Access Database Design

Technical Support Services

Office of Information Technology, West Virginia University

OIT Help Desk – (304) 293-4444

http://oit.wvu.edu/training/classmat/db/

Last revised: June 26, 2008

Table of Contents

Course Description

This is the second in a series of workshops about Microsoft Access 2007. It deals specifically with database design and maintenance.

The purpose of this installment is to expose you to the considerations involved in designing your databases and to introduce you to the various options that Access provides for importing and exporting data as well as the utilities provided for maintaining your databases.

Our goal is to assist you to learn the software, understand some basic concepts, and show you some tips and techniques so you can develop your database management/programming skills over time.

The six classes in the Access workshop series are:

- Introduction to Access
- Access Database Design
- Access Queries
- Access Form Design
- Access Reports
- Access Macros and Database Utilities

Thank you,

The OIT Technical Support Services Trainers

West Virginia University

Understanding Databases

A database management system (DBMS) is a computer application or program that is used to store, manage, and retrieve data in computer files. It is generally associated with large volumes of data or with files where selective retrieval is desired.

Microsoft Access is one example of a DBMS. While using Access is more complicated than using other components of Microsoft Office, such as Word, Excel, or PowerPoint, Access is nonetheless a relatively low-end DBMS. For larger databases requiring greater security and widely shared access, the University uses the Oracle or MySQL DBMS.

Data Organization

There is a hierarchy of components which constitute the collection of data maintained by any DBMS:

- Database
  - Record
    - Field
      - Value

A **database** is the entire file or collection of files maintained as a unit by the DBMS. The University, like most large organizations, has several databases used to manage its operations: a human resources database of all employees, an inventory database of the organization's physical assets, a financial database of receipts and expenditures, and a student database containing information about students and the courses they take, among others. Individual departments may have their own databases for different purposes, just as individual faculty may have databases for information such as bibliographies of useful works in the faculty member's field or data collected during an experiment.

The database is composed of **records**, which are structured collections of closely related data. The nature of the relationship among the data in a record will depend on the purpose of the database. In a human resources database, each record will contain information about one employee. In a bibliographic database, a record would correspond to a book, journal article, or other written work.

The structure of a record is determined by the set of **fields** from which it is composed. Each field is a place where data with a particular meaning is kept. A human resources database record would include fields for name, address, social security number, date of employment, salary, and other information. A bibliographic database record would include fields for a work's title, authors, publisher, date of publication, and other information.

The content of each field in a record is its **value**—the specific text, number, date, or other information stored in that field of that record.

There are different types of databases, but we will be looking only at **relational databases**, since that is the type of database managed by Access. In a relational database, there is an additional level in the hierarchy of data organization:

Database
- Table
  - Record
    - Field
      - Value

The records of the database are organized into **tables**, each of which expresses a particular relation among the data (hence the name relational database).

In our example of the human resources database, there might be a table for each employee and another for each department in the organization:

Employee_Name	Employee_SSN	Employee_Dept	Employee_HireDate
Smith	123456789	Accounting	11/12/1989
Jones	123456678	English	3/15/1992
Brown	123452233	Math	12/2/1997
Cooper	134263525	English	5/5/1994

Department_Name	Department_Location	Department_Authorized_FTE
Accounting	123 Penny Lane	6
English	456 Chaucer Ct.	8
Math	789 Plane St.	4

Table Design

An important part of designing a relational database, including Access databases, is determining what tables will be used to organize the data.

Consider the simple case of a class list and suppose that we wish to keep track of the class's name, number, and department; the name of the teacher; and the teacher's office number. We could include all of the information in one table, as shown below.

Class_Name	Class_No	Class_Teacher	Class_Dept	Class_Office
Calculus	200	Mr. Brown	Mathematics	G105
Interior Design	304	Ms Smith	Design	G103
Algebra	101	Mr. Brown	Mathematics	G105
Geometry	110	Mr. Jones	Mathematics	G107

However, if we do so, then repeated department and office information must be entered each time the same teacher is assigned to teach another class. If one bit of information changes (e.g., Mr. Brown moves from G105 to G110) then every occurrence of data containing Mr. Brown's information must be located and updated to avoid data inconsistency. A different problem arises if Mr. Brown takes a term off for any reason and so does not teach a class. We would either have to retain a record with Mr. Brown's office and department information but with no values in the class name and number fields (a so-called insertion anomaly), or we would have to remove Mr. Brown from the database altogether (a deletion anomaly).

Instead, we use two tables to represent the data, one with the minimal information needed regarding the class and a second table with information about the teachers that applies regardless of the classes they are teaching.

Class_Name	Class_No	Class_Teacher
Calculus	200	Mr. Brown
Interior Design	304	Ms Smith
Algebra	101	Mr. Brown
Geometry	110	Mr. Jones

Class_Teacher	Class_Dept	Class_Office
Ms Smith	Design	G103
Mr. Brown	Mathematics	G105
Mr. Jones	Mathematics	G107

Example 1

The common field, Class_Teacher, would be used to connect the tables, thus allowing the relation expressed in the one-table design to be recovered from the pair of tables.

Some care must be exercised in decomposing larger tables into smaller ones. For example, suppose that our original table were split up differently, like this:

Class_Name	Class_No	Class_Dept
Calculus	200	Mathematics
Interior Design	304	Design
Algebra	101	Mathematics
Geometry	110	Mathematics

Class_Dept	Class_Teacher	Class_Office
Design	Ms Smith	G103
Mathematics	Mr. Brown	G105
Mathematics	Mr. Jones	G107

Example 2

While this decomposition also avoids the redundancy present in the single table, it gives us no way to determine which individual is teaching each Mathematics class, even though we have the Class_Dept field in common to connect the tables to one another.

We can list now some of the undesirable characteristics that we wish to avoid in designing a relational database:

1. Redundancy. Avoid repeating values unnecessarily. It wastes disk space and slows processing while introducing the possibility of inconsistency in the database if not all occurrences of a value are changed at the same time.
2. Insertion and deletion anomalies. Objects whose existences are independent outside the database should be represented independently inside the database. In our example, teachers are represented in a table separate from classes since there can be teachers who are employed but not currently teaching a class. Likewise, there can be classes in the course catalogue that are not offered every term and thus would not have a teacher during some terms.
3. Loss of information. We should be able to recover from the database any relation that exists among the objects modeled by the database. If a specific teacher teaches a specific class, then that relationship should be evident in the database.

