Microsoft Access XP (2002) Tables

- Access Overview
- Database Navigation
- Table Creation
- Primary Keys
- Data Entry & Editing
- Table Modification
- Setting Table Properties
# Table of Contents

ACCESS 2000 OVERVIEW ........................................................................................................4
TABLES ...................................................................................................................................6
PROPER TABLE CONSTRUCTION ..........................................................................................7
  Break Data Into Separate Columns ....................................................................................7
  Vertical vs. Horizontal Construction ...................................................................................8
  Do Not Create Fields That Can Be Generated With A Calculation ...................................8
  Break Data Into Separate Tables .......................................................................................9
CREATE A NEW DATABASE ..................................................................................................10
CREATING A TABLE .............................................................................................................11
  Creating the Employees Table ...........................................................................................11
  Design View ......................................................................................................................11
  Data Type: .........................................................................................................................12
SETTING A PRIMARY KEY .....................................................................................................13
MORE ON PRIMARY KEYS .....................................................................................................14
  Removing a Primary Key .................................................................................................14
  Assign Multiple Primary Keys .........................................................................................14
  Creating an Input Mask .....................................................................................................15
  Tracking The Employee’s Children ..................................................................................15
  Creating the Children Table .............................................................................................16
TABLE RELATIONS ..............................................................................................................18
  Conditions for Relating Tables .........................................................................................18
  Relating Employees & Children .......................................................................................18
NAVIGATING TABLES ...........................................................................................................20
  Keyboard Navigation ........................................................................................................20
  Mouse Navigation .............................................................................................................20
  View Related Tables .........................................................................................................20
ENTERING DATA ..................................................................................................................21
  Tips on Data Entry ............................................................................................................21
  Entering Dates ..................................................................................................................21
  Entering the Data ..............................................................................................................21
SAVING DATA .......................................................................................................................22
EDITING CELLS ...................................................................................................................22
DELETING RECORDS ............................................................................................................22
ENTERING DATA INTO A SUBORDINATE TABLE .............................................................22
TABLE FORMATTING ..........................................................................................................23
  Adjusting Column Widths ..................................................................................................23
  Sort A Column ..................................................................................................................23
  Change Column Order ......................................................................................................23
  Freeze Columns ...............................................................................................................23
  Unfreeze Columns ...........................................................................................................23
  Hide Columns ..................................................................................................................23
  Unhide Columns ...............................................................................................................23
  Control Gridlines and Cell Effects ...................................................................................24
  Data Entry Mode ..............................................................................................................24
EDIT TABLE STRUCTURE ....................................................................................................25
  Adding New Fields - Safe ...............................................................................................25
  Inserting a Row - Safe ......................................................................................................25
Deleting Fields - Unsafe ................................................................. 25
Renaming Fields - No Longer Unsafe .............................................. 25
Moving Fields - Safe ........................................................................ 26
Changing the Data Type - Possibly Unsafe ...................................... 26
Changing the Field Description - Safe .............................................. 26
Changing Field Format - Safe........................................................... 26
Changing Field Size - Possibly Unsafe ............................................ 26

SET FIELD PROPERTIES .................................................................. 27
Field Size .......................................................................................... 28
Formats ............................................................................................. 28
Caption .............................................................................................. 29
Default Value ................................................................................... 29
Validation Rule .................................................................................. 30
Validation Text .................................................................................. 30
Required ............................................................................................ 31
Allow Zero Length ............................................................................ 31
Indexed ............................................................................................... 31
Unicode Compression .......................................................................... 31
Combo Box: Value List (Manual Creation) ........................................ 32
Combo Box: Referencing a Table (Manual Creation) ......................... 32
Using the Wizard to Create a Combo Box - Value List ....................... 33
Using the Wizard to Create a Combo Box - Table Reference ............... 33
ACCESS XP (2002) OVERVIEW

Access is a relational database that allows users to store, manipulate, and maintain vast amounts of data in a useful manner. Some of the more common tasks in Access include:

- Storing, editing, deleting, & retrieving data.
- Sorting and searching for data based upon a criteria.
- Printing data including mailing labels.
- Performing calculations on data.

Access consists of 5 major objects: Tables, Queries, Forms, Reports, and Pages

Tables: Storage
Access stores data in tables. Everything else created (queries, forms and reports) are either directly or indirectly based upon tables. No other object in Access stores data. You can work on your data within the table or you can create forms to work on it. Separate tables can also be linked together through the use of queries.

Queries: Questions
Queries are for asking questions of a table (or a saved query). For example, you can ask the query to list only those people who have two children, were born before 1/2/67 and make over $50K per year and Access will list only those people who meet that criteria. Access also has “Action” queries that can be used to make global changes to a table. For example, you can construct a query that will raise the credit limit of just those people who live in California by 10%.

Forms: Input & Editing
Forms are for input and editing data and are always based on tables. When you create a form, you are really looking at the data in your table. It merely displays differently on the screen. This means that if you change your data in the table, it changes in the form and if you change the data in the form, it changes in the table. The reasons why you might prefer inputting into a form rather directly into a table are:

- Forms can display pictures whereas tables can only hold the pictures.
- Forms can have calculations, tables cannot.
- You can input into multiple tables from one form.
- Forms can hide security data.

Reports: Output
Reports are for printing such things as mailing labels, summary reports, and calculations. Tables, forms and Queries can all be printed as well, but reports give the user more control over the look of the printout. Reports are always based upon either tables or saved queries.

Pages
Data Access Pages allow you to create web pages that people can use to access your database through a web browser. Note that for this to work, you must be using the Internet Explorer.
Tables
All of your data in Access are stored in Tables. All objects in Access are based upon Tables. Tables can be:
- Used for data entry & editing.
- Printed.
- Linked.
- Sorted & Searched.

Forms
Based off a table, a Form provides another way to look at the data in the table. Data Forms can be:
- Used for data entry & editing.
- Used to display pictures.
- Used to create calculations.
- Printed.
- Sorted & Searched.

Queries
Based on tables, Queries are for asking questions of your tables. Queries can be used to:
- Search for and sort data.
- Create calculations.
- Summarizing data.
- Link tables together.
- Do global editing.
- Create & append to tables.

Reports
Based off a table or query, Reports are for specialized printing. Reports are used to:
- Create mailing labels.
- Group items and show subtotals.
- Display summary info.
- Create calculations.
- Display pictures.

