Microsoft®

Establishing a Relationship in Access 2000

The Richard Stockton College of New Jersey

This Course Covers:

- How to establish a relationship between tables
- How to create a subform to manipulate data from multiple tables
- How to filter data and create reports from multiple tables
CustomGuide.com granted to Computer and Telecommunication Services a license agreement to print an unlimited number of copies of the CustomGuide Courseware materials within Stockton College of New Jersey for training staff, faculty and students.

End users who receive this handout may not reproduce or distribute these materials without permission. Please refer to the copyright notice below for more information.
<table>
<thead>
<tr>
<th>Chapter One: Creating Relational Databases</th>
<th>................................................................. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson 1-1: Understanding Table Relationships</td>
<td>.................................................................................................................. 6</td>
</tr>
<tr>
<td>Lesson 1-2: Creating Relationships between Tables</td>
<td>............................................................................................................... 8</td>
</tr>
<tr>
<td>Lesson 1-3: Enforcing Referential Integrity</td>
<td>...................................................................................................................................... 10</td>
</tr>
<tr>
<td>Lesson 1-4: Testing Referential Integrity and Printing and Deleting Table Relationships</td>
<td>. 12</td>
</tr>
<tr>
<td>Lesson 1-5: Understanding Relationship Types</td>
<td>...................................................................................................................................... 14</td>
</tr>
<tr>
<td>Chapter One Review</td>
<td>...................................................................................................................................... 15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter Two: Working with Forms</th>
<th>............................................................................................................. 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson 2-1: Creating a Subform</td>
<td>.......................................................................................................................................................................................... 18</td>
</tr>
<tr>
<td>Lesson 2-2: Modifying and Working with Subforms</td>
<td>............................................................................................................................................................................................... 20</td>
</tr>
<tr>
<td>Chapter Two Review</td>
<td>......................................................................................................................................................................................... 22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter Three: Working with Queries</th>
<th>............................................................................................................. 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson 3-1: A Quick Review</td>
<td>.......................................................................................................................................................................................... 24</td>
</tr>
<tr>
<td>Lesson 3-2: Creating a Multiple Table Query</td>
<td>............................................................................................................................................................................................... 25</td>
</tr>
<tr>
<td>Chapter Three Review</td>
<td>......................................................................................................................................................................................... 27</td>
</tr>
</tbody>
</table>
Chapter Objectives:

- Understand table relationships
- Create a relationship between two tables
- Understand referential integrity

Chapter Task: Join two related tables

This chapter covers what many people agree is one of the most difficult database concepts—how to create and work with relational databases. A relational database contains two or more tables that are related to each other in some way. For example, a database might contain a Customers table and an Invoices table that contains the customer's orders.

In this chapter you will learn how to link tables in an existing database together in a one-to-many relationship to create a relational database. You will also learn how to enforce referential integrity between those tables to keep records in related fields valid and accurate.

Relational databases can be confusing at first, so we'll take things slowly and explain everything in great detail as we go. Let's get started!
Lesson 1-1: Understanding Table Relationships

There are two basic types of databases:

- **Flat File**: Think of a Rolodex when you think of a flat file database. A flat file database stores all its information—names, addresses, etc—in the same place, just like addresses are stored on a rolodex card. Flat file databases are incredibly simple to create and use, but they're not very powerful or well suited for many business tasks.

- **Relational**: A relational database contains multiple tables that are related through matching fields. Figure 1-1 illustrates the design of a relational database. The database in Figure 1-1 has two tables—one that stores customer names and addresses, and another that stores customer orders. The two tables are related or linked by a common field. Relational databases are very powerful, but developing one takes a lot of skill, practice, and a strong understanding of tables and fields.
Microsoft Access can create either type of database—flat file or relational—however most Access databases tend to be the relational flavor.

