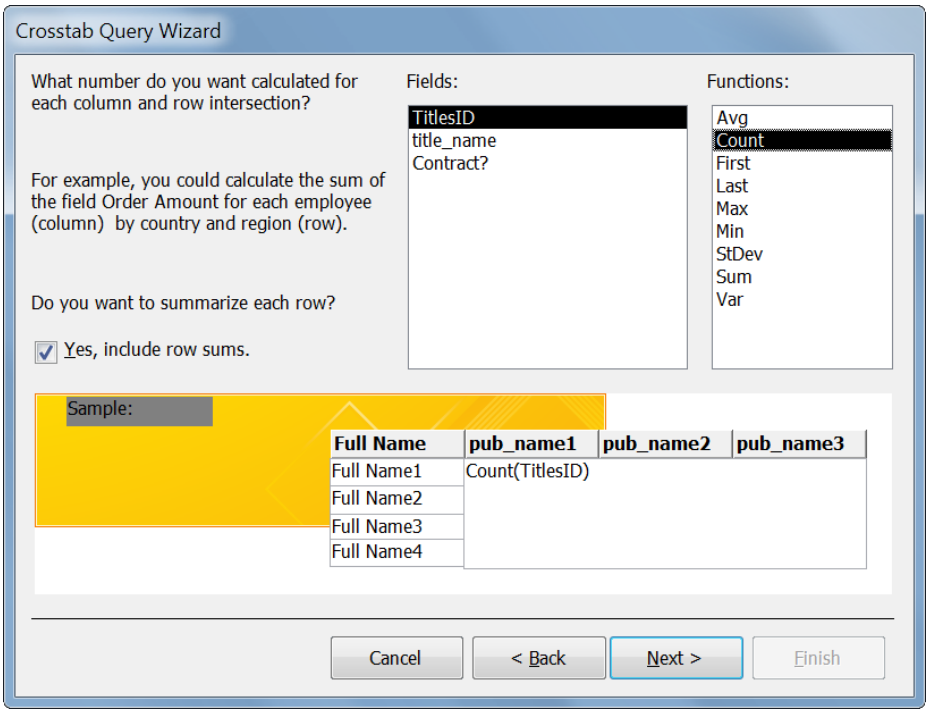
This is where our query comes into shape. You need to select which field you want as the row heading. Select FullName and add it to the list on the right by clicking the arrow button. Press Next.



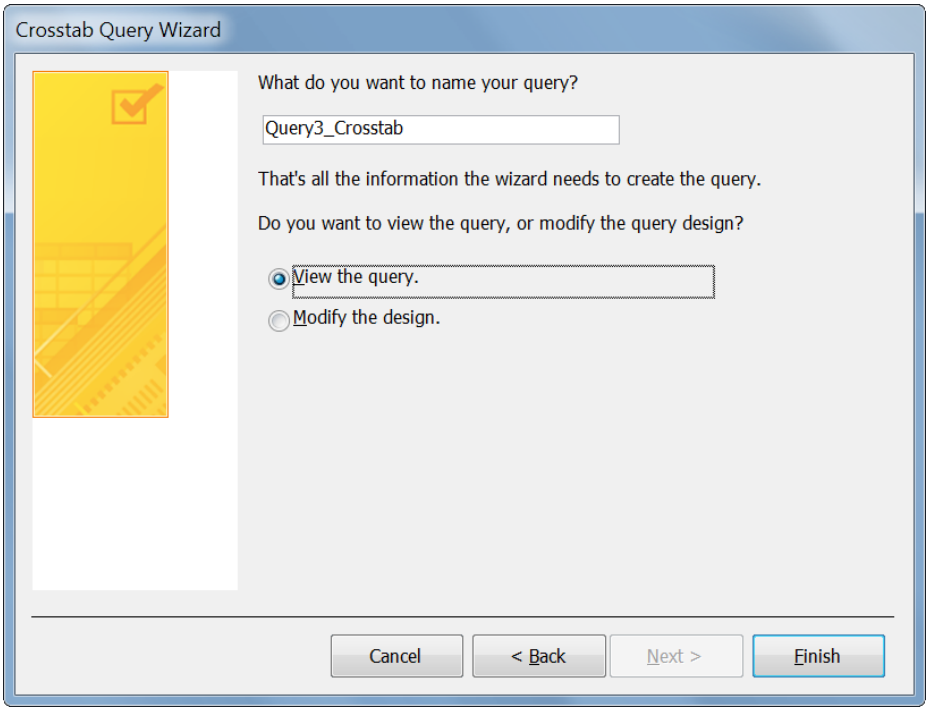
Select pub_name as the column headings and press Next.



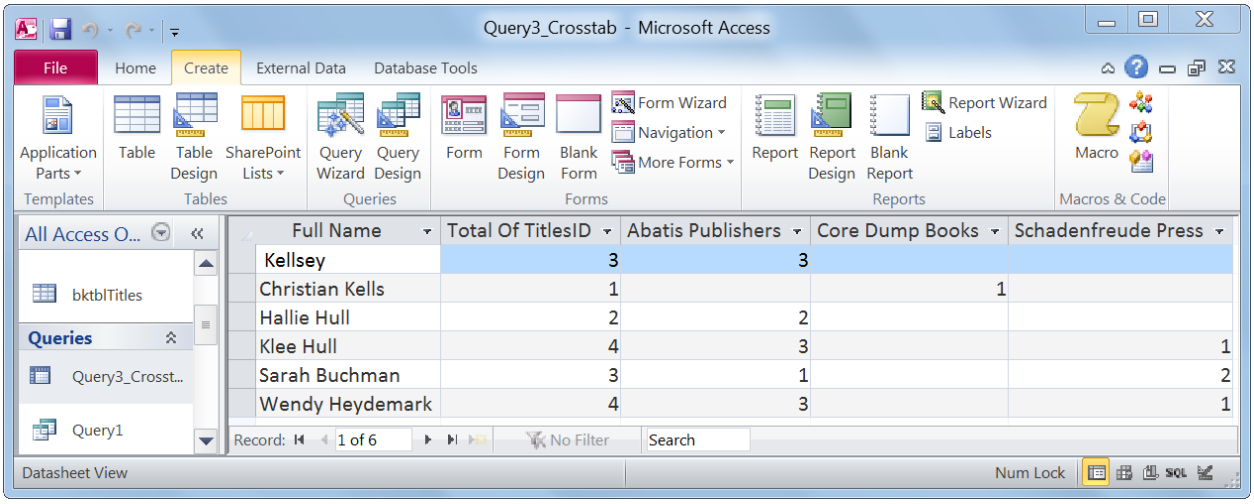
Finally, select which data field needs to be calculated in the query. We would like to calculate the number of titles published for each publisher by each author. Select TitlesID from the fields list and select Count from the functions. Press Next.



Press the Finish button.



Following is what the result of your query should look like. Notice how it grouped all the column and row fields, and then it calculated the number of titles for them. This is a very easy way to organize data.



Exercise 7

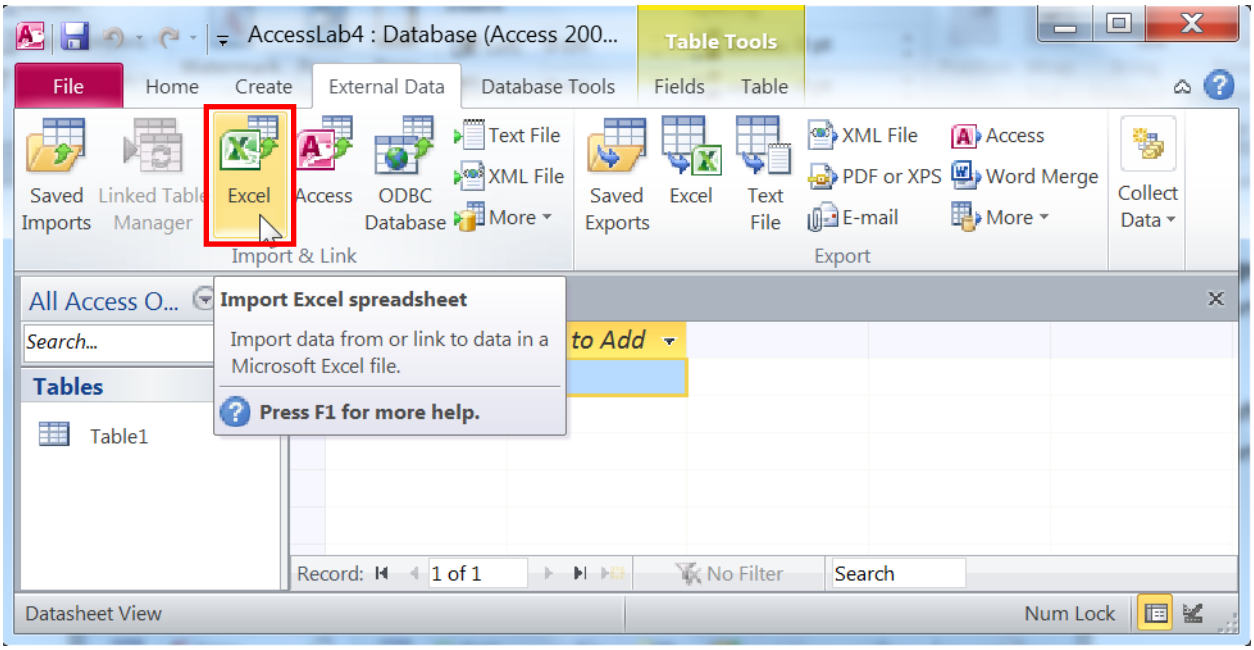
Create a crosstab query that shows the sum of book sales by each advance payment, by each author. In other words, the rows should show an author's first and last name (you can add more than one field) and the column headings should display the advance on each title. Calculate the sum of sales for each row and column intersection.

Lab 4: Make-Table, Union, and Append Queries

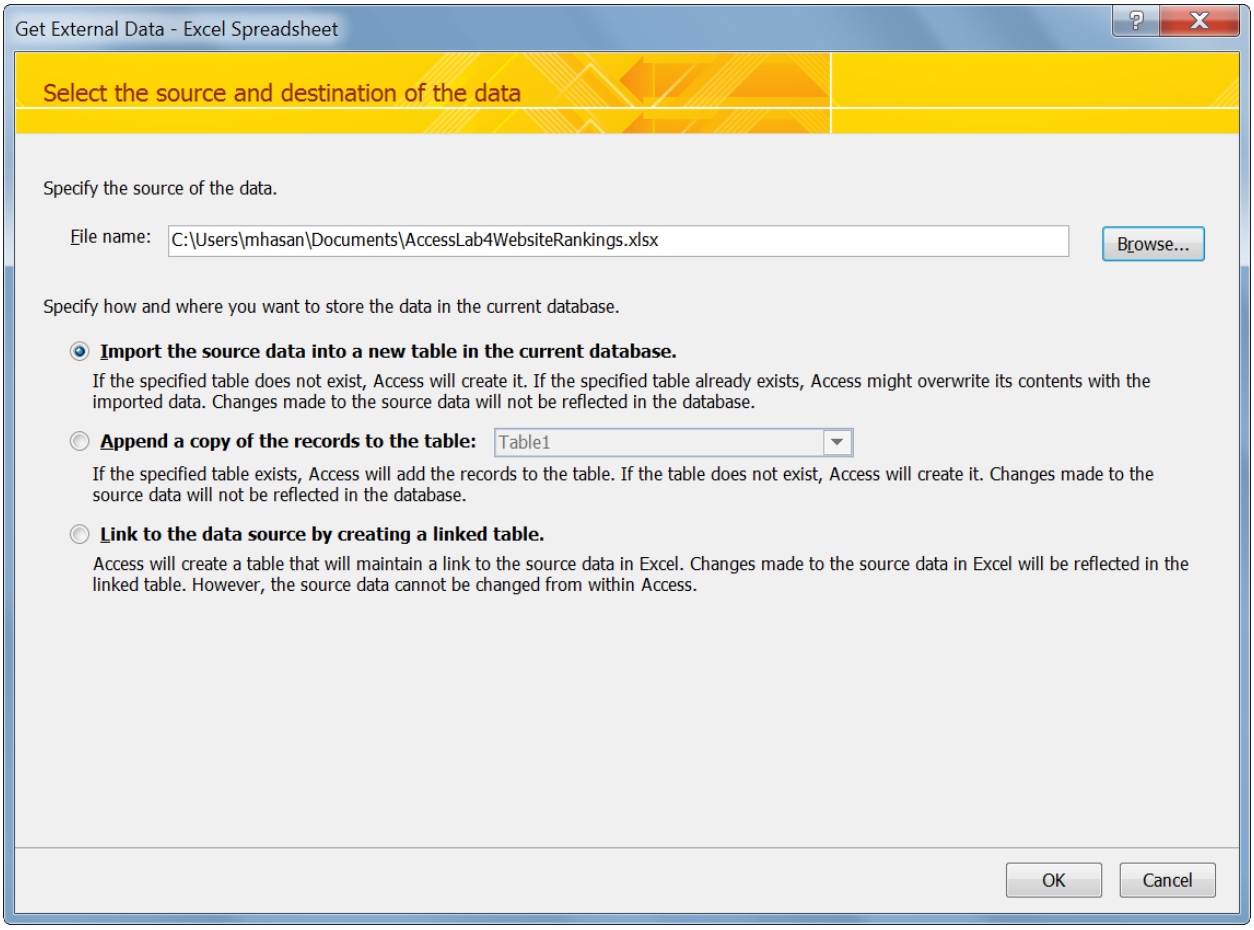
Importing Excel data into Access

Sometimes you start out with a small table in Excel and before you know it, your dataset gets so large that it becomes more manageable if you turn it into a database. This is why it can be useful to import Excel sheets into Access.

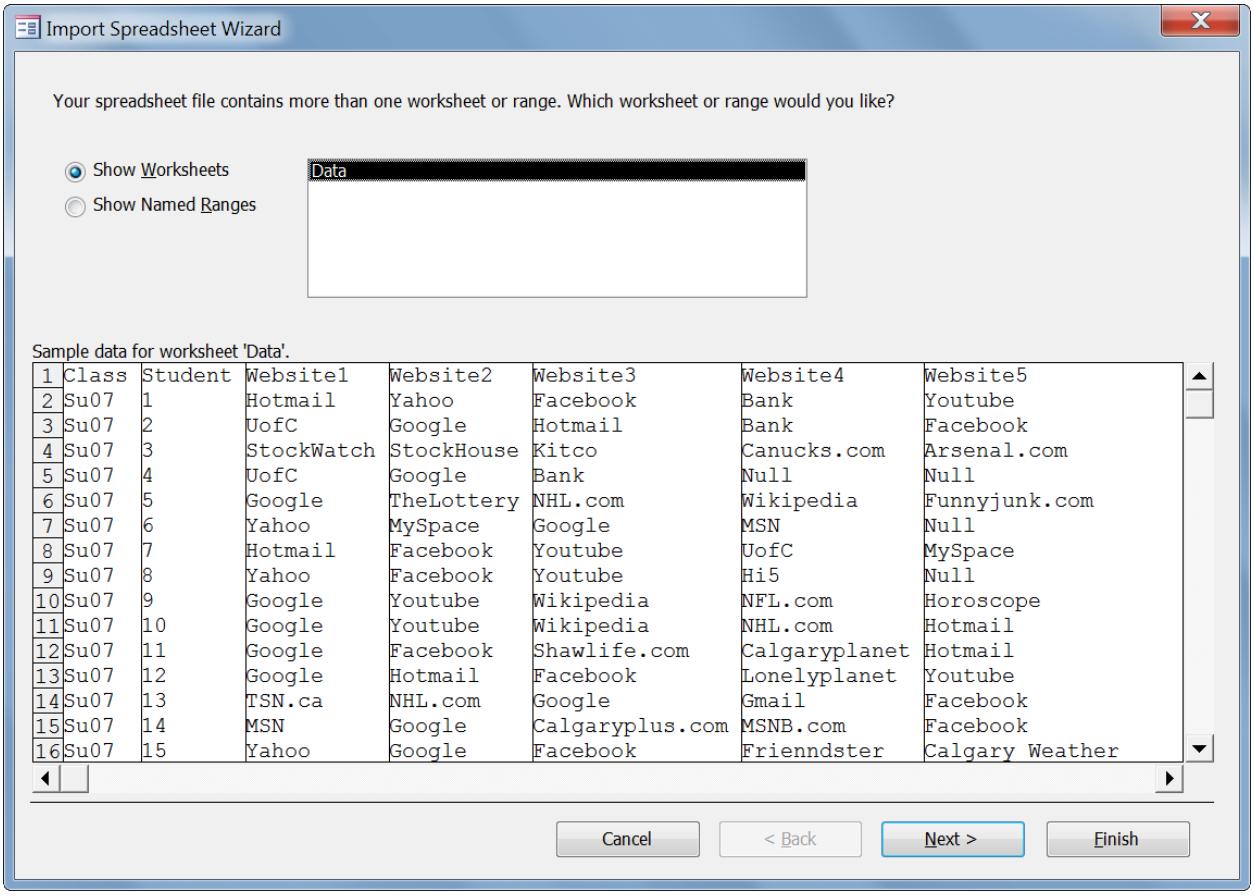
We will be importing the file AccessLab4WebsiteRankings.xslx into a new Database file. Open Microsoft Access and create a new database. In the *External Data* tab, press the *Excel* button from the *Import & Link* group.