Keys and Relationships

Primary and Foreign Keys

For the sake of rapid, unambiguous retrieval of data, each table should include one or more fields that are used to identify uniquely each record stored in the table. This field or combination of fields is called the **primary key** of the table.

The contents of the primary key:

- Cannot be duplicated within the table.
- Cannot be null.
- May consist of more than one field.

The primary key is used frequently so its name should not be very long or hard to remember. Additionally, the size of the key's value affects the speed of database operations. When a table’s data are stored on a computer disk, the DBMS creates an index based on the table’s key to speed the location and retrieval of that data.

The primary key is used to link the table with other tables in the database. When a primary key in one table (known as the *source table*) is linked to another table (known as the *target table*), the connecting field in the target table is called the **foreign key**. A foreign key must have the same structure, data type, and field size as the associated primary key, but it does not have to have the same name. Also, the foreign key in a relationship between two tables need not be a primary key in its own table.

In Example 1 above, we could use the teacher's last name as the primary key in the second table. The same field becomes a foreign key in the class table. It could not be a primary key in the class table because:

- The class table would already have a primary key (probably a combination of class name and number) and a table is restricted to having only one primary key.
- The entry for Mr. Brown is repeated in multiple records of the class table which violates the first rule of the primary key.

Thus in a key pair: the foreign key, while it appears identical to the primary key, does not have to adhere to the same rules as the primary key. It can be duplicated and in some cases it can be null. However, it must have the same structure as the primary key.

Defining Relationships

For example: in a typical classroom environment Students, Teachers, and Classes would be stored in separate tables.

- Students could be related to classes by transcript entries and grades.
- Teachers could be related to classes by teaching assignments.
- Classes could be related to students by enrollment and attendance.

To establish a relationship, we link one table's primary key to a foreign key in another table. But, how do you decide which key to place in which table? To answer this question, we must determine the nature of the relationship.

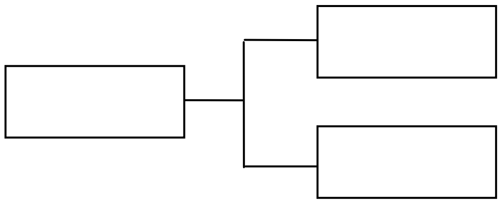
Types of Relationships

There are four basic types of relationships:

- One-to-many
- Many-to-one
- Many-to-many
- One-to-one

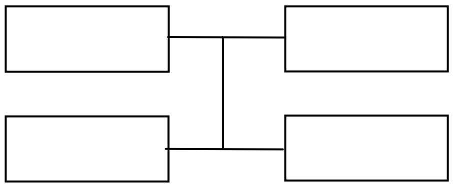
However, the one-to-many relationship is just the many-to-one relationship viewed in the other direction, so we can consider them as one, leaving us with only three that we need to examine.

One-to-Many Relationship (1 - ∞):



The one-to-many relationship is the most common type. As an example, one teacher might teach several classes. That is, the teacher would only have one entry in the Teacher table but may have several entries in the Class table, depending upon teaching load. In this relationship, we always link the primary key of the "one" side of the relationship (the source table) to the foreign key on the "many" side (the target table). In our previous example, the primary key of the teacher would be placed into each of the class records for classes that the teacher taught. A search of the class table using the teacher key would produce a list of classes taught by the teacher.

Many-to-Many Relationship (∞ - ∞):



In the many-to-many relationship, many records from one table would be related to many records in another table. For example: many students would attend each class and each student would attend many classes.

Our first impulse is to design a database that would support this relationship by either:

- Including the primary key for every class attended by a student in the student record, or
- Including the primary key for every student attending a class in the class record.

This approach is faulty since it would result in:

- Huge records in a table with each record containing a large number of similar fields. Some records would contain many blank fields. This could be a significant waste of system resources as well as being difficult to implement.
- Inefficient searching and retrieval since a search would have to examine several fields in each record.
- Data that is hard to update since you would have to search the length of each record to find the spot to insert the next cross-reference key.

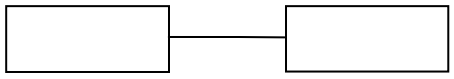
To avoid these problems, we create a relationship between the tables that uses a third table as a common target for both of the other tables. This would break down the many-to-many relationship and convert it into two one-to-many relationships. We would then link the primary keys from each of the original tables (both treated as sources) into the foreign keys in the third table (common target). Additionally, data that is common to the relationship between the tables (such as grades or status codes) could also be stored in the target table to avoid duplication.

In our example, we might create a class detail table called ATTEND that contains the Class ID and the Student ID for each student attending each class. The new table would have a one-to-many relationship with students and another one-to-many relationship with classes. That is:

- For each Class ID, there are potentially many records for attendees, and
- For each Student ID, there are potentially many records for enrolled or completed classes.

This new table could also hold data fields that are common to the relationship such as students' grades, attendance records, and their class status.

One to One Relationship (1-1):



A one-to-one relationship normally indicates a design flaw in the database or a trivial relationship. In most cases, the two tables in question could be combined without any problems. However, if the combined table would have a large amount of empty space, then perhaps they should not be combined for efficiency's sake. The rule therefore, is to strive for the minimum number of data tables without affecting the efficient use of data storage.

Database Design Process

The following procedure is a step-by-step guide to designing and implementing a database. It is, however, only a guide. The exact steps involved in your particular application will be dependent upon the way you organize your data and what you wish to accomplish.

Step 1: Determine the purpose of the database.

- Ask yourself what information you want from the database and determine what data you need to store to provide that information.
- Consider how the data items (fields) will be grouped together within subjects (tables).
- Talk to the users of the database. What data items do they currently collect? What reports do they want the system to produce? What questions do they want the database to answer?

Step 2: Determine the tables you need.

- Look at the way data items are grouped together naturally about a specific subject. Examine each item and ask what it is about.
- Look for ways to avoid storing or updating the same item in two places.
- Look for ways to eliminate redundant data.

Step 3: Determine the fields you need.