Still fuzzy about how relational databases work? To see an example of a relational database look at the diagram illustrated in Figure 1-1. This database tracks customers and their orders. To store this information the database uses two tables: Customers and Orders. Each table contains fields that store similar information. The Customers table contains only information about customers and their addresses. The Orders table contains only information about any orders that were placed—it doesn’t contain any information about the customers. The two tables both have an ID field and it’s this ID field that relates or links the two tables.

Relational databases save storage space by cutting down on duplicate data. For example, the relational database shown in Figure 1-1 stores information in two related tables and eliminates the need to re-enter a customer’s name and address each time they place a new order.

Relational databases require lots of planning ahead. Before you attempt to create your database you should sit down with a trusty pencil and a pad of paper and walk through the following steps:

- **Determine the Purpose of the Database**
  Write down a list of the reports and lists that you want to come out of the database. This may seem a little backward at first, but these reports are the reason you’re creating the database. Making a list of the reports and lists you want to see and then sketch some samples of these reports and lists—be as detailed as possible. This will help determine the tables and fields to include in your database.

- **Write Down the Fields You Need**
  This should be an easy step once you have determined the purpose of your database and have sketched some sample reports and lists.

- **Organize and Group Related Fields into Separate Tables**
  Each Table in the Database should be based on only one subject. By breaking each subject into its own table you avoid redundant information and make the database more organized. The database in Figure 1-1 is broken down into two tables, Customers and Orders, so there isn’t any duplicated data. When you brainstorm try to break down your information as much as possible. If your table contains fields like Item 1, Item 2, Item 3, and so on you should probably break the information up into its own table.

- **Identify and Add the Fields Common to Each Table**
  In Figure 1-1 the Customer table’s ID field links to the Orders Table’s ID field. One of the linked fields should be the table’s primary key. See the Adding a Primary Key to a Table lesson for more information about primary keys.

- **Sketch a Diagram of Your Database**
  Create a diagram of your database similar to the one shown in Figure 1-2. Draw a box for each of your tables and write the table’s field names inside that box. Draw a line between the related fields. Most table relationships are a one to many relationship. This means that a record in one table may be related to one or more records in another table. For example, in Figure 1-1 each record in the tblCustomers table is related to one or more records in the Orders table. This makes sense, since hopefully most customers will place more than one order. You should indicate the one side of the relationship by drawing a 1 on the one side of relationship line and a ∞ (infinity symbol) on the many side of the relationship line.

All this writing and planning may seem like a lot of work, but it’s a critical step in creating a sound database. A carpenter wouldn’t start building a house without their blueprints would they? No—and it’s no different if you’re going to create a good relational database.
Lesson 1-2: Creating Relationships between Tables

Once you begin to understand the difficult concept of relational databases, the process of actually linking the tables in a database is rather simple. You link related tables by connecting the table’s common fields in Access’s Relationship window, shown in Figure 1-6. The Relationship window lets you view, create, and modify relationships among tables in a database.

Keep the following rules in mind when you link two tables together:

- **Linked fields should be (almost) identical**
  Related fields must have the same data type and field size, and they must contain the same kind of information. Related fields don’t have to have the same field name—but they should so that things don’t get confusing. The most common problem people have when they are trying to link two tables is caused fields with data types and/or sizes.

- **The primary key in one table is usually linked with a matching field in the other table**
  Notice in Figure 1-6 that the tblCustomers table's primary key CustomerID links to the CustomerID field in the Customer tblTours table.
• Fields related to an AutoNumber primary key filed must be Number fields with the Long Integer Field Size.

OK you’re ready to create a relationship between the tables in your database. Here’s how to do it:

1. **Open the Lesson 5 database.**
   To view and create relationships between tables you need to display the Relationships window.

2. **Click the Relationships button on the toolbar.**
   The Relationships window appears. If any relationships exist between the tables in your database, each of these tables will appear in a small box with lines connecting the table’s linked fields.

3. **If the Show table dialog box doesn’t appear when you open the Relationships window click the Show Table button on the toolbar.**
   First you have to add the tables that you want to relate using the Show Table dialog box, as shown in Figure 1-3. In this exercise you want to relate the Customers, Tours, and tblCustomerTours tables, so you will need to add these tables to the Relationships window.