Reports can be based off a saved query. This is useful to filter out unwanted data.

Multiple tables can be linked through queries.
TABLES

A table is the first object you must create in your database. The example below displays the makeup of a typical table.

<table>
<thead>
<tr>
<th>CANDY ID</th>
<th>BRAND NAME</th>
<th>MANUFACTURER</th>
<th>COST PER UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mars Bars</td>
<td>Nabisco</td>
<td>$50.00</td>
</tr>
<tr>
<td>2</td>
<td>Milky Way</td>
<td>Nabisco</td>
<td>$25.00</td>
</tr>
<tr>
<td>3</td>
<td>Wacko's</td>
<td>Hershey</td>
<td>$100.00</td>
</tr>
<tr>
<td>4</td>
<td>Zig Zags</td>
<td>General Foods</td>
<td>$25.00</td>
</tr>
<tr>
<td>5</td>
<td>Paydays</td>
<td>Hormell</td>
<td>$40.00</td>
</tr>
</tbody>
</table>

Each row is called a “Record”.

The titles at the top of a column are known as a “Field Heading”.

Each box is called a “Field”.

Each row is called a “Record”.

The titles at the top of a column are known as a “Field Heading”.

Each box is called a “Field”.
PROPER TABLE CONSTRUCTION

How you construct your tables will determine what you can do with them. Therefore, it is vitally important to construct your tables correctly prior to entering data into it. The process of organizing data in a database is known as “Normalization”. There are normalization guidelines which are covered below. Generally, these guidelines should be followed unless real world scenarios make following them too cumbersome.

Break Data Into Separate Columns

Data should be broken up into separate fields as much as possible. This is necessary to aid searches, sorting, and creating form letters and mailing labels.

Note that in the Address field, we left the house number in the same field with the street name. For most users this is acceptable but if you were a real-estate agent, you might want to separate these out further. How much you separate your data into separate columns often depends upon what you plan to do with the data.

<table>
<thead>
<tr>
<th>Salutation</th>
<th>First Name</th>
<th>Last Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zipcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr.</td>
<td>Jonathan</td>
<td>Macias</td>
<td>500 West Hope</td>
<td>Los Angeles</td>
<td>CA</td>
<td>90804</td>
</tr>
<tr>
<td>Mrs.</td>
<td>June</td>
<td>Smith</td>
<td>3987 Elm</td>
<td>Anaheim</td>
<td>CA</td>
<td>90803</td>
</tr>
</tbody>
</table>

CORRECT

Breaking data into separate fields makes the data more versatile. For example, if you wish to make mailing labels for only those people who live in Los Angeles, the task is easier if there is a field called City which contains the names of cities but nothing else. Sorting is performed by columns; therefore, if first an last name were in the same column, you would not be able to sort by last name.

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mr. Jonathan Macias</td>
<td>500 West Hope Los Angeles, CA 90804</td>
</tr>
<tr>
<td>Mrs. June Smith</td>
<td>3987 Elm Anaheim, CA 90803</td>
</tr>
</tbody>
</table>

INCORRECT

The data is lumped together making sorting, searches, and form letters very difficult to accomplish. For example, by placing the entire name in one column, it makes it very difficult to sort by last name.
**Vertical vs. Horizontal Construction**

The examples below show two ways that orders could be tracked. The example on the left shows the correct way to construct the table while that on the right shows the incorrect way.

The example on the right is incorrect for several reasons:

- If any new products are sold, the table must be redesigned to include the new product; but with the correct example, the new product will simply be added to the bottom of the list.
- Calculations are more limited in the incorrect example.
- The maximum number of columns is 255. Therefore, if you have more than 255 products, the table on the right will prove insufficient.

<table>
<thead>
<tr>
<th>Order #</th>
<th>Candy Name</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Snickers</td>
<td>200</td>
</tr>
<tr>
<td>1</td>
<td>M&amp;M's</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>Mars Bars</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>Snickers</td>
<td>600</td>
</tr>
</tbody>
</table>

**CORRECT**

<table>
<thead>
<tr>
<th>Order #</th>
<th>Mars Bars</th>
<th>Snickers</th>
<th>M&amp;M's</th>
<th>Wonka Bars</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>200</td>
<td>300</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>600</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**INCORRECT**

**Do Not Create Fields That Can Be Generated With A Calculation**

Calculations can easily be created in Queries, Forms, and Reports. For this reason, it is unnecessary to place fields in a table that could be arrived at by using a calculation.

For example, the table below includes three fields: **Cost Per Case**, **Items Per Case**, and **Cost Per Item**. **Cost Per Case** is how much it costs for a case of candy bars and **Items Per Case** tells us how many candy bars are in a case. The last field **Cost Per Item** is unnecessary because we could easily calculate this information in a query, form, or report by dividing the **Cost Per Case** by **Items Per Case** without wasting input time or file space.

<table>
<thead>
<tr>
<th>BRAND NAME</th>
<th>COST PER CASE</th>
<th>ITEMS PER CASE</th>
<th>COST PER ITEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mars Bars</td>
<td>$50.00</td>
<td>150</td>
<td>$0.33</td>
</tr>
<tr>
<td>Milky Way</td>
<td>$25.00</td>
<td>50</td>
<td>$0.50</td>
</tr>
<tr>
<td>Wacko's</td>
<td>$100.00</td>
<td>200</td>
<td>$0.50</td>
</tr>
<tr>
<td>Zig Zags</td>
<td>$25.00</td>
<td>50</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

**INCORRECT**
**Break Data Into Separate Tables**

One of the more powerful aspects of Access is its ability to link or relate separate tables to one another. This prevents duplication of data. Duplication wastes time, slows down searches, and wastes space on hard drives.

