

By Susan Sales Harkins

Access stores one of your most important assets--your data--in tables. Now, truthfully, the table you see onscreen is just a graphic representation, but you use this representation to determine how the rest of your database uses and reacts to your data. For this reason, it's important to spend time up front assigning the appropriate properties and data types to suit the data. If the tables aren't right, everything else is suspect.

1 Use table level properties to increase efficiency

Access tables offer several properties that remain with the data. You set the property just once, at the table level, and bound objects inherit those settings:

- ◆ **Format:** Determines how Access displays the data.
- ◆ **Caption:** Access displays this text in a control's corresponding label box.
- ◆ **Input Mask:** Forces users to enter data in a specific order or manner.
- ◆ **Default Value:** Access automatically populates a control with this value for new records.
- ◆ **Description:** Documents at the table level. Access displays this text in the status bar when the control has focus.
- ◆ **Validation Rule:** An expression that sets conditions that a value must meet for Access to accept the value.
- ◆ **Validation Text:** A text message that Access displays if an entry fails to meet the Validation Rule expression. Use this to provide clues for entering appropriate data.

You're not stuck with these property settings. At the form or report level, you can override any of these properties for just that form or report. In all cases, you save time and your data exhibits continuity from object to object.

2 Name fields appropriately

A field name should reflect a field's purpose and describe the data it stores. The field names `FirstName`, `LastName`, and `ZipCode` are self-documenting and easy to manage. There's no guessing. You know exactly what type of data should be in that field. There are, however, a few rules you must follow when choosing field names:

- ◆ A field name can consist of up to 64 characters--but only letters, numbers, and spaces.
- ◆ Don't use reserved words to name fields, or any object, for that matter. For a list of reserved words, search Help for "reserved words."

If you aren't bound by an in-house naming convention, consider creating one of your own.

3 Don't use spaces in field names

Although you might be tempted to use spaces in your field names, don't. Spaces can be difficult to work with, especially in SQL statements and VBA code. If you use them, you must remember to enclose the reference in brackets, which is a nuisance. In addition, if you upgrade the database to SQL Server or export the data to use in another application, those space characters most assuredly will cause problems.

If you want field headings and subsequent objects (see # 1) to display more natural text, use the field's `Caption` property. For instance, if you name a field `LastName`, use the `Caption` property to display *Last Name*. Just remember that the `Caption` property is for display only. When referencing the field, you must always use the field's actual name.

Caption properties come with their own set of behavioral problems. If you're going to use the Caption property, plan for the following pitfalls:

- ◆ Access ignores the AS keyword (alias) in a SQL statement if the underlying field has a Caption setting.
- ◆ A Caption property setting won't make it to the results of a Make Table query.
- ◆ DAO and ADO field objects return a field's Caption property and not the underlying field's name.

4 Don't waste data type effort

When storing numeric data, you might be tempted to assign a Number data type. Before you do, consider the data's purpose. For instance, you won't use street number or zip code values in mathematical equations. When this is the case, store the data as Text. You'll save a bit on memory, but more importantly, the data type is true to the data's purpose. If you should need to use a text value as a true numeric value, use the Val() function in the form

```
=Val(field)
```

where *field* represents the Text field that's storing numeric values or a literal value.

5 Use the most appropriate field size

With today's powerful systems, assigning the most appropriate field size isn't as urgent as it once was. However, as a matter of good practice, developers still limit field size. It's your first step to validating data. For instance, let's say you store state abbreviations in a Text field with Field Size setting of 2. If the user enters ARK instead of AK, Access will reject the entry. Of course, the field size property rejects only entries that are too big. It can't spot typos or other mistakes. The field would still accept other invalid entries, such as "A" or "K6."

This works with numeric fields too; it just isn't as obvious. For instance, a Byte field accepts values 0 through 255, while the Long Integer accepts values from -2,147,483,648 to 2,147,483,647.

Choose the appropriate data type and field size property for each field. Always choose the smallest data type and field size that will accommodate the largest possible value in that field.

6 Choose indexes carefully

Access uses an index to sort data in logical order, according to data type. The right indexing can improve performance. The wrong index will slow things down. By far, the most common mistake is to set too many indexes. Because Access updates the index each time you add or change a record, too many indexes will affect performance. Fortunately, you can apply a few indexing guidelines that will help:

- ◆ Remember that a primary key automatically sorts--that's an internal action you can't control. In most tables, it's the only index you really need. It's called a unique index because every value in the field must be unique.
- ◆ As a general rule, consider indexing a table's foreign key.
- ◆ Avoid an additional index on a table that you'll be frequently updating, unless one of the above conditions applies.
- ◆ If the table has a primary key, consider an index on a second field only when you're working with large amounts of data, you plan to search or sort by that field often, or the field contains mostly unique values.

In short, index any field you want to sort, search, or join to another table to speed up tasks. Just remember that each index increases the size of the database, and too many indexes will slow things down. If you're working with small to reasonable amounts of data, indexes--beyond the primary key--usually aren't necessary.