4. **Click the tblCustomers table and click Add.**
   The tblCustomers table appears in the Relationships window.

5. **Add the tblTours and tblCustomerTours tables to the Relationships window.**
   You can close the Show Table window when you have finished adding all the tables that you want to relate to one another.

6. **Click Close to close the Show Table dialog box.**
   You’re ready to start relating the tables you added. Relating tables may sound difficult but it’s really nothing more than dragging and dropping the field you want to use to link one table to the other. Before you can drag and drop the matching field from one table to the other you have to make sure that the linking fields in both tables are visible.

7. **Drag the CustomerID field from the tblCustomers table to the CustomerID field in the tblCustomerTours table, as shown in Figure 1-5.**
   Dragging a field from one table to another in the Relationships window links the two tables using the selected field.

   **NOTE:** OK so dragging and dropping isn’t quite that easy. Access is very picky about where you point, click, drag, and drop. You need be very accurate and drag the pointer right next to the field you’re linking to.

   The Relationships dialog box appears as shown in Figure 1-4. What’s especially important here is the Enforce Referential Integrity check box. Referential integrity helps you avoid “orphan” records and maintains database accuracy. For example, checking the Referential Integrity box would ensure that you could not enter an invoice for a customer in an Invoice table unless that same customer existed in a tblCustomers table. We’ll discuss referential integrity more later in the chapter.

8. **Click Create to create the relationship between the Customers and tblCustomerTours tables.**
   The Relationships dialog box closes and a line appears between the two tables’ CustomerID fields, indicating that the tables are linked, as shown in Figure 1-6.

Congratulations! You’ve just linked two related tables together!
Lesson 1-3: Enforcing Referential Integrity

When you create a relationship between two tables, it is usually a good idea to enforce referential integrity. So what does this technical-sounding phrase mean? Referential integrity keeps records in related fields valid and accurate. Referential integrity ensures that you don’t accidentally change or delete related data in one table but not in the other. For example, say you were using two related Social Security fields to link two tables. If you changed the Social Security number in one record, referential integrity would automatically change the Social Security number in the other related records.

Access is very picky about when you can set referential integrity. You can only use referential integrity when all of the following conditions are met:

- One of the linked fields is a primary key
- The related fields are the same data type and size. (If you are using an AutoNumber field you can relate it to a Number field with a Long Integer Field size.)
- Both tables are in the same Access database.
- You can’t have a record in a related table unless a matching record already exists in the primary table. Orphan data in a related table is the most common problem people encounter when attempting to establish referential integrity.

Once you have established referential integrity, the following rules are set:

- You can’t add a record to a related table unless a matching record already exists in the primary table.
Chapter One: Creating Relational Database

- You can’t change the value of a primary key in the primary table if matching records exist in the related table (unless you select the Cascade Update Related Fields option.)
- You can’t delete a record from a primary table if matching records exist in a related table (unless you select the Cascade Delete Related Fields option.)

In this lesson you will learn how to enforce referential integrity. You should still have the Relationships window open from the previous lesson to complete this lesson.

First let’s establish a relationship between the tblTours table and the tblCustomerTours table…

1. Drag the TourID field from the tblTours table to the TourID field in the tblCustomerTours table.
   The Relationships dialog box appears. We want to enforce referential integrity so that you can’t enter a TourID in the tblCustomerTours table unless that TourID number already exists in the tblTours table.

2. Check the Enforce Referential Integrity box.
   This will enforce referential integrity between the tables. If you get an error message it’s because your tables and fields don’t meet all the required conditions listed on the previous page.

   There are two other very important boxes in the Relationships dialog box. They are:
   - **Cascade Update Related Fields**
     When you change data in the main field of one table Access will automatically update the matching data in the related table.
   - **Cascade Delete Related Records**
     When you delete a record in the main table, Access will automatically delete any matching records in the related table.

   These are both very powerful options—think twice before using them.