**INCORRECT**

<table>
<thead>
<tr>
<th>Order #</th>
<th>Customer ID</th>
<th>Order Date</th>
<th>Ship Date</th>
<th>Ship Method</th>
<th>Item</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB1</td>
<td>08-Jun-97</td>
<td>12-Jun-97</td>
<td>UPS</td>
<td>Mars Bars</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>AB1</td>
<td>08-Jun-97</td>
<td>12-Jun-97</td>
<td>UPS</td>
<td>Snickers</td>
<td>400</td>
</tr>
<tr>
<td>1</td>
<td>AB1</td>
<td>08-Jun-97</td>
<td>12-Jun-97</td>
<td>UPS</td>
<td>Baby Ruth</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>CC500</td>
<td>05-Jul-97</td>
<td>16-Jul-97</td>
<td>Fed Ex</td>
<td>M&amp;M's</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>CC500</td>
<td>05-Jul-97</td>
<td>16-Jul-97</td>
<td>Fed Ex</td>
<td>Wonka Bars</td>
<td>200</td>
</tr>
</tbody>
</table>

In the incorrect example, an orders table keeps track of orders. In order number 1, three different items were ordered, in order number 2, two different items were ordered. The gray area shows the information that was duplicated. This information should have been broken up into two separate tables.

**CORRECT**

**Order # | Customer ID | Order Date | Ship Date | Ship Method | Item        | Quantity**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB1</td>
<td>08-Jun-97</td>
<td>12-Jun-97</td>
<td>UPS</td>
<td>Mars Bars</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>CC500</td>
<td>05-Jul-97</td>
<td>16-Jul-97</td>
<td>Fed Ex</td>
<td>M&amp;M's</td>
<td>1000</td>
</tr>
</tbody>
</table>

**Order # | Item        | Quantity**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mars Bars</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>Snickers</td>
<td>400</td>
</tr>
<tr>
<td>1</td>
<td>Baby Ruth</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>M&amp;M's</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>Wonka Bars</td>
<td>200</td>
</tr>
</tbody>
</table>

In the correct example, the data has been broken into two separate tables. The first table contains the shipping information and the second table contains the items which were purchased. The two tables can be collected through their common field **Order #**. For data input, a form can be created that enters the information into both tables from one screen.
CREATE A NEW DATABASE

There are several methods of creating a new table including using wizards which will guide you; however, we will use "Design View" to create our table. We are using this method because you will learn more about how to structure a table and set its properties. Further, if you did you the wizard to create a table and wanted to modify its structure, you would still have to use Design View.

1. Click on the Access icon from the Start menu: Microsoft Access

2. Select Blank Database

3. Select the Save in location and enter a file name.

4. Click on "Create" (The Database window opens.)

When you create and save Tables, Queries, Forms, Reports, Pages, Macros, and Modules, they are saved as Objects beneath the tab that pertains to what they are.
CREATING A TABLE

Since queries, forms, & reports must be based upon a table, a table must be the first object you create in your database.

Creating the Employees Table

We will create a table which tracks the names and addresses of our employees.

1. In the Database window, click on the Tables tab.
2. Click [Create table in Design view]

Design View

"Design View" is where the structure of the table is defined. This includes creating field names, and setting data types and properties. The user must set two things: Field Name and Data Type. Descriptions are optional. Additionally, there are General Properties which are optional for each data type.

Field Name:
The Field Name is that which appears at the top of a column. It may include letters, numbers, and some special characters such as the # sign or the underscore _. Field names can be 64-characters in length.

Description:
The Description is optional. If you enter a description, when data is being entered into the field, the description entered will appear in the lower left hand corner of the screen. Descriptions are useful for giving the data entry person tips or instructions on how to enter data.
Data Type:
The Data Type determines what type of data can be typed into the field (letters, numbers, dates, etc.), and what can be done with the field (linking, sorting, calculations.)
The most commonly used data types are as follows:

Text: Text fields can contain any type of keyboard character. The maximum amount of characters allowable is 255. Text fields can be searched, sorted and linked. You cannot perform calculations on numbers entered in a text field. Candidates for text fields are: names, addresses, zip codes, phone numbers, product names, etc.

Memo: Memo can contain any type of keyboard character. The maximum allowable size of a memo field is 65,535 characters. Memo fields can be searched but they cannot be sorted, linked, or used in calculations. Memo fields should only be used for such items as: comments, remarks, descriptions, etc.

Number: Number fields can only contain numbers. Numbers in number fields can be used in calculations. The maximum size of a number field depends upon the Field Size selected in the Properties section of “Design View”.

Currency: Currency fields can only contain numbers. Numbers in currency fields can be used in calculations. Currency fields can contain up to 15 digits to the left of the decimal and 4 digits to the right.

Date/Time: Date/Time fields can contain either a date or a time. Dates and times can be used in calculations. Examples of date/time fields are Hire Date, Date of Birth, Order Date, Time In, etc.

Autonumber: Autonumber fields are counter fields that place numbers in the field for the user automatically. Note that if you delete a record, the field will not renumber.

Yes/No: Yes/No fields offer one of two possibilities. Examples of Yes/No fields are: Patient is allergic to penicillin (yes or no), and This is a current employee (yes or no).

OLE Object: OLE Object fields are for holding nonstandard items such as Excel spreadsheets, pictures, and sounds. The table can hold these objects but the objects will only display in forms or reports.

Hyperlink: Hyperlink fields contain web address. If the web address is activated, the computer’s browser will start and the user will be taken to the address specified.

Lookup Wizard: This is not actually a field type but a method of creating a drop down list in a field.
3. Create the table shown below.
   - Type in the Field Name and Description yourself.
   - When entering the Data Types, click in the data type column and then click the down arrow to get a list of possible field types. Or, press the initial letter of the data type to jump to that data type. For example, pressing “T” will select the “Text” data type.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN</td>
<td>Text</td>
<td>Access will enter the dashes for you</td>
</tr>
<tr>
<td>First Name</td>
<td>Text</td>
<td>Use Initial Caps and then lower case letters</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
<td>Use Initial Caps and then lower case letters</td>
</tr>
<tr>
<td>Address</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Zip Code</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Home Phone</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Salary</td>
<td>Currency</td>
<td></td>
</tr>
<tr>
<td>Number of Dependents</td>
<td>Number</td>
<td></td>
</tr>
<tr>
<td>Hire Date</td>
<td>Date/Time</td>
<td></td>
</tr>
<tr>
<td>Web Address</td>
<td>Hyperlink</td>
<td></td>
</tr>
<tr>
<td>E-Mail Address</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Bonded</td>
<td>Yes/No</td>
<td></td>
</tr>
<tr>
<td>Remarks</td>
<td>Memo</td>
<td></td>
</tr>
<tr>
<td>Photo</td>
<td>OLE Object</td>
<td></td>
</tr>
</tbody>
</table>

**Setting A Primary Key**

The purpose of a Primary Key is to prevent record duplication and/or to create a relationship where one table controls the contents of another table. Tables do not have to have a primary key. Multiple primary keys are possible. The table’s contents and any relationships it may have with other tables will determine if it should be keyed or not.

