

Querying Your Tables

Queries allow you to answer questions about your data, to extract specific information from tables, and to change selected data in various ways. In fact, the ability to perform queries is a key reason for using database management programs—rather than spreadsheets or word processing programs—to manage large amounts of related data. In this chapter you'll learn how to use the Query Wizards and the graphical query by example (QBE) tools in Access to query your data with relative ease.

As you read through this chapter, keep in mind that many techniques for customizing datasheet view and creating filters apply to queries as well. So if you haven't worked through Chapter 9, you should do so before diving into queries. You also should understand the basic table design concepts (Chapter 6) and know how to edit in datasheet view (Chapter 8).

What Queries Let You Do

Queries let you see the data you want, in the order you want it. They also allow you to perform calculations on your data; to create sources of data for forms, reports, charts, and other queries; to make global changes to tables; and to create new tables.

**TIP**

If the main reason you're creating a query is to use it as a source of records for a multitable form or report, you may not need to set up a query at all. Often, the Form Wizard or Report Wizard offers a faster way to simultaneously design the form or report and create an SQL statement that defines the record source for you. Chapters 11 and 12 describe these versatile Wizards in more detail. Of course, if you have many records, a form or report will operate faster if it uses a saved query (rather than an SQL statement) as its record source.

When you run most types of queries or apply a filter, Access collects the data you ask for in a *dynaset*. Although the dynaset looks and acts like a table, it's actually a dynamic or "live" view of one or more tables. Therefore, changes that you make after running a query usually will affect the underlying tables in the database itself.

**NOTE**

A dynaset is an updatable type of *recordset*, which is any collection of records that you can treat as an object. Some types of queries—such as those that create cross-tabs and other summaries—generate recordsets in which some or all fields are not updatable.

Types of Queries

You can create several types of queries as summarized below:

Select Query The most commonly used query type, a Select query lets you select records, create new calculated fields, and summarize your data. Select queries are similar to the filters discussed in Chapter 9.

However, unlike filters, Select queries also let you:

- Query more than one table
- Create new calculated fields
- Summarize and group your data
- Choose which fields to show or hide

Crosstab Query Groups data into categories and displays values in a spreadsheet-like format with summary totals. You can use Crosstab queries to compare values and see trends in your data, to view summary data such as monthly, quarterly, or yearly sales figures, and to answer questions such as *Who has ordered how many of what?* Crosstabs are especially useful as the basis for reports and charts.

Make Table Query Creates a new table from a dynaset. Use Make Table queries to create a backup copy of a table, save a history table of old records you'll be deleting from another table, select a subset of data to report on, or create a table for exporting to other applications.

Update Query Lets you make global changes to data in one or more tables. These queries offer a powerful, fast, and consistent way to change many records in one fell swoop. For example, you can use an Update query to increase the price of all tether ball items by 25 percent, or to empty out certain fields.

Append Query Adds records from one or more tables to the end of an existing table. Append queries are especially useful for adding old records to the end of a history table. You can then convert the Append query to a Delete query (described next) and delete the old records from the original table.

Delete Query Deletes a group of records from one or more tables. For instance, you can delete all customer records for people who haven't bought anything in five years or more. Or, after appending data to a history table, you can use a Delete query to remove the old records from the original table.

Pass-Through Query Strictly for SQL mavens, Pass-Through queries send commands directly to a SQL database server using the syntax required by that server. For information on this topic, search for *Queries, Pass-Through Queries* in the Access Help Index.

Data-Definition Query Also for SQL mavens, data-definition queries use SQL language statements to create or change database objects in the current database. To learn about this topic, search for *Queries, Data-Definition Queries* in the Access Help Index.

Union Query Another one for SQL mavens, Union queries use SQL language statements to combine corresponding fields from two or more tables or queries into one field. See *Queries, Union Queries* in the Access Help Index for details.

Access offers two ways to create queries: the Query Wizards and the "from scratch" methods. We'll look at the Query Wizards first.

Using Query Wizards

Query Wizards offer a quick and easy way to perform the special-purpose queries listed below:

Simple Query Wizard Creates a Select query for one or more tables. The resulting query can do simple selection, or it can calculate sums, averages, counts, and other types of totals.

Crosstab Query Wizard Creates a crosstab for a single table or query.

Find Duplicates Query Wizard Finds duplicate records in a single table or query.

Find Unmatched Query Wizard Finds records in one table that have no related records in another. For example, you can use this Wizard to create queries that find customers who haven't placed orders or to find employees who haven't sold orders to any customers.

Often the Query Wizards will be able to set up the perfect query for your needs. But even if the query that the Wizard creates is only "almost good enough," using the Wizards can save you time in the initial query design. You can always switch to design view (discussed soon) and tweak the Wizard-generated query as needed.

To use the Query Wizards:

1. Click on the drop-down arrow next to the New Object toolbar button (the second-to-last button on the toolbar) from any window in an open database and then choose Query. Or click on the Queries tab from the database window and then click on the New button, or choose Insert ► Query.
2. Double-click on one of the options described above (Simple Query Wizard, Crosstab Query Wizard, Find Duplicates Query Wizard, or Find Unmatched Query Wizard) in the New Query dialog box.
3. Follow the Wizard's instructions and complete the dialog boxes that appear. (See Chapter 3 for information about basic techniques you can use with Access Wizards.)

In a flash, you'll have a new query that you can either use as is or customize as needed.

Creating, Running, Saving, and Changing a Query

Now let's look at the basic steps for creating queries from scratch and for running, saving, and changing queries. Later in the chapter you'll learn how to refine the basics to set up any query you want.

Creating a Query from Scratch

The Query Wizards can create the specialized queries discussed in the previous section. To create other types of queries, however, you'll need to start from scratch. Here are the steps to follow:

1. Click on the drop-down arrow next to the New Object toolbar button (second-to-last button on the toolbar) from any window in an open database, and then choose Query. Or click on the Queries tab from the database window and then click on the New button, or choose Insert ► Query.

2. Double-click on Design View in the New Query dialog box. A *query design window* named Query1 : Select Query opens and the Show Table dialog box appears (see Figure 10.1).



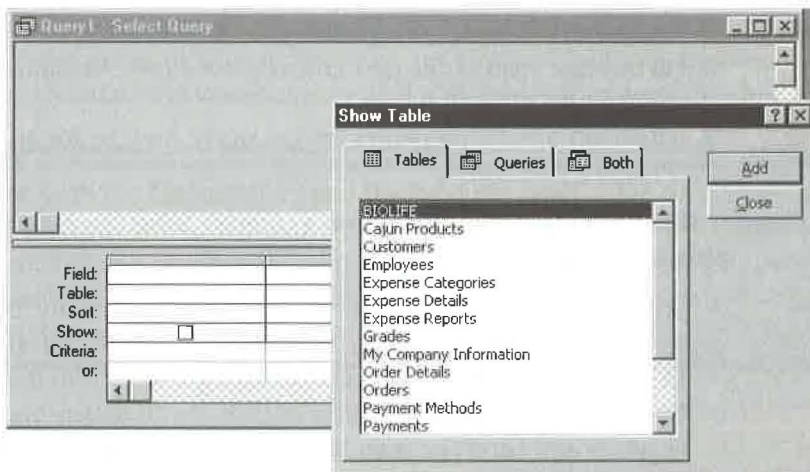
If you already know which table you want to query, go to the database window, click on the Tables tab, click on the table you want to create the new query from, click on the drop-down arrow next to the New Object toolbar button, click on Query, and then double-click on Design View. The Show Table dialog box won't appear when you follow these steps, but you can display it (if necessary) by choosing Query ► Show Table.

3. Use any of these techniques to add the tables you want to query:

- **To choose which objects to list in the Show Table dialog box**, click on the Tables, Queries, or Both tabs at the top of the Show Table dialog box.
- **To add an object to the query design window**, double-click on the object you want to add, or highlight the object and then click on the Add button.
- **To add multiple adjacent objects to the query design window**, click on the first object you want to select; then Shift-click on the last object you want (or drag your mouse through the object names). Click on Add.
- **To add multiple nonadjacent objects to the query design window**, click on the first object you want to select; then hold down the Ctrl key and click on each remaining object. Click on Add.

FIGURE 10.1

The Select Query window and Show Table dialog box appear when you create a new query from scratch.



4. Repeat step 3 as needed. When you're done, click on Close. Join lines will appear automatically if you created relationships between tables (Chapter 6) or if Access can figure out the relationships on its own. You also can join the tables manually, as described later in this chapter.
5. Click on the drop-down list next to the Query Type toolbar button (shown below) to pick the type of query you want to set up or choose options from the Query menu. Your options are Select (the default and most commonly used query type), Crosstab, Make Table, Update, Append, and Delete. (See "About the Query Design Window's Toolbar" for details about the toolbar buttons.)



To convert one type of query to another, simply choose the appropriate type from the Query Type toolbar button or the Query menu.

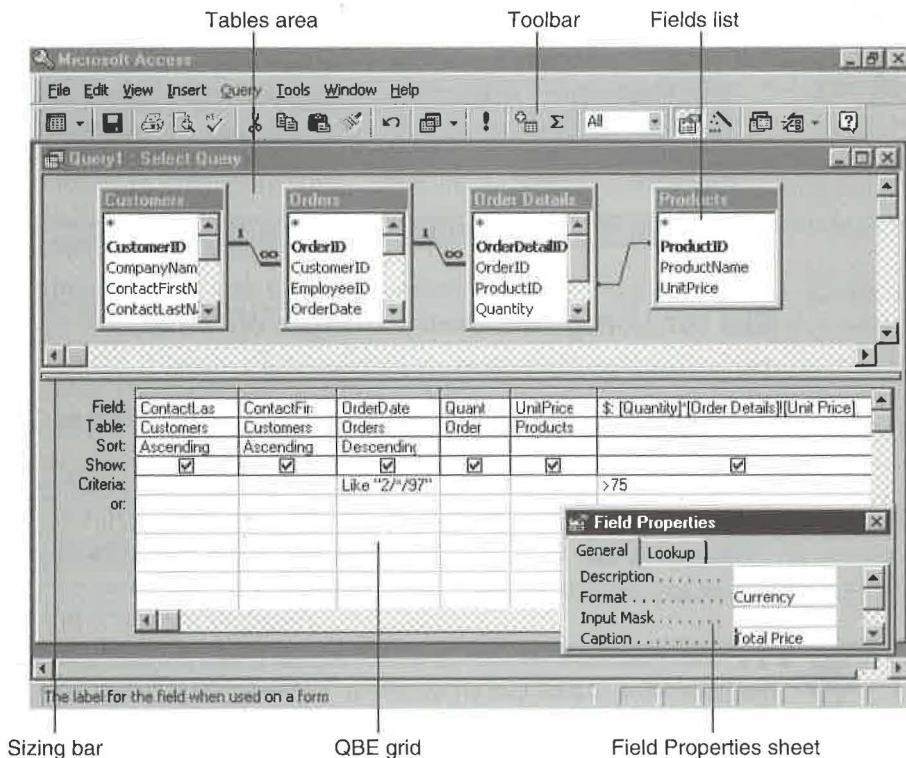
6. Double-click on fields that you want to display in the dynaset or use in selection criteria in the tables area. Or drag the fields from the tables area to the QBE grid. Or select the fields from the drop-down list in the Field row of the QBE grid.
7. Specify any selection criteria that you want to use for isolating records in the dynaset in the Criteria rows under the appropriate column in the QBE grid. The techniques are similar to those for designing a filter (see Chapter 9).
8. Fill in other areas of the QBE grid as needed (see "Refining Your Query" later in the chapter).
9. (Optional) Specify properties for the query itself or for an individual field. See "Refining Your Query."

The example in Figure 10.2 shows a complex Select query that's based on several related tables. This query will

- Select records in which the order was placed in February 1997 and the quantity times the unit price ($[\text{Quantity}] * [\text{Order Details}]![\text{Unit Price}]$) is greater than \$75.
- Display the ContactLastName and ContactFirstName fields from the Customers table (see the Field and Table rows in Figure 10.2), the OrderDate from the Orders table, the Quantity and UnitPrice from the Order Details table, a calculated field named \$,

FIGURE 10.2

A Select query to find orders for a specified date and dollar amount of items.



and the ProductName field from the Products table. In the dynaset, the \$ field will have the column heading Total Price and will appear in Currency format (see the Field Properties sheet in the figure).

- Sort the results by ContactLastName and ContactFirstName (in ascending order) and OrderDate (in descending order) for multiple orders from the same customer.

NOTE

Because both the Order Details and the Products tables have a UnitPrice field, we had to explicitly tell Access to use the UnitPrice field in the Order Details table when multiplying the Quantity times the Unit Price in the query. To specify the table name and field name, use the format [TableName].[FieldName], as in [Order Details].[UnitPrice].