Browse to where your downloaded AccessLab4WebsiteRankings.xslx file and press OK.

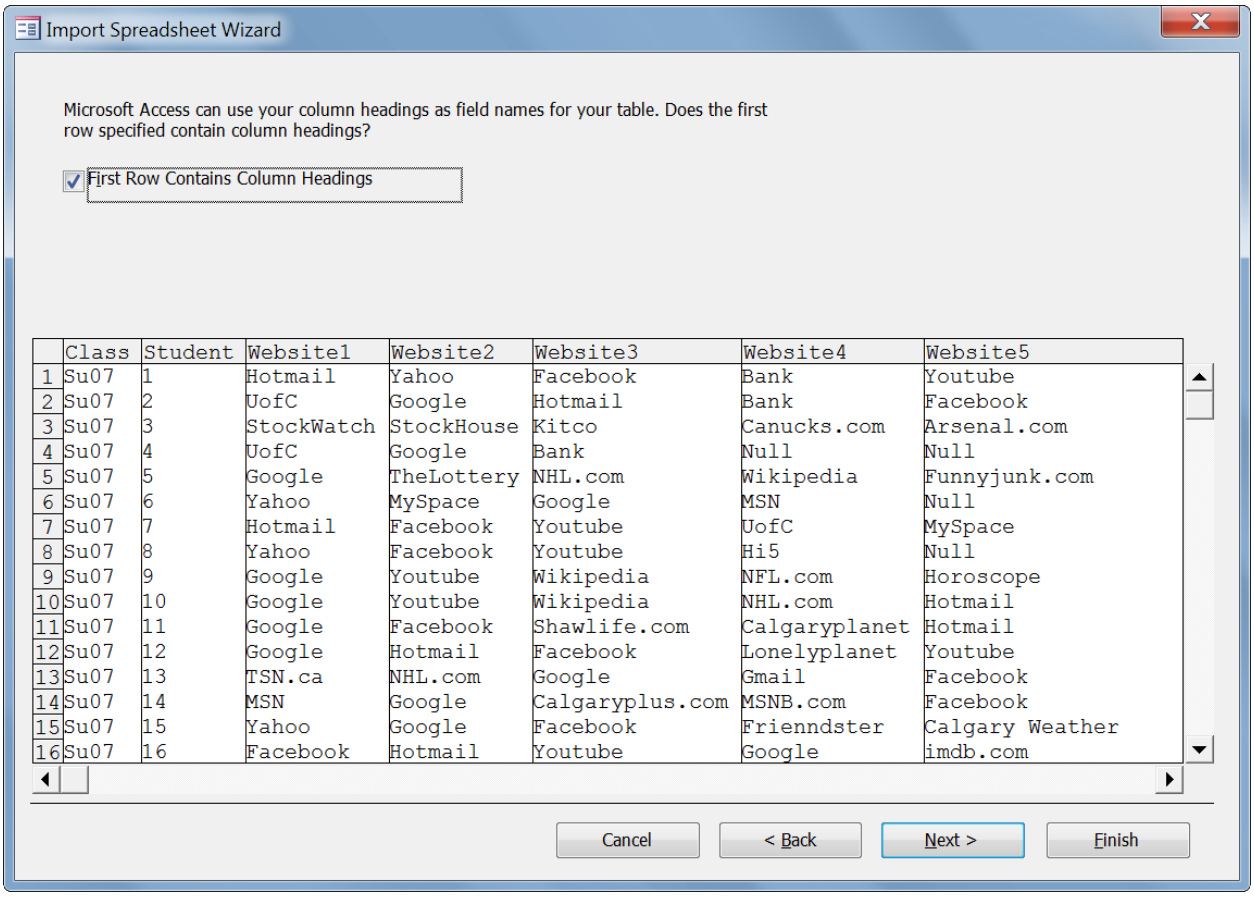


The following dialog box is useful when you have more than one sheet in your Excel file. Press Next.



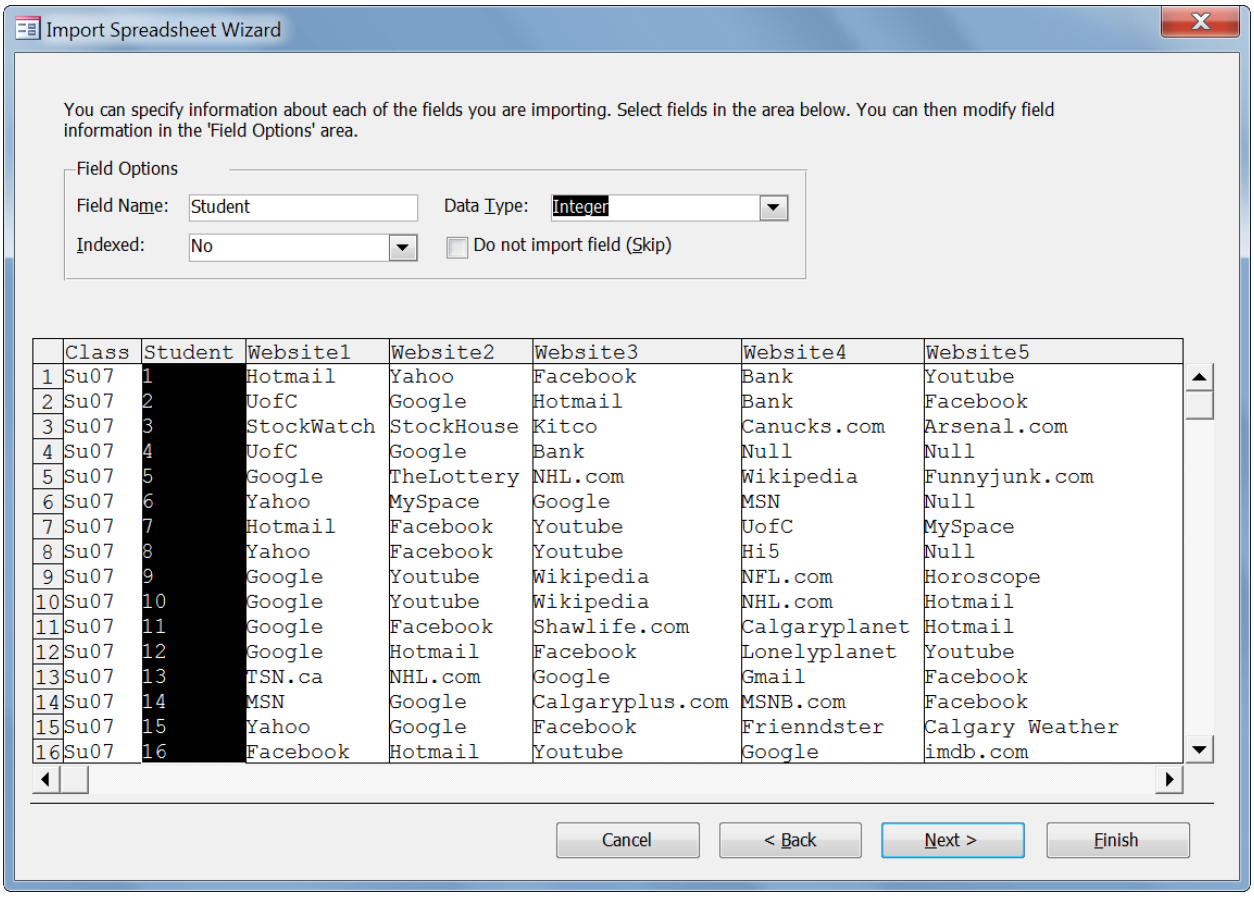
	Class	Student	Website1	Website2	Website3	Website4	Website5
2	Su07	1	Hotmail	Yahoo	Facebook	Bank	Youtube
3	Su07	2	UofC	Google	Hotmail	Bank	Facebook
4	Su07	3	StockWatch	StockHouse	Kitco	Canucks.com	Arsenal.com
5	Su07	4	UofC	Google	Bank	Null	Null
6	Su07	5	Google	TheLottery	NHL.com	Wikipedia	Funnyjunk.com
7	Su07	6	Yahoo	MySpace	Google	MSN	Null
8	Su07	7	Hotmail	Facebook	Youtube	UofC	MySpace
9	Su07	8	Yahoo	Facebook	Youtube	Hi5	Null
10	Su07	9	Google	Youtube	Wikipedia	NFL.com	Horoscope
11	Su07	10	Google	Youtube	Wikipedia	NHL.com	Hotmail
12	Su07	11	Google	Facebook	Shawlife.com	Calgaryplanet	Hotmail
13	Su07	12	Google	Hotmail	Facebook	Lonelyplanet	Youtube
14	Su07	13	TSN.ca	NHL.com	Google	Gmail	Facebook
15	Su07	14	MSN	Google	Calgaryplus.com	MSNB.com	Facebook
16	Su07	15	Yahoo	Google	Facebook	Frienndster	Calgary Weather

Select the *First Row Contains Column Headings* checkbox and press Next. This will set our column headings as field names in the table that will be created.



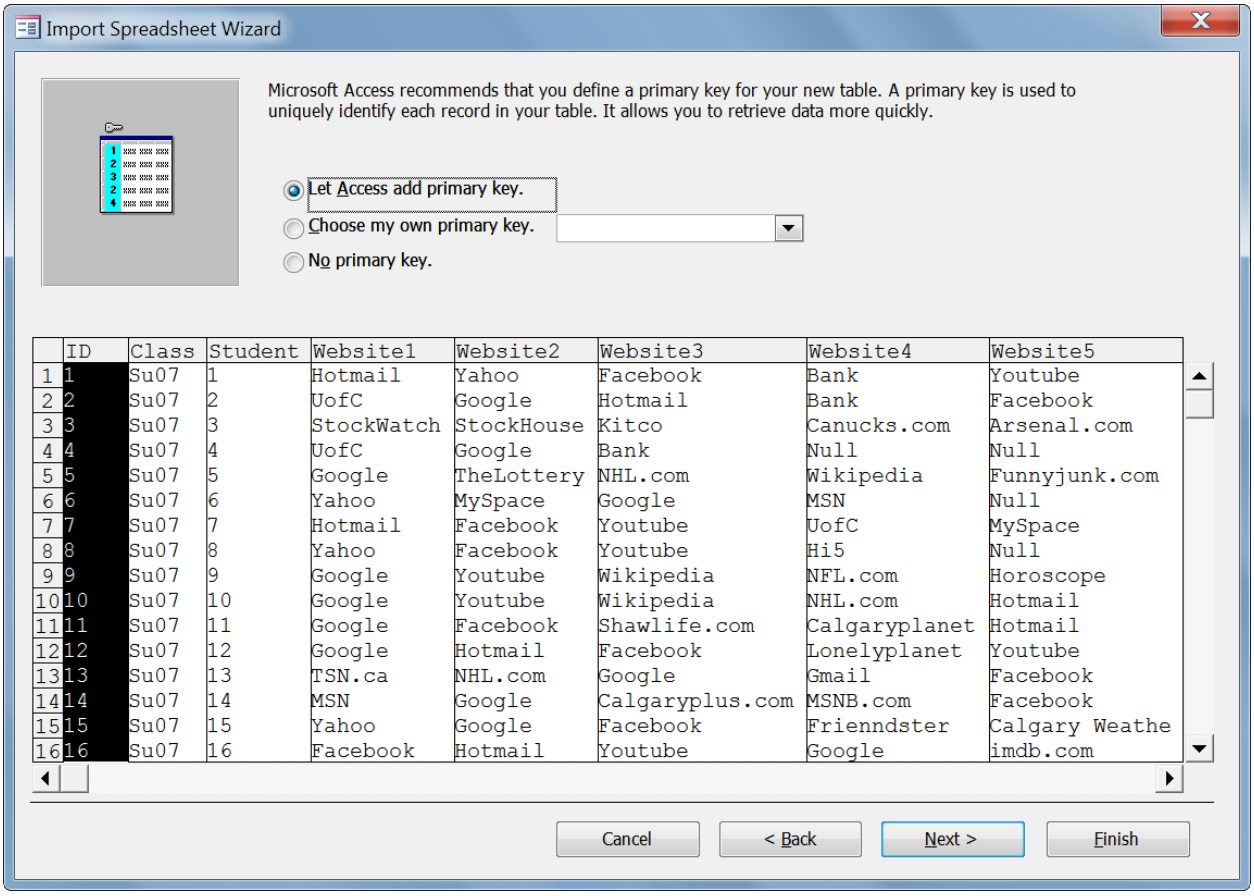
	Class	Student	Website1	Website2	Website3	Website4	Website5
1	Su07	1	Hotmail	Yahoo	Facebook	Bank	Youtube
2	Su07	2	UofC	Google	Hotmail	Bank	Facebook
3	Su07	3	StockWatch	StockHouse	Kitco	Canucks.com	Arsenal.com
4	Su07	4	UofC	Google	Bank	Null	Null
5	Su07	5	Google	TheLottery	NHL.com	Wikipedia	Funnyjunk.com
6	Su07	6	Yahoo	MySpace	Google	MSN	Null
7	Su07	7	Hotmail	Facebook	Youtube	UofC	MySpace
8	Su07	8	Yahoo	Facebook	Youtube	Hi5	Null
9	Su07	9	Google	Youtube	Wikipedia	NFL.com	Horoscope
10	Su07	10	Google	Youtube	Wikipedia	NHL.com	Hotmail
11	Su07	11	Google	Facebook	Shawlife.com	Calgaryplanet	Hotmail
12	Su07	12	Google	Hotmail	Facebook	Lonelyplanet	Youtube
13	Su07	13	TSN.ca	NHL.com	Google	Gmail	Facebook
14	Su07	14	MSN	Google	Calgaryplus.com	MSNB.com	Facebook
15	Su07	15	Yahoo	Google	Facebook	Frienndster	Calgary Weather
16	Su07	16	Facebook	Hotmail	Youtube	Google	imdb.com

The next dialog box allows you to set data types of fields. Select the Student field from the table and change its data type to Integer. Press Next.