As a general rule, key the field (column) in your table that you know will contain unique values and assign it as a primary key. Fields that are good candidates for primary keys are fields containing: Social Security Numbers, Employee ID numbers, Customer ID’s, Product Codes, etc. Fields that would be bad candidates for primary keys would be first or last name because you may have someone with the same first or last name.

The properties of a field that has been assigned as a primary key are:
- **Unique** (True): duplicates are not allowed in the field. (i.e. you can’t have two different records with the same social security number if the Primary Key was assigned to the SSN field.)
- **Ignore Nulls** (False): You cannot save the record until data is entered into the keyed field.

Additionally:
- The table will automatically be sorted by the primary key (although you can easily resort by another field)
- The table can be related to another table using the primary key to control how data is entered.
1. Click on the **SSN** row.
2. Click the “**Primary Key**” button. (Note that a key appears next to the field name.
3. Click on the “**Save**” icon.
4. Name the table “**Employees**” and click “Ok”

**More On Primary Keys**

**Removing a Primary Key**

A. Click on the field containing the primary key.
B. Click on the “**Primary Key**” icon on the toolbar:

**Assign Multiple Primary Keys**

Sometimes, one field alone does not contain unique values but when paired with another field, the combination of the two will produce unique values.

In the Table below, **Order ID** alone is a bad candidate for primary key because people order more than one item in the same order and therefore, the same Order ID will be listed more than once. **Candy ID** is a bad candidate as well because, the same candy will most likely be ordered by someone else.

However, the combination of **Order ID** and **Candy ID** as primary keys will produce unique records. In a single order, the same Candy ID is not ordered more than once.

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Candy ID</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>88</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>500</td>
</tr>
</tbody>
</table>

**To Assign Multiple Primary Keys:**

A. Be in “**Design View**” of the table to have its keys set.
B. Click on the gray box to the left of one of the fields that is to be part of the set.
C. Hold down the “**CONTROL**” key on the keyboard.
D. Click on the gray box to the left of the other field that is to be part of the set.
E. Click on the “**Primary Key**” icon on the toolbar.
Creating an Input Mask

An Input Mask controls two aspects of a field:
- It enters characters for you (the dashes in a social security number for example.)
- It forces the user to type the exact number of required characters, no more, no less.

1. Click on the SSN field.
2. Click in the Input Mask property.
3. Click on the input mask wizard button: 
4. Select: Social Security Number and click “Next”.
5. Click “Next” again to not change the input mask.
6. Select the option: With the symbols in the mask.
7. Click “Finish”.
8. Save and close the table.

Tracking The Employee’s Children

In our fictional company, we like to send our employee’s children cards on their birthdays; therefore, we need to keep track of the children’s names and birthdays. We will relate the “Children” table to the “Employees” table through the parent’s SSN number; however, note that under certain circumstances, this can create the two potential problems listed below.

Problem of Employing Both Parents
Note that this presents a potential problem if both of a child’s parents work for our company. Conceivable, the same child could be entered into the table more than once. For this reason, we will need to assign an ID number to each child. In our example we will assign each child a social security number. The structure of such a relationship is shown to the right. The two tables are connected via one of the parent’s SSN numbers. This type of relationship is known as a “One-to-Many” because the parent’s SSN number can only exist once in the “Employees” table but can exist many times in the “Children” table. (They may have more than one child.)
Problem of Relating Both Parents
Under most circumstances, the relationship above works fine but what if for some reason you wanted to connect both parents that work for your company to the children? Further, most children do not have social security numbers. This problem requires not only creating a Child ID field, but also creating a third table. The window to the right shows how such a relationship would look. Each child is entered into the “Children” table once and each employee is entered into the “Employees” table once. The “Parent-Child” table (which connects the two) tracks which children belong to which parents. See the example below.

<table>
<thead>
<tr>
<th>Child ID</th>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Marsha</td>
<td>Brady</td>
</tr>
<tr>
<td>2</td>
<td>Jan</td>
<td>Brady</td>
</tr>
<tr>
<td>3</td>
<td>Cindy</td>
<td>Brady</td>
</tr>
<tr>
<td>4</td>
<td>Laurie</td>
<td>Partridge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>Child ID</th>
<th>SSN</th>
<th>Child ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>121-21-2121</td>
<td>1</td>
<td>121-21-2121</td>
<td>1</td>
</tr>
<tr>
<td>121-21-2121</td>
<td>2</td>
<td>787-87-8787</td>
<td>1</td>
</tr>
<tr>
<td>121-21-2121</td>
<td>3</td>
<td>787-87-8787</td>
<td>2</td>
</tr>
<tr>
<td>787-87-8787</td>
<td>1</td>
<td>787-87-8787</td>
<td>2</td>
</tr>
<tr>
<td>787-87-8787</td>
<td>3</td>
<td>565-65-6565</td>
<td>4</td>
</tr>
</tbody>
</table>

Creating the Children Table
To make matters simple, we will assume that we do not hire the spouses of current employees and create a two table structure

1. In the Database window, click on the Tables tab.
2. Double-click Create table in Design view
3. Create the table shown below.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child ID</td>
<td>AutoNumber</td>
</tr>
<tr>
<td>SSN</td>
<td>Text</td>
</tr>
<tr>
<td>First Name</td>
<td>Text</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
</tr>
<tr>
<td>Gender</td>
<td>Text</td>
</tr>
<tr>
<td>DOB</td>
<td>Date/Time</td>
</tr>
</tbody>
</table>
Set the Primary Key
1. Click on the Child ID field.

2. Click the “Primary Key” button.

5. Click on the “Save” icon.
6. Name the table “Children” and click “Ok”.

Create the Input Mask
Our two tables are connected through their common field of SSN. The properties of these two fields must match; therefore, because the SSN field in the “Employees” table uses an input mask, the “Children” table must as well.