3. Check both the Cascade Update Related Fields and Cascade Delete Related Records boxes.
   Go ahead and create the relationship.

4. Click Create to create the relationship between the Customers Tours and tblTours tables.
   Access creates the relationship between the two tables and enforces referential integrity between them. Notice the join line between the Customers Tours and tblTours table looks a little different. This relationship indicates that referential integrity is being enforced between the two tables and that the tables have a one-to-many relationship (more about that later).

   You can establish referential integrity in an existing relationship. Here’s how:

5. Double-click the relationship line that connects the tblCustomers table and the tblCustomerTours table.
   **NOTE:** Clicking that tiny little line requires lots of mouse dexterity. You have to place the arrow so that its tip just touches the line before you double-click.

   The Relationships dialog box appears.

6. Uncheck both the Cascade Update Related Fields and Cascade Delete Related Records boxes.
   Now you can save the changes to the relationship.

7. Click OK to modify the relationship between the tblTours table and the tblCustomerTours table.


In the next lesson you will get to test the results of your new referential integrity settings.
In this lesson you will get to test the referential integrity rules you established in the previous lesson. You just went over it, but since this referential integrity stuff is a little tricky and confusing at first let’s review…

Once you have established referential integrity, the following rules are set:

• You can’t add a record to a related table unless a matching record already exists in the primary table.

• You can’t change the value of a primary key in the primary table if matching records exist in the related table (unless you select the Cascade Update Related Fields option.)

• You can’t delete a record from a primary table if matching records existing in a related table (unless you select the Cascade Delete Related Records option.)

Think you have a better understanding referential integrity yet? No? This lesson will give you a chance to work with some related tables where referential integrity has been set. You will also learn how to delete a relationship between two tables.

1. **Open the tblCustomers table in Datasheet view.**
   First let’s try deleting a new record to the tblCustomerTours table.

2. **Click on the record selector for the record with the CustomerID 1 (Antonio Rommero) and click the Delete Record button on the toolbar.**
   Because you enable both referential integrity and cascaded deletes in the previous lesson, Access displays the dialog box shown in [Figure 1-9](#). Access wants you to confirm your deletion, as deleting the customer record will delete any related data for the customer in the tblCustomerTours table.

3. **Click Yes.**
   Access deletes the record from the Customer table and any related tours for the customer in the tblCustomerTours table. If you hadn’t enforced referential integrity between these tables you would be left with one or more “orphan” records for Antonio Rommero in the tblCustomerTours table.
4. Close the tblCustomers table and open the tblCustomerTours table in Datasheet view.

Yikes! The tblCustomerTours table contains a lot of ID number fields, such as CustomerID and TourID. Unless you have a printed list of these ID numbers (or a super-human memory) you won’t know what CustomerID to enter. (By the way this is why you will want to use a lot of forms and queries in relational database—to hide all the technical inner-working of a database from your hapless users.) Anyway let’s try adding a new record to the tblCustomerTours table.

5. Click the New Record button on the Record Navigation bar.

Access adds a new blank record to the table.

6. Enter a new record using the following information:

<table>
<thead>
<tr>
<th>CustomerID</th>
<th>TourID</th>
<th>EmployeeID</th>
<th>Date</th>
<th>Number of Tickets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4/2/00</td>
<td>1</td>
</tr>
</tbody>
</table>

Ship Via: FedEx  Smoker: No  First Class: No  Cost: $450

Before you finish adding the record, it’s very important that you remember that you deleted the CustomerID 1 record back in Steps 2 and 3. Since you enforced referential integrity between the two related CustomerID fields watch what happens when you try to add the record...

7. Finish adding the record by pressing <Ctrl> + <Enter> or clicking in any other record when you have finished entering the records information.

Because a CustomerID 1 doesn’t exist in the tblCustomers table, a dialog box appears informing you that Access can’t add the record because doing so would violate referential integrity.

8. Click OK to close the dialog box.

You can cancel the addition of the new record.