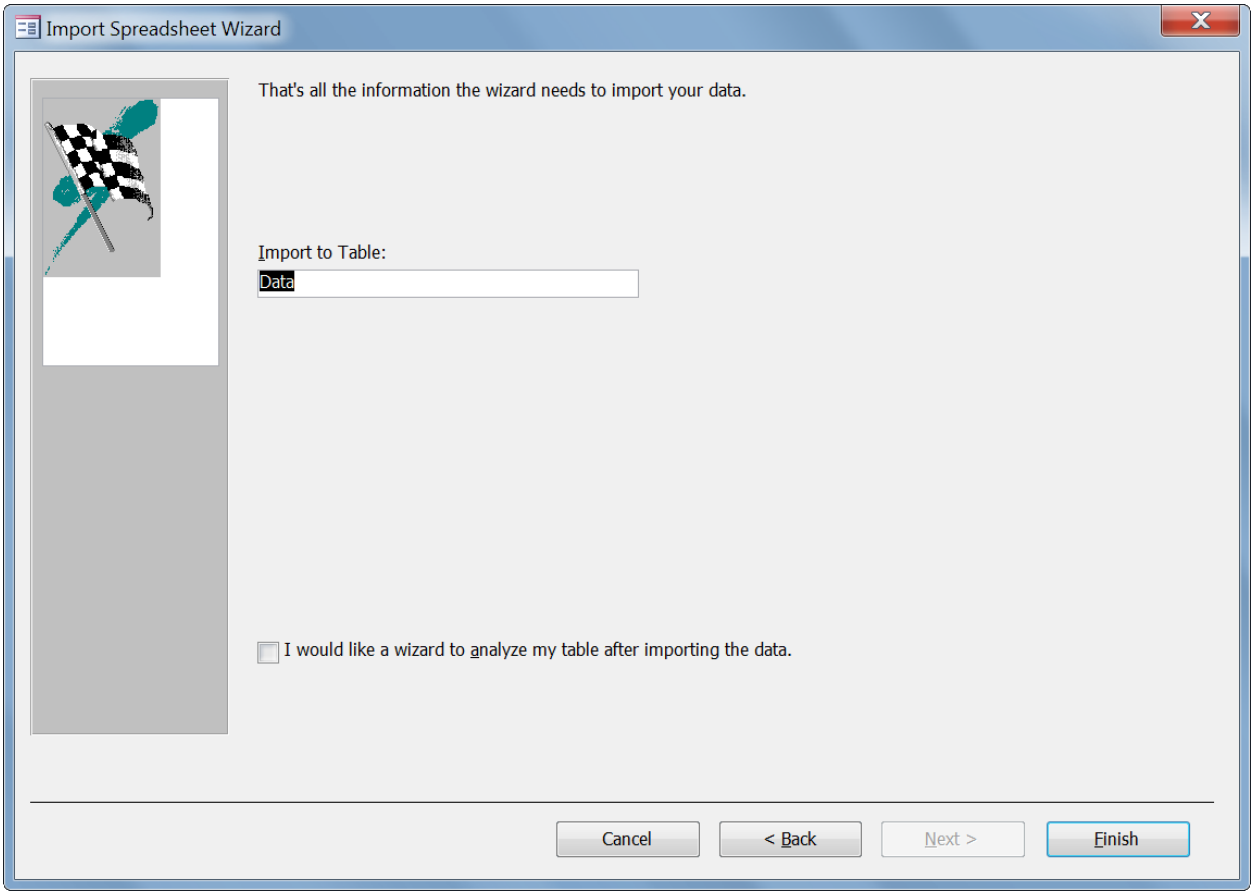
	Class	Student	Website1	Website2	Website3	Website4	Website5
1	Su07	1	Hotmail	Yahoo	Facebook	Bank	Youtube
2	Su07	2	UofC	Google	Hotmail	Bank	Facebook
3	Su07	3	StockWatch	StockHouse	Kitco	Canucks.com	Arsenal.com
4	Su07	4	UofC	Google	Bank	Null	Null
5	Su07	5	Google	TheLottery	NHL.com	Wikipedia	Funnyjunk.com
6	Su07	6	Yahoo	MySpace	Google	MSN	Null
7	Su07	7	Hotmail	Facebook	Youtube	UofC	MySpace
8	Su07	8	Yahoo	Facebook	Youtube	Hi5	Null
9	Su07	9	Google	Youtube	Wikipedia	NFL.com	Horoscope
10	Su07	10	Google	Youtube	Wikipedia	NHL.com	Hotmail
11	Su07	11	Google	Facebook	Shawlife.com	Calgaryplanet	Hotmail
12	Su07	12	Google	Hotmail	Facebook	Lonelyplanet	Youtube
13	Su07	13	TSN.ca	NHL.com	Google	Gmail	Facebook
14	Su07	14	MSN	Google	Calgaryplus.com	MSNB.com	Facebook
15	Su07	15	Yahoo	Google	Facebook	Frienndster	Calgary Weather
16	Su07	16	Facebook	Hotmail	Youtube	Google	imdb.com

The next screen allows us to create a new primary key. The choices are: *Let Access add primary key*, I *Choose my own primary key*, or *No primary key*. In this case, we will let Access create a new field for the primary key. Press Next.



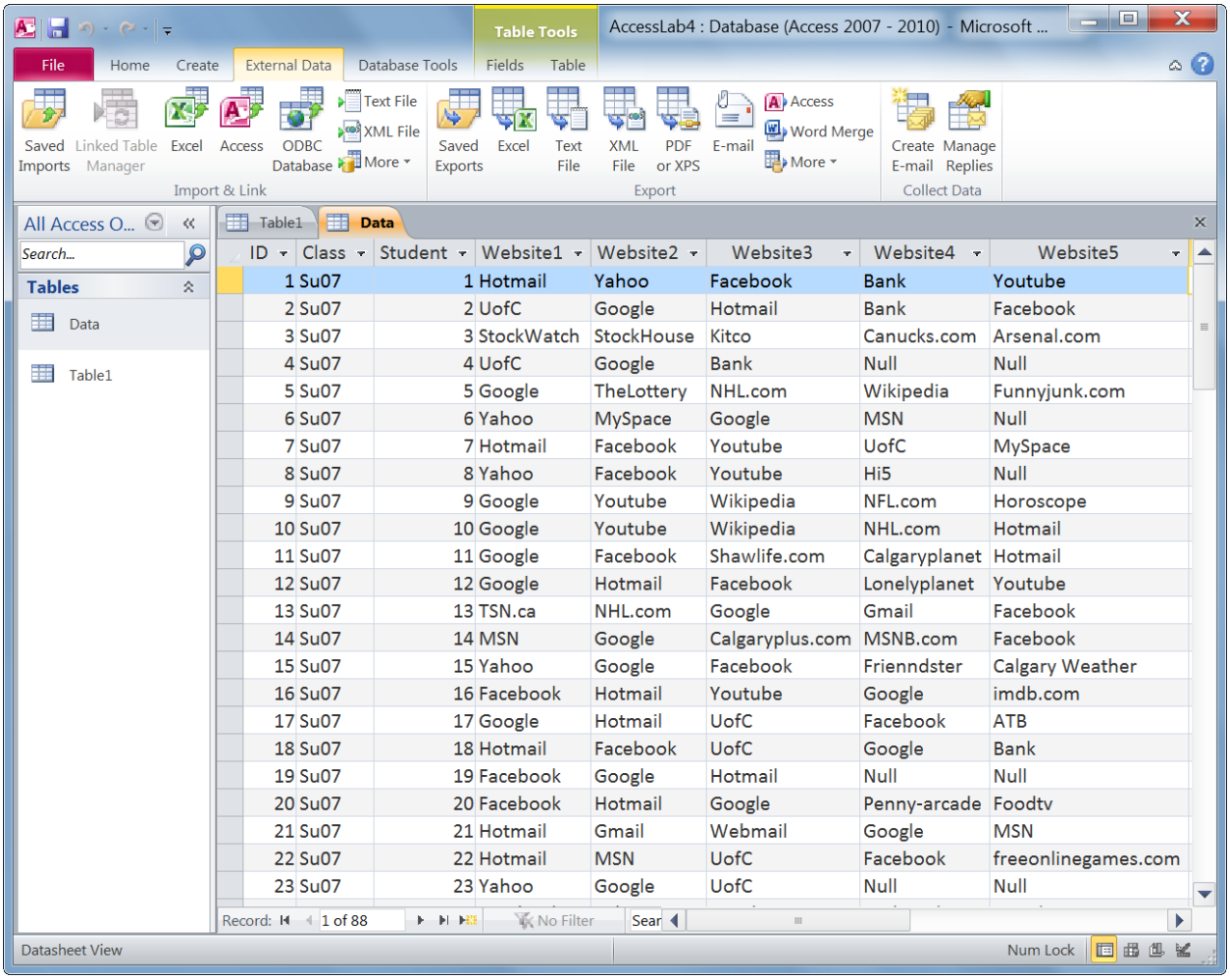
	ID	Class	Student	Website1	Website2	Website3	Website4	Website5
1	1	Su07	1	Hotmail	Yahoo	Facebook	Bank	Youtube
2	2	Su07	2	UofC	Google	Hotmail	Bank	Facebook
3	3	Su07	3	StockWatch	StockHouse	Kitco	Canucks.com	Arsenal.com
4	4	Su07	4	UofC	Google	Bank	Null	Null
5	5	Su07	5	Google	TheLottery	NHL.com	Wikipedia	Funnyjunk.com
6	6	Su07	6	Yahoo	MySpace	Google	MSN	Null
7	7	Su07	7	Hotmail	Facebook	Youtube	UofC	MySpace
8	8	Su07	8	Yahoo	Facebook	Youtube	Hi5	Null
9	9	Su07	9	Google	Youtube	Wikipedia	NFL.com	Horoscope
10	10	Su07	10	Google	Youtube	Wikipedia	NHL.com	Hotmail
11	11	Su07	11	Google	Facebook	Shawlife.com	Calgaryplanet	Hotmail
12	12	Su07	12	Google	Hotmail	Facebook	Lonelyplanet	Youtube
13	13	Su07	13	TSN.ca	NHL.com	Google	Gmail	Facebook
14	14	Su07	14	MSN	Google	Calgaryplus.com	MSNB.com	Facebook
15	15	Su07	15	Yahoo	Google	Facebook	Frienndster	Calgary Weathe
16	16	Su07	16	Facebook	Hotmail	Youtube	Google	imdb.com

Name your table as *Data* and then click on the Finish button.



Next, Access will ask if you want to save the import steps. Press the Close button.

Following is what the Data table looks like.

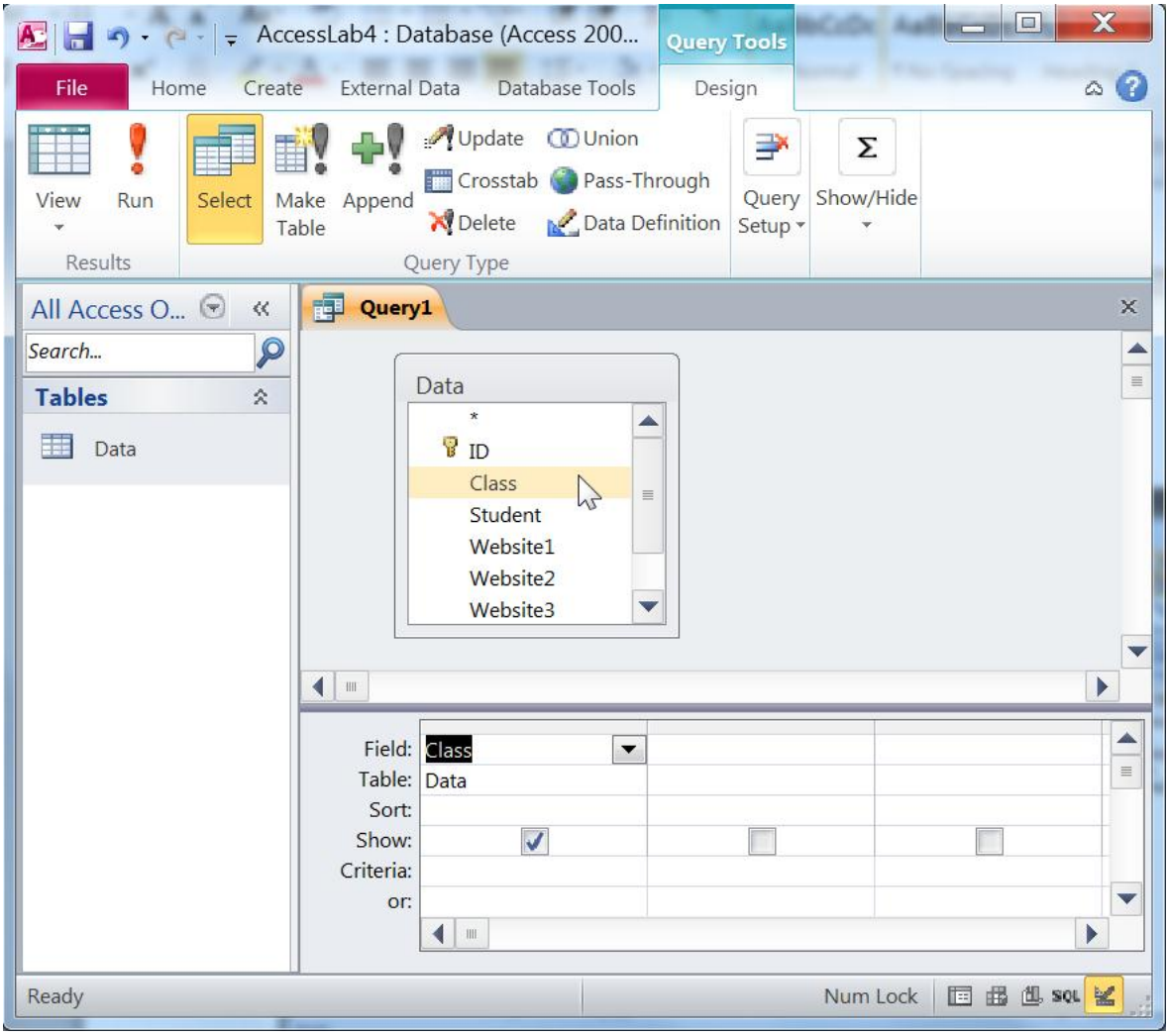


Make-table queries

Make table queries obtain data from one or more tables or queries and then create a new table out of those data.

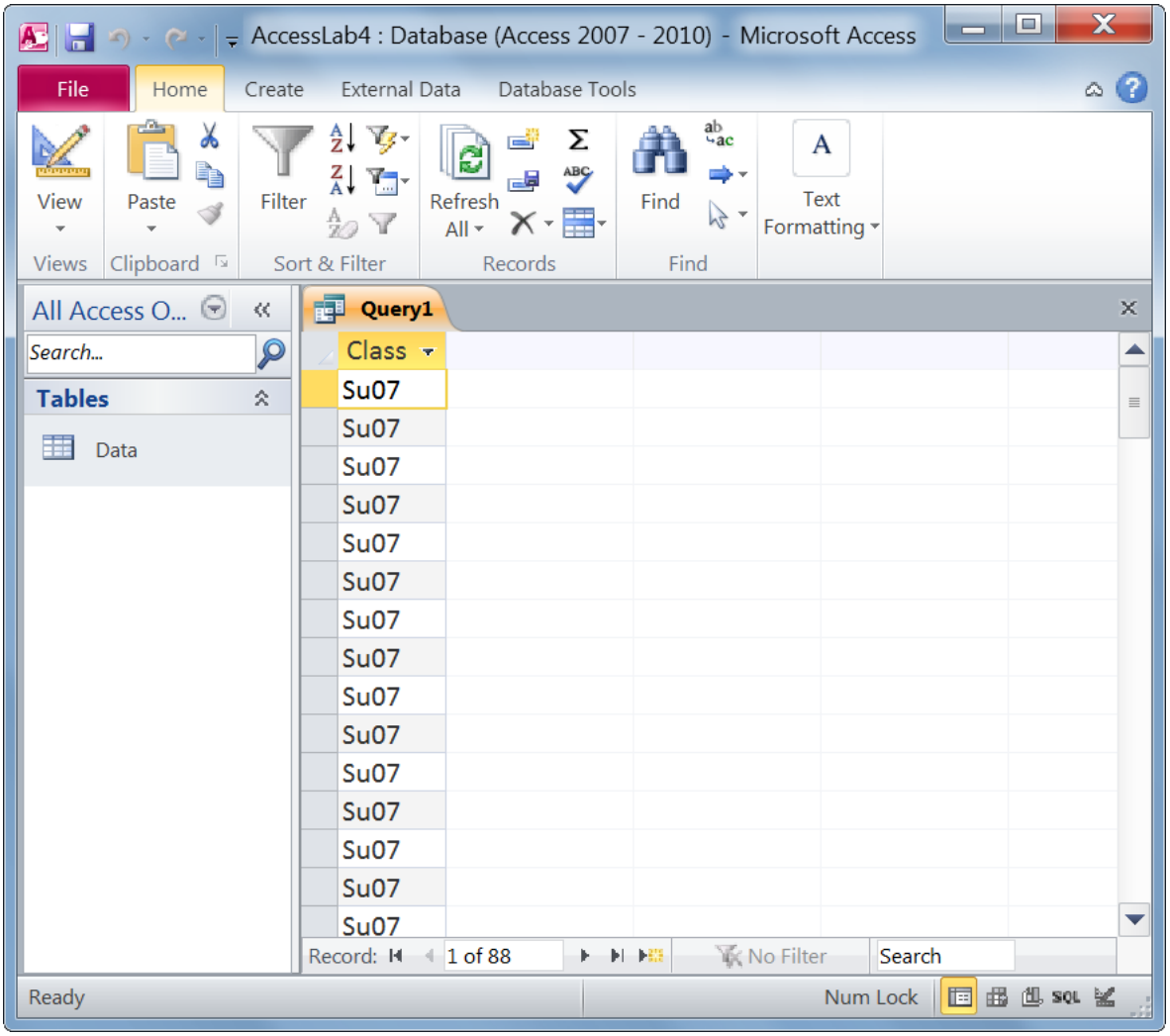
Let us create a Class table for the different semesters in our data. The first step in creating a make-table query is making a Select query with the data that you want in your new table.