1. Click on the SSN field.
2. Click in the Input Mask property.
3. Click on the input mask wizard button:
4. Select: Social Security Number and click “Next”.
5. Click “Next” again to not change the input mask.
6. Select the option: With the symbols in the mask.
7. Click “Finish”.
8. Click on the “Save” icon.
9. Close the table.
TABLE RELATIONS

One of the problems inherent in placing data into separate tables has to do with keeping the data synchronized, input and editing. Having to open two or more tables to perform data entry and editing can be tedious and prone to mistakes if the appropriate changes are not made to all tables involved. Fortunately, these problems can be solved with the use of “Table Relations”. Not only does relating separate tables allow one table to dictate what can be entered into the other table, but a Main/Subform can be created where data can be entered into two tables from one screen. (Main/Subforms are covered in the Access: Forms workshop.)

Conditions for Relating Tables

To relate tables, the conditions below must exist. Note that table relations should be set-up prior to any data entered into any of the tables involved.

Master/Subordinate: When two or more tables are related, there must be at least one “Master” table. When set up to its fullest extent, the master table dictates what can be typed into the other tables which are known as subordinates. (See Referential Integrity below).

Primary Keyed Field: The tables are linked through their common fields. Further, the “Master” table’s linking field must be a primary key field. The subordinate table’s linking fields do not have to be primary keys.

Same Data Type: The Data Type for linking fields must the same or similar. The following field types are compatible:
- Text to Text
- Number to Number
- AutoNumber to Number (Field size on the number field must be set to “Long Integer”)

Relating Employees & Children

We will relate the “Employee” (Master) table to the “Children” (subordinate) table through their common field of SSN.

1. Close any open tables.
2. From the toolbar, click on the “Relationships” icon:

3. Click on the “Show Table” icon on the toolbar:
4. Select the table: “Employees” and click on “Add”.
5. Select the table: “Children” and click on “Add”.
6. Close the Show Table window.

The tables are related by using the mouse to connect the field which is common to both of the tables. The direction the mouse is dragged determines which table is the master and which is the subordinate. Always drag from the master table to the subordinate table.
7. Click on SSN in the “Employees” table & drag it onto SSN in the “Children” table, then release.
8. Select the Following: Enforce Referential Integrity
   Cascade Update Related Fields
   Cascade Delete Related Fields
9. Click on “Create”.
10. Save and close the Relationships window.

---

Enforce Referential Integrity:
When checked, two or more tables can be related in such a way that data cannot be typed into the subordinate table without first being typed into the master table. For example, we would not be able to enter an SSN in the “Children” table unless we first place the same SSN value in the “Employees” table. (This option must be check to enabled the other two options)

Cascade Delete Related Tables:
If checked, when a record is deleted from the master table, all related records in the subordinate table are also deleted. For example, if a parent is deleted in the “Employees” table then all of their children will delete from the “Children” table.

Cascade Update Related Tables:
If checked, when the linking field in the master table is edited, all of the matching linking fields in the subordinate table are changed as well. For example, if you were to change a parent’s SSN in the “Employees” table, their corresponding SSN in the “Children” table would change as well.

The parent’s SSN exists only once in the “Employees” table but may exist many times in the “Children” table because they may have more than one child.
NAVIGATING TABLES

This section covers how to move through fields and rows in a table.
1. Click on the Table tab.
2. Double click the Employees table to open it.

Keyboard Navigation

<table>
<thead>
<tr>
<th>Keystroke</th>
<th>Cursor Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up/Down Arrows</td>
<td>Moves one record up or down.</td>
</tr>
<tr>
<td>Left/Right Arrows</td>
<td>Moves one field left or right.</td>
</tr>
<tr>
<td>HOME</td>
<td>Move to the far left of a record.</td>
</tr>
<tr>
<td>END</td>
<td>Moves to the far right of a record.</td>
</tr>
<tr>
<td>CONTROL + HOME</td>
<td>Moves to the top of the table.</td>
</tr>
<tr>
<td>CONTROL + END</td>
<td>Moves to the bottom of the table</td>
</tr>
</tbody>
</table>

Mouse Navigation

Mouse navigation keys are in the lower left hand corner of the screen.

View Related Tables

Because relationships were created where Employees has a subordinate table, clicking the plus sign to the left of a record will display corresponding records from the subordinate table.
ENTERING DATA

Data can be entered into a table in several different ways:
- It can be entered directly into the table.
- It can be entered into a table through use of a form.
- It can be “Appended” from another table using an “Action” query.

**Tips on Data Entry**

<table>
<thead>
<tr>
<th>Action</th>
<th>Key(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy the field above</td>
<td>CONTROL '</td>
</tr>
<tr>
<td>Insert the current date</td>
<td>CONTROL ;</td>
</tr>
<tr>
<td>Insert the current time</td>
<td>CONTROL :</td>
</tr>
<tr>
<td>Zoom in on a field</td>
<td>SHIFT F2</td>
</tr>
<tr>
<td>Toggle a YES/NO field</td>
<td>SPACEBAR</td>
</tr>
<tr>
<td>Toggle between highlighted cell and cursor</td>
<td>F2</td>
</tr>
<tr>
<td>Saving before completing a record</td>
<td>SHIFT ENTER</td>
</tr>
</tbody>
</table>

**Entering Dates**

In date fields, there is a break point at 1929. If all four digits are not used when typing in the year, 12/31/29 and less will default to 20xx, i.e. 12/31/2029, and 1/1/30 and above will default to 19xx, i.e. 1/1/1930.

**Entering the Data**

1. Click on the “Add Record” button on the toolbar:
2. Begin entering the employee’s record.

<table>
<thead>
<tr>
<th>Web Address</th>
<th>E-Mail Address</th>
<th>Bonded</th>
<th>Remarks</th>
<th>Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.usc.edu">http://www.usc.edu</a></td>
<td><a href="mailto:tommy@usc.edu">tommy@usc.edu</a></td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Web Address** is a hyperlink field that if clicked, will launch a web page.

**Bonded** is a YES/NO field. Press the “SPACE” key to toggle between checked and unchecked.

**Remarks:**
Press “F2” to zoom into any field.

**Photo** - To insert a photo file:
1. Right mouse click the field.
2. Select: Insert Object
3. Select: Create From File
4. Click “Browse” to find the file.
5. Once the file is located, click “Ok” then “Ok” again.

Note that the photo will only display in a form or report.
SAVING DATA
Access automatically saves your data as you type. When you press enter in the last column of a record, the record is saved. If you are not going to type in the last few fields and do not wish to keep pressing enter to move through the fields to reach the last field, or if you edit a record, press “SHIFT” + “ENTER” on your keyboard to save the record.