Figure 10.3 shows the dynaset produced by this query.

FIGURE 10.3

The dynaset produced by the query in Figure 10.2.

Field Properties shown in Figure 10.2 control this field's caption and format.

Contact Last N	Contact First N	Order Date	Quantity	Unit Price	Total Price	Product Name
Schurnack	Susita	2/16/97	2	\$127.45	\$254.90	Billiard balls
Stumingi	Wilbur	2/2/97	2	\$45.55	\$91.10	Crystal ball

Tips for Using the Query Design Window

Here are some tips and shortcuts for designing a query:

- **To clear all the items from the QBE grid**, choose **Edit ► Clear Grid**. Caution: You cannot undo Clear Grid.
- **To expand the editing area for an input cell in the QBE grid**, click in the cell you want to expand and then press **Shift+F2**. Or right-click on the cell and choose **Zoom**. In the Zoom Box that appears, edit the text as needed and then click on **OK** to close the box.

TIP The Zoom Box is available anywhere an expanded input area might be handy, including property sheets, a table's design or datasheet view, the QBE grid, the filter window, and the macro window.

- **To see which table each field in the QBE grid is from or to hide that information**, choose **View ► Table Names** or right-click on the QBE grid and choose **Table Names**. When you select (check) the **View ► Table Names** option or the **Table Names** shortcut menu option, a **Table** row will appear in the QBE grid (see Figure 10.2). When you deselect the option, this row is hidden.
- **To change a field name in the QBE grid**, click to the left of the field name's first letter in the QBE grid and then enter the new name followed by a colon. We used this technique to change the original name of an expression field to **\$** in Figure 10.2. The new field name will be used in the dynaset unless you've specified a **Caption** in the **Field Properties** sheet (see Figures 10.2 and 10.3).



- **To add tables to the query design window**, click on the Show Table toolbar button (shown at left); or right-click on an empty place in the tables area and choose Show Table; or choose Query ► Show Table; or open the database window (F11) and then choose Window ► Tile Vertically so that you can see the database and query design windows. Next click on the database window's Tables tab and then drag the table name from the database window to the tables area of the query design window.
- **To remove a table from the query design window**, click on the table and then press the Delete key. (This step doesn't remove the table from the database, only from the query design window.)
- **When a query gets too complicated**, consider using two queries. Create, test, and save the first query. Then, with the first query open in the query design or datasheet window, click on the drop-down arrow next to the New Object toolbar button, choose Query, and then double-click on Design View. Now design, test, and save your second query (which will be based on the first one). In the future, you can just run the second query to get the results you want.

For help with designing queries, look up any of the subtopics under *Queries* in the Access Help Index. Or go to the Help Contents, open the *Working With Queries* book, open the *Creating A Query* book, and explore subtopics there (a great jumping off point is the *Create A Query* subtopic).

Viewing the Dynaset

You can view the dynaset anytime:



- Click on the Query View toolbar button (shown at left) or choose View ► Datasheet View.

TIP

If you need to cancel a query while it's running, press Ctrl+Break. (This cancel procedure may not work for very large queries.)

The dynaset will appear in Datasheet view (see Figure 10.3). From here you can do any of the following:

- **Edit records in the dynaset.** Your changes will update the underlying tables.
- **Customize the dynaset's appearance.** Use the same techniques you use to customize a datasheet (see Chapter 8). You can save your layout changes when you save the query.
- **Sort the dynaset.** Use the same techniques you use to sort a table (see Chapter 9).

- **Filter the dynaset.** Use the same techniques you use to filter a table (see Chapter 9).
- **Preview and print the dynaset as for any other table.** To preview the data, click on the Print Preview toolbar button or choose File ► Print Preview. When you're ready to print the data, click on the Print toolbar button or choose File ► Print. Chapter 9 offers more details on printing.
- **Return to the query design window.** To do so, click on the Query View toolbar button (shown at left) or choose View ► Design View. Your query will reappear in the query design window exactly as you left it.



Viewing the SQL Statement behind a Query

If you're familiar with SQL (or you'd like to become better acquainted), you can view or edit your query in SQL view. To switch to SQL view, open the query in design or datasheet view. Then click on the drop-down arrow next to the Query View toolbar button and choose SQL View; or choose View ► SQL View. View or change the SQL statement as needed and then click on the Query View toolbar button to see the results in datasheet view.



WARNING

Structured Query Language (SQL) is used behind the scenes in queries and other Access objects. However, it's not for the faint of heart or the inexperienced database user. Be careful when changing SQL statements; if you make mistakes, your queries and other objects may not work properly. For more about SQL, look up *SQL Statements, Using In Queries* in the Access Help Index.

Running an Action Query

If you've designed an action query (Make Table, Update, Append, or Delete query), switching to datasheet view will let you see which records will be affected when you *run* the query. But no changes will be made to any tables, and no new tables will be created. Previewing in datasheet view gives you a chance to look before you leap.

When you're sure the dynaset shows exactly which records the action should act on, you can run the query this way:



1. Return to the query design window and click on the Run toolbar button (shown at left), or choose Query ► Run.
2. Respond to any prompts from the action query to confirm your changes. (There's more about action queries later in the chapter.)

**TIP**

If the action query's confirmation messages annoy you, or if you're not getting confirmation messages, choose **Tools** ► **Options** and then click on the **Edit/Find** tab. To make sure confirmation messages appear (the *safest* choice), check the **Action Queries** box under **Confirm**. To suppress those messages, clear the **Action Queries** box. Click on **OK** to close the **Options** dialog box.

The action query will do its job, and you'll be returned to the query design window.

**NOTE**

Running a **Select** query or **Crosstab** query is the same as viewing the query's datasheet.

For help with viewing and running action queries, look up subtopics under *Action Queries* in the Access Help Index.

Saving Your Query

To save a new or changed query, use any of these techniques:

- **If you're in the query or datasheet window**, click on the **Save** toolbar button (shown at left) or choose **File** ► **Save** or press **Ctrl+S**.
- **If you're done with the query for now**, close the query design window or datasheet view (for example, choose **File** ► **Close** or press **Ctrl+W**). When asked if you want to save your changes, click on **Yes**.

If this query is new, you'll be prompted for a query name. Type a standard Access name (up to 64 characters including blank spaces) and then click on **OK**. Note that Access won't let you save the query with the same name as an existing table or query.

When you save a query, Access saves *only* the query design, not the resulting record set. That way, the query will operate on whatever data is in your tables at the time you run the query.

Opening a Saved Query

To reopen a saved query:

1. Start in the database window and click on the **Queries** tab.



2. Do one of the following:


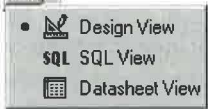



- **To view the query's dynaset or run an action query**, double-click on the query name, or highlight the query name and then click on Open.
- **To open the query's design window**, highlight the query name and then click on Design.

After opening the query, you can switch between datasheet view and design view by clicking on the Query View toolbar button.

About the Query Design Window's Toolbar











As you saw in Figure 10.2, the query design window's toolbar contains many buttons. Proceeding from left to right on the toolbar, Table 10.1 briefly describes what each one does. (In Table 10.1 we've omitted the standard Copy, Cut, and Paste buttons, which appear on most Access toolbars, and the Print, Print Preview, Spelling, and Format Painter buttons, which are dimmed and unavailable.)

TABLE 10.1: THE QUERY WINDOW'S MOST IMPORTANT TOOLBAR BUTTONS

BUTTON	BUTTON NAME	WHAT IT DOES
	Query View (Design View)	Switches to the query window, where you can design or change your query.
	Query View (SQL View)	Switches to SQL view, where you can use SQL statements to design or change your query.
	Query View (Datasheet View)	Switches to datasheet view, where you can view the dynaset. If you're designing an action query, use this button to preview which records the query will affect.
	Save	Saves your latest changes to the query.
	Undo	Undoes your most recent change to a criterion entry (when you are still in that field).















Continued ▶

TABLE 10.1: THE QUERY WINDOW'S MOST IMPORTANT TOOLBAR BUTTONS (CONTINUED)

BUTTON	BUTTON NAME	WHAT IT DOES
	Query Type (Select Query)	Displays the QBE grid for a Select query.
	Query Type (Crosstab Query)	Displays the QBE grid for a Crosstab query.
	Query Type (Make- Table Query)	Displays the QBE grid for a Make-Table query.
	Query Type (Update Query)	Displays the QBE grid for an Update query.
	Query Type (Append Query)	Displays the QBE grid for an Append query.
	Query Type (Delete Query)	Displays the QBE grid for a Delete query.
	Run	Runs an action query. For Select and Crosstab queries, this button has the same effect as the Query View (Datasheet View) button.
	Show Table	Lets you add more tables to the tables area of the query window.
	Totals	Displays a Total row in the QBE grid. Use this row to specify how data will be grouped and summarized.
	Top Values	Lets you choose whether to return a specified number of records, a percentage of records, or all values. Access uses the left-most sorted field to choose which top values to display. You can choose a setting from the Top Values button's drop-down list, or type a number (e.g., 25) or a percentage (e.g., 47%) into the Top Values combo box.

Continued

TABLE 10.1: THE QUERY WINDOW'S MOST IMPORTANT TOOLBAR BUTTONS (CONTINUED)

BUTTON	BUTTON NAME	WHAT IT DOES
	Properties	Opens the property sheet, where you can change field or query properties.
	Build	Opens the Expression Builder, which makes it easier to enter complicated expressions.
	Database Window	Opens the database window.
 <ul style="list-style-type: none">  AutoForm  AutoReport  Table  Query  Form  Report  Macro  Module  Class Module 	New Object (AutoForm)	Creates a new automatic form that's based on the current query.
	New Object (AutoReport)	Creates a new automatic report that's based on the current query.
	New Object (Table)	Lets you create a new table. The table is not based on the current query.
	New Object (Query)	Lets you create a new query that's based on the current query.
	New Object (Form)	Lets you create a new form that's based on the current query.
	New Object (Report)	Lets you create a new report that's based on the current query.
	New Object (Macro)	Lets you create a new macro. The macro is not based on the current query.
	New Object (Module)	Lets you create a new module. The module is not based on the current query.
	New Object (Class Module)	Lets you create a new class module not based on a form or report.
	Office Assistant	Opens the Office Assistant.

Refining Your Query

In the next few sections, we'll explain how to refine your queries.

Filling In the QBE Grid

Filling in the QBE grid is perhaps the trickiest part of designing a query. Figure 10.2 shows a typical Select query and its QBE grid. The rows that appear in the grid depend on the type of query you're designing. For example, a Delete query will not have a Sort row.

Here are some techniques you can use to fill the rows in the QBE grid for a Select query. You'll find more examples in later sections.

- **In the Field row**, enter the fields you want to work with. Use double-clicking or dragging to add fields from the fields list (see Chapter 9). Or to tell Access to display all fields even if the table structure changes, use the asterisk (*) field name (see "Using the Asterisk in a QBE Grid" below). Or double-click on the table name to select all fields and then drag any field to the Field row (all fields will come along for the ride). Or use the drop-down list in the Field row to select the field you want. You also can create new calculated fields in the Field row (see "Totals, Averages, and Other Calculations").
- **In the Sort row**, choose the sort order you want for each field. Your options are Ascending, Descending, or Not Sorted. Like filters (Chapter 9), query fields are sorted in left-to-right order, so you may need to reposition columns in the grid.

Using the Asterisk in a QBE Grid















You can use the asterisk in a QBE grid to tell Access that you want all fields to appear in the dynaset. The resulting dynaset will *always* contain all the fields in the table, even if you add or delete fields in the table structure later.

To add the asterisk to a Field column in the QBE grid, double-click on the asterisk (*) at the top of the appropriate fields list or drag the asterisk to the grid. The Field column will show the table name followed by a period and the asterisk, like this:

Customers.*

If you use the asterisk, you still can sort on, select, group by, calculate with, and do other operations with a specific field. To do so, add the appropriate field to the QBE grid. Then to prevent that field from appearing twice in the dynaset, deselect the field's Show box.

TABLE 10.1: THE QUERY WINDOW'S MOST IMPORTANT TOOLBAR BUTTONS (CONTINUED)

BUTTON	BUTTON NAME	WHAT IT DOES
	Properties	Opens the property sheet, where you can change field or query properties.
	Build	Opens the Expression Builder, which makes it easier to enter complicated expressions.
	Database Window	Opens the database window.
 <ul style="list-style-type: none">  AutoForm  AutoReport  Table  Query  Form  Report  Macro  Module  Class Module 	New Object (AutoForm)	Creates a new automatic form that's based on the current query.
	New Object (AutoReport)	Creates a new automatic report that's based on the current query.
	New Object (Table)	Lets you create a new table. The table is not based on the current query.
	New Object (Query)	Lets you create a new query that's based on the current query.
	New Object (Form)	Lets you create a new form that's based on the current query.
	New Object (Report)	Lets you create a new report that's based on the current query.
	New Object (Macro)	Lets you create a new macro. The macro is not based on the current query.
	New Object (Module)	Lets you create a new module. The module is not based on the current query.
	New Object (Class Module)	Lets you create a new class module not based on a form or report.
	Office Assistant	Opens the Office Assistant.