Create a new query and add the Data table to the design view. Add the Class field to the query.

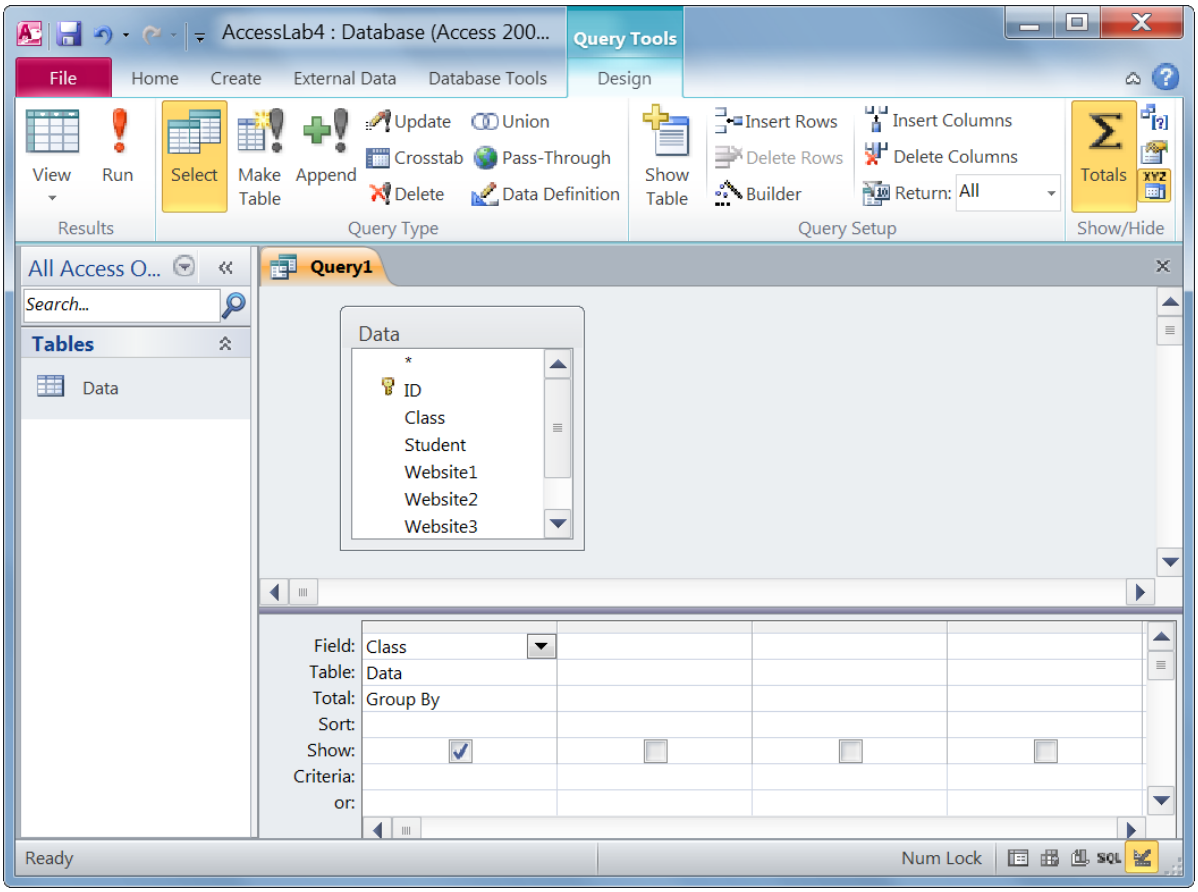


The advantage of creating a Select query first is that you can test it to see if the correct data is going to be inserted into the new table. Run the query to see the result.

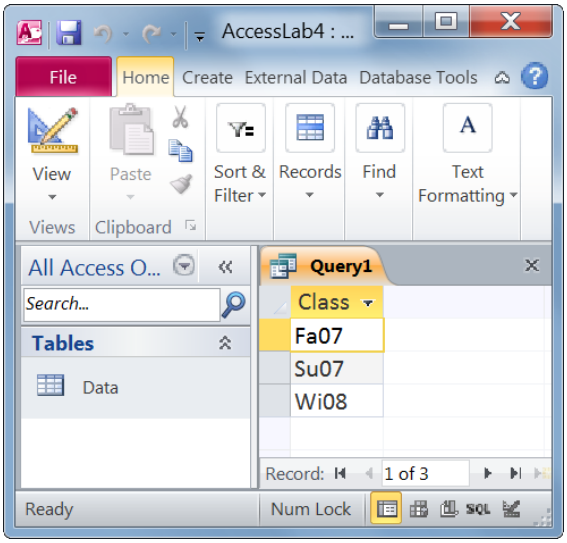
As you can see, the following is not the data we want in our Class table.



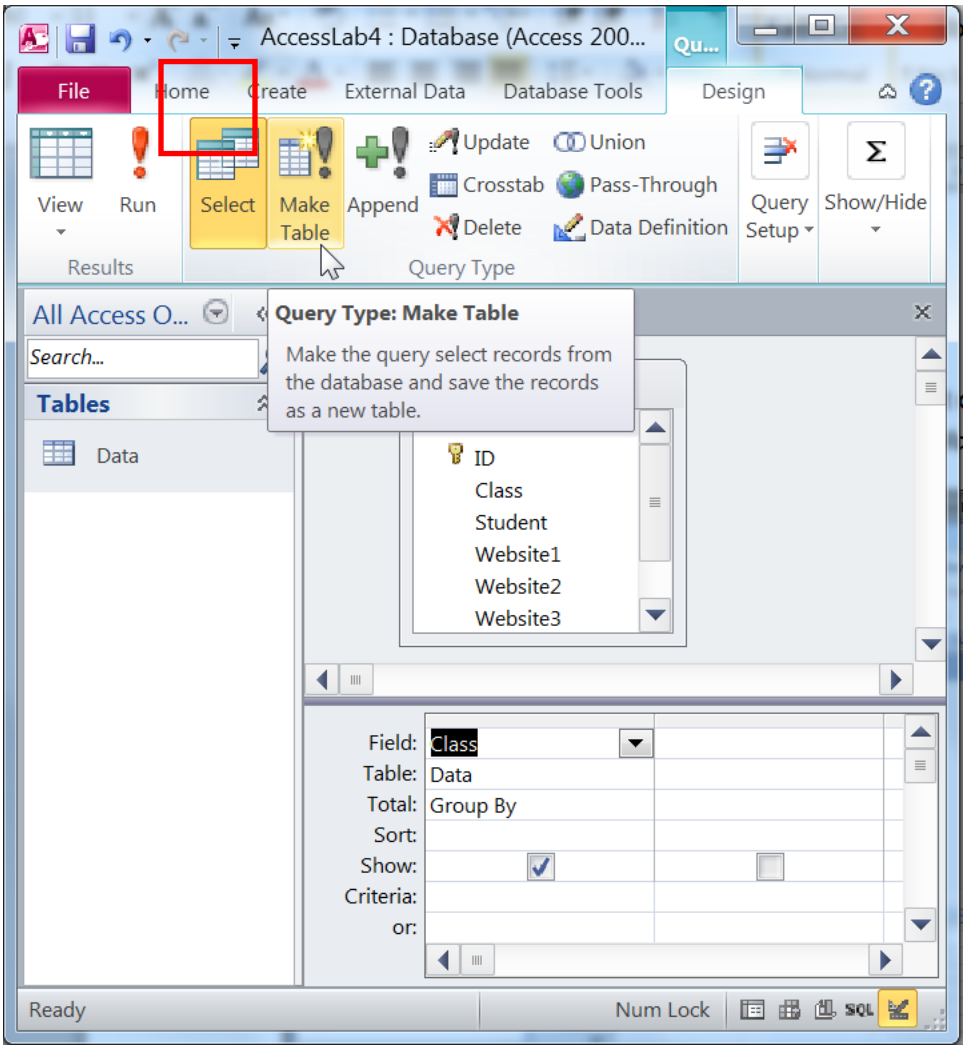
The easiest way we can get the data we want is by grouping. Go back to the design view of the query and group by the Class field.



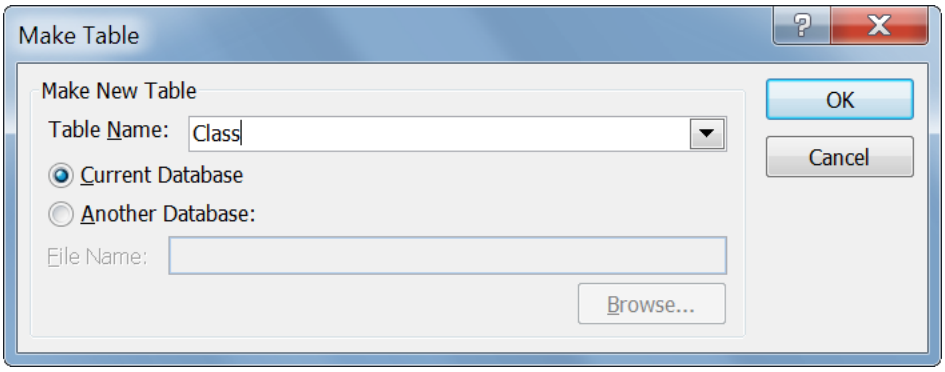
Now the results make sense.



After you are comfortable with the results of your Select query, go back to the Design view and change it into a Make-table query. You do this by pressing the *Make Table* button in the *Query Type* group.

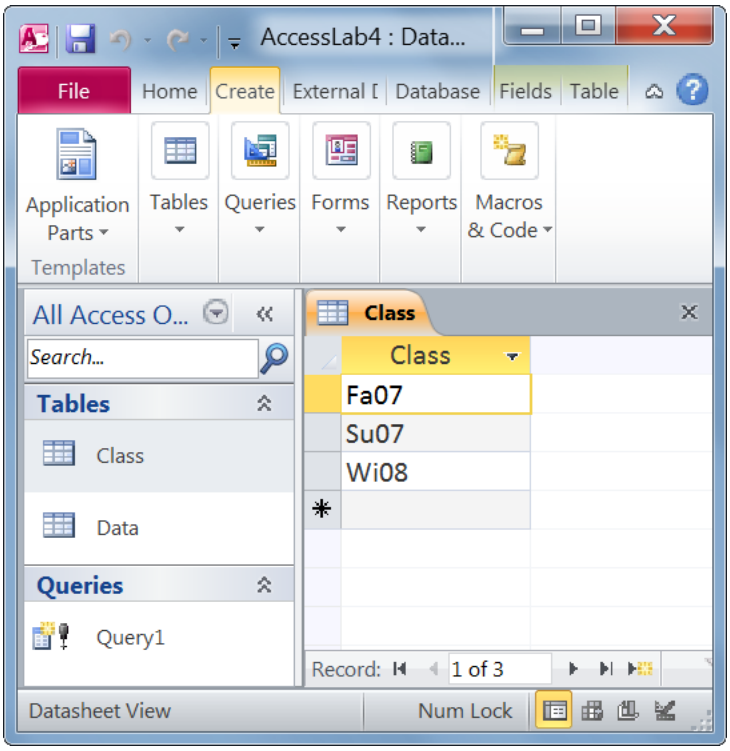


Enter the table name and press OK.



Now to make the table, click on the *Run* button in the *Results* group. Press the *Yes* button on the warning message that follows.

Now, following is what the new table looks like.



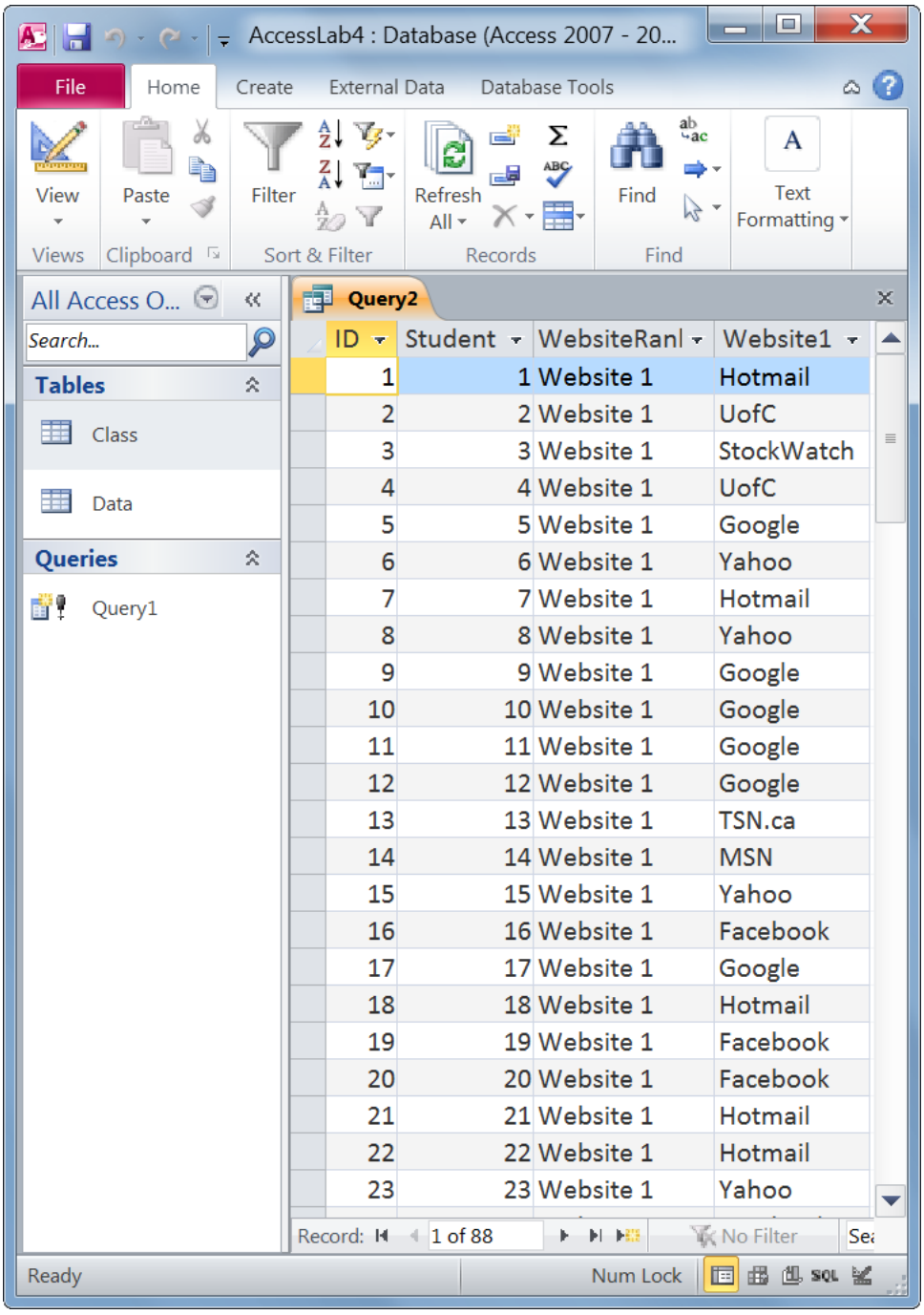
Exercise 1

Set the primary key of the Class table as the Class field.

