# Access Tutorial 1: Introduction to Microsoft Access

The purpose of these tutorials is not to teach you Microsoft Access, but rather to teach you some generic information systems concepts and skills using Access. Of course, as a side effect, you will learn a great deal about the software—enough to write your own useful applications. However, keep in mind that Access is an enormously complex, nearly-industrial-strength software development environment. The material here only scrapes the surface of Access development and database programming.

## 1.1 Introduction: What is Access?

Microsoft Access is a relational database management system (DBMS). At the most basic level, a DBMS is a program that facilitates the storage and retrieval of structured information on a computer's hard drive. Examples of well-know industrial-strength relational DBMSes include

- Oracle
- Microsoft SQL Server
- IBM DB2
- Informix

Well-know PC-based ("desktop") relational DBMSes include

- Microsoft Access
- Microsoft FoxPro
- Borland dBase

### 1.1.1 The many faces of Access

Microsoft generally likes to incorporate as many features as possible into its products. For example, the Access package contains the following elements:

- a **relational database system** that supports two industry standard query languages: Structured Query Language (SQL) and Query By Example (QBE);

- a full-featured **procedural programming language**—essentially a subset of Visual Basic,
- a simplified procedural **macro language** unique to Access;
- a **rapid application development environment** complete with visual form and report development tools;
- a sprinkling of **objected-oriented extensions**; and,
- various **wizards and builders** to make development easier.

For new users, these "multiple personalities" can be a source of enormous frustration. The problem is that each personality is based on a different set of assumptions and a different view of computing. For instance,

- the relational database personality expects you to view your application as sets of data;
- the procedural programming personality expects you to view your application as commands to be executed sequentially;
- the object-oriented personality expects you to view your application as objects which encapsulate state and behavior information.

Microsoft makes no effort to provide an overall logical integration of these personalities (indeed, it is unlikely that such an integration is possible). Instead, it is up to you as a developer to pick and choose the best approach to implementing your application.

Since there are often several vastly different ways to implement a particular feature in Access, recognizing the different personalities and exploiting the best features (and avoiding the pitfalls) of each are important skills for Access developers.

The advantage of these multiple personalities is that it is possible to use Access to learn about an enormous range of information systems concepts without having to interact with a large number of “single-personality” tools, for example:

- Oracle for relational databases
- PowerBuilder for rapid applications development,
- SmallTalk for object-oriented programming.

Keep this advantage in mind as we switch back and forth between personalities and different computing paradigms.

### 1.1.2 What is in an Access database file?

Although the term “database” typically refers to a collection of related data tables, an Access database includes more than just data. In addition to tables, an Access database file contains several different types of **database objects**:

- saved queries for organizing data,
- forms for interacting with the data on screen,
- reports for printing results,
- macros and Visual Basic programs for extending the functionality of database applications.

All these database objects are stored in a single file named `<filename>.mdb`. When you are running Access, a temporary “locking” file named `<filename>.ldb` is also created. You can safely ignore the *.ldb file; everything of value is in the *.mdb file.

## 1.2 Learning objectives

❒ How do I get started?

❒ How do I determine the version I am using?

❒ How do I create or edit a database object?

❒ What is the database window and what does it contain?

❒ How do I import an Excel spreadsheet?

❒ How do I delete or rename database objects?

❐ How do I get help from the on-line help system?

❐ How do I compact a database to save space?

## 1.3 Tutorial exercises

In this tutorial, you will start by creating a new database file.

### 1.3.1 Starting Access

- To start Access, you double click the Access icon ( for version 8.0 and 7.0 or for version 2.0) from within Microsoft Windows.

If you are working in the Commerce PC Lab, you will be working with Access version 2.0. If you are working at home, you will able be to tell what version you are using by watching the screen “splash” as the program loads. Alternatively, select *Help > About Access* from the main menu to see which version you are using.