Refining Your Query

In the next few sections, we'll explain how to refine your queries.

Filling In the QBE Grid

Filling in the QBE grid is perhaps the trickiest part of designing a query. Figure 10.2 shows a typical Select query and its QBE grid. The rows that appear in the grid depend on the type of query you're designing. For example, a Delete query will not have a Sort row.

Here are some techniques you can use to fill the rows in the QBE grid for a Select query. You'll find more examples in later sections.

- **In the Field row**, enter the fields you want to work with. Use double-clicking or dragging to add fields from the fields list (see Chapter 9). Or to tell Access to display all fields even if the table structure changes, use the asterisk (*) field name (see "Using the Asterisk in a QBE Grid" below). Or double-click on the table name to select all fields and then drag any field to the Field row (all fields will come along for the ride). Or use the drop-down list in the Field row to select the field you want. You also can create new calculated fields in the Field row (see "Totals, Averages, and Other Calculations").
- **In the Sort row**, choose the sort order you want for each field. Your options are Ascending, Descending, or Not Sorted. Like filters (Chapter 9), query fields are sorted in left-to-right order, so you may need to reposition columns in the grid.

Using the Asterisk in a QBE Grid

You can use the asterisk in a QBE grid to tell Access that you want all fields to appear in the dynaset. The resulting dynaset will *always* contain all the fields in the table, even if you add or delete fields in the table structure later.

To add the asterisk to a Field column in the QBE grid, double-click on the asterisk (*) at the top of the appropriate fields list or drag the asterisk to the grid. The Field column will show the table name followed by a period and the asterisk, like this:

Customers.*

If you use the asterisk, you still can sort on, select, group by, calculate with, and do other operations with a specific field. To do so, add the appropriate field to the QBE grid. Then to prevent that field from appearing twice in the dynaset, deselect the field's Show box.

TIP When you design Make Table, Delete, or Append queries, consider using the asterisk so that the query always operates on the current structure of the underlying table, and the operation includes all the fields.

Changing Field Properties

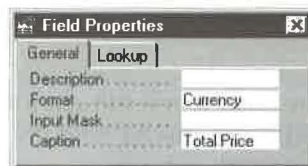
Field properties govern the appearance of fields in the dynaset. By default, fields added to the QBE grid will inherit properties of the underlying table. However, calculated fields, such as the field `[:Quantity] * [Order Details]![UnitPrice]` shown back in Figure 10.2, do not inherit properties. To assign or change field properties:

1. Click in the field you want to change. (If the Show box is checked, you won't be able to view or set field properties.)
2. Click on the Properties toolbar button (shown at left) if the Field Properties sheet isn't visible. Or right-click in the field and choose Properties. Or choose View ► Properties from the menus.
3. Click on the tab that contains the properties you want to change and then change the properties as needed.



To change field properties in another field, simply click in that field and change the properties as needed. When you're done using the Field Properties sheet, click on the Properties button again or click on the Close button at the upper-right corner of the Field Properties sheet.

Shown below is the Field Properties dialog box for the field that multiplies the quantity ordered field (`[Quantity]`) by the unit price field (`[Order Details]![UnitPrice]`) back in Figure 10.2. Here we've changed the format to Currency and set the caption to Total Price. When Access displays the dynaset, this calculated field's numbers will appear as dollar amounts (for example, \$35.00) and the column heading will be Total Price (refer back to Figure 10.3).



NOTE

If you don't specify a field's Caption property, Access will use the field name that's assigned in the QBE grid as the column heading in the dynaset, unless the field has a caption in the underlying table.

To find out what a property does, click in a property box on the property sheet and then press F1, or press Shift+F1 (or click on the Help toolbar button) and then click on the property you're interested in.

Changing Query Properties

You also can change properties of the entire query. The steps are the same as for changing field properties, except that in step 1 you click on an empty place in the tables area at the top of the query design window. The resulting Query Properties sheet looks like this for a Select query:



Viewing Unique Values

When you query a single table, Access normally shows all records that match what you're looking for. Sometimes, however, you'll want to eliminate duplicates by changing the Unique Values property in the Query Properties sheet from No to Yes.

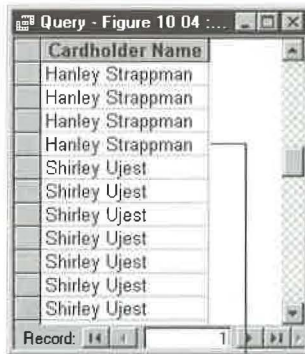
Figure 10.4 illustrates this point. In order to see which credit card numbers are represented in the Payments table, we've queried to show nonempty CardHoldersName fields only. Notice that all records, including duplicate values, appear on the left side of

FIGURE 10.4

The Unique Values property controls whether duplicate records appear in a single-table query.



Query shows nonblank CardHoldersName fields.



A normal query on one table includes duplicate values.



Setting the Unique Values property to Yes hides duplicates.

the figure. To get rid of those duplicates (see right side of the figure), we changed the query's Unique Values property to Yes.

Here are some tips for working with unique values:

- Make sure the QBE grid includes only the field or combination of fields you want to view unique values in. For instance, to see unique credit card numbers, include only the CreditCardNumber field. To see unique combinations of credit card name and credit card number (for example, to show customers who paid with more than one credit card), include both the CardHoldersName and CreditCard-Number fields.
- All property settings affect the current query design window. To see duplicate records again, change Unique Values back to No.
- Use a Totals query if you want to count duplicate values in your data. See "Totals, Averages, and Other Calculations" later in the chapter.

NOTE

When you query multiple related tables, Access normally returns the unique records (that is, the Unique Records property is Yes), but not the unique values. To see the duplicate records in a multitable query, change the query's Unique Records and Unique Values properties to No. To see only the unique records and values in a multitable query, change the Unique Values properties to Yes (the Unique Records property changes to No automatically).

Displaying Top Values

If you want Access to return only the top n records in the list, or the top n percent of the records, you can change the Top Values property in the Query Properties sheet or the Top Values combo box on the toolbar. Suppose you want to display the top 10 payments in the Payments table. The query shown in Figure 10.5 will return the first 10 records, plus any records in which all the fields match the values in the 10th record. Note that the field used to display the Top n values must be the leftmost Sort field in the QBE grid, and that you must sort that field in Descending order if you want to see the top-most values. (To see the bottom n values, change the Sort order to Ascending.) Figure 10.6 shows the results of the query in Figure 10.5.

TIP

To return the highest or lowest values without displaying any duplicate records, set the Unique Values property in the Query Properties sheet to Yes.

FIGURE 10.5

A query for displaying the top 10 values.

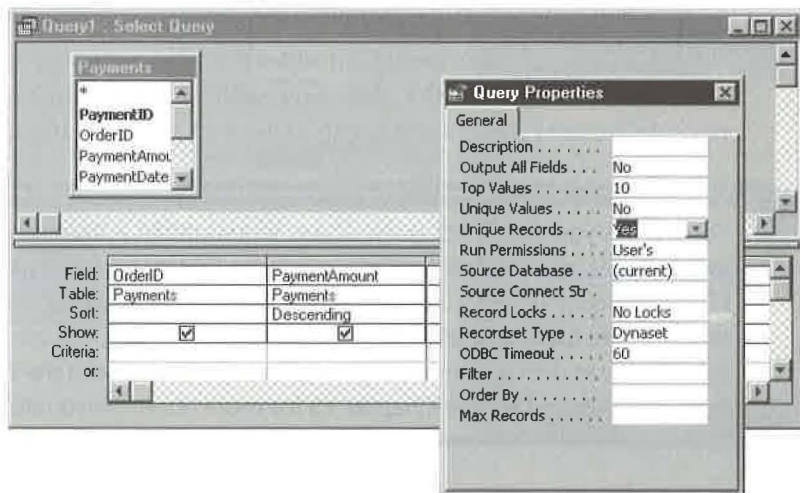


FIGURE 10.6

The results of showing top 10 values.

Order ID	Payment Amount
13	\$110.00
3	\$100.00
2	\$60.00
1	\$35.00
11	\$30.00
23	\$25.00
8	\$25.00
17	\$24.00
22	\$12.00
7	\$12.00

Joining Tables

You can join tables to limit the records in your dynaset or to display data from several related tables at once. For example, if you join the Customers and Orders tables, the dynaset will show only related records (typically, customers who have unfilled orders in the Orders table). Similarly, if you join the Employees and Orders tables, the dynaset will show only the employees associated with orders in the Orders table.

Access will join tables for you automatically when you add them to the query design window:

- If you've established a relationship between those tables (see Chapter 6).
- If tables include fields that have the same name and the same (or compatible) data type. For example, because the Customers and Orders tables both have a numeric field named CustomerID, Access can join them automatically.

Join lines appear between joined tables in the query design window to show you what kind of relationship the tables have, which fields are joined, and whether the relationship enforces referential integrity (refer back to Figure 10.2).

TIP

Sometimes you might not want Access to join tables automatically unless you've purposefully set up relationships between those tables via the Relationships window (Tools ► Relationships). To prevent automatic joins based on a "best guess" about which tables and fields are related, choose Tools ► Options and click on the Tables/Queries tab in the Options dialog box. Then deselect (clear) the Enable AutoJoin checkbox and click on OK. See Chapter 15 for more about personalizing Access.

You can join tables in three ways—inner join (also called equi-join), outer join, or self-join—as described in the following sections.

Inner Joins

In an *inner join*, records in the joined tables must have the same values for fields that are joined. This type of join answers questions such as, Which customers have placed new orders? You saw an inner join in Figures 10.2 and 10.3. When Access joins tables automatically on the basis of field names and data types, it always creates an inner join.

Outer Joins

In an *outer join*, all records from one table are added to the dynaset even if no matching values occur in the joined field from the other table. Records from the second table are combined with those from the first table *only* if matches occur in the joined field. The two types of outer joins are left outer joins and right outer joins.

Using the Employees and Orders example, a *left outer join* will show all employees and any orders they've sold (order information for some employees will be blank), as shown in Figure 10.7.

A *right outer join* will show all order numbers and the employees who sold them (employees who didn't sell orders won't appear in the dynaset), as shown in Figure 10.8.

FIGURE 10.7

A left outer join of the Employees and Orders table, along with the resulting dynaset.

This left outer join query...

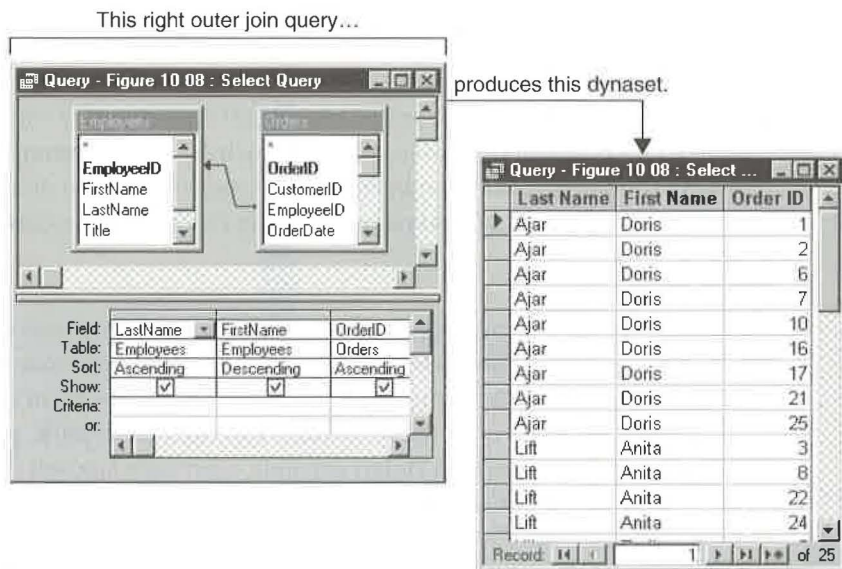
produces this dynaset.

Last Name	First Name	Order ID
Ajar	Doris	1
Ajar	Doris	2
Ajar	Doris	6
Ajar	Doris	7
Ajar	Doris	10
Ajar	Doris	16
Ajar	Doris	17
Ajar	Doris	21
Ajar	Doris	25
Clozure	Lane	
Honcho	Ed	
Lift	Anita	3
Lift	Anita	8

Record: 1 of 29

FIGURE 10.8

A right outer join of the *Employees* and *Orders* table, along with the resulting dynaset.