Exercise 2

Create 5 select queries – one for each website column in the Data table. Each query result should show the ID field from the Data table, a new field named "WebsiteRank" which contains either Website1 or Website2 etc. This will be fixed for each query. Each query should also show the Student ID, as well as

the website name from that column. To make things clear, following should be the result of the first query:



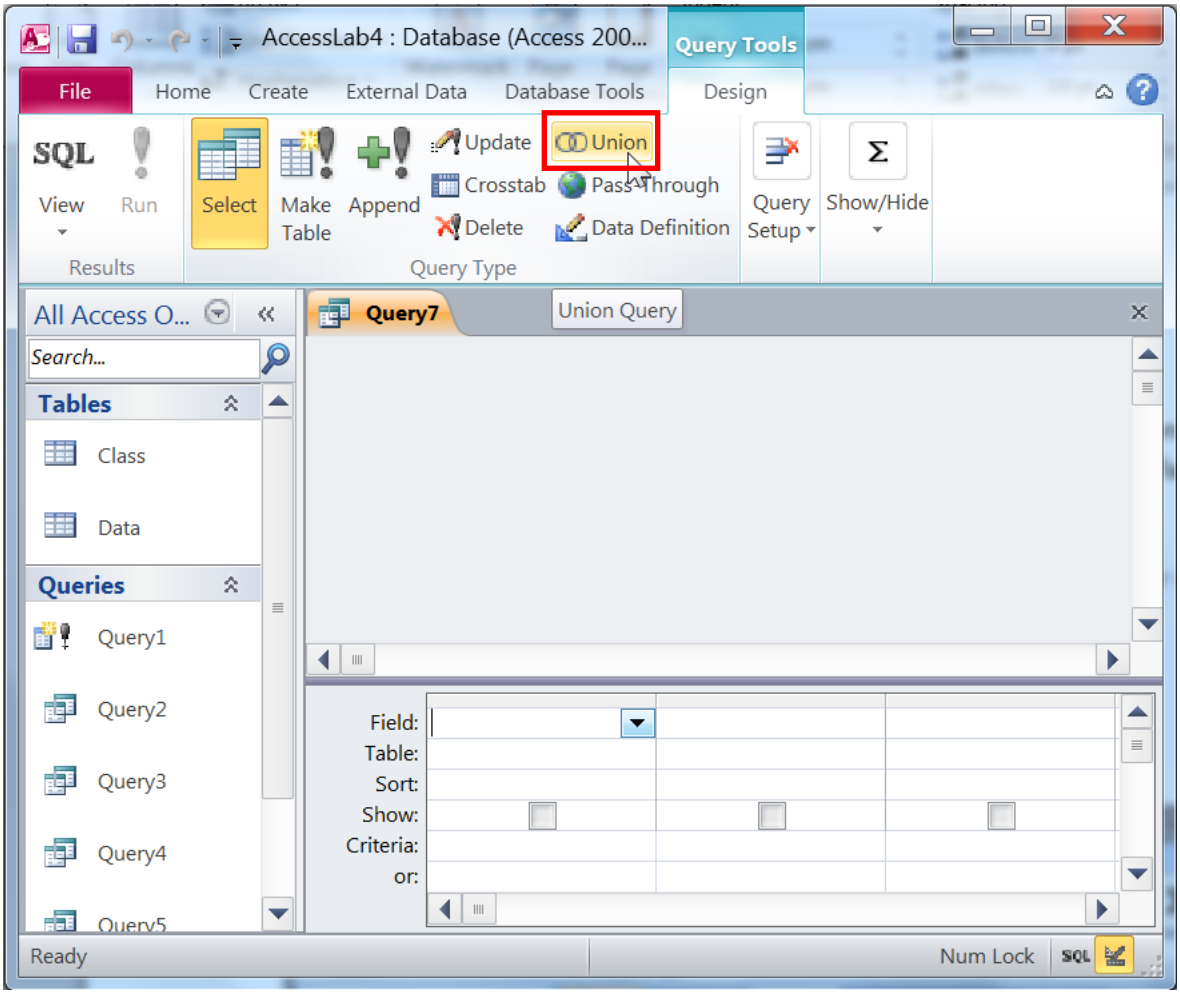
Union queries

Union queries allow you to combine multiple queries into the result of one query. Note that the Select queries you would like to combine must have the same number of fields in the result. These fields must be in the same order and have matching data types.

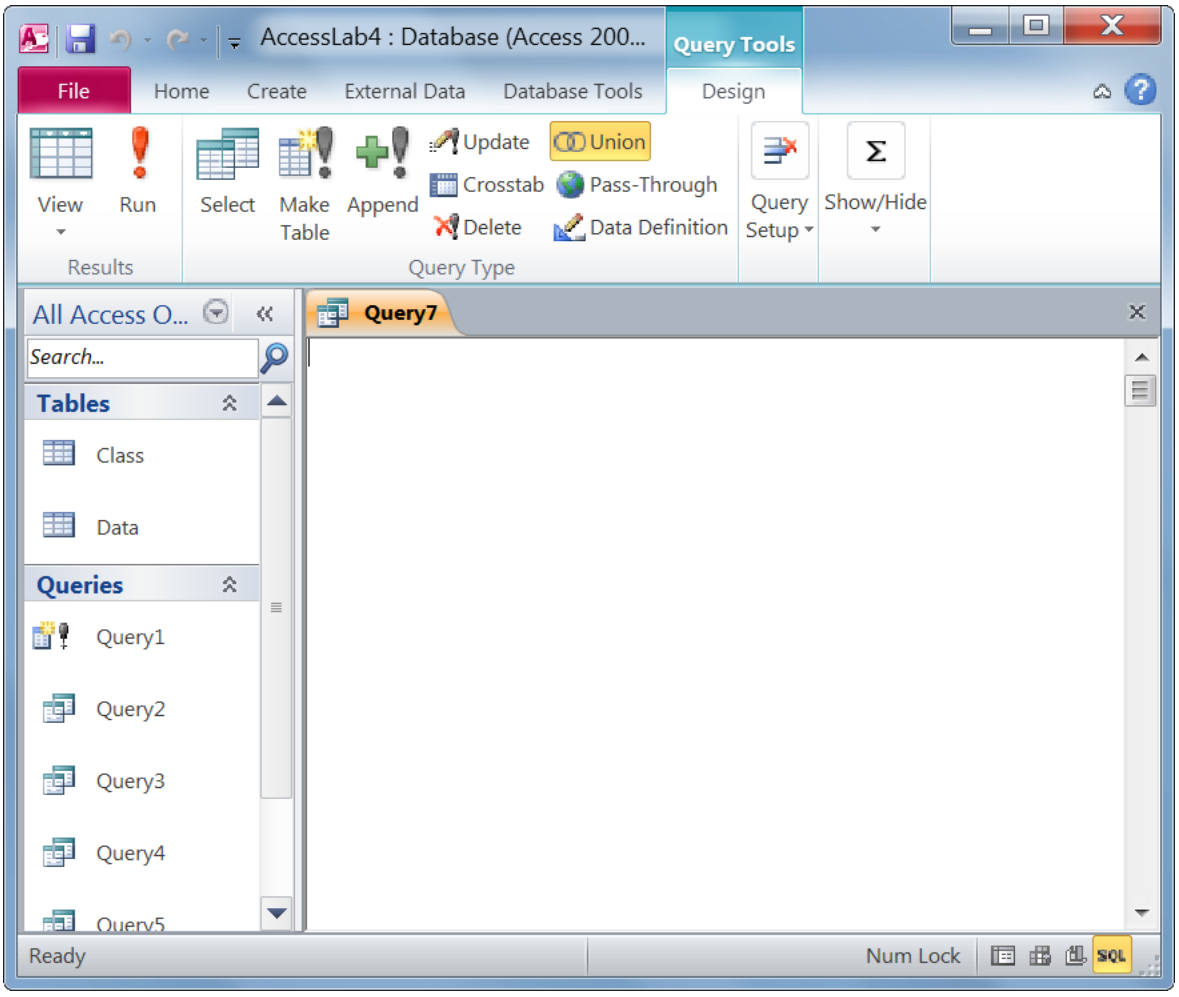
Let us create a union query to combine the 5 queries you created for the website rankings. Union queries must be done in SQL (Structured Query Language).

Create a new query and do not add any tables to the design view.

In the *Query Type* group, click on the *Union* button.

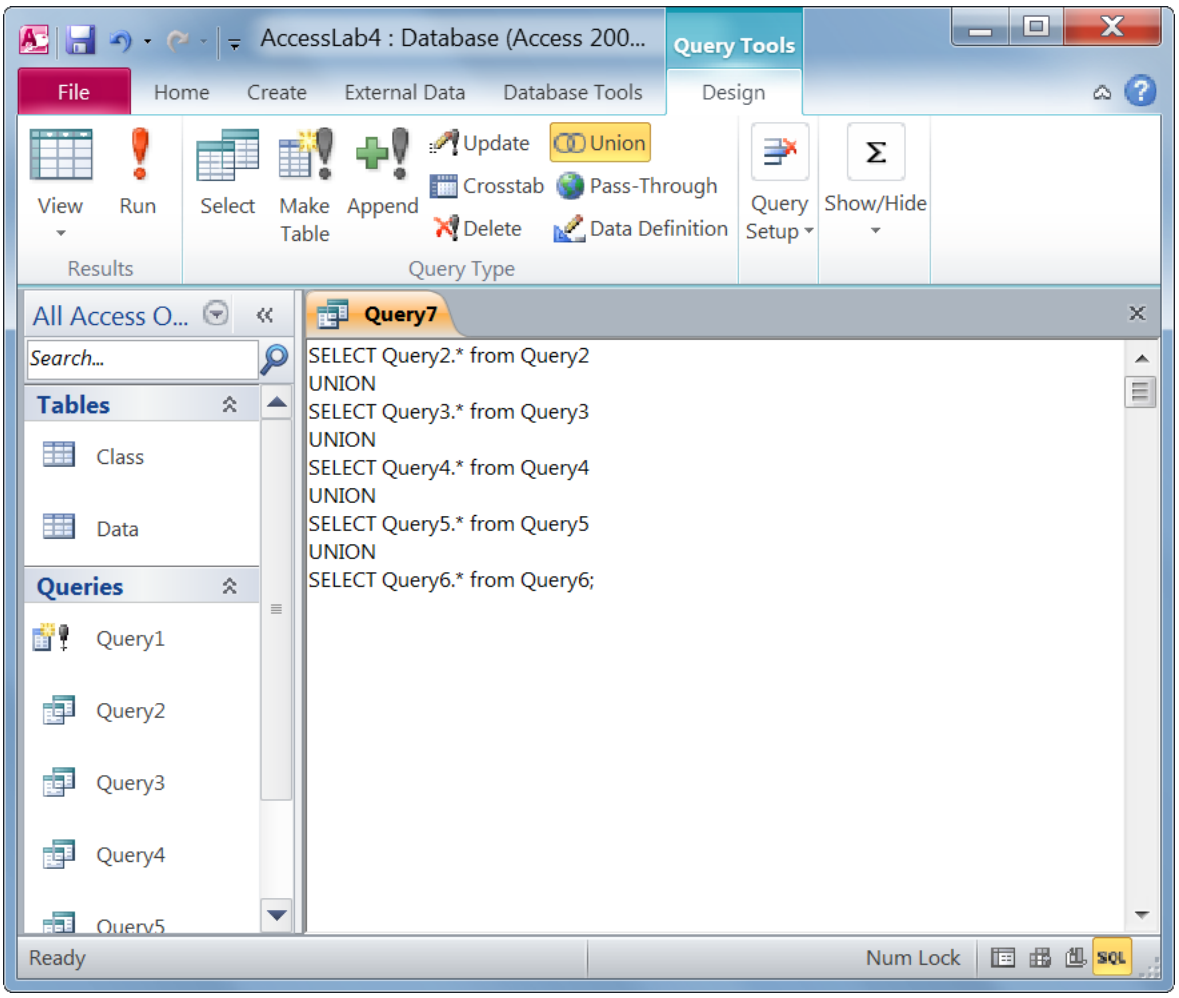


You then get a blank screen, where you will type in the SQL code for the union query.

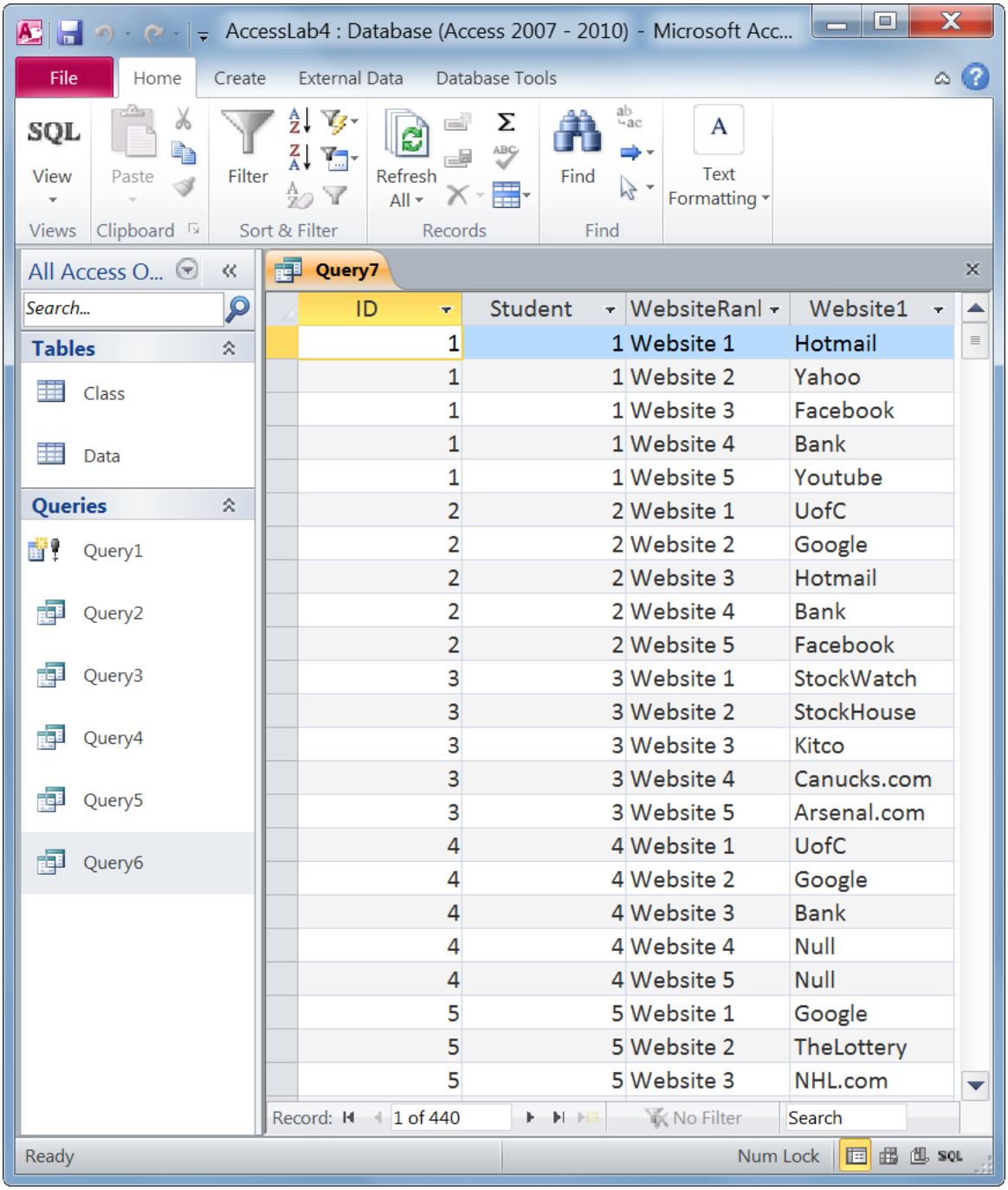


Type in the following:

```
SELECT Query2.* from Query2
UNION
SELECT Query3.* from Query3
UNION
SELECT Query4.* from Query4
UNION
SELECT Query5.* from Query5
UNION
SELECT Query6.* from Query6;
```


AccessLab4 : Database (Access 200...
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Query Setup
Show/Hide
Results
Query Type
All Access O...
Search...
Tables
Class
Data
Queries
Query1
Query2
Query3
Query4
Query5
Query7
SELECT Query2.* from Query2
UNION
SELECT Query3.* from Query3
UNION
SELECT Query4.* from Query4
UNION
SELECT Query5.* from Query5
UNION
SELECT Query6.* from Query6;
Ready
Num Lock
SQL

This will combine all the fields in each of the five queries into one query. Run your query.