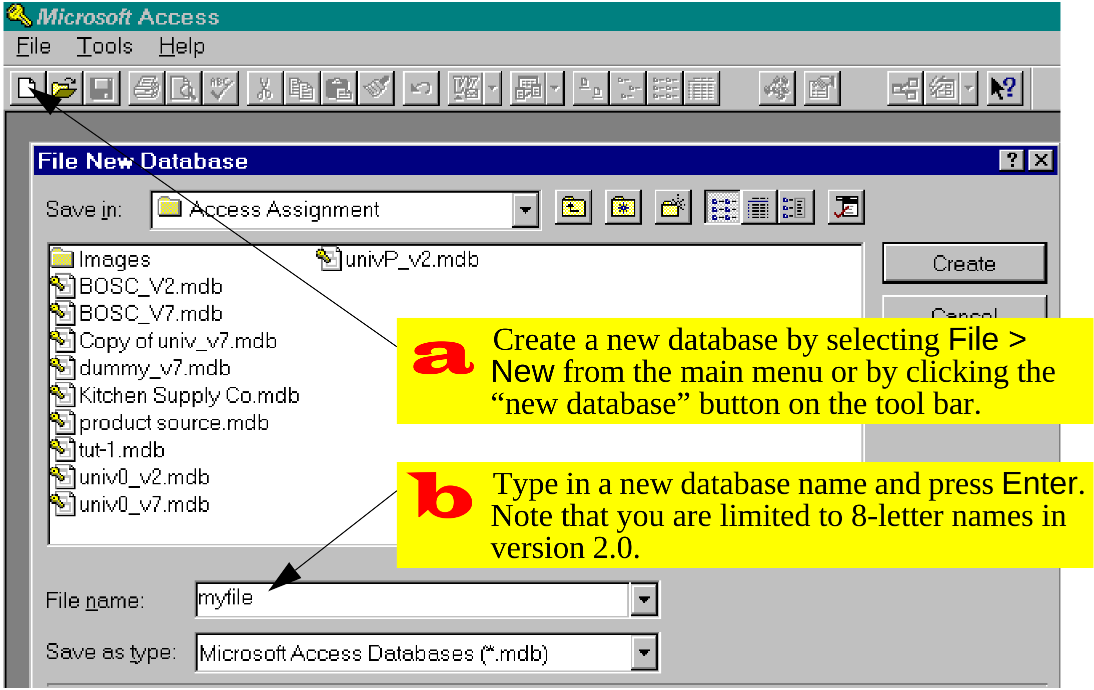
All the screen shots in these tutorials are taken from Access version 7.0 (released as part of Office 95). Although there are some important differences between version 2.0 and version 7.0, the concepts covered here are the same for both. Version 8.0 (released as part of Office 97) is only slightly different from version 7.0.

Whenever the instructions given in the tutorial differ significantly from version 7.0, a warning box such as this is used.

### 1.3.2 Creating a new database

- Follow the directions in Figure 1.1 to create a new database file called `myfile.mdb`.

**FIGURE 1.1: Select the name and location of your new (empty) database.**



Save in: Access Assignment

File name: myfile

Save as type: Microsoft Access Databases (*.mdb)

**a** Create a new database by selecting File > New from the main menu or by clicking the “new database” button on the tool bar.

**b** Type in a new database name and press Enter. Note that you are limited to 8-letter names in version 2.0.

- Examine the main features of the database window—including the tabs for viewing the different database objects—as shown in Figure 1.2.

### 1.3.3 Opening an existing database

Since an empty database file is not particularly interesting, you are provided with an existing database file containing information about university courses. For the remainder of this tutorial, we will use a file called `univ0_v7.mdb`, which is available from the tutorial's Internet site.

If you are using version 2.0, you will need to use the `univ0_v2.mdb` database instead. Although you can open a version 2.0 database with version 7.0, you cannot open a version 7.0 database with version 2.0. Importing and exporting across versions is possible, however.

If you are using version 8.0, you can use either `univ0_v2.mdb` or `univ0_v7.mdb` for the tutorials. When you open the file, Access will ask you if you want to convert it to version 8.0. Select yes and provide a new name for the converted file (e.g., `univ0_v8.mdb`)