Self-Joins

In a *self-join*, the table is joined to itself. The self-join can be handy when one field within the table refers to another field within the same table. Consider, for example, the revised *Employee* table (named *Employees With Supervisors*) shown below.

Employee ID	First Name	Last Name	Title	SupervisorID	Wo
1	Anita	Lift	Sales Representative	4	(206)
2	Andrew	Van Noslo	Vice President, Sales	9	(206)
3	Emily	Niktog	Sales Representative	2	(206)
4	Albert	Niktog	Sales Manager	2	(206)
5	Doris	Ajar	Sales Representative	4	(206)
6	Lane	Clozure	Office Manager	9	(206)
7	Joseph	Swipple	Vice President, Marketing	9	(206)
8	Helen	Wheels	Product Evangelist	9	(206)
9	Ed	Honcho	President	9	(206)
*	(AutoNumber)			0	

Figure 10.9 shows a query that produces a list of supervisors and the employees they manage. To clarify the meaning of each column in the dynaset, we assigned captions to each field. Looking at the dynaset, it's easy to see this company's pecking order: President Ed Honcho manages Lane Clozure, Joseph Swipple, Andrew Van Noslo, and Helen

FIGURE 10.9

This self-join query on an Employee table shows supervisors and their subordinates. Although not obvious from this example, the table on the left is named *Employees With Supervisors*; the self-join table on the right is named *Employees With Supervisors_1*.

This self-join query...

produces this dynaset.

Field:	LastName	FirstName	Title	LastName	FirstName	Title
Table:	Employees	Employee1	Empl	Employees	Employee	Empl
Sort:	Ascending	Ascending		Ascending	Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:						
or:						

	This Boss	>>>	>>>	Manages	<<<	<<<
▶	Honcho	Ed	President	Clozure	Lane	Office Manager
	Honcho	Ed	President	Honcho	Ed	President
	Honcho	Ed	President	Swipple	Joseph	Vice President, Marketing
	Honcho	Ed	President	Van Noslo	Andrew	Vice President, Sales
	Honcho	Ed	President	Wheels	Helen	Product Evangelist
	Niktog	Albert	Sales Manager	Ajar	Doris	Sales Representative
	Niktog	Albert	Sales Manager	Lift	Anita	Sales Representative
	Van Noslo	Andrew	Vice President, Sales	Niktog	Albert	Sales Manager
	Van Noslo	Andrew	Vice President, Sales	Niktog	Emily	Sales Representative

Wheels; Sales Manager Albert Niktog manages Doris Ajar and Anita Lift; Vice President (Sales) Andrew Van Noslo manages Albert Niktog and Emily Niktog.

To create a self-join, add the same table to the query design window twice (Access will add an underscore followed by a number to the second table name, as in *Employees With Supervisors_1*). Then create an inner join between appropriate fields in the two tables (see the next section). In Figure 10.9, for example, we joined the *EmployeeID* field from the *Employees With Supervisors* table on the left to the *SupervisorID* field from the *Employees With Supervisors_1* table on the right.

Defining an Inner Join

To define any type of join, you first define an inner join:

1. Start in the query design window and add any tables you want to join. If join lines appear automatically, you're done.
2. Drag the field name from one table onto a field name containing the same kind of data in another table to join tables manually. Usually you'll want to drag a primary key (which appears in bold).
3. Release the mouse button.

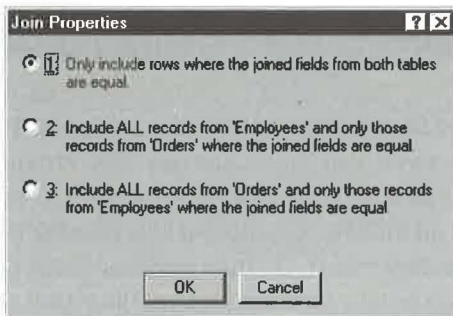
Inner join lines will connect the joined fields in each table as shown here:



Changing the Join Type

To change the type of join between two tables:

1. Right-click on the join line you want to change and choose Join Properties. Or double-click on the thin part of the join line. You'll see the Join Properties dialog box:



NOTE

If your mouse isn't on the thin part of the join line—or the join line isn't bold when you double-click on it—the Query Properties sheet may appear instead of the Join Properties dialog box. If this happens to you, close the Query Properties sheet and try again.

1. Choose the option you want in the Join Properties dialog box. Option 1 produces an inner join. Option 2 creates a left outer join. Option 3 makes a right outer join.
2. Click on OK. Access will update the appearance of the join line to reflect your choice in step 2.

Figures 10.7 and 10.8 show left and right outer join lines, respectively.

**TIP**

If you find yourself creating manual joins often, you can establish a permanent relationship between the tables. To do so, return to the database window and choose Tools ► Relationships, or click on the Relationships toolbar button; then define the type of join you want (see Chapter 6).

Deleting a Join

Deleting a join is easy:

1. Click on the thin part of the join line you want to delete so that it appears bold.
2. Press the Delete key.

**WARNING**

If the join line isn't bold when you press Delete, you may delete a table from the query design window. If that happens, you'll need to add the table (Query ► Show Table) and its fields again.

For help with joining tables, look up *Joins* and its subtopics in the Access Help Index.

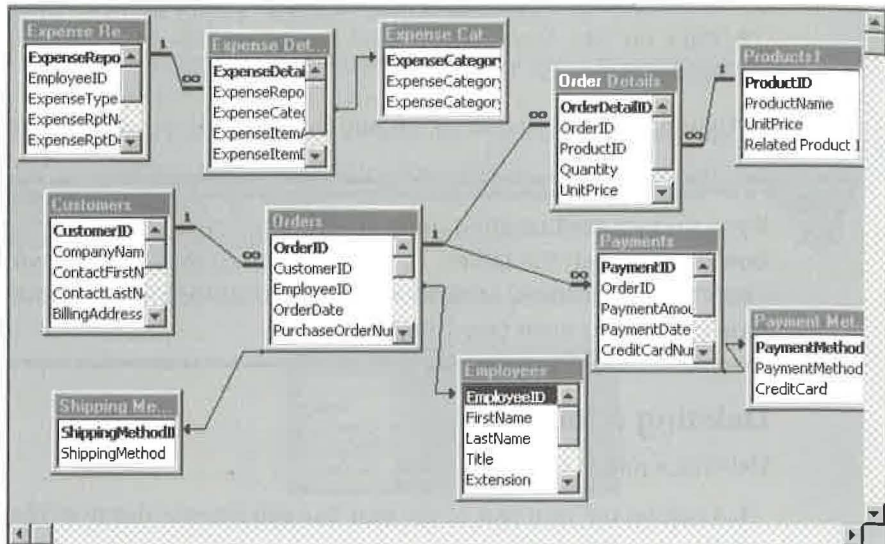
Showing an Entire Database Structure

As explained in Chapter 6, you can use the Relationships window to get “the big picture” of all the tables in your database and how they're related to one another. Here's a quick review of how to show the entire database structure:

1. Return to the database window and choose Tools ► Relationships or click on the Relationships toolbar button.
2. Choose Relationships ► Show All or click on the Show All Relationships toolbar button to show all the relationships.
3. If necessary, rearrange the tables on the screen so the join lines don't look like a pile of spaghetti (see Figure 10.10).

FIGURE 10.10

The Relationships window for the Order Entry database.



Creating AutoLookup Queries

Suppose a customer calls to place an order. Being a lazy typist, you don't want to reenter customer information that's already in the Customers table. To save your typing fingers, you can create a special Select query called an *AutoLookup query*. Then simply enter a valid CustomerID, and Access will fill in the customer information automatically.

AutoLookup works in queries in which two tables have a one-to-many relationship and the join field on the "one" side of the relationship has a unique index. That is, the join field on the one side must either be a primary key or have its Indexed property set to Yes (No Duplicates).

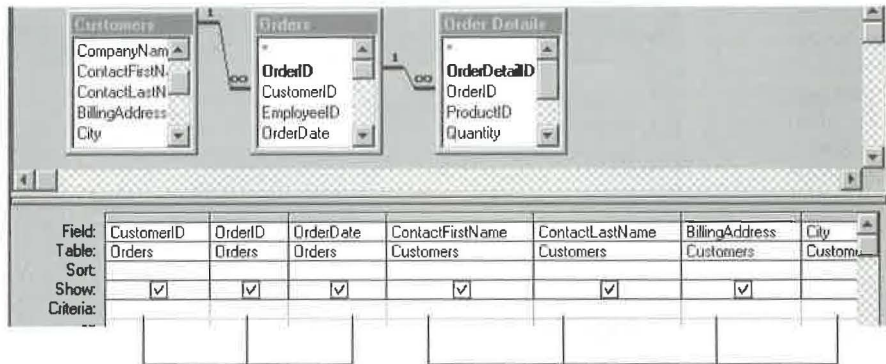
NOTE

Do not confuse a lookup field (Chapter 6) with an AutoLookup query. A lookup field displays a list of values that makes it easier to enter data into one specific field. An AutoLookup query automatically fills in corresponding data on the one side of a one-to-many query when you enter new data in the join field on the many side. As Figure 10.11 shows, you can use a lookup field (for example, CustomerID) in an AutoLookup query.

In the classic example of Customers and Orders, the CustomerID field in the Customers table is a primary key on the one side of the relationship. The CustomerID field in the Orders table is on the many side of the relationship.

FIGURE 10.11

An AutoLookup query uses the join field from the many side of the relationship to look up and fill in information from the one side automatically.



These fields come from the many side.

These fields come from the one side.

Here's the secret to designing an AutoLookup query: Put the join field from the many side of the query into the QBE grid. This field is the *Autolookup* field. For example, add the CustomerID field from the Orders table (not the Customers table) to the grid.

NOTE

For AutoLookup to work, you must be able to update the Autolookup field from the many side of the query. If you try to change a field that can't be updated, Access will display an explanatory message in the status bar.

Figure 10.11 shows an AutoLookup query in which CustomerID (from the Orders table) is the Autolookup field for customer information. Figure 10.12 shows the dynaset after we chose RNAA Associates from the drop-down list for the CustomerID field in the empty row at the bottom of the datasheet. As soon as we selected a company name from the Autolookup field, Access filled in the corresponding customer information automatically.

WARNING

When designing queries for data entry purposes, be sure to include required fields and validated fields, and be sure to check their Show boxes on the QBE grid. If the query doesn't include such fields, Access may display error messages about bad or missing data in those fields; unfortunately, the person entering the data won't be able to fix the problem because the fields aren't there. A catch-22 situation if there ever was one!

FIGURE 10.12

The dynaset created by the AutoLookup query in Figure 10.11. After we chose RNAA Associates from the CustomerID drop-down list, Access filled in the customer information automatically.

Customer ID	Order ID	Order Date	ContactF	Contact Las	Billing Address
WorldWide Widgets	17	2/11/95	Shirley	Ujest	187 Suffolk Ln.
WorldWide Widgets	18	2/28/95	Shirley	Ujest	187 Suffolk Ln.
WorldWide Widgets	19	3/15/95	Shirley	Ujest	187 Suffolk Ln.
WorldWide Widgets	19	3/15/95	Shirley	Ujest	187 Suffolk Ln.
WorldWide Widgets	20	4/1/95	Shirley	Ujest	187 Suffolk Ln.
University of the Elite	21	2/1/95	Wilbur	Stumingi	P. O. Box 555
University of the Elite	22	4/1/95	Wilbur	Stumingi	P. O. Box 555
University of the Elite	23	5/7/95	Wilbur	Stumingi	P. O. Box 555
University of the Elite	24	6/28/95	Wilbur	Stumingi	P. O. Box 555
University of the Elite	25	7/15/95	Wilbur	Stumingi	P. O. Box 555
RNAA Associates	26		Hanley	Strappman	2743 Bering St.

When data is entered into the lookup field...