Now that we have the union query, we could either make a new table out of it or append it to an existing table.

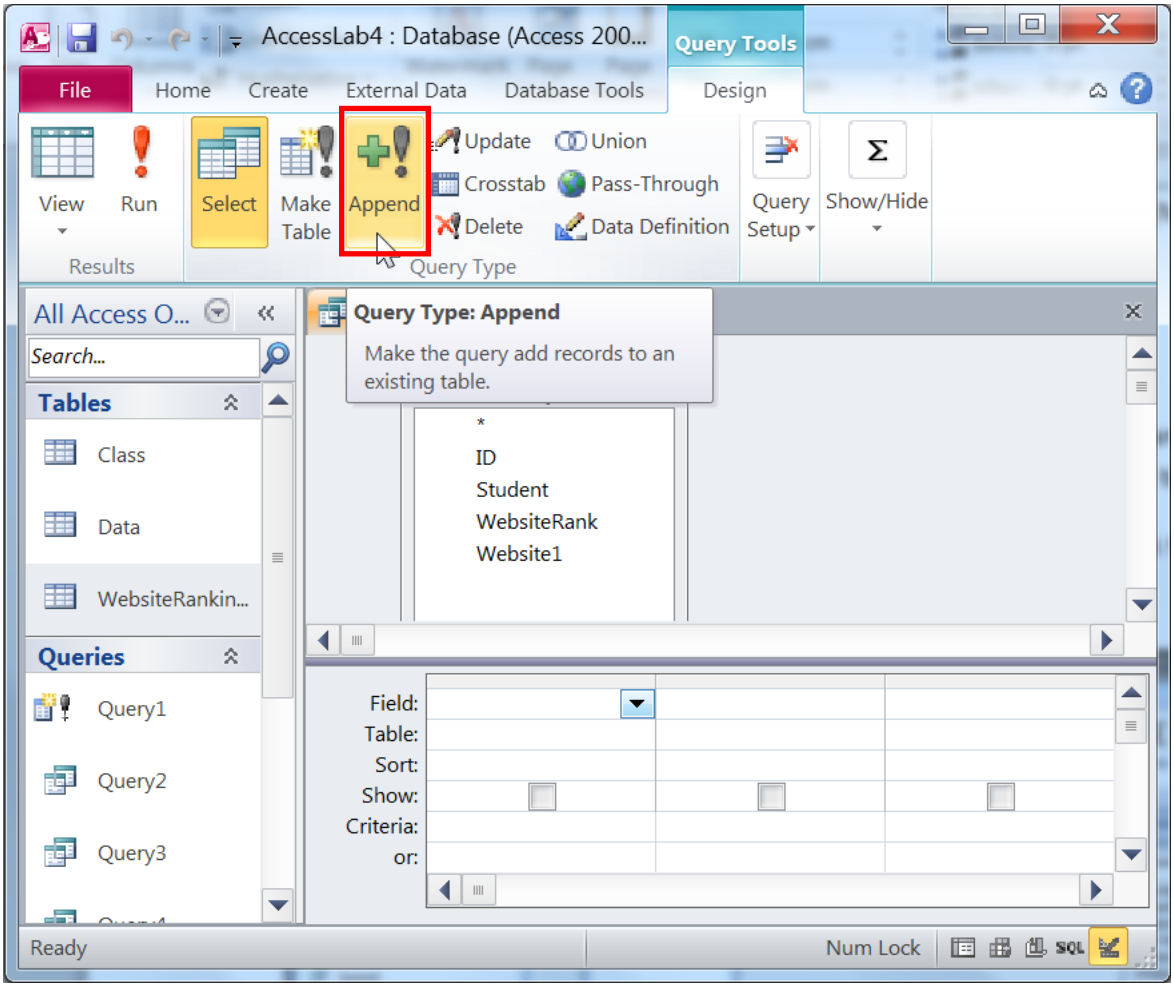
Exercise 3

Create a new blank table named WebsiteRankings. The table should have the following fields: WebsiteRankID (AutoNumber) as the primary key, ID (Number), Student (Number), WebsiteRank (Text), and WebsiteName (Text).

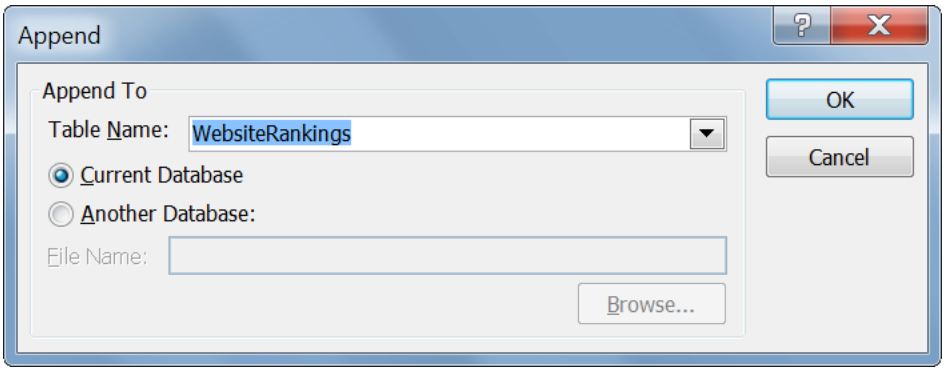
Append queries

When you already have an existing table, you can add records to it using an *Append* query. We are going to use an Append query to add the result of the union query we created to the WebsiteRankings table.

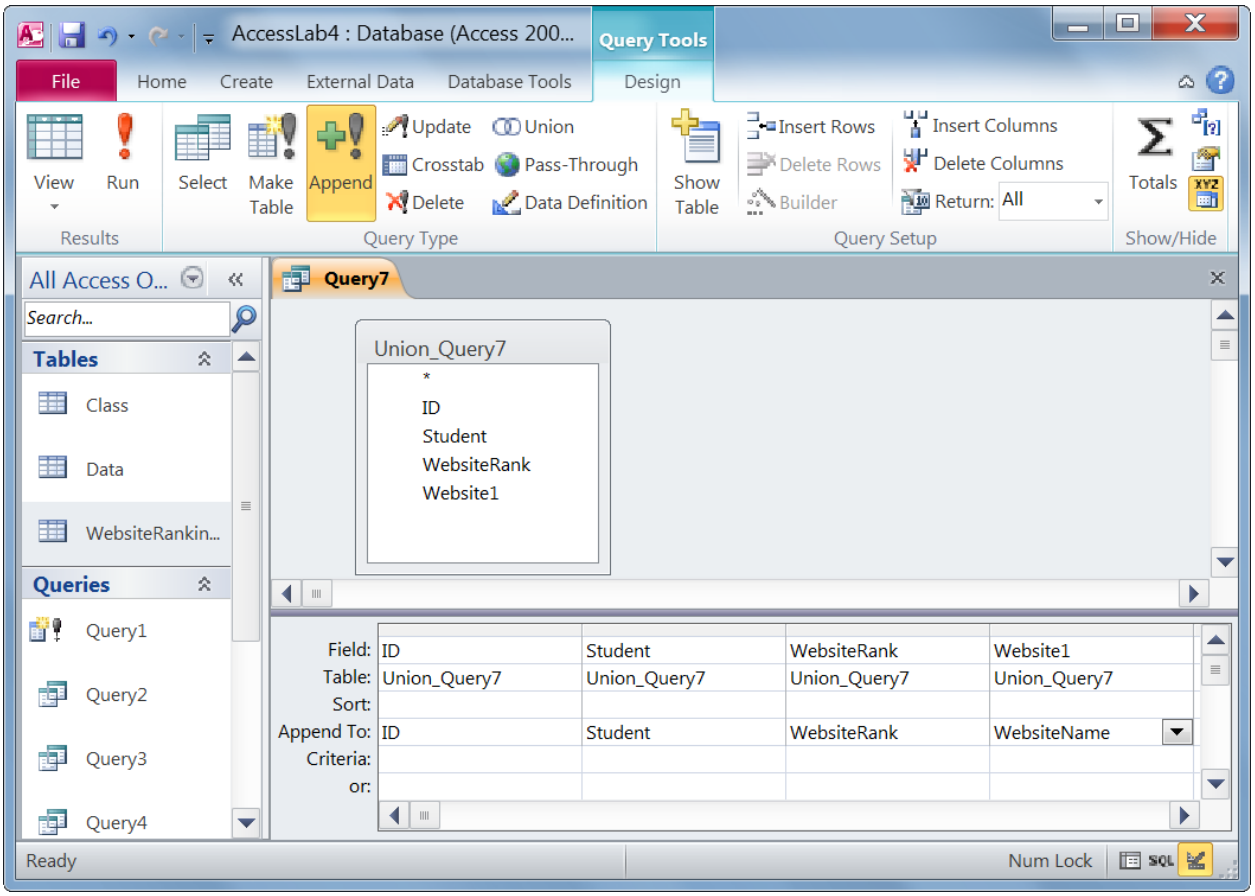
Create a new query and add the union query into the design view. Click on the *Append* button in the *Query Type* group.



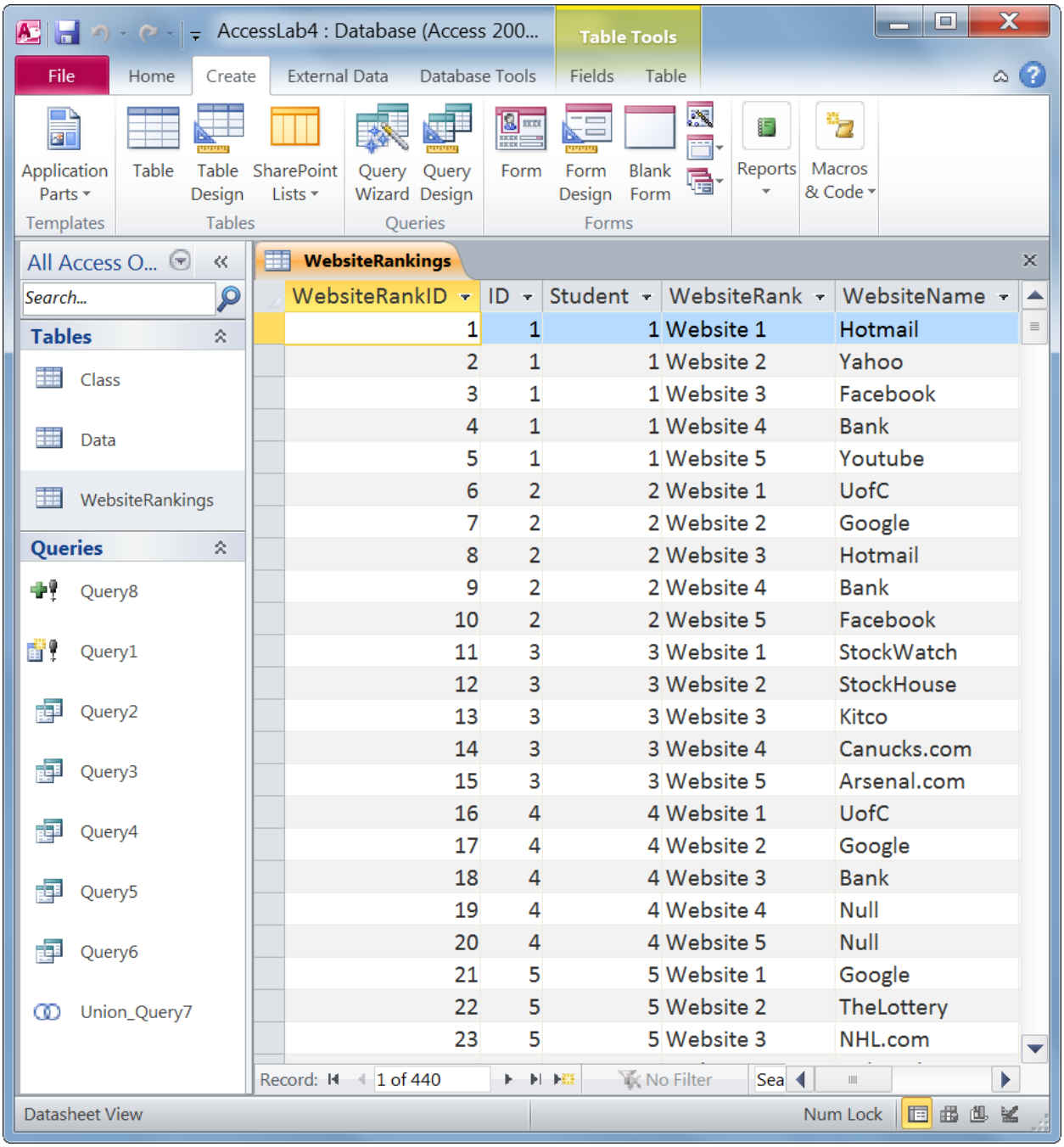
Select the table WebiteRankings from the menu and then click on the OK button.



Now all we need to do is match the union query fields to the table fields.



