

Helpful Techniques for Access 2000



(408) 433-0990

<http://www.mentortraining.com>

Establishing Relationships

Table relations are necessary for the integrity of the database and must be established.

Referential integrity is the means by which relationships between tables are enforced. Referential integrity means that a parent record cannot be deleted if any child records depend on its existence. For example, a customer who has any orders in the Orders table should not be deleted. Referential integrity in Access has two options:

- *Cascade deletes*: this option should very seldom, if ever, be used. Checking this option will delete any child record when the parent is deleted.
- *Cascade updates*: this option is primarily used when lookup tables in relationships are involved.

How to establish relationships:

1. Click on the table tab of the container to select it.
2. Right click on the table tab.
3. Select "Relationship..."
4. If this is the first time relationships are being established, select all the tables and click the Add button. If this is *not* the first time, click on the toolbar button with the yellow plus sign, select the tables to add, and then click the Add button.
5. Click and drag with the mouse from the primary key (parent table) to the foreign key (child table). Access really doesn't care which direction you drag but you should know which are the parent and child tables. The relationship dialog will appear.
6. Make sure that the key fields showing in the 'Table/Query' and 'Related Table/Query' match your selection of primary and foreign key.
7. Click on 'Enforce referential integrity.'
8. Ignore 'Join type' button. Joins are better enforced at the query level as needed.
9. If the parent table is a single column lookup table then click on 'Cascade update related fields.'
10. Click the 'Create' button.
11. Repeat as necessary.

Generating lookup tables from existing data

Lookup tables are used to restrict data input in main and transaction (relation) tables. They generally are not more than one to three columns wide. Sometimes main tables can also act as a lookup table. For example, when filling in an order in the 'Orders' table we could restrict the customer number column to existing values in the 'Customers' table by having the 'Customers' table act as a lookup in the 'Orders' table.

Only authorized personnel should have access to the data in lookups.

Every proper noun stored in a database is a candidate for a lookup table and should be.

How to create lookup tables from existing data:

1. Click on the Query tab in the container.
2. Select 'New.'
3. Select the table containing the column to be converted to a lookup table.
4. Select View>Properties. The property sheet will open. The title bar will read 'Query properties.'
5. Change the 'Unique Values' property to *Yes*. Make sure you're not changing the 'Unique Records' property by mistake.
6. Drag the column to be converted to a lookup table into the grid.
7. Select Query>Make table. The Make Table dialog will appear
8. Type in the name for the new lookup table.
9. Click on the button that has the maroon exclamation point. This button runs action queries and will generate the table.
10. If any other columns from the same table need to be converted, delete the current column from the grid, drag in the new column, and repeat step 7 to give this table a name. You will not need to reset the query property.
11. For each table created, open it in Design View and assign a primary key.
12. Include each table in a relationship.

Creating combo boxes in tables

Every foreign key is a candidate for a lookup column. In other words, for every lookup table you create you will create at least one lookup column. Each field storing a value from a lookup table must be a *combo box*. A combo box is also known as a pick list or drop down list. This ensures that data entered into main and transaction (relation) tables will be accurate and consistent. Restrict free-form typing to a minimum.

There is a wizard that helps to create a lookup field in a table, but it restricts the user to choosing an existing table or query. This is not acceptable for two reasons:

1. Combo boxes are generally sorted to help the user pick from the list. Unfortunately, the lookup tables are sorted by Primary key order, which may or may not be the order you wish to use in your combo box.
2. Using combo boxes based on a stored query is unacceptable if there is any chance the query could be deleted. Also data integrity within the table is based on an external object. Most database experts would cringe at the thought.

The only acceptable alternative is to create the pick list based on SQL, which means that you yourself must set the appropriate properties.

How to create combo boxes in tables:

1. Open the table in Design view.
2. Click on the column to be converted.
3. In *Field properties* (bottom half of table) select 'Lookup.'
4. Click in the field labeled *Display control* and select 'Combo box.' The properties will appear.
5. Do not change *Row Source type*. This property should always be 'Table/Query.'
6. The *Row Source* property is one of the most important of all properties. When you click in this field you will see a button with three dots on it at the far right of the field. Anytime you see one of these buttons it means you have a builder at your disposal. A builder helps you create the property.
7. Click on the button with three dots and a standard query window appears.
8. Select the lookup table and click on 'Close.'
9. Make sure you drag the primary key field into the first slot of the grid. Drag in any other fields for your combo box in the order you want them displayed.
10. Sort the Query.
11. Click on the X in the upper right hand corner of the window to close it. A dialog box appears that asks if you want to save the changes made to the SQL statement.
12. Click on *Yes* in the dialog box.
13. An SQL statement appears in the property.
14. Select the *Bound Column* property and leave a *1*.

When you make a selection from a combo box, the *Bound Column* property tells Access which column value to store in the current table.

15. Select the *Column Count* property and type in the number of columns you wish to display in the combo box. This should equal the number of displayed columns in the query you built with the row Source property builder.
16. Select *Column heads* and change it to *Yes*. This tells Access to display column headings for each column in your combo box. The heading are based on fields names, caption properties for field names or field names generated in the query.
17. Select *Column Widths* and type in a comma-separated list of widths in inches. If a column width is set to zero, it doesn't appear in the combo box. You may have to play with this property to get the combo box columns to display as desired.

If the primary key is a random, system generated number, you can hide the primary key column by setting its width to zero. Be aware that Access converts the comma-separated list into a semi-colon-separated list with inch marks. If you edit the list, be sure you leave this syntax intact. An example of this setting for a three column combo box where the first column is the primary key would look like this: *0. 1.25,2*. This setting would produce a two column combo box where the first visible column would be 1.25 inches wide, the second would be 2 inches wide and the primary key column is invisible.

Don't make the mistake of not including the primary key column in the combo box-thinking that if we don't see it, we don't need it. Access needs to have this value available so that it knows what to store in the current table. This property works closely with the *List Width* property as described below.

18. Select *List Rows* and enter how many rows you want the list box to display. The default is 8, but I usually change it to 12. I prefer longer lists to reduce scrolling.
19. Select *List Width* and change this setting to the sum of the individual column width specified in the *Column Width* property. Never leave this setting on *Auto* for a multi-column list as the combo box invariably displays an annoying horizontal scroll bar. A well-behaved combo box has only a vertical scroll bar.
20. Select *Limit to List* and change this to YES. The default setting is NO. If you leave this set on NO, you've basically enabled garbage data entry.

MentorTraining
LEARNING THAT CLICKS™

(408) 433-0990

<http://www.mentortraining.com>