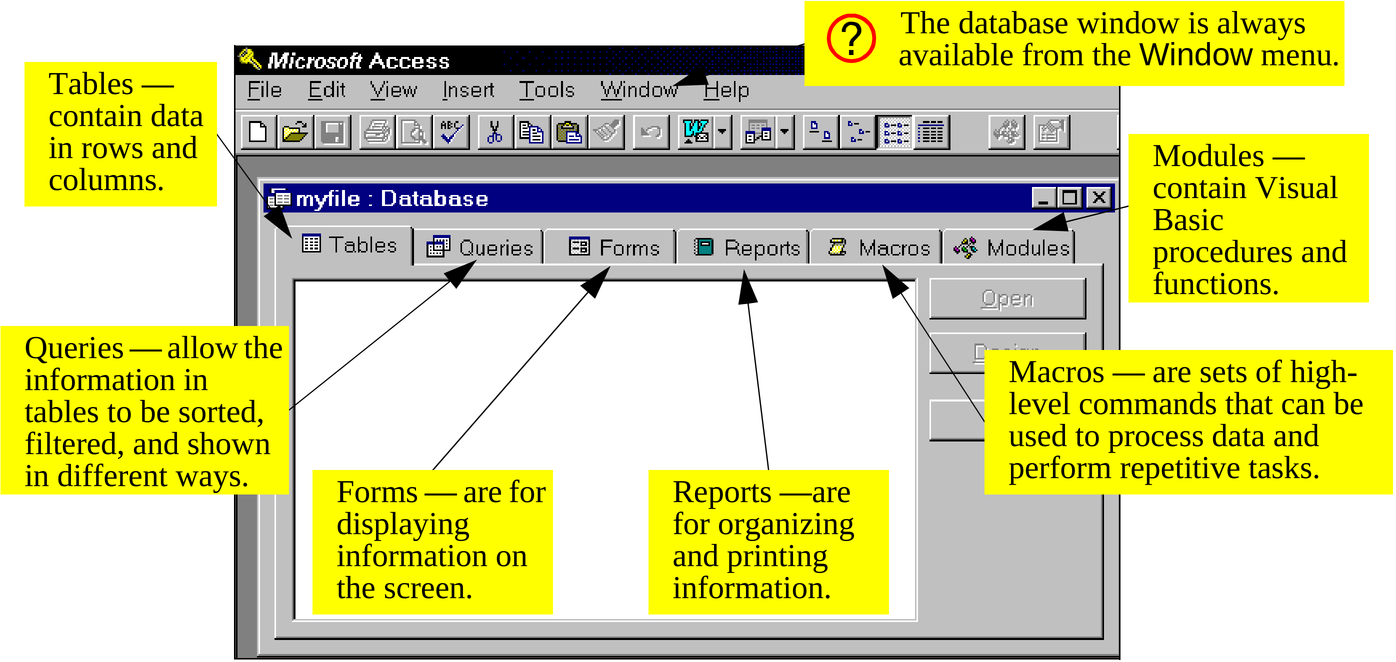
- Open the `univ0_vx.mdb` file and examine the contents of the `Sections` table, as shown in Figure 1.3.

### 1.3.4 Importing data from other applications

Access makes it easy to import data from other applications. In this section, you will create a new table using data from an Excel spreadsheet.

- Select *File > Get External Data > Import* from the main menu and import the `depts.xls` spread-

**FIGURE 1.2: The *database window* contains all the *database objects* for a particular application.**



The database window is always available from the Window menu.

Tables — contain data in rows and columns.

Modules — contain Visual Basic procedures and functions.

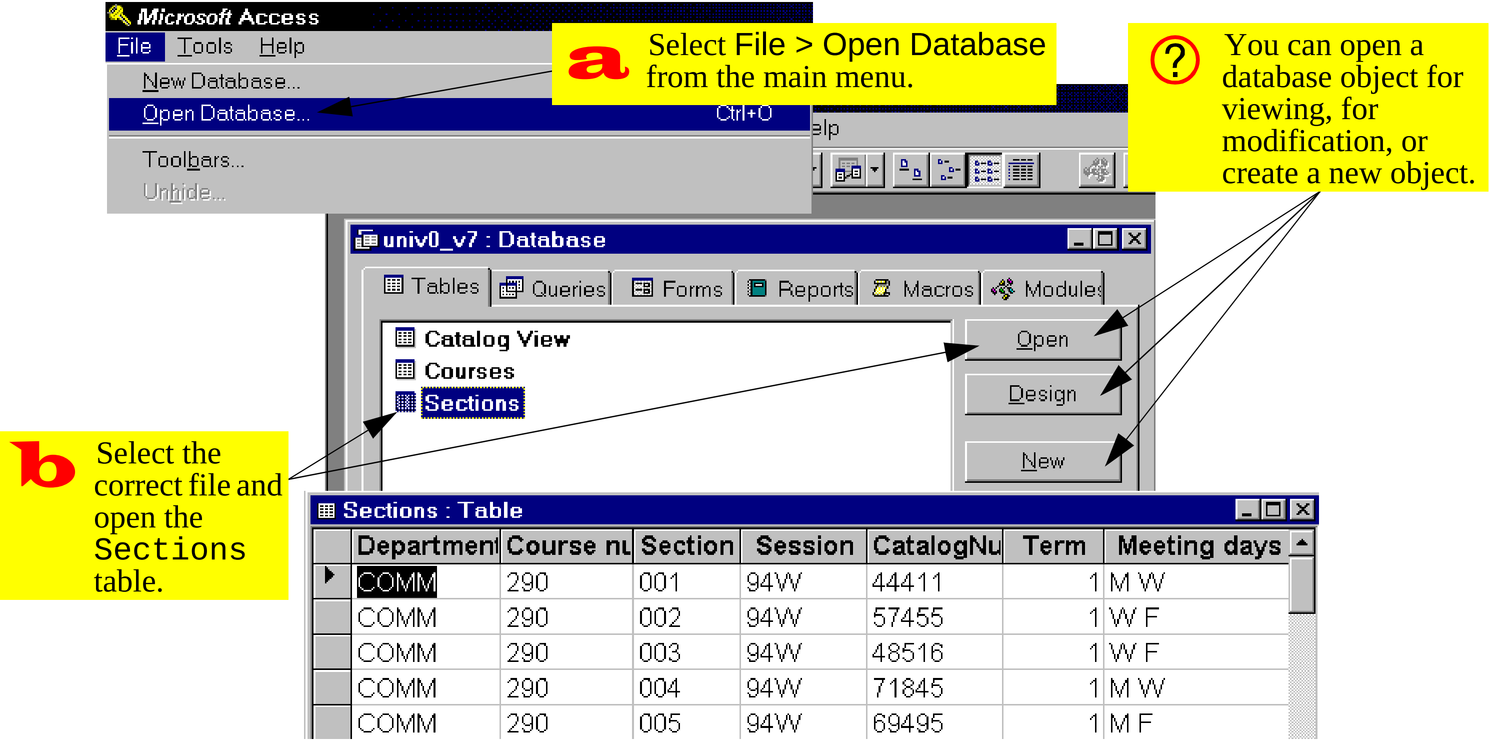
Queries — allow the information in tables to be sorted, filtered, and shown in different ways.