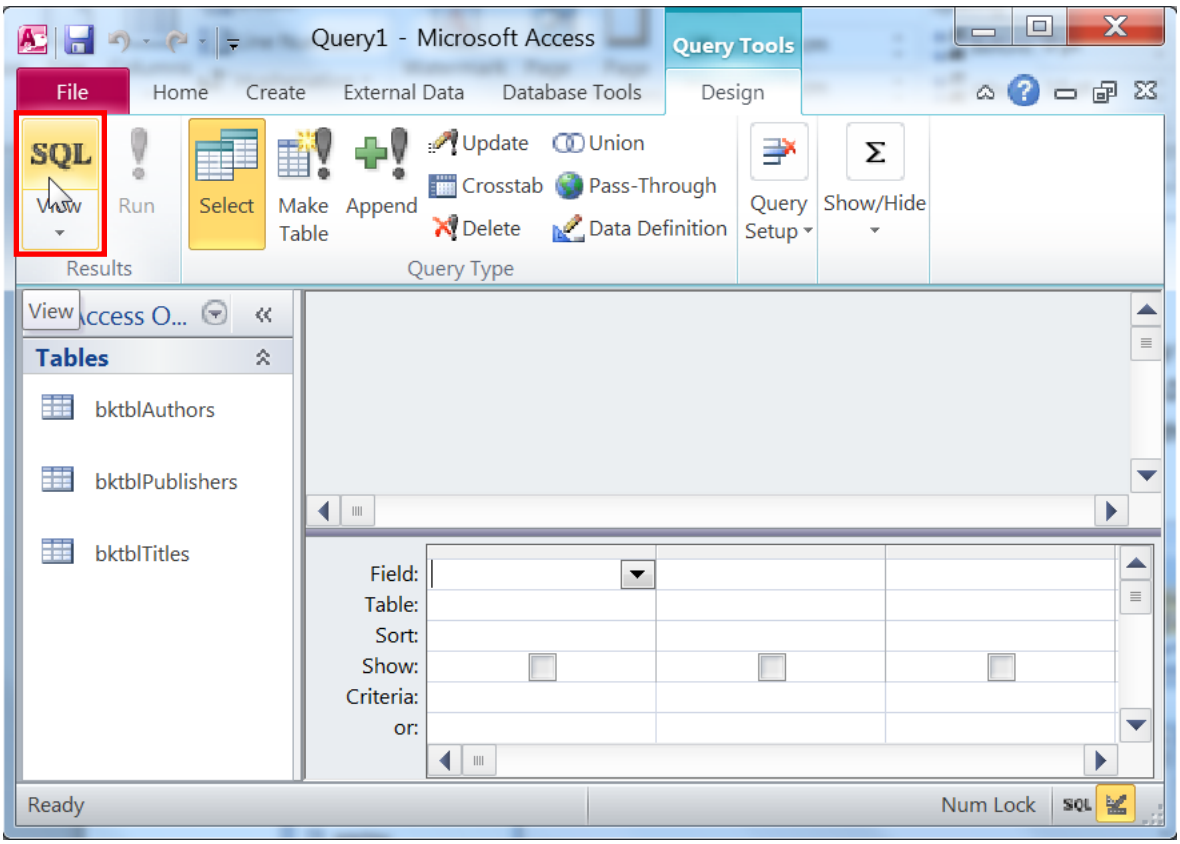
Run the query and press the Yes button of the warning message. The fields will get appended to the WebsiteRankings table. Now, the WebsiteRankings table looks as follows:



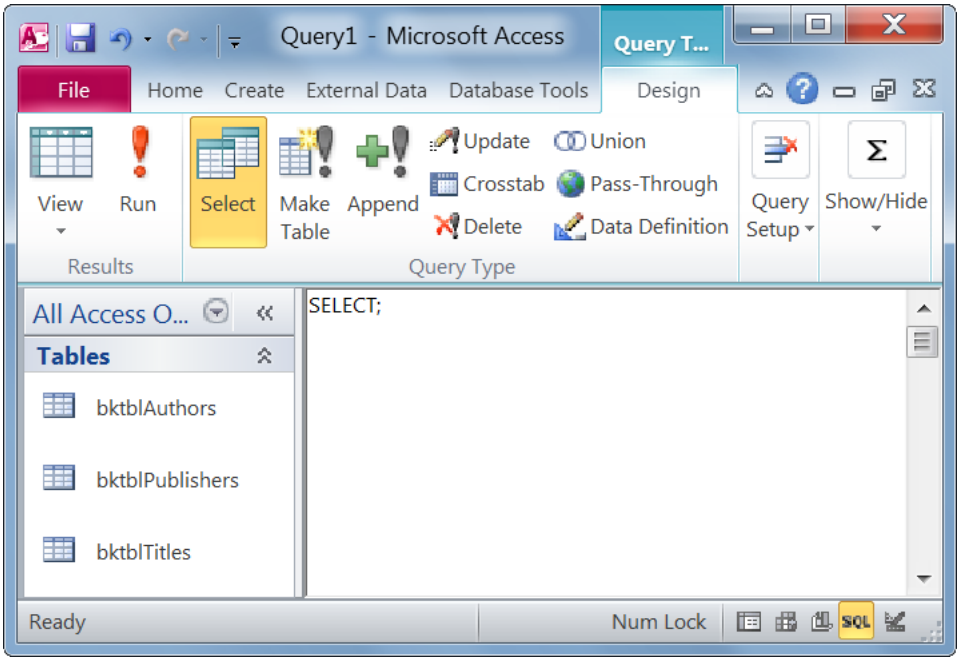
Lab 5: SQL in Access

Select queries

The SELECT statement is the most basic phrase in the Structured Query Language (SQL). Let us make a query that returns all records from a table using SQL. Open file AccessLab5.accdb and then click on *Query Design*. Close the *Show Table* dialog box without selecting any table. To go to the SQL view, click on the button labeled *SQL* in the *Results* group.



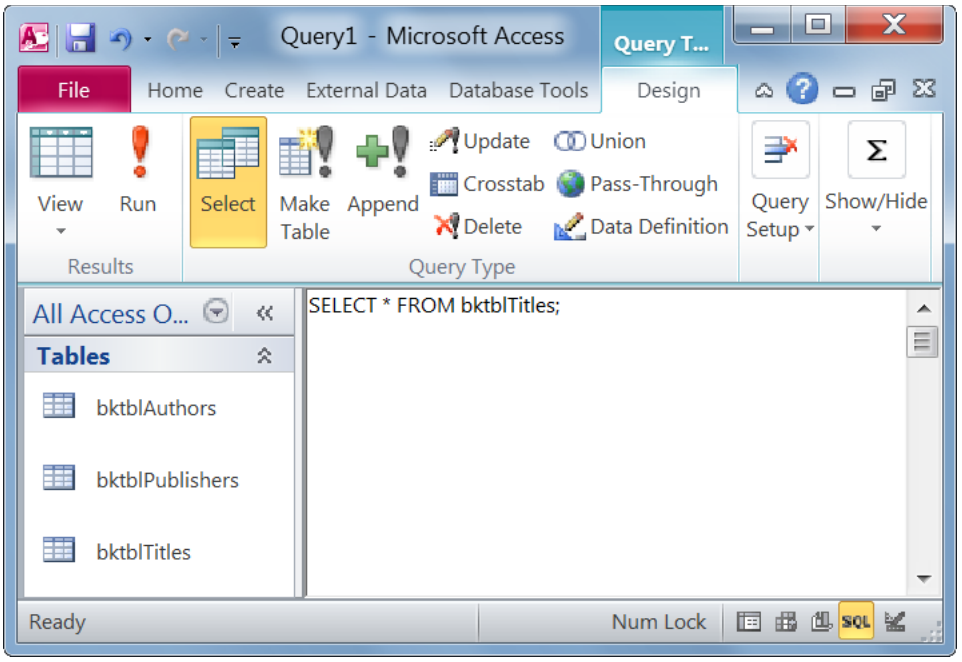
Since we are creating a SELECT query, Access knows enough to start it out for us.



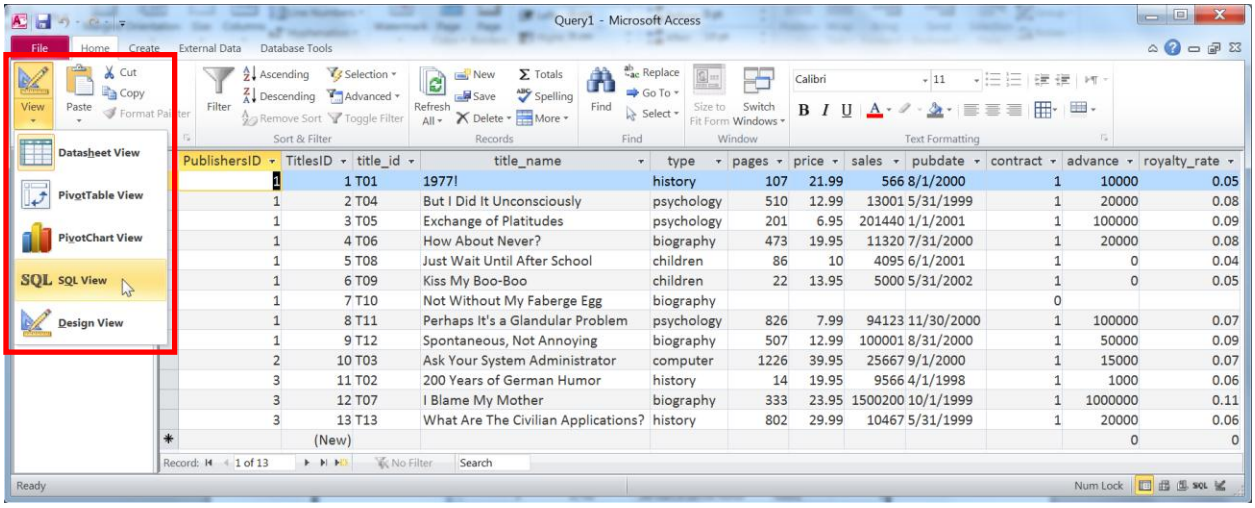
To get a query that shows all the records and fields in bktblTitles, type in (do not forget the semicolon):

SELECT * FROM bktblTitles;

Then, run the query.



Following is the result of the query. To get back to the SQL view, press the *View* button in the *Views* group and select *SQL View*.



Exercise 1

Create a query (Query2) in SQL that shows a *title name* and *type* from bktblTitles.

Exercise 2

Create a query (Query3) in SQL that displays a *title name* along with book's *publisher* and its author's *last name*. Do not worry if there are duplicated entries at this point.

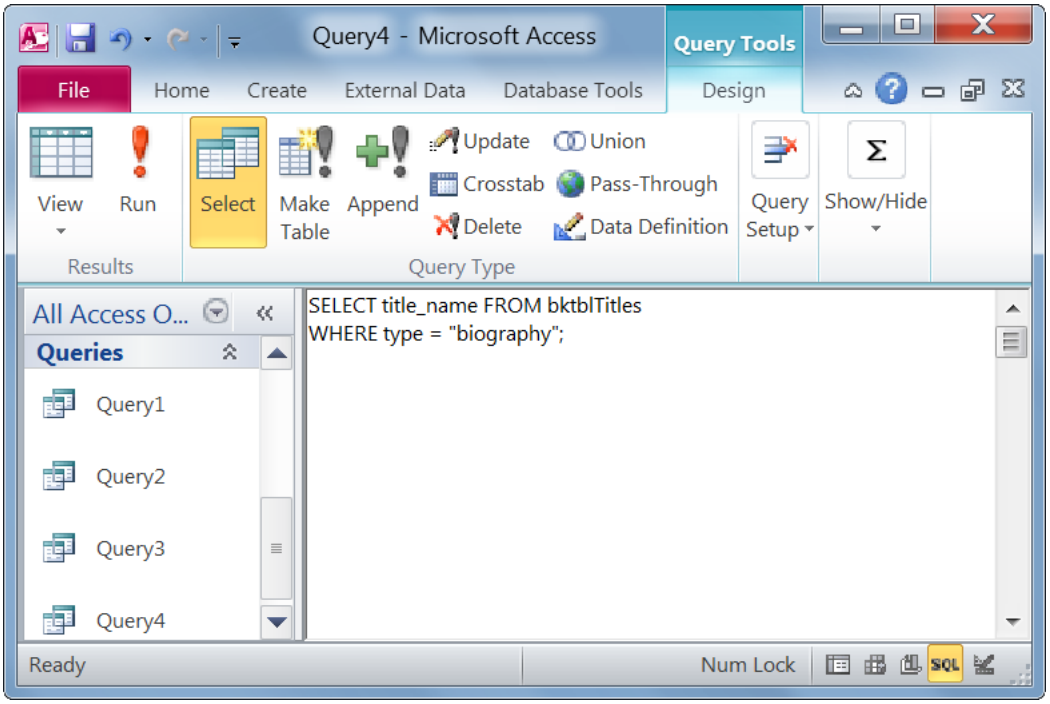
The WHERE clause

The WHERE clause in an SQL SELECT query provides a similar function to the Criteria field in the Access query Design View. WHERE allows you to set criteria that must be met by the records to be shown in the query result. WHERE also allows you to set relationships between fields in your queries.

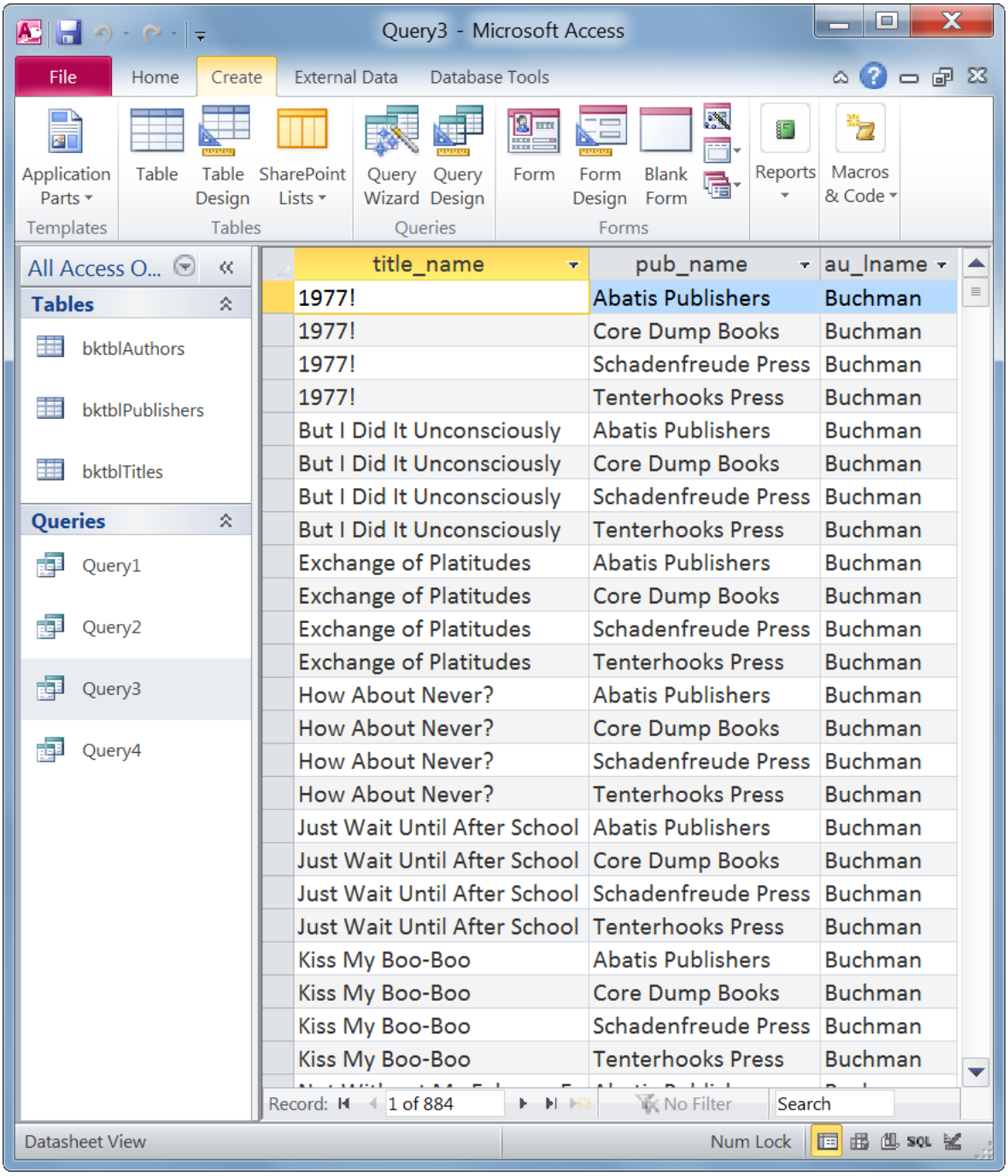
Create a new query (Query4) and open its SQL view. Let us create a query that returns the titles of books with types "biography". Type in:

```
SELECT title_name FROM bktblTitles
WHERE type = "biography";
```

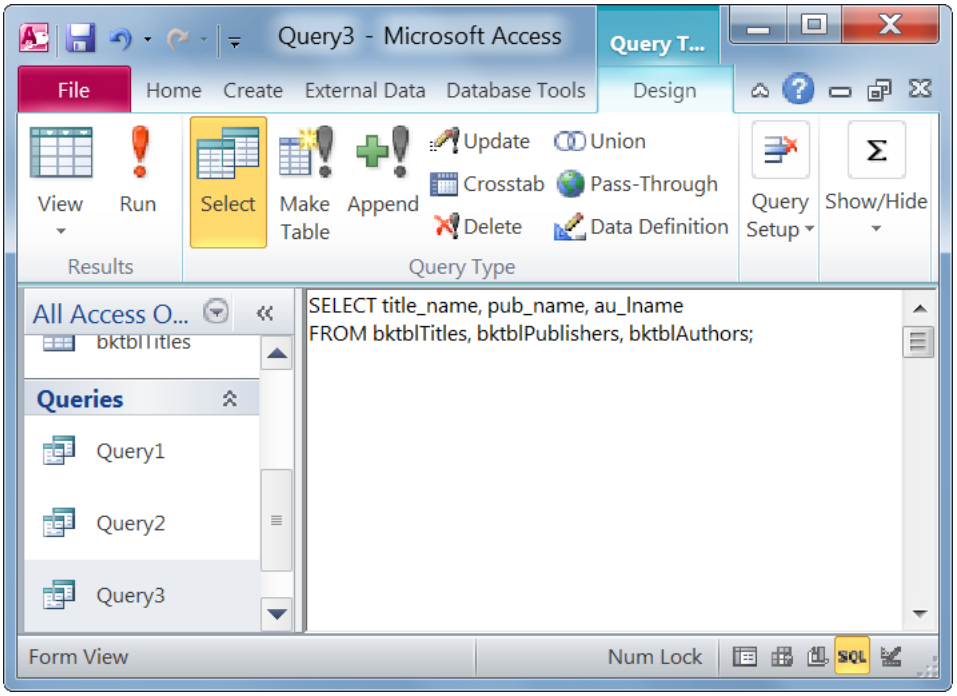
Then run the query.