Access fills in these fields automatically

Prompting for Selection Criteria

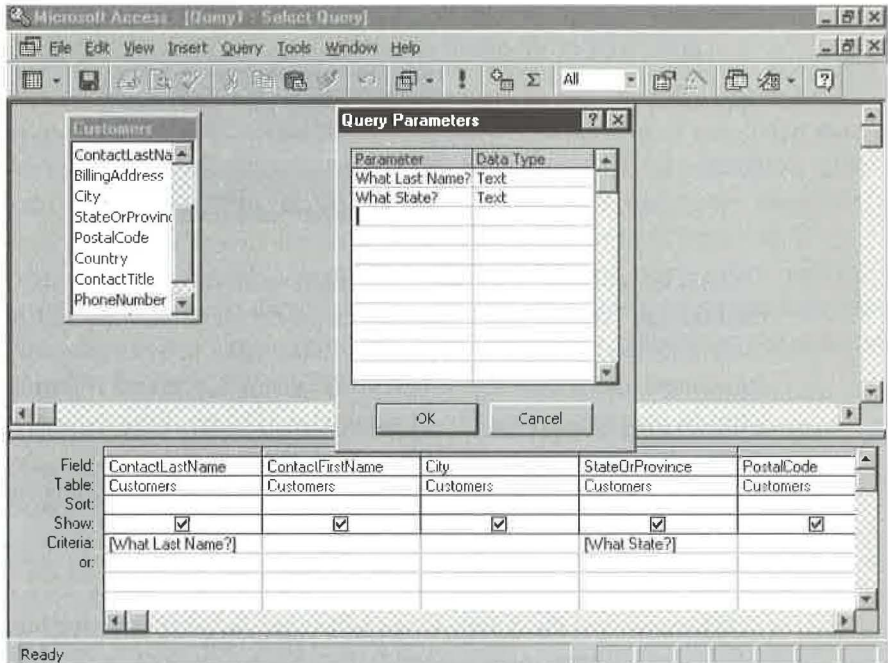
If you find yourself changing the criteria each time you run a Select query, consider using *parameters* instead of specific values in the criteria. A parameter acts as a sort of placeholder that you can fill in when you run the query.

To define a parameter:

1. Create your query normally but omit the criteria for now.
2. Click on the Criteria cell of the field you want to search. In place of the data you're searching for, type a parameter in square brackets. The parameter can't be the same as an existing field name (though it can include the field name) and is best worded as a question or a prompt that tells the user what to do. Repeat this step for each field that you want to define as a parameter.
3. Choose Query ► Parameters or right-click on an empty place in the tables area of the query design window and choose Parameters. You'll see the Query Parameters dialog box (Figure 10.13 shows a completed example).
4. Type the first parameter you want Access to prompt for *exactly* as you typed it into the Criteria cell in the Parameter column, but without the square brackets. (If the spellings don't match exactly, Access will prompt for both spellings when you run the query and you won't get the results you want.)
5. Choose a data type for the parameter in the Data Type column. This should match (or be compatible with) the data type of the field the parameter is in. (Access usually fills this in automatically when you tab to or click in the Data Type cell.)

FIGURE 10.13

This query uses two parameters to prompt for criteria when you run the query.



- Repeat steps 4 and 5 for each parameter. When you're done, click on OK.



As a shortcut, you often can skip steps 3 through 6 and not bother to define any parameters in the Query Parameters dialog box. When you run the query, it will display the prompts you entered in step 2 in order from the leftmost prompt to the rightmost prompt. However, you *should not* skip these steps if (1) you're using parameters in a Crosstab query or in a parameter query that a Crosstab query or chart is based on, or (2) you're creating a parameter for a field with the Yes/No data type or for fields that come from a table in an external database. For Crosstab queries you also must set the ColumnHeadings property in the Query Properties sheet.

Figure 10.13 shows a query that uses parameters. Whenever you run this query, Access first displays this message:



You can type any name you want, such as **Adams**, and click on OK. Next you'll see the message "What State?" This time, type a state name, such as **CA**, and click on OK. (Access always prompts for parameters in the order in which you list them in the Query Parameters dialog box. If you've left that dialog box blank, it prompts in order from leftmost parameter to rightmost parameter.) The resulting dynaset will show only California customers whose last names are Adams.

You also can use the parameter as part of a search criterion. For instance, you can replace the [What Last Name?] criterion in Figure 10.13 with

```
Like [What Last Name?]&"*"
```

When you run the query, Access will show the same message. But your entry will be substituted with the more general "Like" expression. So if you enter **Adams**, Access will convert that entry to **Like Adams*** and display records that begin with the letters *adams* (Adams, Adamson, Adams and Lee, and so on).

NOTE

The & operator concatenates strings on either side of it. Thus, *Like "*" & [What Name?]&"*"* translates into *Like *Adams** if you enter *Adams* in the Enter Parameter Value dialog box. This example matches records such as *McAdams*, *Adams*, *Adamson*, and so on.

The bottom line is that you can use parameters anywhere you'd type normal text or a field name in a Criteria cell.

As usual, help is available for setting up parameter queries. In the Access Help Index, a good place to start is the *Queries, Parameter* topic.

Using Totals, Averages, and Other Calculations

Access provides several ways to perform calculations on your data:

- You can perform calculations in Criteria cells, as discussed in Chapter 9.
- You can perform a calculation and display the results in a calculated field that isn't stored in the underlying tables. Back in Figure 10.2, for example, a calculated field named `$` multiplied the `[Quantity]` field times the `[Order Details]![Unit Price]` field. The results appear in the Total Price column of Figure 10.3.
- You can group records according to certain fields and display summary calculations for the grouped data. This lets you determine how many items were ordered on various dates, how many orders were placed by each customer, the dollar value of orders for each product, and so on.

You'll learn how to use calculated fields and data summaries in the following sections.

Using Calculated Fields

To create a calculated field, click in an empty Field cell in the QBE grid and type an expression that will do the calculation. When referring to other fields in the QBE grid, be sure to enclose their names in square brackets. If more than one table in your query has the same field name, you must specify both the table name and the field name, using the format `[TableName]![FieldName]`. For instance, to multiply the quantity field by the unit price field of the Order Details table, type `[Quantity]*[Order Details]![UnitPrice]` in an empty Field cell.

You can give the calculated field its own heading in the dynaset and use that heading as a field name in still other calculations. Just precede the calculation with the heading you want to use, followed by a colon, like this:

```
ExtPrice:[Quantity]*[Order Details]![UnitPrice]
```

Now you can use the `ExtPrice` calculated field in another calculated field. This example calculates the price with sales tax:

```
WithTax:[ExtPrice]*(1+[SalesTaxRate])
```

Here's the QBE grid for these calculations:

Field:	ProductName	SalesTaxRate	ExtPrice: [Quantity][Order Details]![UnitPrice]	WithTax: [ExtPrice]*(1+[SalesTaxRate])
Table:	Products	Orders		
Sort:	Ascending	Ascending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				

And here are the results:

Product Name	Sales Tax Rate	ExtPrice	WithTax
Baseball	6.00%	\$8.75	\$9.27
Baseball	7.25%	\$17.50	\$18.77
Basketball	0.00%	\$4.95	\$4.95
Basketball	5.00%	\$9.90	\$10.40
Basketball	6.00%	\$9.90	\$10.49
Basketball	7.25%	\$9.90	\$10.62
Basketball	7.25%	\$4.95	\$5.31
Billiard balls	0.00%	\$127.45	\$127.45
Billiard balls	0.00%	\$254.90	\$254.90
Billiard balls	5.25%	\$127.45	\$134.14
Crystal ball	0.00%	\$45.55	\$45.55
Crystal ball	0.00%	\$45.55	\$45.55
Crystal ball	0.00%	\$45.55	\$45.55
Crystal ball	0.00%	\$45.55	\$45.55
Crystal ball	0.00%	\$136.65	\$136.65

You can use the & operator to join text and fields into a single calculated field. When using this operator, be sure to enclose any text, including blank spaces, in quotation marks.

Let's look at some examples of calculated fields that use the & operator to display values from the Employee table's FirstName and LastName fields in a single column in the query result.

Assuming the FirstName is *Helen* and the LastName is *Wheels*, the calculated field shown below will display the result *Helen Wheels*:

```
[FirstName]&" "&[LastName]
```

To display the result as *Wheels, Helen* use this calculated field:

```
[LastName]& ", " & [FirstName]
```

And if you're in an expansive mood, you can use this calculated field:

```
"Hey "&[FirstName]&" "&[LastName]&"! How are ya?"
```

to display

```
Hey Helen Wheels! How are ya?
```

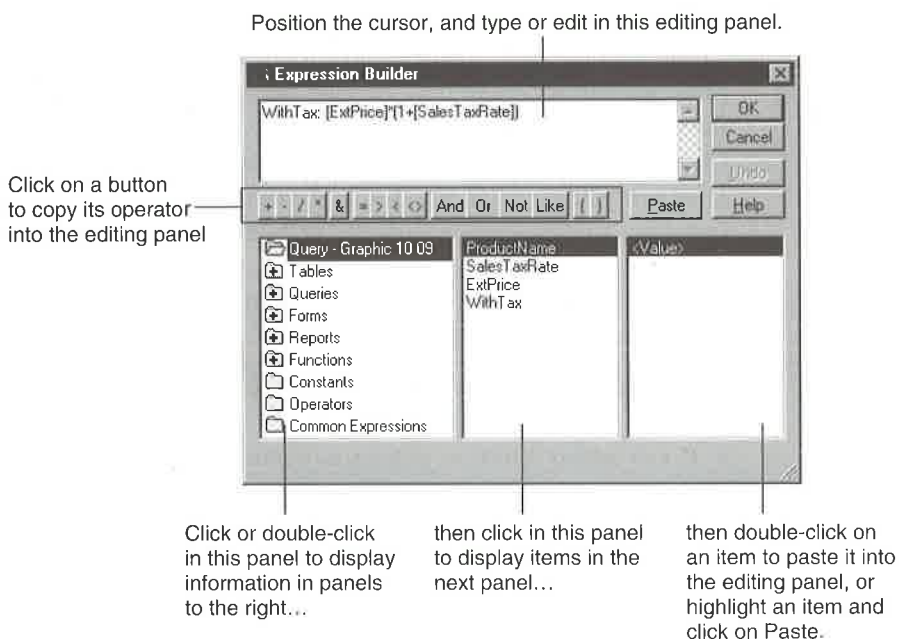
The following tips will help you make the most of calculated fields:

- **If you don't provide a name for a calculated field**, Access will assign a meaningless name, such as Expr1, Expr2, Expr3, and so on. You can change these to more useful names as needed.

- **If you misspell a field name within square brackets**, Access will treat the field as a parameter when you run the query. You'll need to click on Cancel in the Enter Parameter Value dialog box and return to query design view to correct the faulty field name.
- **If you want to change the properties of the calculated field** (to control the caption and format displayed in the dynaset), right-click on the field, choose Properties, and define the properties you want to use.
- **If you are typing lengthy calculations**, use Zoom (press Shift+F2) to expand the input area as described earlier in the chapter.
- **If you need to enter complicated calculations or criteria into a cell**, use Expression Builder (see Figure 10.14) to help. To open Expression Builder, click in the cell you want to edit and then click on the Build toolbar button (shown at left); or right-click on the cell and choose Build. You can then type in and change the expression directly in the editing panel at the top of the dialog box. Or to have Expression Builder enter field names, operators, and expressions for you, position the cursor in the editing panel, then click and double-click on items in the three lower panels of the dialog box or click on the operator buttons above the panels. (For more help with Expression Builder, click on the Help button in the dialog box.) When you're ready to leave Expression Builder and save your expression, click on OK.

**FIGURE 10.14**

The Expression Builder.



TIP

You can't change the contents of a calculated field in a dynaset. But you *can* change other fields. The calculated field will be recalculated automatically.

Access provides literally dozens of functions that are useful in calculated fields. Some of the most useful are discussed under "Creating Your Own Calculation Expressions" later in this chapter. To find out about any Access function, search for *Functions, Reference Topics* in the Access Help Index or look up a specific function name, such as *Avg Function*.

Summarizing Your Data

Suppose you want to know the number of customers in Los Angeles or the average price of products purchased in February. For jobs like these, you need *summary calculations*. Summary calculations differ from calculated fields in that they compute some value, such as a sum or average, on *multiple records* within a table.

TIP

For a truly easy way to summarize your data, use the Simple Query Wizard, described earlier in this chapter.

To perform a summary calculation:

1. Create a simple Select query as usual. Drag any fields you want to group by, or perform calculations on, to the QBE grid. Do not include the asterisk (*) field name in the QBE grid.
2. Click on the Totals toolbar button (shown at left). Or choose View ► Totals. A new Total row will appear in the QBE grid, with *Group By* in each column.
3. Leave the Total row set to *Group By* for fields that you want to use for grouping (or categorizing, or subtotaling).
4. Choose one of the summary options listed in Table 10.2 in the Total cell for the field you want to summarize.
5. Add calculated fields, sorting specifications, and selection criteria as needed. Then...
 - If you added selection criteria to columns that you don't want to group by, change the Total cell for those columns to *Where*.
 - If you created calculated fields in columns that you don't want to group by, change the Total cell for those columns to *Expression*.
6. Click on the Query View toolbar button to see the results.