One final word on indexes: Don't use an index to sort. That isn't their purpose and you might not end up with the results you need.

7 Beware of AutoNumber limitations

Use an AutoNumber field to generate a unique number for your records. Many people use these fields as the table's primary key, which is fine. This type of key is called a surrogate key. In later versions of Access, you can generate an incremental or random value. Incremental values are fine for most tables. You probably won't use random values unless you're using replication to synchronize multiple databases.

Remember that a table can have only one AutoNumber field. That might limit the way you can use that table later. For instance, if you need to merge two tables with AutoNumber fields in a Make Table query, you might run into trouble. It's important to keep this limitation in mind when adding an AutoNumber to a table.

8 Change the starting value of an AutoNumber field

By default, an AutoNumber field starts with the value 1 and increments by 1 for each record. Occasionally, you may need to start a new table with an AutoNumber value other than 1. There's no built-in property that lets you do this, but you can force a value:

1. Create the table and add an AutoNumber field but don't enter any records.
2. Create a second table with one Long Integer Number field. Give the field the same name as the AutoNumber field in the data table (step 1).
3. In the single field table (step 2), enter the value that is 1 less than the value you want the AutoNumber field to begin with. For instance, if the first AutoNumber value must be 100, enter 99.
4. Run an append query to append the record in the single-field table to the data table.
5. Delete the single-field table.

When you enter the first record into your data table, the AutoNumber field will generate a value that is 1 more than the value you entered into the single-field table (step 3).

9 Use analysis tools

Access has two tools that will help you refine your design. First, the Table Analyzer Wizard reviews a table and recommends changes where appropriate. Second, the Performance Analyzer reviews your entire database and makes suggestions for improving the design. Usually, you'll want to apply the wizards' advice.

To run either wizard, choose Analyzer from the Tools menu and choose the appropriate item.

10 Don't forget table properties

Like fields, tables have properties that define the table's purpose. Most are self-explanatory, and the defaults are usually adequate. To access these properties, open the table in Design view and then choose Properties from the View menu. Here are a couple you should know about:

- ◆ **Order By:** Specifies a sorting order that Access applies when you open the table. Simply enter the name of the field by which you want to sort the records. If you don't use this setting, Access sorts by the primary key. If there's no primary key, Access displays records in data entry order. It won't matter much at the table level, since users don't view tables. However, like many field properties, bound objects inherit the property.
- ◆ **Subdatasheet Name:** Determines whether subdatasheets display related records. Many people find subdatasheets annoying. If that's you, set this property to [None] to disable it.

11 Be aware of the limits

Although you may never reach any of the limits set for tables, you should know they exist:

- ◆ Table names can contain up to 64 characters.
- ◆ Field names can contain up to 64 characters.
- ◆ Each table can contain up to 255 fields.
- ◆ Table size is limited to 2 gigabytes minus the space needed for the system objects.
- ◆ A Text field stores up to 255 characters.
- ◆ A Memo field stores up to 65,535 characters when you enter data via the user interface. It'll accept up to 1 gigabyte if you enter the data programmatically.
- ◆ An OLE field supports up to 1 gigabyte.
- ◆ Each table can have up to 32 indexes.
- ◆ Each index can comprise up to 10 fields.
- ◆ A Validation Rule expression stores up to 2,045 characters; the Validation Text property can have up to 255 characters.
- ◆ The Description property stores up to 255 characters.
- ◆ Each record can store up to 4,000 (for 2003) characters. (Version 2002 is 2,000.)

Susan Sales Harkins is an independent consultant and the author of several articles and books on database technologies. Her most recent book is *Mastering Microsoft SQL Server 2005 Express*, with Mike Gunderloy, published by Sybex. Other collaborations with Gunderloy are *Automating Microsoft Access 2003 with VBA*, *Upgrader's Guide to Microsoft Office System 2003*, *ICDL Exam Cram 2*, and *Absolute Beginner's Guide to Microsoft Access 2003*, all published by Que. Currently, Susan volunteers as the Publications Director for Database Advisors at <http://www.databaseadvisors.com>. You can reach her at ssharkins@setel.com.

Additional resources

- TechRepublic's [Downloads RSS Feed](#) **XML**
- Sign up for TechRepublic's [Downloads Weekly Update](#) newsletter
- Sign up for our [Microsoft Office Suite newsletter](#)
- Check out all of TechRepublic's [free newsletters](#)
- ["70+ Keyboard shortcuts to move faster in Microsoft Access"](#) (TechRepublic download)
- ["Design Access 2003 forms for efficient and accurate data entry"](#) (TechRepublic download)
- ["Access reference: Working with tables"](#) (TechRepublic download)
-

Version history

Version: 1.0

Published: March 8, 2007

Tell us what you think

TechRepublic downloads are designed to help you get your job done as painlessly and effectively as possible. Because we're continually looking for ways to improve the usefulness of these tools, we need your feedback. Please take a minute to [drop us a line](#) and tell us how well this download worked for you and offer your suggestions for improvement.

Thanks!

—The TechRepublic Downloads Team