- List the data items that will be stored in each table. Do not attempt to define relationships and pointer fields at this point, simply list them.
- List the characteristics (or properties) of each item.
- Examine each item carefully to see if it is really related to the table subject. If not, consider moving it to another table. If another logical table does not exist, re-examine the table structure.
- Do not include derived or calculated data fields in a table. Make sure you have all of the basic data needed to calculate a value when it is needed. For example, don’t store a person’s age in a table, since it would need to be updated each year. Instead, store the person’s birth date and compute the age when it is needed. Exception: Store a calculated value if it is extremely complex to calculate, requires a large amount of calculation time, and is used frequently.
- Make sure that you have all of the data you need. Re-examine all the desired input forms and reports to make sure that you are collecting everything you need. Re-think the questions that you will ask of your database. Do you have everything you need to answer them?
- Store information in its smallest logical parts. For example, put a person’s first and last name in different fields so that each can be retrieved separately for sorting or for reports. But put all parts of a street address in the same field unless you have some special need to use the street name separately from the number on that street.

Step 4: Determine the relationships:

- Select the primary key fields. Make them unique and easy to remember for retrievals. Keep the field size small to increase speed without distorting data.
- Set up relationships between tables. Add foreign key fields where necessary.
- Convert all relationships to one-to-many by adding linking tables where necessary.

Step 5: Refine the design:

- *Turn on the computer*. It is important that the preceding tasks be accomplished off-line. If you start creating your database without first thinking through the design, you will generally focus more on the way you implement instead of an efficient design.
- Create the database structure as defined and enter a few data records into each table to test your procedures, data entry forms, and reports.
- Did you forget any fields? Did you select a good primary key for each table? Are you repeating data frequently as you enter it into a table? Do you have tables with many fields, most of them empty?

Step 6: Repeat the design process as necessary to minimize or eliminate questions or problems.

Common Design Problems:

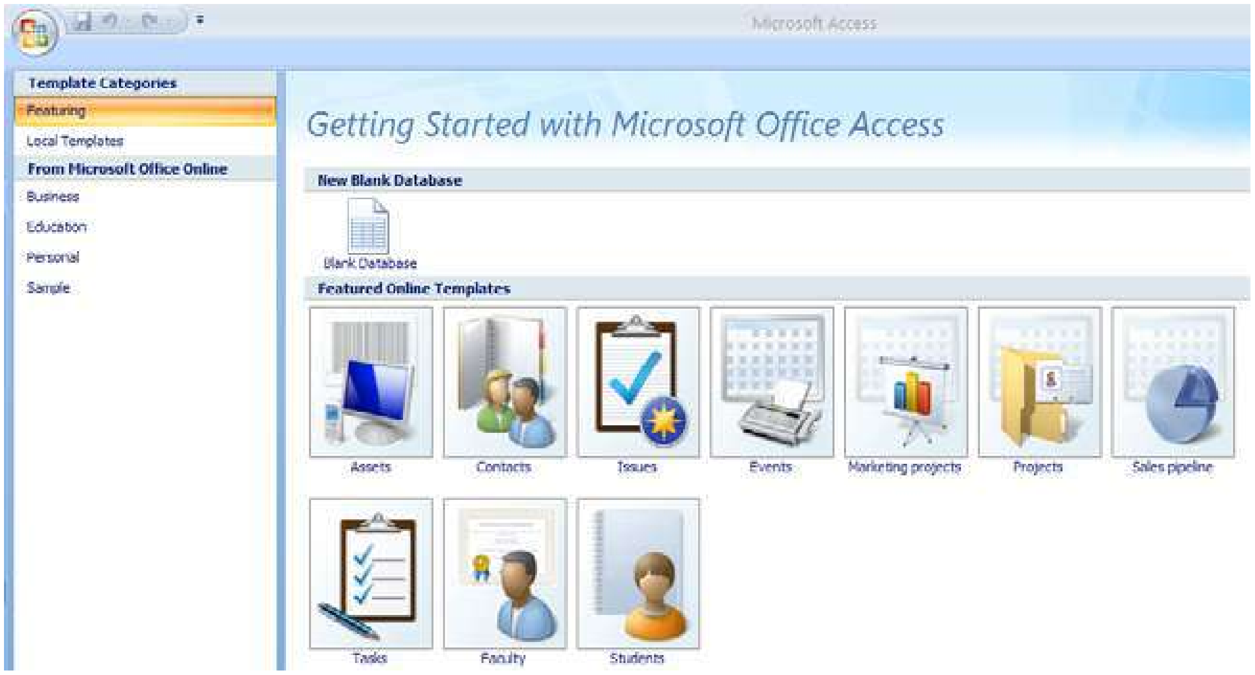
- You have a table with several fields that do not relate to the subject. Redistribute fields into tables defined by subject.
- You have a large number of records with many blank fields. Break the table into two or more subject-oriented tables.
- You have the same field repeated in several tables. Try consolidating all of the information relating to a specific subject into a single table.

Creating a New Database

Access provides two principal methods for creating a new database. You can start completely from scratch with a blank database, or you can start with a database template (a number of which are provided by Microsoft) and modify it to meet your specific needs. We will devote most of our time to the first method, but let's briefly consider the second method.

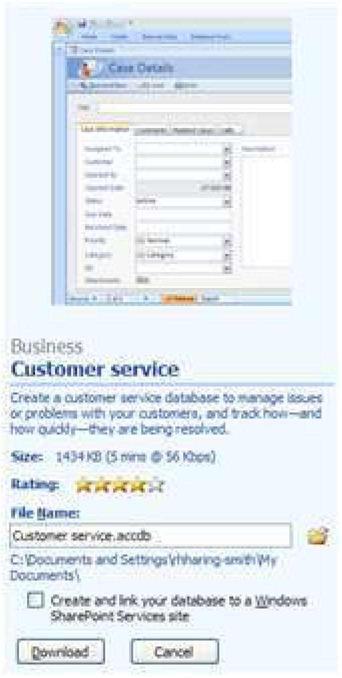
Creating a Database from a Template

When you start Access, you see the following screen, in part:



The icons under the heading "Featured Online Templates" are among the templates upon which you can base your new database. You can click on one of the icons in order to download the template from Microsoft or select one of the categories listed to the left under "From Microsoft Office Online" to see a somewhat broader selection of templates in each of the categories.

When you select a template from any of these sources, a description of it will appear on the right side of the Access window, including a Download button that you can click to retrieve the template after specifying your own name for the database and choosing the location on your computer where it will be stored. (The software will check that you are using a valid copy of Access before the download proceeds.) An example description appears to the right.



Business

Customer service

Create a customer service database to manage issues or problems with your customers, and track how—and how quickly—they are being resolved.