Macros — are sets of high-level commands that can be used to process data and perform repetitive tasks.

Forms — are for displaying information on the screen.

Reports —are for organizing and printing information.

**FIGURE 1.3:** **Open the `univ0_vx.mdb` file for the version of Access that you are using and then open the `Sections` table**



**a** Select File > Open Database from the main menu.

**?** You can open a database object for viewing, for modification, or create a new object.

**b** Select the correct file and open the `Sections` table.

| | Department | Course nu | Section | Session | CatalogNu | Term | Meeting days |
|---|---|---|---|---|---|---|---|
| ▶ | COMM | 290 | 001 | 94W | 44411 | 1 | M W |
| | COMM | 290 | 002 | 94W | 57455 | 1 | W F |
| | COMM | 290 | 003 | 94W | 48516 | 1 | W F |
| | COMM | 290 | 004 | 94W | 71845 | 1 | M W |
| | COMM | 290 | 005 | 94W | 69495 | 1 | M F |

sheet as a new table called `Departments` (see Figure 1.4).

In version 2.0, the menu structure is slightly different. As such, you must use *File > Import*.

- Use the import wizard specify the basic import parameters. You should accept all the defaults provided by the wizard except for those shown in Figure 1.5.
- Double click the `Departments` table to ensure it was imported correctly.

If you make a mistake, you can rename or delete a table (or any database object in the database window) by selecting it and right-clicking (pressing the right mouse button once).