TABLE 10.2: SUMMARY OPERATORS USED IN QUERIES

SUMMARY OPERATOR	COMPUTES	WORKS WITH DATA TYPES
Avg	Average	AutoNumber, Currency, Date/Time, Number, Yes/No
Count	Number of non-blank values	All
First	Value in first record	All
Last	Value in last record	All
Max	Highest (largest) value	AutoNumber, Currency, Date/Time, Number, Text, Yes/No
Min	Lowest (smallest) value	AutoNumber, Currency, Date/Time, Number, Text, Yes/No
StdDev	Standard deviation	AutoNumber, Currency, Date/Time, Number, Yes/No
Sum	Total	AutoNumber, Currency, Date/Time, Number, Yes/No
Var	Variance	AutoNumber, Currency, Date/Time, Number, Yes/No

Remember these points as you design summary queries:

- Don't be concerned if Access changes your Totals queries slightly after you save, close, and reopen them. The queries will still work fine! For example, Access might change the summary operator in the Totals cell from *Avg* to *Expression* and transform the Field cell from an entry such as this:

Avg: [Order Details]![UnitPrice]

to an aggregate function such as this:

Avg: Avg([Order Details]![UnitPrice])

- You can't update the fields in the snapshot created by a Totals query.

NOTE

Unlike a Select query (which creates a *dynaset* that's usually updatable), a Totals query creates a *snapshot*. Although a snapshot looks the same as a dynaset on the screen, you cannot update any of the data in a snapshot.

- You can summarize as many fields as you wish, and you can include multiple copies of the same field in the QBE grid. Here's an example that displays the number of products and the average, minimum, and maximum price of products ordered in February:

Field:	OrderDate	Count: ProductName	Avg: [Order Details].[UnitPrice]	Min: [Order Details].[UnitPrice]	Max: [Order Details].[UnitPrice]
Table:	Orders	Products			
Total:	Group By	Count	Avg	Min	Max
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	Like "2/*/95"				

Order Date	Count	Avg	Min	Max
2/1/95	9	\$25.26	\$4.95	\$127.45
2/11/95	4	\$6.08	\$4.95	\$6.75
2/18/95	2	\$26.15	\$6.75	\$45.55
2/25/95	1	\$5.65	\$5.65	\$5.65
2/28/95	1	\$17.85	\$17.85	\$17.85

- To base calculations on all records in the table, do not use *Group By* in any column. For instance, to count the customers in the Customers table, put just the CustomerID field in the QBE grid and use the Count operator.
- To base calculations on groups of records (that is, to *subtotal* a group of records), use *Group By* in the Total cell for all fields that make up the group. This example counts the number of customers in each city and state/province:

Field:	City	StateOrProvince	CustomerID
Table:	Customers	Customers	Customers
Total:	Group By	Group By	Count
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			

- You can sort on summarized fields to rank them. For instance, if you're calculating the average unit price for products sold each month, sort the UnitPrice field in descending order to rank products from highest average price to lowest, as shown in Figure 10.15. In Figure 10.15, we used the Format ([OrderDate], "mmm") function to extract the month name from the order date.
- You can specify search criteria for fields that you're grouping by, as you would in any query. This limits the display to the groups you chose in the Criteria cells of the Group By field or fields.
- To apply a search criterion *after* calculating the summary, enter that criterion in the field where you're doing the calculation. The example below calculates the

FIGURE 10.15

Average Price (descending) by month ordered. The function `Format([OrderDate], "mmm")` extracts the month name from the order date.

Field:	Month: <code>Format([OrderDate], "mmm")</code>	Avg: <code>[Order Details]![UnitPrice]</code>
Table:		
Total:		
Group By:	Avg	
Sort:	Descending	
Show:	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		

Month	Avg
July	\$127.45
March	\$29.99
May	\$24.27
June	\$24.27
February	\$19.26
April	\$12.11

average price of products sold in each month and displays (in descending order) only those with an average price greater than \$15.

Field:	Month: <code>Format([OrderDate], "mmm")</code>	Avg: <code>[Order Details]![UnitPrice]</code>
Table:		
Total:		
Group By:	Avg	
Sort:	Descending	
Show:	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		>15

Month	Avg
July	\$127.45
March	\$29.99
May	\$24.27
June	\$24.27
February	\$19.26

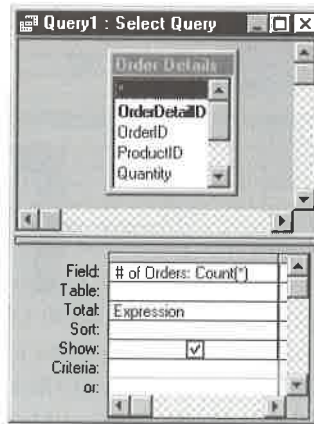
- To apply a search criterion *before* the calculation takes place, use the *Where* operator in the Total cell for the field you want to select first. You also must deselect the Show box for that field. The example below limits the average price calculations to products sold to customers outside the state of California. Compare the numbers

shown below (which exclude the purchases by California customers) with the averages shown in Figure 10.15 (which include those purchases):

Field:	Month: Format([OrderDate],"mmm")	Avg: [Order Details].[UnitPrice]	StateOrProvince
Table:			Customers
Total:	Group By	Avg	Where
Sort:		Descending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			<>"ca"

Month	Avg
July	\$127.45
March	\$47.35
June	\$32.25
May	\$28.93
February	\$20.32
April	\$8.19

- Summary calculations normally exclude blank records, but they will include zero values in a numeric field. If you need to count all records, including blanks, use **Count(*)** in the column's field cell, as shown below:



TIP Assuming you don't mind including records that contain null values, the calculation **Count(*)** runs much faster than **Count([fieldname])**.

- You can summarize calculated fields. However, the expression in each calculated field must refer to “real” fields, not to other calculated fields. For example, if you’ve defined the calculated field

ExtPrice: [Quantity]*[UnitPrice]

and want to calculate the average sales tax, you can’t use the expression

AvgSalesTax: [ExtPrice]*1.05

for the field you’re averaging. Instead you must repeat the calculation, like this:

AvgSalesTax: [Quantity]*[UnitPrice]*1.05.

For more help with calculating totals, look up *Totals In Queries* in the Access Help Index.

Creating Your Own Calculation Expressions

You can use Access functions to design your own summary expressions for analyzing data. Here’s how:

1. Activate totaling in the query design window by clicking on the Totals toolbar button.
2. Start in a blank column of the QBE grid. If you’re using an *aggregate function* or *domain aggregate function* (described just below), choose *Expression* in the column’s Total cell. For *formatting functions*, set the Total cell to *Group By*.
3. Enter an expression that uses the function you want in the Field cell of the column. Some of the most useful functions are listed in Table 10.3.

NOTE

You can use Zoom (Shift+F2) when entering lengthy functions into the Field cell. Always type expressions on a single line (even if they don’t appear on one line in this book).

4. Fill in other rows and columns in the QBE grid as needed.
5. Run the query as usual.

TABLE 10.3: USEFUL ACCESS FUNCTIONS

FUNCTIONS	WHAT THEY DO	EXAMPLE
Aggregate Functions		
Avg, Count, Max, Min, StDev, Sum, and Var	Perform statistical calculations on a group of records in the current table.	<i>Avg([Quantity]*[UnitPrice])</i> averages the quantity times the unit price for a group of records.
Domain Aggregate Functions		
Davg, Dcount, Dmax, DMin, DStDev, DStDevP, Dsum, DVar, and DVarP	Perform statistical calculations on all records in any table or query, overriding <i>Group By</i> expressions.	<i>DSum("[Quantity]*[UnitPrice]", "OrderDetails")</i> shows the total quantity times the unit price for all records in the Order Details table.
Formatting Functions		
Ccur, CDbl, Cint, CLng, CSng, CStr, and Cvar	Convert expressions from one format to another. You can use these functions instead of changing a field's properties.	<i>CCur(Avg([Quantity]*[UnitPrice]))</i> converts to currency format the average quantity times the unit price for a group of records.
Format	Formats a number, date, time, or string.	<i>Format([OrderDate], "mmm")</i> displays the month portion of the OrderDate field.
Left	Returns the leftmost <i>n</i> characters in a string.	<i>Left([LastName], 1)</i> displays the first letter of the LastName field.

Continued 

TABLE 10.3: USEFUL ACCESS FUNCTIONS (CONTINUED)

FUNCTIONS	WHAT THEY DO	EXAMPLE
Mid	Returns a string that's part of another string.	<i>Mid([PhoneNumber], 1, 3)</i> displays the area code portion of a phone number field. This example assumes three-character United States area codes in which the parentheses aren't stored in the table.
Right	Returns the rightmost <i>n</i> characters in a string.	<i>Right([Ship CSZ], 5)</i> displays the last five characters of a hypothetical Ship CSZ field (a combined City, State, and Zip Code field). This example assumes five-character United States zip codes.

You'll find that the following types of functions are especially useful for summarizing data:

Aggregate Functions Calculate statistics on a group of records in the current table.
Examples: Count, Avg, Sum, Min, Max, and Var.

Domain Aggregate Functions Calculate statistics on an entire table or query, overriding any Group By expressions.

Examples: DCount, DAvg, DSum, DMin, DMax, and DVar.

Formatting Functions Return specified parts of the data or format data in specific ways.

Examples: Left, Right, Mid, and Format. These functions are useful with many types of calculated fields, not just summary calculations.

For a complete list of Access functions, search for *Functions, Reference Topics* in the Access Help Index. You also can use the Access Help Index to look up individual functions by name.

Sample Summary Calculation Expressions

For practice, let's look at some sample summary calculations. The example in Figure 10.16 groups and sorts records by Month, and it calculates both the value of orders in each month and each month's contribution to the whole, expressed as a percentage.

In Figure 10.16, we used the calculation shown below to isolate the month number portion of the order date; then we sorted this column in ascending order and hid it from view (you can't see this calculation in Figure 10.16 because it has scrolled off the right edge of the screen). This trick puts the months in January, February, March, April, etc. order, rather than displaying them in alphabetical order by month name (April, February, July, June, etc.).

```
DatePart("m",OrderDate)
```

The calculation in the first column shows the month name portion of the order date:

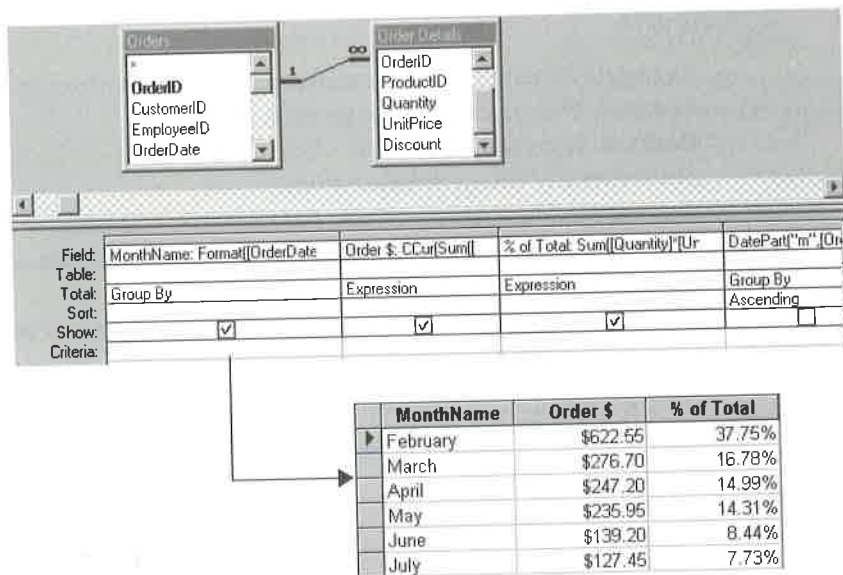
```
MonthName: Format([OrderDate], "mmm")
```

We used this calculation in the second column of the QBE grid:

```
Order $:CCur(Sum([Quantity]*[UnitPrice]))
```

FIGURE 10.16

This example uses the Sum and Dsum functions to calculate each month's contribution to total sales. It also uses the DatePart function to extract the month number from the OrderDate, so that we can present the results sorted chronologically by month rather than alphabetically by month name.



The calculation in the QBE grid's third column is

```
% of Total:Sum([Quantity]*[UnitPrice])/
Dsum("[Quantity]*[UnitPrice]","Order Details")
```

We used the Field Properties sheet to format the % *Of Total* column as a percentage.

You can replace "mmm" in the Format function to group by other time intervals. For example, "ww" groups by week of the year (1-54); "q" groups by quarter; "yyyy" groups by year; and "hh" groups by hour.