Size: 1434 KB (5 mins @ 56 Kbps)

Rating: ★★★★☆

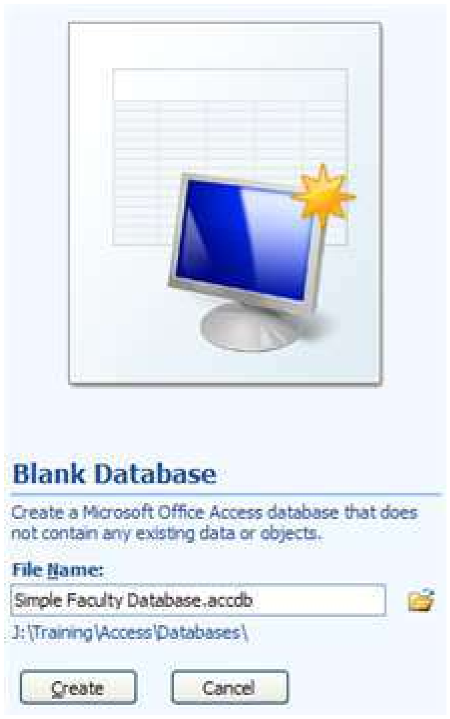
File Name: Customer service.accdb

C:\Documents and Settings\rhharing-smith\My Documents\

☐ Create and link your database to a Windows SharePoint Services site

Download   Cancel

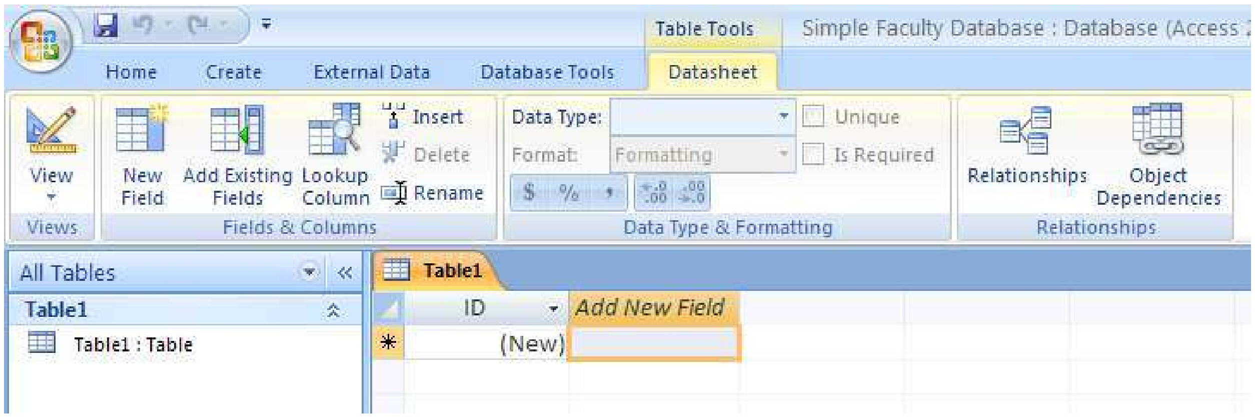
When the download is complete, the file will open to show you a complete but empty database, including tables, queries, forms, reports, and macros as needed. You can use the database as is by populating the tables with your data, or you can first modify the database design by adding or removing objects, changing field names or their properties, and otherwise tailoring the database template to your specific needs. The methods by which you'll make such changes are just the same ones you've been hearing about, or will hear about, in this series of workshops. For this workshop, we'll study the creation, population, and modification of tables using a database we build from scratch.


Blank Database
Create a Microsoft Office Access database that does not contain any existing data or objects.
File Name:
Simple Faculty Database.accdb
J:\Training\Access\Databases\
Create
Cancel

Creating a Blank Database

Creating a blank database is very simple. Click on the "Blank Database" icon on the opening screen for Access. In the pane that opens on the right side of the Access window, give the database a name and indicate where you want to store it (by clicking on the folder icon to the right of the name and navigating to your chosen location). Then click on the Create button.

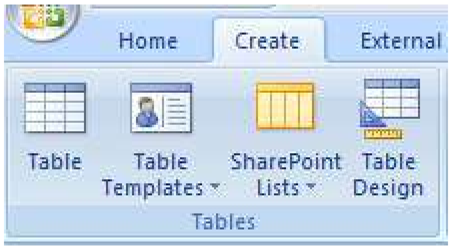
The database will open with a single object, a table named Table1, as illustrated below.



From the design work that you've done off-line, you should have a good idea of the tables that you need to create, the fields to be stored in each table, and the relationships between the tables. Now you'll implement that design using the tools provided by Access. For your first table, you can rename and modify Table1 using Design View, or you can discard Table1 and use either of the other methods of creating tables described below.

Creating Tables

Access provides at least four methods for creating tables. We'll look at the three that you can use without having a SharePoint server available to you. All are initiated from the Create ribbon, a portion of which is shown to the right.



Data Entry

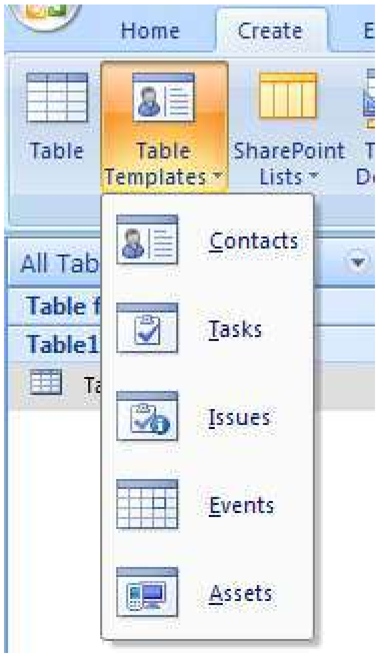
You can create a table in Access simply by entering data into a blank table by typing it in or by copying and pasting from some other source, such as an Excel spreadsheet. Position yourself to do so by clicking on the "Table" icon at the left end of the Create ribbon. Your initial view of the table will look just like the new blank database shown above.

Start in the shaded box under "Add New Field" and type the value of the first field in the first record of your table.

- Press the Tab key to move to a new field in the same record.
- Press Enter to move to the next record.

When you first close the table, Access will ask you to name it and will establish other table properties based on the values present there. Field names will be Field1, Field2, and so forth. Once created, the table can be modified in design view.

Table Templates



The table templates menu on the Create ribbon offers a small collection of pre-defined tables, shown to the right. If there is a table in the list that comes close to meeting your needs, you can select it to create the table and then modify the table in Design View so that it matches your design precisely.