EDITING CELLS
Data editing can be performed from within tables, forms, or queries.

- Press “F2” to toggle the cursor between highlighting the entire field to a single character cursor.
- After editing a cell, press “SHIFT” + “ENTER” to save your edit (or move the cursor to another row.)

DELETING RECORDS
Records can be deleted from within tables, forms or queries. Note that you can not undo a delete.
- To delete a record, click on the gray box to the left of the record and press the “DELETE” key on the keyboard. (or Right-Click and choose “Delete Record”)

ENTERING DATA INTO A SUBORDINATE TABLE
If the tables have been related, rather than opening the subordinate table to enter data into it, you can perform data entry into it from its master table.

1. From within the “Employees” table, click the plus sign next to the parent whose children you wish to enter.
2. Enter the child data using the same techniques you did in the parent table. Note that the Employee ID is placed into the child table for you.
3. Click the minus sign to collapse the child table.

<table>
<thead>
<tr>
<th>SSN</th>
<th>First Name</th>
<th>Last Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>ZipCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ 121-21-2121</td>
<td>Carol</td>
<td>Brady</td>
<td>500 South St</td>
<td>Long Beach</td>
<td>CA</td>
<td>90909</td>
</tr>
<tr>
<td>+ 565-65-6565</td>
<td>Shirley</td>
<td>Partridge</td>
<td>300 Morning St</td>
<td>Los Angeles</td>
<td>CA</td>
<td>90805</td>
</tr>
<tr>
<td>- 686-86-8666</td>
<td>Samantha</td>
<td>Stevens</td>
<td>300 Morning St</td>
<td>New York</td>
<td>NY</td>
<td>90847</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Date of Birth</th>
<th>Gender</th>
<th>Address</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabitha</td>
<td>Stevens</td>
<td>5/7/1999</td>
<td>Female</td>
<td>300 Morning St</td>
<td>New York</td>
</tr>
<tr>
<td>+ 787-87-8767</td>
<td>Mike</td>
<td>Brady</td>
<td>500 South St</td>
<td>Long Beach</td>
<td>CA</td>
</tr>
</tbody>
</table>

* *
TABLE FORMATTING

The actions below change how a table displays data but do not change the actual structure of the table. Structural changes should be done in “Table Design View”.

Adjusting Column Widths
To adjust the display width of a column (not the number of characters it can contain), place the cursor in the groove between the field names and click and drag the mouse to the left or the right. Double click a column to make it automatically find the best fit.

Sort A Column
1. Click anywhere in the column to be sorted.
2. Click one of the two sort icons on the toolbar.

Change Column Order
To change the order columns are in:
1. Click on the field name of the column to be moved and let go of the mouse button.
2. Click and drag the field name to the left or right.

Freeze Columns
This is useful when a table has a lot of columns and you wish a particular column to always remain visible on the screen while scrolling left and right.
1. Click in the column to be frozen.
2. From the menu, click on: FORMAT - FREEZE COLUMNS

The column will be moved to the left side of the screen. When you scroll left and right, it will remain locked there until you unfreeze columns.

Unfreeze Columns
1. From the menu, click on: FORMAT - UNFREEZE ALL COLUMNS
The column will remain on the left side.

Hide Columns
Hiding a column is useful when you don’t want the column to print or if it contains sensitive data.
1. Click in the column to be hidden.
2. From the menu, click on: FORMAT - HIDE COLUMNS

Unhide Columns
1. From the menu, click on: FORMAT - UNHIDE COLUMNS
2. Checking any blank box will display that particular field.
3. When complete, click on “Close”.

Access_2002-Tables.doc  April 22, 2003  Page 23 of 33
Control Gridlines and Cell Effects

These options allow you to display/suppress gridlines, change cell and gridline color, and select different cell effects.

1. From the menu, click on: **FORMAT - DATASHEET**
2. Select the desired options in the window below and click on “Ok”.

Data Entry Mode

Data Entry Mode allows you to add new records to a table but does not display any previous records in the table. This is useful if you do not want to accidentally erase any old records or do not wish the data entry person to see any of the previous data.

1. Open the table.
2. From the menu, click on: **RECORD - DATA ENTRY**
3. Type your data as usual. To see all records, close the table and reopen it.
EDIT TABLE STRUCTURE

This section discusses how to safely redesign a table once data has been placed in it.

1. From the *Database* window, select the table you wish to modify.
2. Click on the ![Design](Design.png) icon

### Adding New Fields - Safe

Adding a new field to your table is completely safe. You cannot harm existing data. To add a new field at the bottom of the table:

1. Click below the last field & type in a new field name and specify a type.

### Inserting a Row - Safe

Inserting a row is safe and is usually performed when a field is to be inserted between two existing fields.

1. Click in the field below where the new row is to be.
2. Click on the ![Insert Rows](InsertRows.png) icon on the toolbar:
   (Don’t confuse it with the ![Delete Rows](DeleteRows.png) icon to its right.)

### Deleting Fields - Unsafe

Deleting a field will permanently delete all data stored in its column (assuming you save the table). For example, if you delete the field called *Last Name*, then you will lose all of your employee’s last names. To delete a field:

1. Click on the gray box to the left of the field’s name (the row will turn black).
2. Press the **DELETE** key on your keyboard (or use the **Delete Rows** ![Delete Rows](DeleteRows.png) icon).
3. Click on **Yes** at the warning message to delete, **No** to cancel the deletion.
   (Note that none of your changes will occur if you do not save the table.)

### Renaming Fields - No Longer Unsafe

In previous versions of Access, if you had any saved queries, forms, or reports that used a field in a calculation, changing the field’s name in table design view would cause the calculation to error; however, this is no longer the case with Access XP (or 2000).

1. To rename a field, simply retype the field name.
**Moving Fields - Safe**
Changing the order of fields is completely safe. To change field order:
1. Click on the gray box to the left of the fieldname and let go of the mouse.
2. Click and drag the gray button up or down.

**Changing the Data Type - Possibly Unsafe**
The data type controls that type of characters that are allowed in a field and what can be done with the data; therefore, changing the type may have severe consequences on your data. For example, if you change from a “Currency”, “Number”, or “Date” field to a “Text” field, you will no longer be able to do calculations on numbers. You should also be aware that if someone has created Relationships based on that field, you may be destroying the link between the tables.

