

1.1 Introduction: Why work through these lessons?

The lessons contained in this tutorial are designed to teach you about building information systems to solve practical business problems. Although much of the content relates to MICROSOFT products—and in particular, MICROSOFT ACCESS—the overall pedagogical objectives are much broader than simply teaching you ACCESS. Specifically, the lessons have been designed to:

1. Demystify important information technology concepts that arise in databases, programming, and the Internet.
2. Provide practical guidance on how to model real-world business problems and implement solutions using database technology.
3. Give the you confidence to build simple information systems to solve your own problems.

As a by-product of working through the lessons, you will certainly learn a great deal about using the MICROSOFT ACCESS database package. ACCESS is used because it is cheap (relative to other database systems), powerful enough to be a

good teaching tool, and ubiquitous. However, much of what you learn will apply to any relational database system.



Will these tutorials help be become a MICROSOFT CERTIFIED PROFESSIONAL?¹ The short answer is “yes and no”. The long answer is in [Section 1.1.3](#).

1.1.1 Betwixt dummies and developers

These tutorials have emerged from my experience as a teacher of information systems in a business school and also as a day-to-day user of information technology to solve my own problems. Thus, these tutorials address what I think “business people” (broadly defined to include you) need to know about the nuts-and-bolts of technology. The target audience of these lessons is people who are not information technology professionals, but who nonetheless

¹ Those unfamiliar with MICROSOFT’s certification programs may wish to visit the MICROSOFT web site and search under “certification”. The basic concept is that MICROSOFT has successfully incubated a massive third-party training market for its products. To ensure a minimum standard of quality across these programs, MICROSOFT has established curricula and an certification examination process.



need to use information technology to solve non-trivial, real-world problems.

Unfortunately, a conspicuous gap exists between the resources available for “dummies” and those available for “developers”. The target audience for the “dummies” books tends to be **end-users** of a technology—for example, clerical staff who use ACCESS-based applications. Although the dummy books have enormous appeal and are an excellent way to get started, the material tends to focus on mechanical skills, such as formatting, using wizards, and so on. Unfortunately, knowing how to create simple tables and put bold headings on reports will only take you so far.

At the other end of the spectrum are resources targeted at developers—people who make their living writing software and building information systems. The titles of these books often contain terms such as “developer’s bible” or “secrets unleashed” and the books themselves run to a thousand pages or so (not including material on the inevitable companion CD-ROM). The problem with these resources is that *we* are all not full-time developers. We have neither the time nor the inclination to (say) use the undocumented features of WINDOWS API to write a control system for a nuclear submarine.

These tutorials are meant fill the gap betwixt dummies and developers. The lessons begin

slowly to accommodate people who know nothing about information systems or MICROSOFT ACCESS. As the lessons progress, however, theories and techniques are introduced that enable one to go far beyond a simple user’s view of databases.



If you are not being modest and you really know *nothing* about computers, a dummy book may be a prerequisite investment. The dummy books are good at teaching you the basics of using the mouse, resizing and moving windows, and so on. In the lessons that follow, basic computer skills are assumed.

1.1.2 Make versus buy

By the time you finish the final lesson in these tutorials, you will have accomplished the following:

1. **Designed** a relational database for storing information about a small wholesale business.
2. **Built** a graphical application on top of the database to simplify order entry and minimize redundancy and errors.
3. **Transformed** your raw data into a more useful format to support decision-making.



4. **Web-enabled** the business application so customers can place their own orders over the Internet.

In short, these tutorials will teach you how to build a functional business system from scratch. Keep in mind, however, that there are people who devote their entire careers to the art and science of building functioning business systems from scratch. The knowledge you gain here should not be considered a substitute for hiring a professional when it comes time to implement a mission-critical business systems. Instead, the knowledge you gain by working through these tutorials is more appropriately used in one or more of the following ways:

1. **Enable you to build *non-mission critical systems*** – If you want to build a quick-and-dirty system to accomplish Task *X* and the failure to accomplish Task *X* does not endanger your ability to feed your family, by all means, take a crack at it. We learn by doing. If Task *X* is mission critical, however, you should hire a professional.
2. **Permit you to communicate with software professionals** – Good software professionals are hard to find. Good software professionals who also understand the complexities of *your* business problems are virtually impossible to find. Thus, the ability to communicate with your consultants or internal IS staff *in their own*

language can yield a massive payoff in the long run. You will get the system you require without being condescended to or exploited to by techno-bullies.

3. **Provide a foundation for becoming an IT professional**—Given the shortage of good software professionals referred to above, there is ample opportunity to become one. These tutorials are a good place to start.

1.1.3 Certification

A deliberate decision has been made to avoid tying these tutorials in with the MICROSOFT certification curricula. There are two reasons for this. First, as stated above, the goal of these tutorials is to teach broad information systems and technology concepts that transcend a single vendor. As such, we are more interested in solving business problems than memorizing keystroke sequences and the intricacies of MICROSOFT's "component object model".

Second, as professional educators, we reserve the right to speak freely and editorialize whenever we feel the urge. That is, if we encounter a particular feature of ACCESS that is dumb, we want the freedom to write: "This is a dumb feature." In other words, we wish to be as unbiased and truthful as we can.¹

¹ If for some reason Redmond threatens to litigate, we will, of course, cave-in without hesitation.

Naturally, there overlap between the material in this tutorial and any MICROSOFT approved training program (especially the MICROSOFT OFFICE USER SPECIALIST—or MOUS—program). In addition, the concepts we are addressing (e.g., relational databases, programming, web development) are standardized (more or less) so nothing here contradicts the official MICROSOFT training program (if it does, please let us know and we will instruct MICROSOFT to make the necessary corrections). Thus, although these tutorials are not designed to help you cram for a certification program, they certainly will not hinder your preparation.