### 1.3.5 Getting help

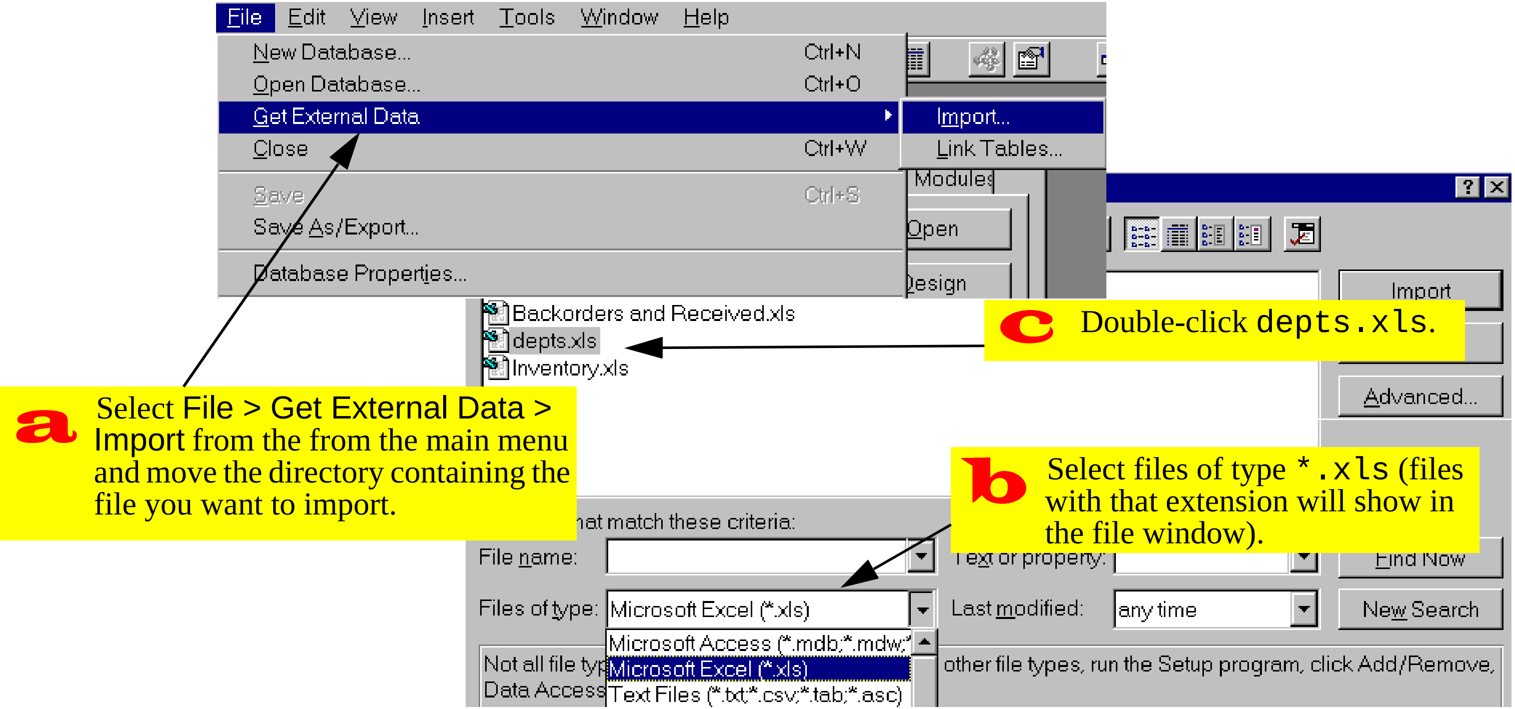
A recent trend in commercial software (especially from Microsoft) is a reliance on on-line help and documentation in lieu of printed manuals. As a consequence, a good understanding of how to use the on-line help system is essential for learning any new software. In this section, you will use Access' on-line help system to tell you how to compact a database.

- Press *F1* to invoke the on-line help system. Find information on compacting a database, as shown in Figure 1.6.
- Familiarize yourself with the basic elements of the help window as shown in Figure 1.7.

### 1.3.6 Compacting your database

- Follow the directions provided by the on-line help window shown in Figure 1.7 to compact your database.

**FIGURE 1.4: Import the `dept.xls` spreadsheet as a table called `Departments`.**



hat match these criteria:

File name:

Files of type: Microsoft Excel (*.xls)

Microsoft Access (*.mdb;*.mdw;*
Microsoft Excel (*.xls)
Text Files (*.txt;*.csv;*.tab;*.asc)

Text or property:

Last modified: any time

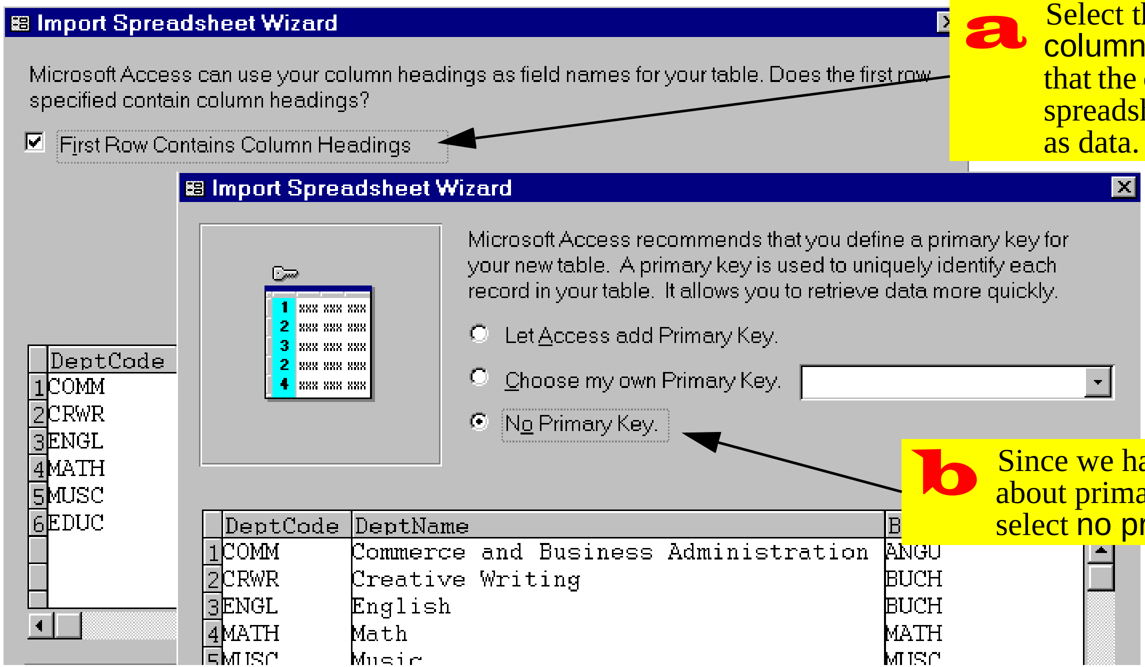
Not all file typ ... other file types, run the Setup program, click Add/Remove,
Data Access

**a** Select File > Get External Data > Import from the from the main menu and move the directory containing the file you want to import.

**b** Select files of type `*.xls` (files with that extension will show in the file window).

**c** Double-click `depts.xls`.

**FIGURE 1.5:** Use the spreadsheet import wizard to import the Excel file.



Import Spreadsheet Wizard

Microsoft Access can use your column headings as field names for your table. Does the first row specified contain column headings?

☑ First Row Contains Column Headings

| | DeptCode |
|---|---|
| 1 | COMM |
| 2 | CRWR |
| 3 | ENGL |
| 4 | MATH |
| 5 | MUSC |
| 6 | EDUC |

Import Spreadsheet Wizard

Microsoft Access recommends that you define a primary key for your new table. A primary key is used to uniquely identify each record in your table. It allows you to retrieve data more quickly.

○ Let Access add Primary Key.

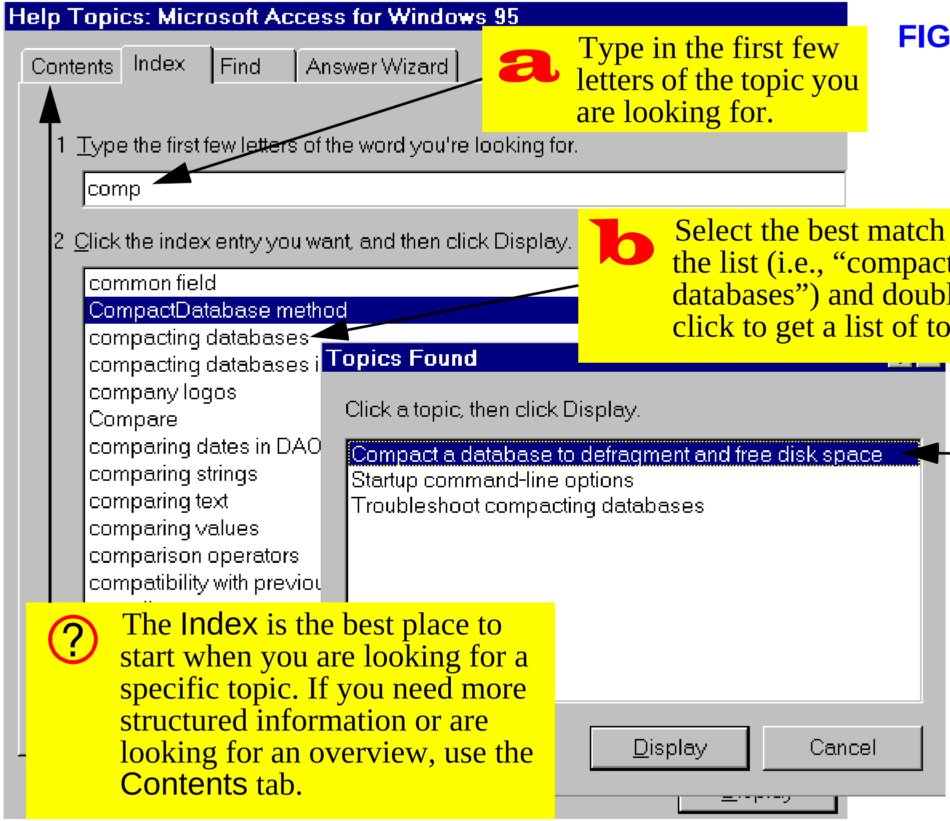
○ Choose my own Primary Key.