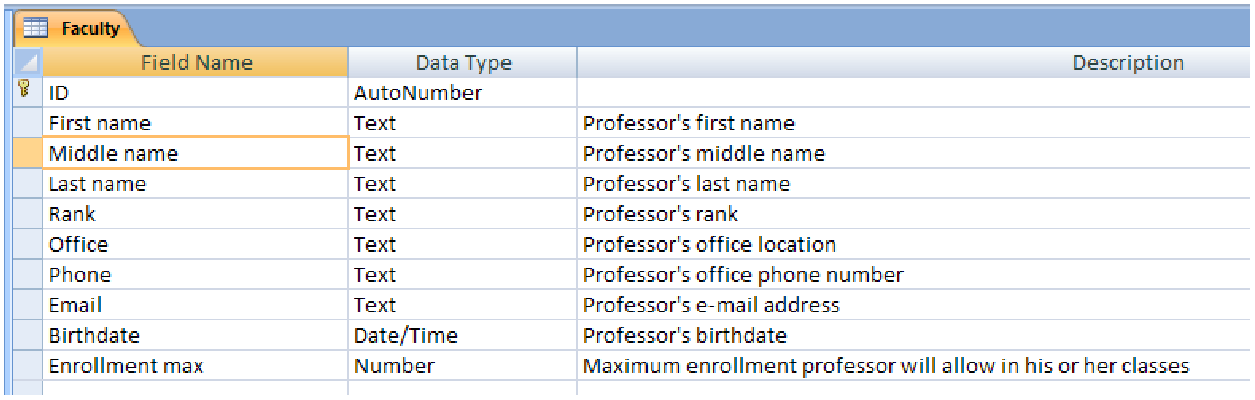
Table Design

If there is one table creation method with which you must be familiar, it is Table Design, because it provides the means by which you can adjust a table's properties, however the table was originally created. When you click on the Table Design icon in the Create ribbon, you are presented with a worksheet for specifying the fields in the new table. At a minimum, you must supply the name of each field and select the type of data it will hold. You can also provide a description of the field and specify other properties such as input mask and validation.

Table Design View

You can open the Design View of any table by selecting the table and then clicking on the Design View icon at the left end of the Home ribbon. In Design View, you can add or remove fields and change the properties of fields.

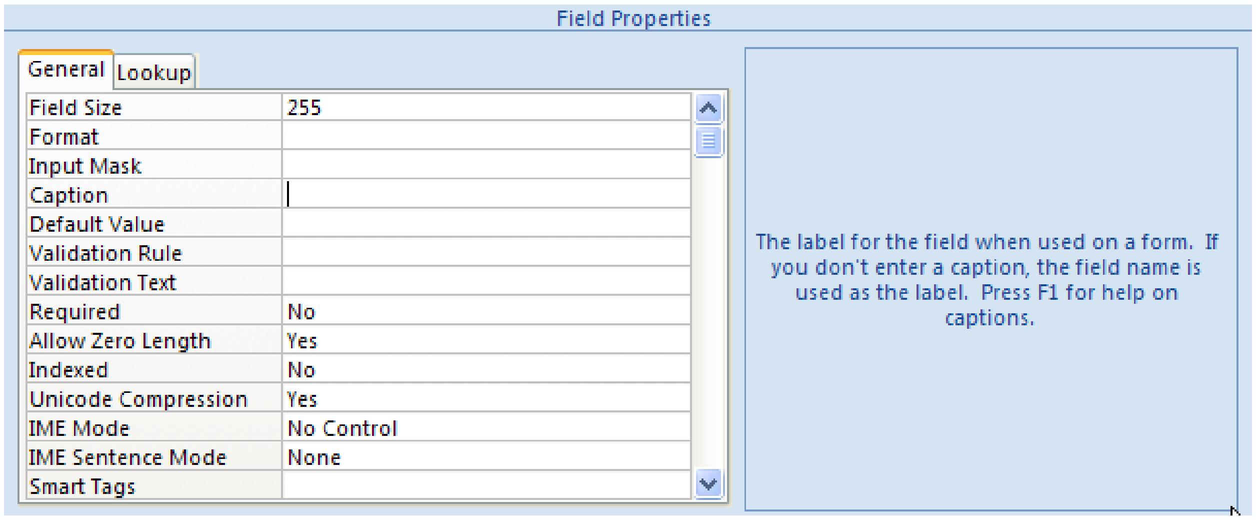
The main part of the Design View pane lists the names, types, and descriptions (if any) of the fields in the table, as illustrated below.



Faculty

	Field Name	Data Type	Description
🔑	ID	AutoNumber	
	First name	Text	Professor's first name
	Middle name	Text	Professor's middle name
	Last name	Text	Professor's last name
	Rank	Text	Professor's rank
	Office	Text	Professor's office location
	Phone	Text	Professor's office phone number
	Email	Text	Professor's e-mail address
	Birthdate	Date/Time	Professor's birthdate
	Enrollment max	Number	Maximum enrollment professor will allow in his or her classes

When a field is selected, as Middle Name is in the illustration above, the Property Sheet for that field will be displayed at the bottom of the Design View pane.



Field Properties

General | Lookup

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

The label for the field when used on a form. If you don't enter a caption, the field name is used as the label. Press F1 for help on captions.

You can adjust the properties to meet the needs of your database. When you select one of the properties, a brief description of it is displayed in the blue area to the right. Further information is available by pressing the F1 key to access context-sensitive help.

In this example, you might change the Field Size of 255 characters, since that is much larger than anyone's middle name.