1.2 Structure of the lessons

The tutorials are organized in the following manner:

1. **Scenario** — You are provided with a business scenario in [Lesson 2](#) to work through. The scenario provides a common thread through all of the lessons and thus the lessons should be considered cumulative (i.e., you should finish Lesson 5 before starting Lesson 6, and so on).
2. **Lessons** — Each lesson consists of different sections:
 - a) **Introductory comments and learning objectives** — The opening section of each lesson is meant to give you some

idea of what you will accomplish in the lesson and why it is important.

- b) **Tutorial exercises** — The exercise are the meat of the lesson since you learn by doing. All the steps that require action on your part are marked with an arrow (➡).



If you are an extremely action-oriented learner, you can scan the text for the arrows, follow the instructions, and read the ancillary chit-chat on an as-needed basis.

- c) **Discussion** — During the course of the lesson, technical and theoretical issues may arise. Rather than break of the flow of the exercises with long detailed explanations, you are encouraged to suppress your natural curiosity, work through the exercises, and then read the discussion at the end of the lesson. Hopefully the material in the discussion will fill in the important gaps in your understanding.
- d) **Application to the project** — The general problem that occurs when one works through the steps of a tutorial is that no real learning occurs. In other words, you may remember the sequence of keystrokes and mouse



jiggles required to complete a task, but you cannot generalize the skill to other tasks. In the “application to the project” sections, you are on your own. You are given tasks that you must accomplish before you can move on to the next lesson. If this sounds a lot like work, that is because it is work. No pain, no gain, right?

- Background lessons** — We are firm believers that there is nothing more practical than a good theory. Databases and programming languages are based on explicit theories, and unless you understand the critical theoretical principles that underlie these technologies, you are going to find it hard to generalize the skills you learn here to your own problems. Background lessons—such as this one—are intended to fortify your theoretical knowledge. Since there are no exercises in a background lessons, you are encouraged to take your hands off the keyboard and simply read.

1.3 Typographical conventions

The following is a brief list of symbols and typographical conventions used in the tutorials.

1.3.1 Warnings, tricks, and tips



Important warnings are marked with an exclamation mark. It is important that you heed these warnings to avoid common problems.



Other clarifications, recommendations, and trivia are marked with a question mark. These sections are typically time-saving tips or explanations of alternative ways of doing things.


HINT: In the “application to the project” sections, hints to help you complete the steps are indicated with the **hint** symbol.


1.3.2 Version differences

All the screen shots and videos in these tutorials are taken from ACCESS version 8.0 (released as part of MICROSOFT OFFICE 97). Although there are some important differences between the various versions of ACCESS (i.e., version 2.0, version 7.0, version 8.0 and ACCESS 2000) the underlying concepts remain the same in all versions.



Whenever the instructions given in the tutorial differ significantly from version 8.0 (ACCESS 97), a warning box such as this is used.

 Similarly, the “2” marker indicates that the procedure for version 2.0 differs from the procedure described in the lesson.

 This set of tutorials does not cover ACCESS 2000¹. However, with a few notable exceptions, the differences between ACCESS 8.0 and ACCESS 2000 are cosmetic. As such, it is possible to complete the lessons using ACCESS 2000.

1.3.3 Exercises

➔ As discussed above, tutorial exercises are indented and marked with an arrow.

1.3.4 Important terms and hyperlinks

If a term is important, it is marked in **bold text**. If a word in the electronic version of this lesson is a hyperlink to another location, it appears in [blue](#). Clicking on a hyperlink takes you directly to the new location.

1.3.5 Menu commands

In some cases, you are asked to use the mouse to execute a series of menu commands (e.g., **File** → **Save As**). What this means is that you

select **File** from the main menu, followed by **Save As**.


1.3.6 Programming code

In some exercises, you are asked to type in programming code, such as VISUAL BASIC (VB) or STRUCTURED QUERY LANGUAGE (SQL). The code you enter is shown in a **monospaced font**.

1.3.6.1 The “new line” marker

When multiple lines of programming code are shown, the “new line” marker (**NL**) is used to indicate the start of a new line (you start a new line by pressing the **Enter** key). For example, each of the **set** statements below should be typed on a single line regardless of the line breaks that appear in this document:

```
NL Set dbCurr =
    DBEngine.Workspaces(0).Databases(0)
NL Set rsBackOrders =
    dbCurr.OpenRecordset("BackOrders",
    dbOpenDynaset)
```

 Unlike most modern languages, the main programming language you will use to develop your application, VISUAL BASIC FOR APPLICATIONS (VBA), requires each statement to be on its own line. However, the narrow columns used in the lessons result in line wrapping. If you type VBA code in its line-wrapped form, you will

¹ The ACCESS 2000 tutorials are under construction. See 2np.org for details.



encounter errors. Hence the use of the **NL** marker.

1.3.6.2 New code

When code is added to an existing program, the old code is shown in a lighter typeface whereas the new code is bolded. For example, the third statement below is new:

```
NL Set dbCurr =  
    DBEngine.Workspaces(0).Databases(0)  
NL Set rsBackOrders =  
    dbCurr.OpenRecordset("BackOrders",  
    dbOpenDynaset)  
NL Set qdf =  
    dbCurr.QueryDefs!pqryBackOrderChang  
    es
```

Making a distinction between new code and existing code makes it easier for you to find the location of the changes in your own programs much faster.

1.4 Questions, queries comments

Much of the material in these tutorials emerged from student questions. If you find something difficult to understand, or if you encounter a problem in your own work that is not covered in the tutorials, please let us know. We will do our best to continuously grow and upgrade the material.