● No Primary Key.

| | DeptCode | DeptName | B |
|---|---|---|---|
| 1 | COMM | Commerce and Business Administration | ANGU |
| 2 | CRWR | Creative Writing | BUCH |
| 3 | ENGL | English | BUCH |
| 4 | MATH | Math | MATH |
| 5 | MUSC | Music | MUSC |

**a** Select the first row contains column headings option so that the column headings in the spreadsheet are not interpreted as data.

**b** Since we have not talked about primary keys yet, select no primary key.



Help Topics: Microsoft Access for Windows 95

1 Type the first few letters of the word you're looking for.

comp

2 Click the index entry you want, and then click Display.

common field
CompactDatabase method
compacting databases
compacting databases i
company logos
Compare
comparing dates in DAO
comparing strings
comparing text
comparing values
comparison operators
compatibility with previou

Topics Found

Click a topic, then click Display.

Compact a database to defragment and free disk space
Startup command-line options
Troubleshoot compacting databases

**a** Type in the first few letters of the topic you are looking for.

**b** Select the best match from the list (i.e., “compacting databases”) and double-click to get a list of topics.

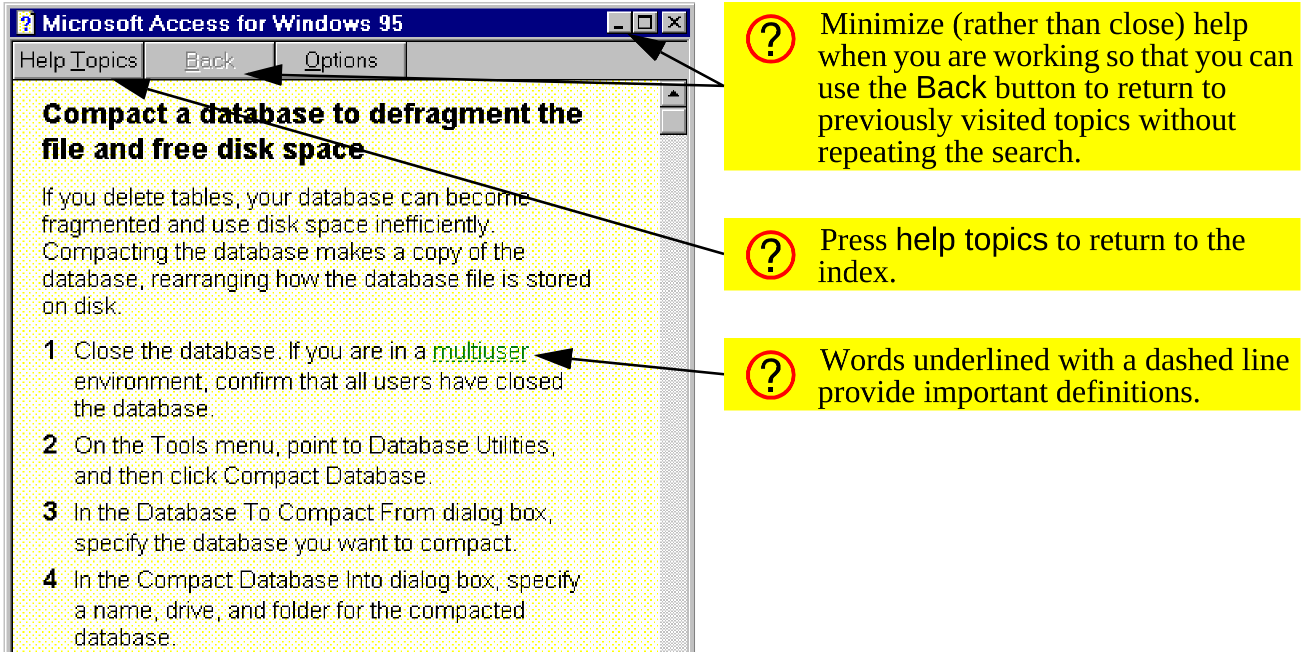
**c** Double click the most promising entry in this list to get the actual help topic.

? The Index is the best place to start when you are looking for a specific topic. If you need more structured information or are looking for an overview, use the Contents tab.

**FIGURE 1.6: Use the help system to find information on a specific topic**

? For most students, the help system in Access version 2.0 is easier to navigate. Use the “cue cards” in version 2.0 to get step-by-step instructions for many operations.