Now let us get back to the query that was created in Exercise 2. You will notice that the query result is listing all possible combinations of authors, books, and publishers. This is because the query is not taking into account the relationships between the tables.


Query3 - Microsoft Access
File
Home
Create
External Data
Database Tools
Application Parts
Templates
Table
Table Design
SharePoint Lists
Tables
Query Wizard
Query Design
Queries
Form
Form Design
Blank Form
Forms
Reports
Macros & Code
All Access O...
Tables
bktblAuthors
bktblPublishers
bktblTitles
Queries
Query1
Query2
Query3
Query4
title_name
pub_name
au_lname
1977! Abatis Publishers Buchman
1977! Core Dump Books Buchman
1977! Schadenfreude Press Buchman
1977! Tenterhooks Press Buchman
But I Did It Unconsciously Abatis Publishers Buchman
But I Did It Unconsciously Core Dump Books Buchman
But I Did It Unconsciously Schadenfreude Press Buchman
But I Did It Unconsciously Tenterhooks Press Buchman
Exchange of Platitudes Abatis Publishers Buchman
Exchange of Platitudes Core Dump Books Buchman
Exchange of Platitudes Schadenfreude Press Buchman
Exchange of Platitudes Tenterhooks Press Buchman
How About Never? Abatis Publishers Buchman
How About Never? Core Dump Books Buchman
How About Never? Schadenfreude Press Buchman
How About Never? Tenterhooks Press Buchman
Just Wait Until After School Abatis Publishers Buchman
Just Wait Until After School Core Dump Books Buchman
Just Wait Until After School Schadenfreude Press Buchman
Just Wait Until After School Tenterhooks Press Buchman
Kiss My Boo-Boo Abatis Publishers Buchman
Kiss My Boo-Boo Core Dump Books Buchman
Kiss My Boo-Boo Schadenfreude Press Buchman
Kiss My Boo-Boo Tenterhooks Press Buchman
Record: 1 of 884
No Filter
Search
Datasheet View
Num Lock
SQL

The easiest way to set relationships in SQL is by using the WHERE clause. Go to the SQL view of Query3.



Now modify the SQL code so that it reads:

SELECT title_name, pub_name, au_lname
FROM bktblTitles, bktblPublishers, bktblAuthors
WHERE bktblTitles.PublishersID = bktblPublishers.PublishersID;

Run the query. Now the relationship between the bktblTitles and bktblPublishers tables has been taken into consideration.

Exercise 3

Modify Query3 so that the relationship between the book title and author tables is taken into consideration. Use the AND keyword.

Exercise 4

Create a new query (Query5) using SQL that displays book title ID, book title name, publisher's name, and author's last name. Only show entries if the state of a publishing company is "CA". Make sure the relationships are included in the WHERE clause.

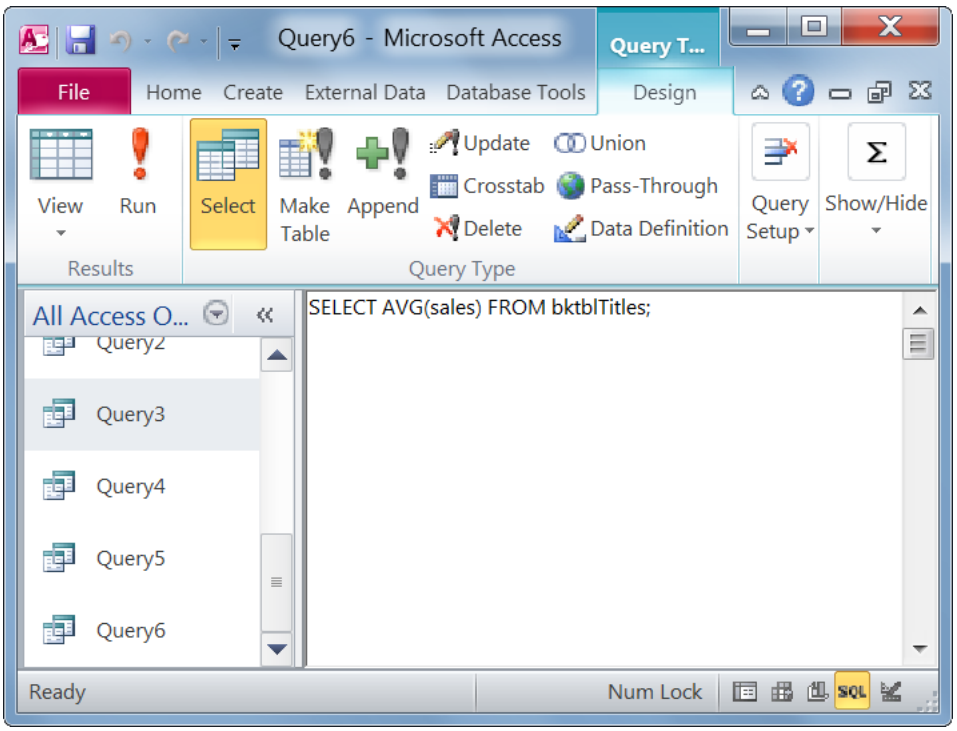
Aggregate functions in SQL

You can use functions like avg, count, sum, min, and max to perform calculations in sets of records using SQL.

Create a new query (Query6) and open it in SQL view.

Type in:

SELECT AVG(sales) FROM bktblTitles;

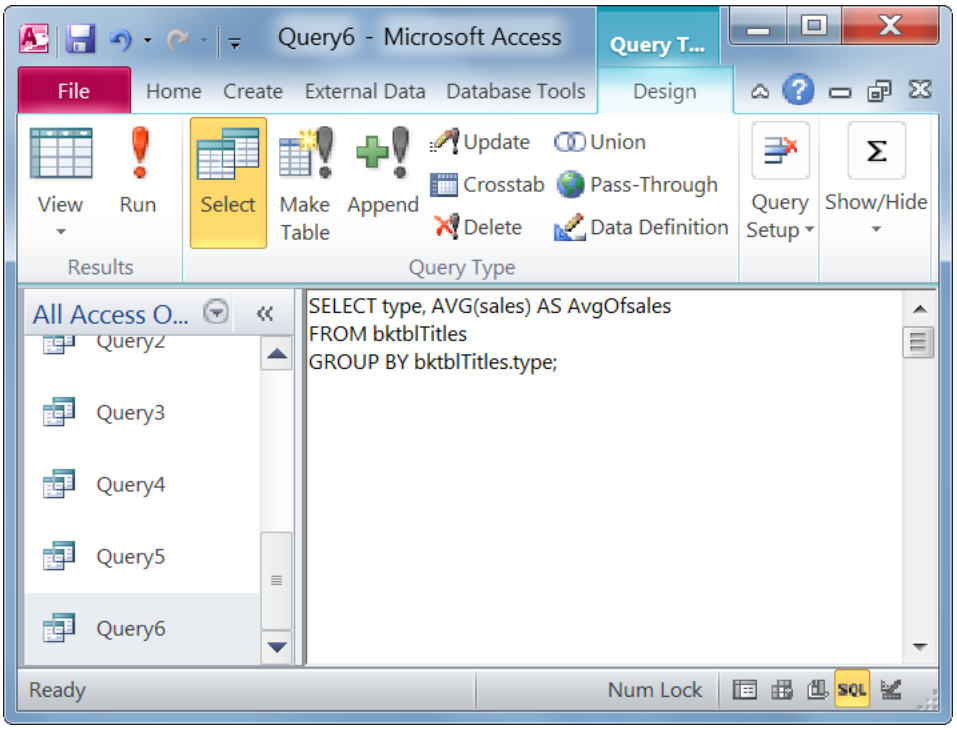


Run the query.

We can also perform these aggregate functions on groups of records that we define, using the GROUP BY clause. Modify Query6 so that it reads:

```
SELECT type, AVG(sales) AS AvgOfsales
FROM bktblTitles
GROUP BY bktblTitles.type;
```

The GROUP BY clause in SQL has exactly the same function as the Group by field in the Access query Design View. The AS clause allows you to modify the name of the aggregate function result.

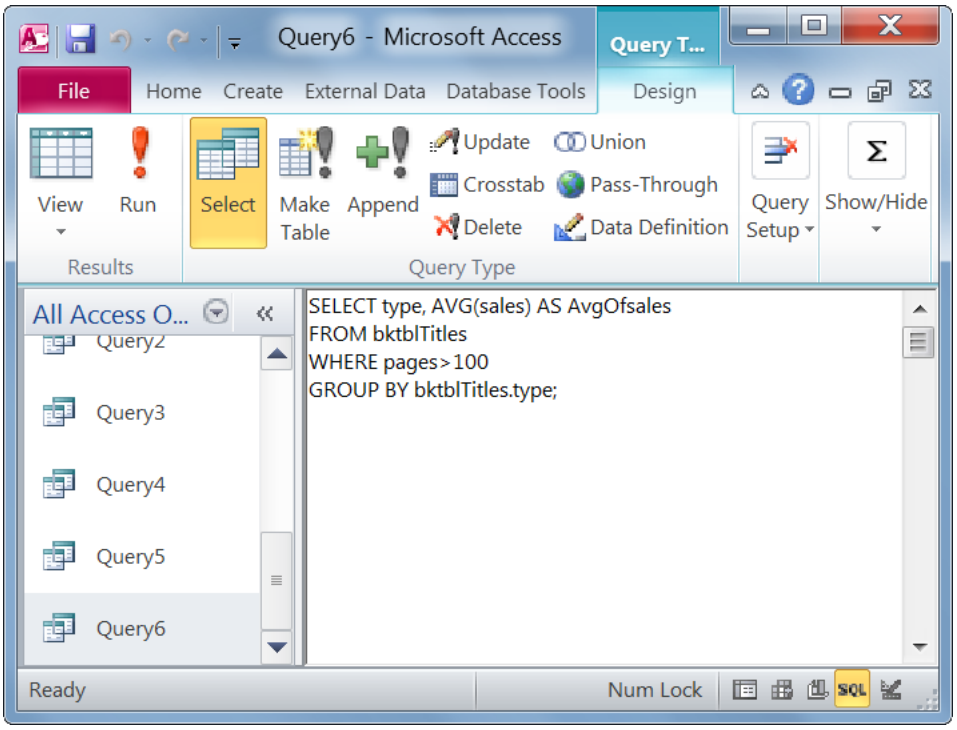


To set criteria in an aggregate query, use the HAVING clause instead of the WHERE clause. The HAVING clause selects which grouped records are displayed.

The HAVING clause sets criteria on groups of records and the WHERE clause sets criteria on individual records. You can have both in one query but you cannot use HAVING without a GROUP BY clause.

For instance, if you would like to filter the above query so that it only includes records with pages more than100 you would use the WHERE clause as shown below.

```
SELECT type, AVG(sales) AS AvgOfsales
FROM bktblTitles
WHERE pages>100
GROUP BY bktblTitles.type;
```

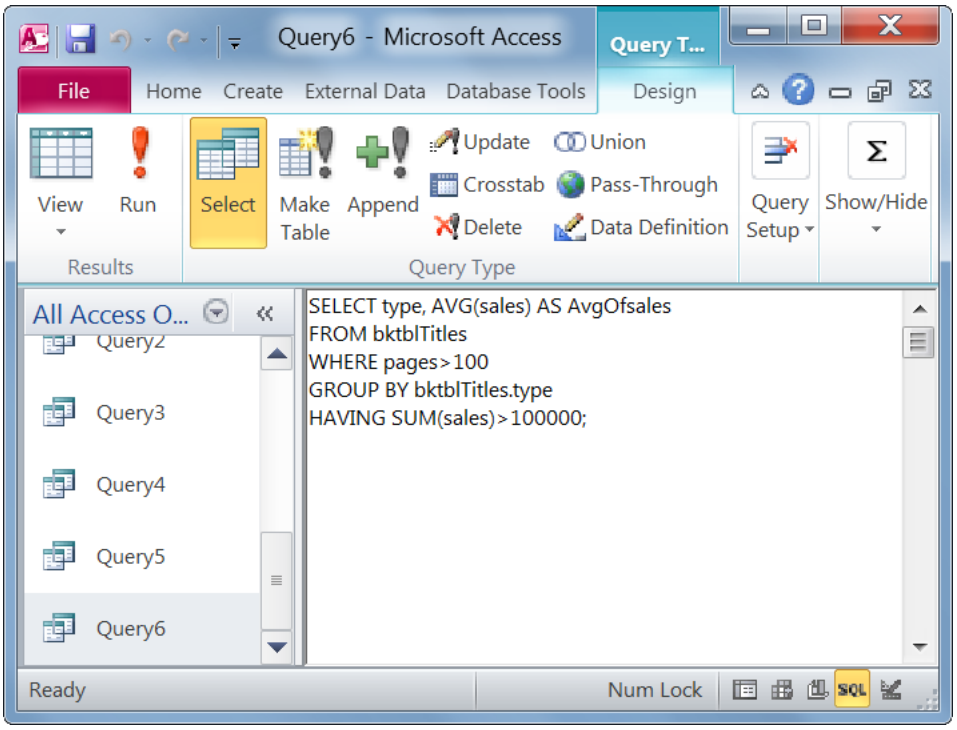


The pages >100 criterion is one that is conducted on individual records, before the average is calculated. In other words, the query calculates the average sales only for books with more than 100 pages. Using the HAVING clause instead will result in an error message.

Now if we only want to display groups of records where the sum of sales is greater than 100,000, we need to use the HAVING clause as shown below.

```
SELECT type, AVG(sales) AS AvgOfsales
FROM bktblTitles
WHERE pages>100
GROUP BY bktblTitles.type
HAVING SUM(sales)>100000;
```

This query calculates the average sales for each *bktblTitles.type*. Then the *bktblTitles.type* for which the SUM of sales is less than or equal 100,000 will be filtered out from the result.



Exercise 5

Using SQL, create a new query that displays the number of books each publisher has published. Name the count field CountOfBooks.

Exercise 6

Modify the query in Exercise 5 so that it only includes records of books with authors who live in the state "NY". Only show groups of records with number of pages of titles greater than 100.