**Changing the Field Description - Safe**
Field descriptions are optional and have no effect on how Access functions.

**Changing Field Format - Safe**
Changing a field’s format simply changes how Access displays data on the screen. The data itself is not affected.

**Changing Field Size - Possibly Unsafe**
With “Text” and “Numeric” fields this is dangerous when the field size is decreased because there is a risk of truncating or deleting data that is too large to fit the smaller sized field. Further, if the field is related to a field in another table, Access will not allow you to change the size as long as the relationship still exists. You must first remove the relationship, change the field size, and then add the relationship back in (if desired). Field size is covered in the next section.
SET FIELD PROPERTIES

When in design view of a table, the bottom section of the screen contains the field property settings you can use to further control the data entered into your table. Always click on the field name you are modifying prior to going to the properties area. The Properties area reflects only the currently selected field.

1. In the Database window, select the “Employees” table
2. Click on the “Design” button to go into design view.
3. Click on Home Phone.

**General:** This contains all of the properties for the selected field.

**Lookup:** Allows you to create dropdown boxes in a table that list

The Properties section reflects the properties of the field currently selected above. Therefore, always click on the field name prior to going to the Properties section.
Field Size
You can set the Field Size for Text fields and Number fields. Both have to do with how many characters you can enter into the field and how much space the entry will consume on the drive.

TEXT FIELDS
The maximum size of a text field is 255 characters. However, even if you type only 4 characters in a field, it will take the space of 255 characters on your drive. The Field Size property reduces this from 255 to whatever number the user types in. For example, if you enter 25, the maximum number of characters you can type into a field is 25 and no matter what you type, the space of 25 characters will be taken up on your drive.
- Make the Field Size just large enough for your largest entry.
- If you make it too small, existing data will be truncated.

<table>
<thead>
<tr>
<th>Code</th>
<th>Result in Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>Up to 255 characters; no decimal places; 1 byte storage space.</td>
</tr>
<tr>
<td>Integer</td>
<td>-32,768 to 32,768; no decimal places; 2 bytes storage space.</td>
</tr>
<tr>
<td>Long Integer</td>
<td>-2,147,483,648 to 2,147,438,648; n decimal places; 4 bytes S.S.</td>
</tr>
<tr>
<td>Single</td>
<td>-3.4x10^-3 to 3.4x10^-3; 7 decimal places; 4 bytes storage space.</td>
</tr>
<tr>
<td>Double</td>
<td>-1.797x10^308 to 1.797x10^308; 15 decimal places; 8 bytes storage space.</td>
</tr>
<tr>
<td>Replication ID</td>
<td>If you will be sharing portions of one database.</td>
</tr>
</tbody>
</table>

NUMBER FIELDS

Formats
Format controls how the data will be displayed in the table. The choices vary depending upon the field type:

Text Fields

<table>
<thead>
<tr>
<th>Code</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Will display characters in all UPPERCASE.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Will display characters in all lowercase.</td>
</tr>
<tr>
<td>@[blue]</td>
<td>Will display characters in blue ([cyan], [red], [magenta], [yellow], [green],[white],[black] are other choices.</td>
</tr>
</tbody>
</table>

Number Fields & Currency Fields

<table>
<thead>
<tr>
<th>Code</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Numbers and decimal place. No commas or $ sign.</td>
</tr>
<tr>
<td>Currency</td>
<td>$, commas, and 2 places to the right of the decimal.</td>
</tr>
<tr>
<td>Fixed</td>
<td>2 places to the right of the decimal.</td>
</tr>
<tr>
<td>Standard</td>
<td>commas, and 2 places to the right of the decimal.</td>
</tr>
<tr>
<td>Percent</td>
<td>Moves the cursor two places to the right and adds a % sign.</td>
</tr>
<tr>
<td>Scientific</td>
<td>Displays the number in scientific notation.</td>
</tr>
<tr>
<td>#,##0.00;<a href="#,##0.00">RED</a></td>
<td>Displays negative numbers in red with () around them.</td>
</tr>
</tbody>
</table>

Date/Time Fields

<table>
<thead>
<tr>
<th>Code</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Date</td>
<td>6/19/94  5:34:06 PM</td>
</tr>
<tr>
<td>Long Date</td>
<td>Sunday, June 19, 1994</td>
</tr>
<tr>
<td>Medium Date</td>
<td>19-Jun-94</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Short Date</td>
<td>6/19/94</td>
</tr>
<tr>
<td>Long Time</td>
<td>5:34:06 PM</td>
</tr>
<tr>
<td>Medium Time</td>
<td>5:34 PM</td>
</tr>
<tr>
<td>Short Time</td>
<td>17:34</td>
</tr>
</tbody>
</table>

**Yes/No Fields**

<table>
<thead>
<tr>
<th>Code</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/No</td>
<td></td>
</tr>
<tr>
<td>True/False</td>
<td></td>
</tr>
<tr>
<td>On/Off</td>
<td></td>
</tr>
</tbody>
</table>

**Caption**

A *Caption* provides a way of changing the display name of a column without changing the actual name. To give a column heading a caption, merely type it in the *Caption* box.

**Default Value**

The purpose of a *Default Value* is to save time. A default value is data that will appear in a new field automatically. For example, if you set the default value of a field called *Country* to “United States”, then all new records will already have the text “United States” in the *Country* field. If the country was something other than the default value, the data entry person could merely overtype the default.

**Date Fields**

<table>
<thead>
<tr>
<th>Example</th>
<th>Effect in Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date()</td>
<td>Inserts the current date.</td>
</tr>
<tr>
<td>Date()+7</td>
<td>Inserts a date seven days after of the current date.</td>
</tr>
<tr>
<td>#1/1/97#</td>
<td>Inserts the date 1/1/97.</td>
</tr>
</tbody>
</table>

**Text Fields**

<table>
<thead>
<tr>
<th>Example</th>
<th>Effect in Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>“United States”</td>
<td>Inserts the words United States in the field.</td>
</tr>
</tbody>
</table>

**Number & Currency Fields**

<table>
<thead>
<tr>
<th>Example</th>
<th>Effect in Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Inserts the number 10 in the field.</td>
</tr>
</tbody>
</table>

**Yes/No Fields**

<table>
<thead>
<tr>
<th>Example</th>
<th>Effect in Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The default value is NO.</td>
</tr>
<tr>
<td>-1</td>
<td>The default value is YES</td>
</tr>
</tbody>
</table>
Validation Rule

Though the field type controls whether letters, numbers, or dates can be typed into a cell, a Validation Rule controls what range of letters, numbers, or dates, can be entered. For example, a validation rule can specify that the Cost Per Unit must be between 5 and 60 or that Expiration Dates must be 1 year greater than the current date.