**FIGURE 1.7: Follow the instructions provided by help to compact your database**



**Compact a database to defragment the file and free disk space**

If you delete tables, your database can become fragmented and use disk space inefficiently. Compacting the database makes a copy of the database, rearranging how the database file is stored on disk.

1 Close the database. If you are in a multiuser environment, confirm that all users have closed the database.

2 On the Tools menu, point to Database Utilities, and then click Compact Database.

3 In the Database To Compact From dialog box, specify the database you want to compact.

4 In the Compact Database Into dialog box, specify a name, drive, and folder for the compacted database.

? Minimize (rather than close) help when you are working so that you can use the Back button to return to previously visited topics without repeating the search.

? Press help topics to return to the index.

? Words underlined with a dashed line provide important definitions.

# 1.4 Discussion

## 1.4.1 The database file in Access

The term “database” means different things depending on the DBMS used. For example in dBase IV, a database is a file (`<filename>.dbf`) containing a single table. Forms and reports are also stored as individual files with different extensions. The net result is a clutter of files.

In contrast, an Oracle database has virtually no relationship to individual files or individual projects. For instance, a database may contain many tables from different projects/applications and may also be stored split into one or more files (perhaps on different machines).

Access strikes a convenient balance—all the “objects” (tables, queries, forms, reports, etc.) for a single project/application are stored in a single file.

## 1.4.2 Compacting a database

As the help system points out, Access database files can become highly fragmented and grow to become much larger than you might expect given the amount of data they contain (e.g., multiple megabytes for a handful of records). Compacting the database from time to time eliminates fragmentation and can dramatically reduce the disk space requirement of your database.

## 1.4.3 Renaming a database

It is often the case that you are working with a database and want to save it under a different name or save it on to a different disk drive. However, one command on the *File* menu that is conspicuous by its absence is *Save As*.

However, when compacting your database, Access asks for the name and destination of the compacted file. As a result, the compact database utility can be used as a substitute for the *Save As* command. This is especially useful in situations in which you cannot use the operating system to rename a file (e.g., when you do not have access to the Windows file manager).

### 1.4.4 Developing applications in Access

In general, there are two basic approaches to developing information systems:

- in-depth systems analysis, design, and implementation,
- rapid prototyping (in which analysis, design, and implementation are done iteratively)

Access provides a number of features (such as graphical design tools, wizards, and a high-level macro language) that facilitate rapid prototyping. Since you are going to build a small system and since time is limited, you will use a rapid prototyping approach to build your application. The recommended sequence for prototyping using Access is the following:

1. Model the information of interest in terms of entities and relationships between the entities (this is covered in the lecture portion of the course).
2. Create a table for each entity (Tutorial 2).
3. Specify the relationships between the tables (Tutorial 3).
4. Organize the information in your tables using queries (Tutorial 4, Tutorial 5, Tutorial 10)
5. Create forms and reports to support input and output transactions (Tutorial 6, Tutorial 7).
6. Enhance you forms with input controls (Tutorial 8)
7. Create action queries (Tutorial 11), macros (Tutorial 13), or Visual Basic programs (Tutorial 12, Tutorial 14) to perform the transaction processing functions of the application.

8. Create “triggers” (procedures attached to events) to automate certain repetitive tasks (Tutorial 15).

### 1.4.5 Use of linked tables

Most professional Access developers do not put their tables in the same database file as their queries, forms, reports, and so on. The reason for this is simple: keep the application’s data and interface separate.

Access allows you to use the “linked table” feature to link two database files: one containing all the tables (“data”) and another containing all the interface and logic elements of the application (“interface”). The linked tables from the data file show up in the interface file with little arrows (indicating that they are not actually stored in the interface file).

In this way, you can modify or update the interface file without affecting the actual data in any way. You just copy the new interface file over to the user’s machine, update the links to the data file, and the upgrade is done.

⚠ *Do not* used linked tables in the assignment. The links are dependent on the absolute directory structure. As a result, if the directory structure on your machine is different from that on the marker’s machine, the marker will not be able to use your application without first updating the links (a time consuming process for a large number of assignments).

## 1.5 Application to the assignment

After completing this tutorial you should be ready to create the database file that you will use for the remainder of the course.

1. Create an empty database file called `<your groupID>.mdb`. Remember that your group number consists of eight digits.

2. Import the `inventor.xls` spreadsheet as your `Products` table.
3. Use the compact utility to make a backup copy of your database (use a different name such as `backup.mdb`).