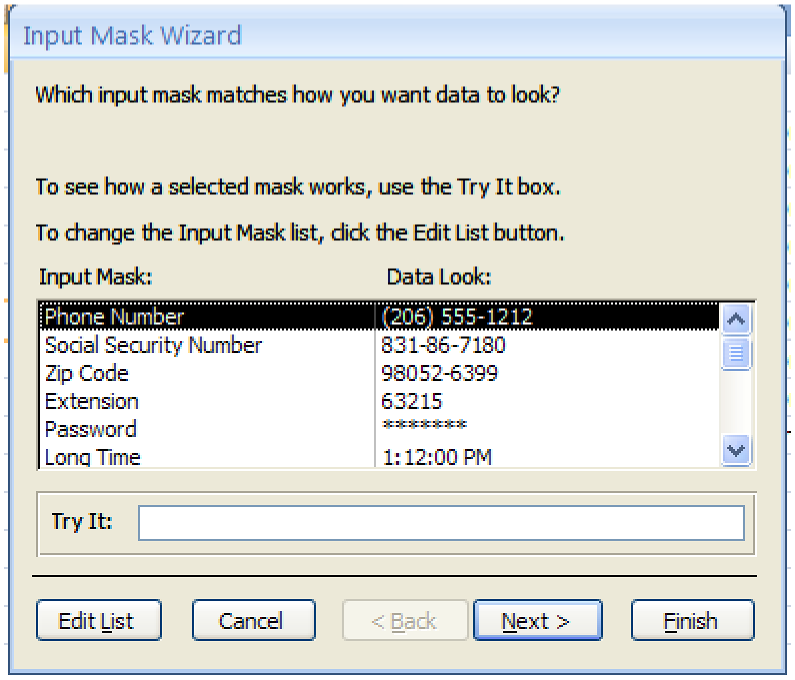
Some of a field's properties can be used to reduce if not eliminate data entry errors by imposing restrictions on what values are permitted in the field. An input mask, validation rule, and other specifications can each help in some circumstances.

Input Mask

You can use an input mask to ensure that commonly used data like phone numbers, dates, and zip codes are properly entered into the database. Here's how:

1. Select the field for which you want to set an input mask by clicking on it in the list of fields.
2. Click in the Input Mask line of the field's properties. An icon containing an ellipsis (…) will appear at the right end of the line.
3. Click on the ellipsis icon to open the Input Mask Wizard.

4. Select one of the standard masks from the list or click on Edit List to create your own mask.
5. Proceed through the remaining steps of the wizard, responding to each question that is posed and clicking on the Next button to move to the next step.
6. When you finish with the wizard, code that represents the choices you made in the wizard will be put in the Input Mask line of the properties sheet. That code will control the appearance of the data entered into the field. In the case of the phone number example used here, it will prevent non-digits or the wrong number of digits from being put into the phone number field.



Input Mask Wizard

Which input mask matches how you want data to look?

To see how a selected mask works, use the Try It box.

To change the Input Mask list, click the Edit List button.

Input Mask:	Data Look:
Phone Number	(206) 555-1212
Social Security Number	831-86-7180
Zip Code	98052-6399
Extension	63215
Password	********
Long Time	1:12:00 PM

Validation Rule and Text

A validation rule for a field can specify conditions that the field's value must satisfy to be accepted into the database. Typically, the conditions are upper and/or lower limits on a numerical value or date. The validation text, if provided, will be used in an error message displayed for database users who attempt to enter an invalid value, so it should help them understand what is required in the field.

To set a validation rule and text:

1. Select the field for which you want to set a validation rule by clicking on it in the list of fields.
2. Click in the Validation Rule line of the field's properties. An icon containing an ellipsis (…) will appear at the right end of the line.
3. Click on the ellipsis icon to open the Expression Builder, or, if you wish, simply type the expression for the rule directly into the properties sheet.
4. Click in the Validation Text line of the field's properties. Type the message that should be displayed if a database user enters an invalid value.

For example, in a database containing information about current employees, you might feel that no birth date should be earlier than 1908. In that case, type "> 1/1/1908" as the validation rule and something like "Birth date can be no earlier than 1908" for the validation text. You may find that Access will adjust the syntax of your validation rule. In this instance, for example, it will be modified to read ">#1/1/1908#".

Note: If you add a validation rule to a table that already has data in it, be sure to click on the "Test Validation Rules" icon in the Design ribbon to determine if any existing data already violates the rule.

Required Data

Some field values may be so critical that you wish to require them in any record of your table. For a table containing data about people, you might insist on having a last name for each person in the database (unless, I suppose, your database includes Cher or Sting). To require a value for a field:

1. Select the field whose value is to be required by clicking on it in the list of fields.
2. Click in the Required line of the field's properties. An icon indicating a drop-down menu will appear at the right end of the line.
3. Click on the drop-down menu icon and select Yes from the list.

Note: If you specify a required value in a table that already has data in it, be sure to click on the "Test Validation Rules" icon in the Design ribbon to determine if any existing data already violates the requirement.

Primary Key

Access insists that each table have a primary key. If you try to save a new table without having specified a primary key, Access will offer to create a key for you. There are, thus, two ways to designate the primary key for a table:

1. If there is no field in the table that can serve as a primary key, you can accept the offer that Access makes when you save a table lacking a key. In that case, a new field will be inserted into the table. Typically, the name of the new field will be ID and its data type will be AutoNumber. As you enter data into a table with such a primary key, Access will supply values in the ID field automatically, beginning with 1 and increasing sequentially.
2. If there is a field that could be a key—such as a WVUID field in a table of WVU employees or students—then you can manually specify it as such. Simply select the field in Design View and click on the Primary Key icon in the Design ribbon.

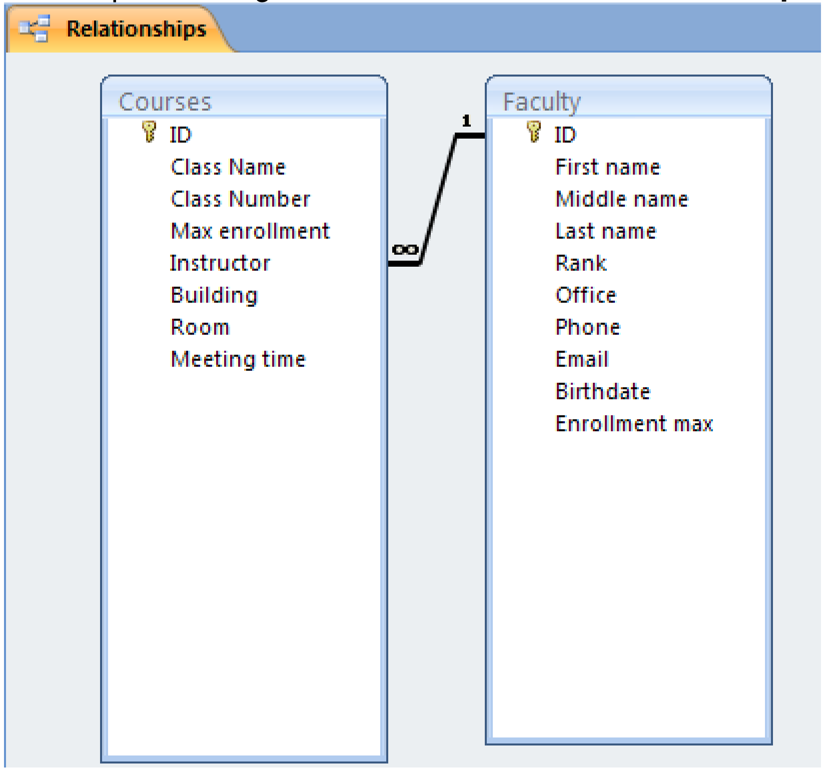
Creating Relationships

The general procedure for creating a relationship in Microsoft Access is to:

- Determine the type of the relationship and identify the source and target tables.
- If necessary, create a linking or cross reference table to serve as the common target thereby resolve a many-to-many relationship.
- Create the foreign key field(s) in the target table if they are not already present.
- Select the **Database Tools** ribbon.
- Click on the **Relationships** icon in the ribbon.
- Add all of the tables involved in the relationship to the window.
- Create the relationship(s) by dragging the primary key(s) from the source(s) and dropping them on the associated foreign key(s) in the target(s).

Relationship Pane

To open the relationship window go to **Database Tools > Relationships**.

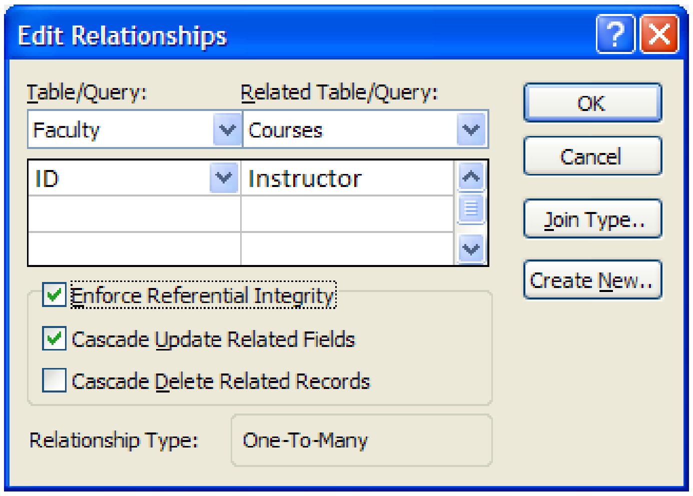


If your database does not have any relationships defined, the **Show Table** dialog box will automatically be displayed. Use it to add the tables you want to relate.

Once you have selected all the tables you want to relate, define a relationship between two tables by **dragging** the field that you want to relate from one table to the related field in the other table.

In most cases, you drag the **primary key** field (which is displayed with a key icon next to it) from one table to a similar field (often with the same name) called the **foreign key** in the other table. The related fields are not required to have the same names (though it's good practice to make them the same since it reminds you where the relationship comes from), but they must have the same data type and contain the same kind of information. In addition, when the matching fields are **Number** fields, they must have the same Field Size property setting.

Once you have created the relationships, the **Edit Relationships** dialog box is displayed. Check the field names displayed in the two columns to ensure they are correct. You can change them if necessary.



Table/Query:	Related Table/Query:
Faculty	Courses
ID	Instructor

[✓] Enforce Referential Integrity
[✓] Cascade Update Related Fields
[ ] Cascade Delete Related Records

Relationship Type: One-To-Many

Referential Integrity

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. Under the rules of referential integrity, you cannot enter data in a foreign key of a target table unless the same data already exists in a primary key in the source table. When you complete the drag-and-drop operations described above, Access provides a dialog box that allows you to set referential integrity.

This means that, if referential integrity is set on the relationship between the teacher table and the class table and a teacher is assigned to teach a course, Access will not allow you to delete that teacher's record in the teacher table. It will also not allow you to assign a teacher to a class if they are not in the teacher table.

If you choose to set referential integrity, then two other options are available for your use:

- Cascading Updates: If you make a change to the primary key in the source table, then that change is cascaded to all related foreign keys in the target table.
- Cascading deletes: If you delete a primary key in the source table then all related records in the target table are also deleted.

Activity 1: Designing an Access table

Open a new blank database in Access. Use Design View to create two related tables either for a database of your own devising or for the data described below. Use the relationship window to establish a relationship between the two tables. What referential integrity settings make sense for your database?

Feel free to devise your own tables or create these two tables using appropriate data types for each field:

Student table: Student_ID, First_name, Last_name, Birthdate, Financial_aid, Dorm_room

Room table: Room_ID, Building, Room_number, Capacity

Populating the Database

Once you have designed the tables for your database and established relationships between them, there are several ways to put data into the tables: typing the data, perhaps with the aid of forms, lookup lists, or other methods of ensuring the accuracy of the data entered; importing data from an existing repository such as a spreadsheet or another database; and linking to an existing file containing data. Some or all of these methods might be used to build a single database.