<table>
<thead>
<tr>
<th>Example</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>“USA” or “Canada” or “UK”</td>
<td>Only USA, Canada, or UK will be accepted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;date()</td>
<td>Entries must be greater than today’s date.</td>
</tr>
<tr>
<td>&gt;date()+30</td>
<td>Entries must be 30 greater than today’s date.</td>
</tr>
<tr>
<td>&gt;=#1/1/95# and &lt;=#1/1/2000#</td>
<td>Entries must be between 1/1/95 and 1/1/2000 inclusively.</td>
</tr>
<tr>
<td>&gt;#5/1/97#</td>
<td>Entries must be greater than 5/1/97</td>
</tr>
<tr>
<td>Between #1/1/95# and #1/1/2000#</td>
<td>Entries must be between 1/1/95 and 1/1/2000 inclusively.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>RESULT IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=5</td>
<td>Entries must be greater or equal to five.</td>
</tr>
<tr>
<td>&lt;=10</td>
<td>Entries must be less than or equal to 10.</td>
</tr>
<tr>
<td>&gt;5 and &lt;10</td>
<td>Entries must be greater than 5 and less than 10.</td>
</tr>
<tr>
<td>Between 5 and 10</td>
<td>Entries must be greater or equal to 5 and less than or equal to 10.</td>
</tr>
<tr>
<td>&gt;=5 and &lt;=10</td>
<td>Entries must be greater or equal to 5 and less than or equal to 10.</td>
</tr>
<tr>
<td>10</td>
<td>Entries must be 10.</td>
</tr>
</tbody>
</table>

When you save your table, Access will ask if you wish to test existing data against the new validation rules. Click “Yes” to check it, click “No” to have the validation rule only affect new records or old fields your are editing.

Violating the Rule

Access will not allow you to leave a field which contains data that violates a validation rule. You have two choices:

- Enter data that does not violate the rule.
- Press “ESCAPE” several times on the keyboard to leave the field unchanged.

Validation Text

This is optional and can be used with your Validation Rule. It is basically the error message that will appear on the user’s screen if they violate the validation rule. For example, if you type: “You are an idiot! Enter a date after today’s date”. That message will appear when they violate your rule.
**Required**
Your choices are either “Yes” or “No”. YES means that the user must enter something into the field before the record can be saved. NO means that the user can leave the field blank.

**Allow Zero Length**
When the Required property for a field is set to yes, data must be typed into the field before the record can be saved. However, double quotes “” could be typed into the field to satisfy the YES requirement while making it still appear to be blank. If you do not want people to be able to enter “”, set Allow Zero Length to: No

**Indexed**
If you index a field, Access will be able to find data within it more quickly when you run a query or the Find command. You should index fields that you query often.

Types of Indexes are:
- Indexed - Duplicates OK
- Indexed - No Duplicates

**Unicode Compression**
Starting with Access 2000, text, memo, and hyperlink fields use UNICODE compression. UNICODE uses two bytes instead of one byte to represent each character. To offset the extra space used, Access can compress the data in these types of fields. By default, UNICODE compression is set to “Yes” to save on disk space.

DROP DOWN BOXES (LOOKUPS)

The options under the Lookup tab in the Properties area allow you to create drop down choices in your table. Lookups are useful when there are a limited number of choices to enter into a field and you wish to ensure that the data entry person selects one of them. Lookups also ensure correct spelling. Lookups are only available in Text, Number, and Yes/No fields. The data displayed with the user clicks a lookup field can be coming from another table or entered into the properties of the lookup. Further, lookups can be created manually or by using a wizard.

Combo Box: Value List (Manual Creation)

We will create a Combo Box (drop down list) that displays choices that were typed into a Value List. The choices on our list will be: “Los Angeles”, “San Diego”, and “San Francisco”

1. Click the Tables tab and select the table Employees.
2. Click on the “Design” button .
3. Click on the City field.
4. Click on the Lookup tab in the Properties area.
5. Set Display Control to: Combo Box
6. Set Row Source Type to: Value List
7. In Row Source, enter the choices in quotes and separated by semicolons: “Los Angeles”;“San Diego”;“Sacramento”
8. Set Limit to List to: “Yes”
9. Save the table and go to “Data View” by clicking the “View” button to test the combo box.

Combo Box: Referencing a Table (Manual Creation)

This example references a table called Cities which lists the names of cities. To work properly, the table containing the cities should list each city only once and in alphabetical order.

Note that if you place more than one column in a reference table you should create queries for each lookup. This would allow for alphabetical listing and prevention of duplicates.

Note that if your reference table/query has multiple columns, you will need to adjust the Bound Column, Column Count, and Column Widths properties in your lookup to display data from the correct column.

The example to the right is referencing a table called Cities.
Using the Wizard to Create a Combo Box - Value List

The wizards can also be used to create a lookup. We will place the items “Boy” and “Girl” on a combo box for the Gender field in the “Children” table.

1. Open the “Children” table in Design view.
2. Click on the Gender field.
3. Change the data type to: “Lookup Wizard…”. (The wizard should start.)
4. Select: I will type the values that I want and click on “Next”.
5. Type “Boy” in the first row then press the down arrow on your keyboard.
6. Type “Girl” in the second row then press the down arrow on your keyboard.
7. Click on “Next”.
8. Label the field: Gender then click “Finish”.

The code is shown to the right.

Using the Wizard to Create a Combo Box - Table Reference

We will use the wizards to reference the City field in the “Cities” table and place the data on a drop down box in the City field of the “Children” table.

1. Open the “Children” table in “Design View”.
2. Click on the City field and change its data type to: “Lookup Wizard…”.
3. Select: I want the lookup column to lookup values in a table or query
4. Click “Next”.
5. Select the “Cities” table then click “Next”.
6. Double click the City field then click “Next”.
7. Adjust the column width as desired then click “Next”.
8. Label the column: City then click “Finish”.

The code is shown to the right.