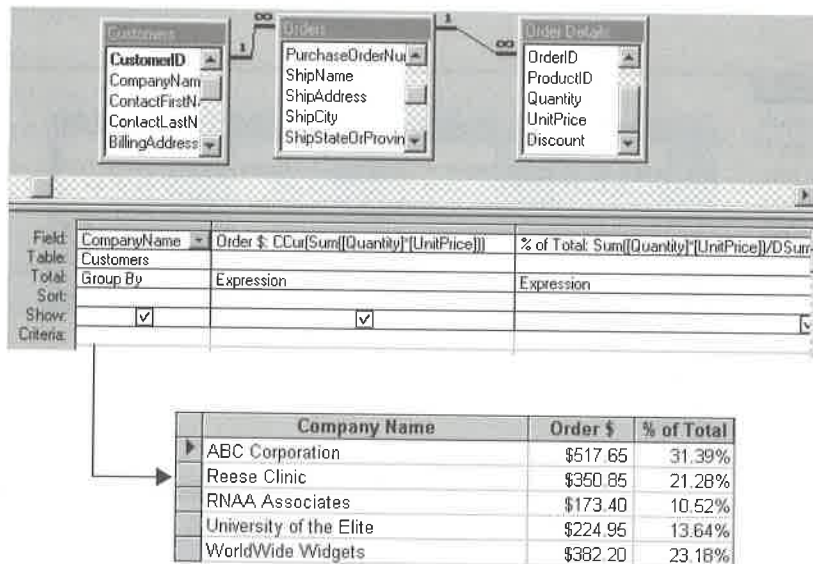
In Figure 10.17 we used the same summary calculations for the second and third columns. However, this time we added the Customers table to the query and grouped orders by CompanyName instead of month.

With little effort, you can modify the example in Figure 10.17 to group orders by the first letter of the CompanyName (though there isn't an especially good reason to do so). Simply replace the expression in the first column's Field cell with this one:

```
FirstLetter:Left([CompanyName],1)
```

FIGURE 10.17

This example uses the Sum and Dsum functions to calculate each company's total sales.



Creating Crosstab Queries

Crosstab queries let you cross-tabulate data in a row-by-column fashion. The example in Figure 10.18 answers the question “Who ordered how many of what?”.



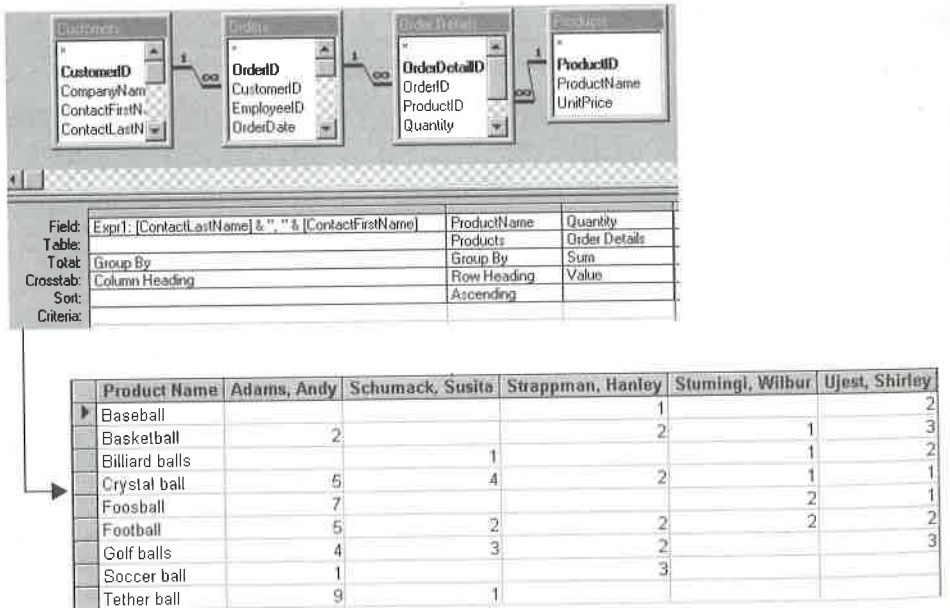
To quickly create crosstabs of data in a single table, use the Crosstabs Query Wizard described earlier in this chapter.

When designing a Crosstab query, you must decide which fields to use for row headings, column headings, and summary values and how you want to summarize the values (for example, as sums or averages). Here are the steps to follow:

1. Start in the query design window and choose the tables you want to use. Put fields (including calculated fields) into the QBE grid. You can specify any search criteria and sorting you need.
2. Click on the drop-down arrow next to the Query Type toolbar button and then choose Crosstab. Or choose Query ► Crosstab.

FIGURE 10.18

A Crosstab query makes it easy to see who ordered how many of what.



3. Choose Row Heading in the Crosstab cell for the field you want to use for row headings. You can designate more than one field, but at least one field must have *Group By* in its Total cell. In place of a Field name, you can use expressions to group values together (see the next step for an example).
4. Choose Column Heading in the Crosstab cell for the field you want to use for column headings. Only one field is allowed, and it must contain *Group By* in its Total cell. In place of a Field name, you can use expressions to group values together. Figure 10.18 shows the expression Expr1: [ContactLastName]&", "&[ContactFirstName] used this way.
5. Choose Value in the Crosstab cell for the field you want to summarize. Then, in the Total cell for the field, choose the type of summary you want (usually *Sum* or *Average*). Don't choose *Group By* for this field.
6. Choose (Not Shown) in the Crosstab cell for those fields to group by additional fields without displaying them in the results.
7. Click on the Query View toolbar button to view the results.

Access automatically determines crosstab column headings from the data in the table and then sorts the headings in left-to-right order across the columns. You can change this behavior with these steps:

1. Create and run the query as described above. Print the results or jot down the column headings, and then return to the query design window.
2. Open the Query Properties sheet (right-click on an empty spot in the tables area and choose Properties or click on the Properties toolbar button).
3. Type the headings you want to use (spelled exactly as they appear in the output) in the order you want them in the Column Headings box. Separate each heading with a semicolon (;) or new line (Ctrl+Enter). Or enclose the headings in double quotes and separate them with commas (see Figure 10.19). For example, type
January;February;March; (or "January","February","March",....)
4. Run the query again to see the revised column headings.

The crosstab shown in Figure 10.19 has two row heading fields (ProductName and StateOrProvince). The column headings are taken from the month portion of the order date. We changed the Query Properties to list only the months February, March, April, May, June, and July (there were no orders in the other months, so we omitted them).

The online Help can guide you through cross-tab setup. Just look up any of the topics under *Crosstab Queries* in the Help Index for details. Or for fast setup, use the Crosstab Wizard to create a Crosstab query; then switch to the query design window and tweak the design as needed.

FIGURE 10.19

Crosstab queries to calculate number of orders by month for each product and state group. We used Query Properties to choose which months appeared and in which order.

The screenshot shows the Microsoft Access interface. On the left, the 'Queries' pane shows a query named 'Orders' based on the 'Customers' table. The 'Query Properties' dialog box is open, showing the 'General' tab. The 'Column Headings' property is set to "February", "March", "April", "May", "June", "July". Below the dialog, a preview of the query results is shown as a table.

Field:	ProductName	StateOrProvince	Months: Format([OrderDate], "mmmm")	Quantity
Table:	Products	Customers		Order Details
Total:	Group By	Group By	Group By	Sum
Crosstab:	Row Heading	Row Heading	Column Heading	Value
Sort:	Ascending	Ascending	Ascending	
Criteria:				

Product Name	State/Prov	February	March	April	May	June	July
Baseball	AK			1			
Baseball	ID	2					
Basketball	AK			2			
Basketball	CA	2					
Basketball	ID	3					
Basketball	WY	1					
Billiard balls	ID	2					
Billiard balls	NM		1				
Billiard balls	WY				1	1	1
Crystal ball	AK			2			
Crystal ball	CA	2	1	2			
Crystal ball	ID		1				
Crystal ball	NM	1			3		
Crystal ball	WY						1
Foosball	CA	4		1		2	
Foosball	ID	1					

Creating Action Queries

Action queries differ from the queries discussed so far because they change or delete data in a table. The four types of action queries are

Update Queries Change data in a group of records.

Append Queries Copy a group of records from one table to another.


Delete Queries Delete records from a table.

Make Table Queries Create a new table from a group of records in another table.

Action queries are very fast and can do a lot of damage if they aren't exactly right. So keep these data saving tips in mind:

- **Always back up your database, or (at least) the tables you're going to change.** To copy a table, go to the database window, click on the Tables tab, and then click on the table you want to copy. Choose Edit ► Copy (Ctrl+C), choose Edit ► Paste (Ctrl+V), type a new name for the table, and then click on OK.

- **For extra safety**, you can design and test a Select query. Then, when you're sure it will act on the proper records, convert it to an action query. To convert from one type of query to another, click on the drop-down arrow next to the Query Type toolbar button, and then choose the appropriate option from the list, or choose equivalent options from the Query menu. (The steps in the sections below take a slightly more streamlined, yet still safe, approach to designing and running action queries.)
- **In the database window**, double-clicking on an action query (or highlighting the query and clicking on Open) runs the query. To warn you that this will happen, an exclamation point and descriptive icon appear beside the query name in the database window (see example below).

 Query - Increase Price By 15%

To get help with action queries, go to the Access Help Index and double-click on the topic *Action Queries, Creating*; then click on the button next to the appropriate "What do you want to do?" topic. Or go to the Help Contents, open the *Working With Queries* book, open the *Creating Action Queries, Crosstab Queries, And Parameter Queries* book, open the *Action Query* book, and then choose a subtopic from the list.

Update Queries

Update queries let you change all records or groups of records very quickly. To create an Update query:

1. Click on the drop-down arrow next to the Query Type toolbar button in the query design window and then choose Update Query. Or choose Query ► Update Query from the menus. The Query Type button changes, as shown at left.
2. Add the tables, include fields you want to update and fields you want to use for selection criteria, and set criteria to select the records you want to update, just as for a Select query.
3. Type a new value for the field in the Update To call for each field you want to change or type an expression that will calculate a new value. If you want to empty out a field, type Null in the Update To cell for that field.
4. Click on the Query View toolbar button (shown at left) to see which records will be updated before you update them or choose View ► Datasheet View. Adjust the query criteria and preview the results until you select the records you want to update.
5. Return to the query design window and click on the Run toolbar button (shown at left) or choose Query ► Run to run the Update query.



- Click on Yes if you want to proceed when prompted with the number of rows that will be updated or click on No to cancel the operation.

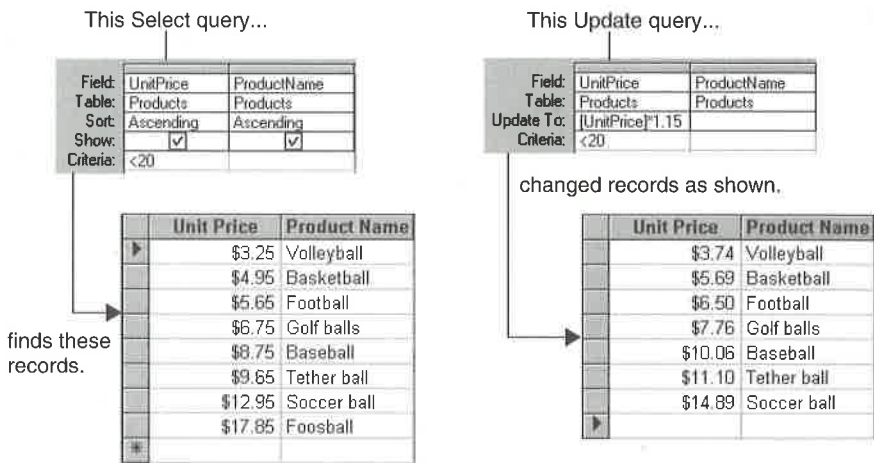
**NOTE**

You can't use an Update query to update totals queries, calculated fields, or locked tables.

The left side of Figure 10.20 shows a Select query and the products it will find. The right side shows the Select query after we converted it to an Update query that raises prices for those items by 15 percent. After running the Update query, we converted it back to a Select query (by clicking on the drop-down arrow next to the Query Type button and choosing Select) and displayed the revised records shown at the bottom right. Some items no longer appear because their prices are now \$20 or more.

FIGURE 10.20

A Select query to search for records with a current price less than \$20, along with the Update query that raises prices of those products by 15 percent.



Understanding Cascaded Updates

Recall from Chapter 6 that you can use the Relationships window to define one-to-many relationships between tables, to enforce referential integrity between such tables, and to cascade updates to related fields automatically. If you use this method, Access will carry over updates to the related field from the one side of the relationship to the related field on the many side, even if your Update query doesn't include tables on the many side.

Suppose we set up a one-to-many relationship and enforced referential integrity between Suppliers (a table that's not in our sample Order Entry database) and Products.