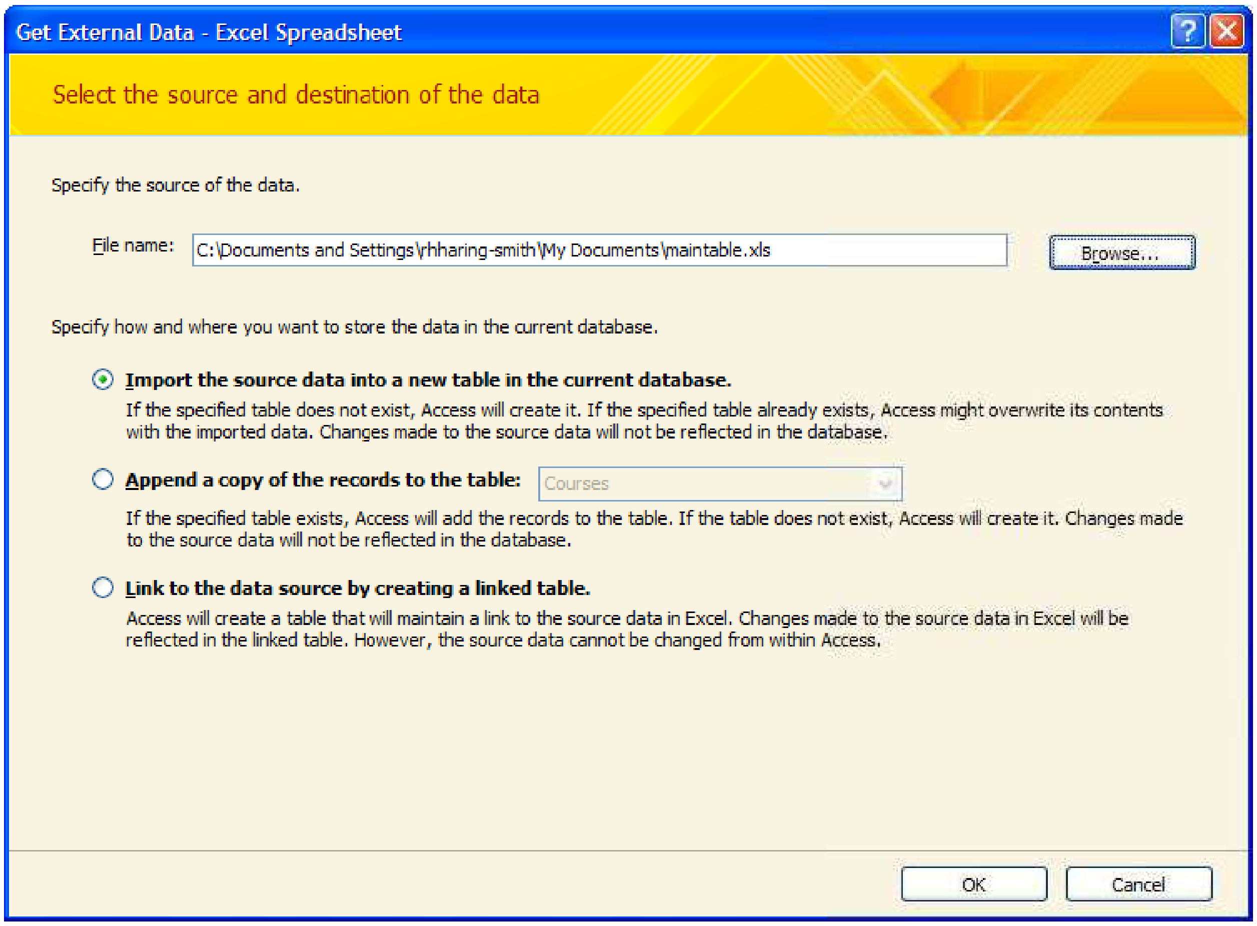
Forms are the topic of a future workshop in this series of Access workshops, but we will touch on the other methods now.

The **External Data** ribbon is the principal locale for tools allowing you to exchange data with other files, whether you are importing or exporting data.

Importing data from an Excel spreadsheet

Open the database.

1. Select the **External Data** ribbon, and click on the **Excel** tool in the Import section of the ribbon.


Get External Data - Excel Spreadsheet
Select the source and destination of the data

Specify the source of the data.

File name: C:\Documents and Settings\rhharing-smith\My Documents\maintable.xls

Specify how and where you want to store the data in the current database.

[✓ **Import the source data into a new table in the current database.**]
If the specified table does not exist, Access will create it. If the specified table already exists, Access might overwrite its contents with the imported data. Changes made to the source data will not be reflected in the database.

Append a copy of the records to the table: Courses
If the specified table exists, Access will add the records to the table. If the table does not exist, Access will create it. Changes made to the source data will not be reflected in the database.

Link to the data source by creating a linked table.
Access will create a table that will maintain a link to the source data in Excel. Changes made to the source data in Excel will be reflected in the linked table. However, the source data cannot be changed from within Access.

2. In the **Get External Data** window, use the Browse button to locate the file to be imported.
3. Choose one of the three options for incorporating the spreadsheet data into the database: (a) importing into a new table, (b) appending to an existing table, or (c) linking to the spreadsheet.
4. Follow the directions in the Import Spreadsheet Wizard in order to select which sheet of the Excel file is to be used, indicate whether it has a header row, specify the type of data in each column to be imported, determine how the primary key will be specified, and name the table in which the data is to be put. If you are importing data into an existing table, then you must be sure that the first row of the sheet contains all of the field names for that table so the import wizard can determine where each value should be stored.

Notes on Importing

- You can import or link all the data from a spreadsheet, or just the data from a named range of cells.
- You can create a **new** table in Access based on the data.
- You can append the data to an existing table as long as your spreadsheet column headings match the table's field names.
  - If you attempt to append a table which contains a primary key field and your data for that field contains null or duplicate values you will experience an error.
- Access attempts to assign the appropriate data type to imported fields, but you should check your fields to make sure that they are set to the data type you want.
  - For example, in an Access database, a phone number or postal code field might be imported as a Number field, but should be changed to a Text field in Microsoft Access because it is unlikely that you will perform any calculations on these types of fields. In the case of zip codes, leading zeroes will be lost if the zip code data is brought into a Number field, but not if it is brought into a Text field. You should also check and set field properties, such as formatting, as necessary.

Notes on Linking

- If you link to a file and that file is deleted or moved from its original location (e.g., it is placed in a different folder or drive) the link to the file will be severed. You then have to use the Linked Table Manager to re-establish the link if the file has not been deleted.
- If you link to a file on a local area network, it is best to use a universal naming convention (**UNC)** path, if you know it, rather than relying on the drive letter of a mapped network drive in Microsoft Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for Microsoft Access to locate the data source that contains the linked table.
- Universal naming convention for files that provides a machine-independent means of specifying the file's location. Rather than using a drive letter and path, a UNC name uses the syntax \\server\share\path\filename.

Should you Import or Link a foreign file?

In making that decision, consider that Access works faster in its own file format and that you can customize that format to meet your needs. This suggests that importing is the way to go. But consider, too, the role of the data in the external file. Do you want to capture that data as it is now or do you want your database to reflect future changes in the external file as they are made? In the former case, import the data; in the latter case, link to it.

In general:

- Research the procedure.
- Make backup copies of all files.
- Test the import or link procedure.
- Test the foreign files in Access thoroughly before discarding backups.

Before you import or link to external data, it is important to check that the data is arranged in an appropriate tabular format, and has the same type of data in each field (column) and the same fields in every row.

Activity 2: Importing an Access table

Use the Access import tool on the External Data ribbon to import a new table into your database from the Faculty database in the OIT_Workshops folder.

Activity 3: Link to a table in another database

Use the Excel import tool on the External Data ribbon to link to a worksheet in one of the files in the Excel subfolder of the OIT_Workshops folder, as specified by your instructor.