Further suppose we selected Cascade Update Related Records on the one side of each relationship. Therefore, if we use an Update query to change the SupplierID for a record in Suppliers, Access automatically updates the corresponding SupplierID in the Products table.

If you've defined Cascade Update Records for related tables, you don't need to create an Update query when you simply want a change to the primary key on the one side to carry over to related records on the many side. Instead, go to form view or datasheet view for the table on the one side and change the data in the appropriate primary key field. Access will update the related field in the many side table automatically.

NOTE

Values in AutoNumber fields can't be changed by editing in a table or form, or by running an update query. In the sample Order Entry database, all the IDs—including CustomerID, EmployeeID, OrderID, and ProductID—on the one side are defined as primary keys with the AutoNumber data type.

Append Queries

An *Append query* copies some or all records from one table (the *source table*) to the bottom of another table (the *target table*). This query is handy when you use separate tables with similar structures to manage data.

For example, you might use a Products table to store current products and an Old-Products table to store discontinued products. Or you might store current, unfilled orders in an Orders table and filled orders in a *history table* named OrdHist.

TIP

To quickly create a new table that has the same structure as an existing table, click on the Tables tab on the database window, click on the table that has the structure you want to copy, choose Edit ► Copy (Ctrl+C), and then choose Edit ► Paste (Ctrl+V). When the Paste Table As dialog box appears, type a new table name (for example, OldProducts), choose Structure Only, and click on OK.

The example below will append records for shipped orders to the bottom of another table that has the same structure.

Field:	ShipDate	Orders.*
Table:	Orders	Orders
Sort:	Ascending	
Append To:		OrdHist.*
Criteria:	Is Not Null	

To create an Append query:



1. Click on the drop-down arrow next to the Query View toolbar button in the query design window and then choose Append Query. Or choose Query ► Append Query from the menus. (After you complete step 3, the Query Type button changes, as shown at left.) You'll see this dialog box:



2. Skip to step 3 if the table is in the same database. If the table you want to append to is in a different database, click on Another Database, press Tab, and specify the drive, folder, and name of that database (for example, `c:\My Documents\Mydata`).



To look for the target database, right-click on the Start button, choose Explore or Find, and then use standard Windows techniques to locate the database. When you find what you're looking for, close Explorer or Find, and then type the appropriate location into the File Name text box below Another Database.

3. Choose the target table from the Table Name drop-down list and then click on OK.
4. Add tables (include fields you want to copy and fields being used for selection criteria) and set criteria to select the records you want to append in the query design window, just as for a Select query.



If the field names in the source table match those in the target table, Access will fill in the appropriate Append To cells for the target table automatically. You can change this if necessary. If you're using the asterisk to copy all fields from the source table, delete field names from the Append To cells of columns that contain selection criteria.

5. Click on the Query View toolbar button (or choose View ► Datasheet View) to preview your changes. Adjust the query and preview the results until you select the records you want to append.
6. Return to the query design window to run the Append query and then click on the Run toolbar button, or choose Query ► Run.
7. Click on Yes if you want to proceed when prompted with the number of records that will be appended or click on No to cancel the operation.

If an error message dialog box appears, respond to its prompts (there's a Help button if you need it), as discussed later in "Troubleshooting Action Queries."

Here are some important points about Append queries:

- When designing an Append query, you work with the source table. Records are *copied* (not moved) from the source table to the target table.
- The two tables must have similar structures and field names, but they needn't have identical structures.
- If the source table has more fields than the target table, extra fields are ignored.
- If the source table has fewer fields than the target table, fields with matching names are updated and any additional fields are left blank.
- Access will copy only those fields that are included in the source table's QBE grid to the target table.
- If the two tables have identical structures, you can use the asterisk instead of field names in both the Field cell and the Append To cell. If the QBE grid also includes fields with selection criteria, leave the Append To cell for those fields blank.
- If the target table has a primary key, the results will appear in sorted order (by primary key) rather than with all the new records at the bottom.
- To have Access assign new AutoNumber values to records as they come into the target table, exclude the AutoNumber field (if any) from the source table's QBE grid.
- To retain AutoNumber values from the source table in the target table, include the AutoNumber field in the source table's QBE grid.

WARNING

If the AutoNumber field is a primary key in the target table, the incoming records must not contain values that generate key violations.

If you need to append only a few records to a table, you may prefer to use Paste Append (see Chapter 8). Chapter 7 explains how to append records to a non-Access database.

Delete Queries

Delete queries let you delete a group of records that meet specific search criteria in one fell swoop. This query, for example, will delete shipped Orders from the Orders table:

Field:	ShipDate	Orders.*
Table:	Orders	Orders
Delete:	Where	From
Criteria:	Is Not Null	

The procedures for setting up a Delete query depend on whether you're deleting records from one table only (or from multiple tables involved in a one-to-one relationship) or from multiple tables that are involved in a one-to-many relationship.



WARNING

You can't undo Delete queries! So be sure to preview the records that will be deleted before you run the query.

Deleting Records from One Table

Let's take the simplest case first: deleting records from a single table or from multiple tables in a one-to-one relationship. Here are the steps to follow:



1. Click on the drop-down arrow next to the Query View toolbar button in the query design window and then choose Delete Query. Or choose Query ► Delete Query from the menus. The Query Type button changes, as shown at left.
2. Add the tables, include fields you'll be using to select specific records, and set criteria to select the records you want to delete, just as for a Select query.
3. Double-click on the asterisk (*) for each table you want to delete records from if you're querying multiple tables.
4. Click on the Query View toolbar button (or choose View ► Datasheet View) to preview your changes. Adjust the query and preview the results until you select the records you want to delete.
5. Return to the query design window to run the Delete query and then click on the Run toolbar button, or choose Query ► Run.
6. If you're prompted with the number of rows that will be deleted, click on Yes if you want to proceed or click on No to cancel the operation.

**TIP**

After running an Append query to append records from a “current” table to a “history” table, you can click on the drop-down arrow next to the Query Type toolbar button and choose Delete to convert the Append query into a Delete query. Then click on the Run toolbar button to delete the old records from the “current” table.

Understanding Cascaded Deletes

If you’ve defined referential integrity between tables involved in a one-to-many relationship and you’ve also selected Cascade Delete Related Records, Access will delete records on the many side of the relationship for you automatically, even if you haven’t included the many side table in the query.

For example, the sample Order Entry database has a one-to-many relationship between Customers and Orders and between Orders and Order Details. The tables on the one side of the relationship have cascaded deletes selected. Therefore, if we create a single-table Delete query to delete a particular Orders record, Access will delete the requested Orders record and its corresponding Order Details records automatically.

Likewise, if we create a single-table Delete query to delete a particular Customer record, Access will delete that customer, along with any corresponding Orders and Order Details records for that customer—deleting, in effect, records in three related tables (which may or may not be what you want).

If you’ve defined Cascade Delete Related Records between related tables, you don’t have to set up a Delete query when you want a deletion on the one side of the relationship to carry over to the many side. Simply go to form view or datasheet view for the table on the one side and delete the appropriate record or records. Access will delete the related records in the many side tables automatically!

**WARNING**

The technique described above creates a delete cascade that will delete records in all related tables that have Cascade Delete Related Records selected in Relationships. Before you delete a record that in turn might delete records in other related table(s), make sure you know how the relationships between your tables are set up and which tables will be affected. See Chapter 6 for information about creating relationships between tables.

Deleting Records from Multiple One-to-Many Tables

If you haven't set up cascaded deletes and want to delete records from multiple tables that are involved in a one-to-many relationship, you need to run two Delete queries, like this:

1. Click on the drop-down arrow next to the Query Type toolbar button in the query design window and then choose Delete Query. Or choose Query ► Delete Query.
2. Add the tables you want to use in the query.
3. Drag the field or fields you want to use for selection criteria and define the criteria as usual from the table on the one side of the relationship (for example, Products).
4. Double-click on the asterisk (*) in the table (or tables) on the many side of the relationship (for example, Orders).
5. Preview the query and run it as usual. This takes care of deleting records on the many side of the relationship.
6. Return to the query design window and delete from the window any tables from the many side (for example, Order Details).
7. Preview the resulting query on the one side table and run it again.

Make-Table Queries

A *Make-Table query* creates an entirely new table from the results of a query. Use Make-Table queries to

- Work with a “frozen” copy of your data (perhaps for printing reports or charting).
- Create an editable copy of the record set that results from a summary, cross-tab, or unique values query.
- Export data to nonrelational applications such as spreadsheets, word processors, and other programs that can't combine data from multiple tables.

To create a Make-Table query:



1. Click on the drop-down arrow next to the Query View toolbar button in the query design window and then choose Make-Table Query. Or choose Query ► Make-Table Query from the menus. (After you complete step 3, the Query Type button changes, as shown at left.) You'll see a dialog box that's similar to the one shown earlier for Append queries.
2. Skip to step 3 if the table you want to create is in the same database. If the table is in a different database, choose Another Database, press Tab, and specify the folder location and name of that database (for example, `c:\My Documents\Mydata`).

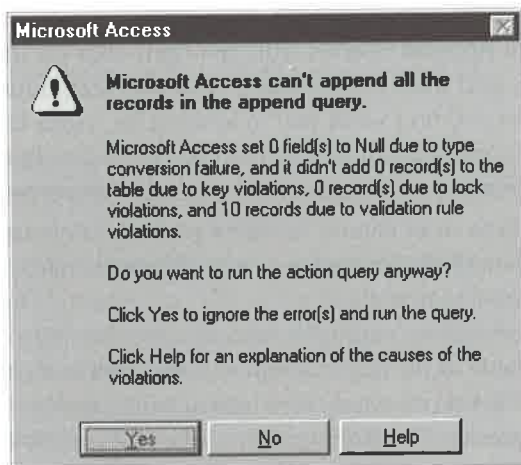


As for Append queries, you can right-click on the Start button, choose Explore or Find, and then use standard Windows techniques to locate the target database. When you find what you're looking for, close Explorer or Find and then type the appropriate location into the File Name box below Another Database.

3. Type in or choose the table you want to make in the Table Name drop-down list and then click on OK. If you choose the name of an existing table, Access will *overwrite* that table.
4. Add tables and fields, and set criteria to select the records and fields for the new table in the query design window, just as for a Select query.
5. Click on the Query View toolbar button to preview your changes. Adjust the query and preview the results until you select the records you want to include in the new table.
6. Return to the query design window and click on the Run toolbar button to run the Make-Table query. If you chose an existing table in step 3, you'll be asked if you want to continue.
7. Click on Yes if you want to delete the old table and continue or No to return to the query design window without deleting the table. (If necessary, you can click on the drop-down arrow next to the Query Type toolbar button, choose Make Table again, and specify a different table.)
8. Click on Yes to proceed when prompted with the number of rows that will be copied or No to abandon the operation.

Troubleshooting Action Queries

Action queries must obey the same rules that apply when you're entering or editing data from the keyboard. If the action query is about to break a rule, you'll see a dialog box. At this point, the query hasn't really been executed (even though the dialog box message implies otherwise). Click on No. Then return to the query or the original data, fix any problems, and rerun the query.



The most common causes of problems are

- An Append query or Update query is trying to enter data that isn't appropriate for the data type of the field. For Append queries, make sure the data types in the target table match the data types of the source table. For Update queries, make sure the update value is the correct data type for the field.
- An Append query or Update query is trying to add records that will cause key violations in a table that has a primary key. Remember, the primary key must have a unique value, and it can't contain null values. In an Append query, the "lost" records that violate the key won't be added to the target table (though, of course, they'll remain in the source table). In an Update query, no changes are made to records that will violate the key.
- Another user has locked records that your action query is trying to change. Your best bet is to cancel the query and wait until no one is using the table(s) involved in your query. Then try again.
- An action query is about to violate a referential integrity relationship between two tables. For instance, a Delete query can't delete records from the one side of a one-to-many relationship if the table on the many side contains related records. (It can, however, delete those records if the relationship allows cascaded deletes, as discussed in Chapter 6 and earlier in this chapter.)

What's New in the Access Zoo?

Most features of queries are the same in Access 97 for Windows 95 as they were in Access95. However, a few things are new:

- Some of the menu command names are now easier to understand. For example, Make Table is now Make-Table Query.
- Several new query properties are available. To find them, right-click in the background area of a query window and select Properties.

Where to Go from Here

In this chapter, you've learned how to create many types of queries to ask questions about and perform calculations on your data, to update data, to delete data, and to add data from one table to another. In the next chapter, you'll learn how to use Form Wizards to generate forms instantly. As you'll discover, forms can display data from tables or queries.



FEATURING

Using Wizards to create forms

428

Opening and using a form

434

Changing the style of a form

441