9. Press <Esc> to cancel the new record, then close the tblCustomerTours table.

Now that you have a better feel for how referential integrity works let’s go back to the Relationships window to quickly cover this lesson’s other two topics—printing and deleting table relationships.

10. Click the Relationships button on the toolbar.

Sometimes you may want a printed hardcopy of the Relationship window.

11. Select File → Print Relationships from the menu.

The Print dialog box appears. You don’t actually need to print anything for now, so…

12. Click Cancel.

Sometimes you may want to delete the relationship between two tables. Access is very restrictive about letting you modify a related table, and often you must temporarily delete the relationship between two tables, modify one of the tables, and then re-connect them. Here’s how to delete a table relationship.

13. Click the join line between the tblTours table and the tblCustomerTours table to select it and press <Delete>.

**NOTE:** Clicking a tiny join line between two tables requires a lot of mouse dexterity—you probably will have to try several times before you get it.

Access asks you to confirm the deletion. Let’s leave the table relationship be for now.

14. Click No and close the Relationships window.
Lesson 1-5: Understanding Relationship Types

When you link two tables together, they form one of three possible relationships. This information is rather technical, but it’s good to know if you’re working with related or linked tables. There isn’t a step-by-step exercise in this lesson—all you have to do is look over Table 1-1: Types of Relationships to get a better understanding of table relationships.

### Table 1-1: Types of Relationships

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One to One</td>
<td>Each record in a table relates to one record in another table. The simplest type of relationship, but they don’t occur very often because it’s usually easier to store such information in one table instead of two.  &lt;br&gt;<strong>Example:</strong> Each customer has one credit report.</td>
</tr>
<tr>
<td>One to Many</td>
<td>Each record in a table relates to one or more records in another table. <strong>This is the most common type of relationship.</strong>  &lt;br&gt;<strong>Example:</strong> Each customer has one or more invoices.</td>
</tr>
<tr>
<td>Many to Many</td>
<td>One or more records in a table relates to one or more records in another table. Many to many relationships are very confusing and thus they aren’t used very often. To create a Many to Many relationship you usually need a third intermediate table that contains the primary keys from each of the two tables in the relationship. Such an intermediate table is called a <strong>junction table</strong>.  &lt;br&gt;<strong>Example:</strong> Each sales representative sells several products and each product is sold by several sales representatives.</td>
</tr>
</tbody>
</table>
Chapter One Review

Lesson Summary

Understanding Table Relationships

• Make sure that you have a good understanding of table relationships.

Creating Relationships between Tables

• To Define Table Relationships: In the Database window, click the Relationships button on the toolbar or right-click any blank area in the Database window and select Relationships. If necessary, click the Show Table button on the toolbar, click the table you want and click Add. Repeat for each for table you want to use in the relationship and click Close when you’re finished. Click the related field in the first table and drag it to the related field in the second table, specify any referential integrity and/or join type options (optional). Click Create to create the relationship.

Enforcing Referential Integrity

• To Enforce Referential Integrity: In the Database window, click the Relationships button on the toolbar or right-click any blank area in the Database window and select Relationships. Double-click the join line for the relationship you want to work with and check the Enforce Referential Integrity box. If you want changes to the primary field of the primary table copied to the related field in the related table, check the Cascade Update Related Fields box or if you want Access to automatically delete orphan records in the related table, check the Cascade Delete Related Records box. Then Click OK.

Testing Referential Integrity and Printing and Deleting Table Relationships

• To Print the Relationships Window: n the Database window, click the Relationships button on the toolbar or right-click any blank area in the Database window and select Relationships. Select File → Print Relationships from the menu.

• To Delete a Table Relationship: Open the Relationships window, and click the join line that connects the tables and press <Delete>.

Quiz

1. You can use fields with different data types to link two tables (True or False?)

2. Which of the following statements is NOT true?
   A. Click the Relationships button on the toolbar to display the table relationships in a database
   B. Once you have added the required tables to the Relationships window, you can relate the tables by clicking the related field in the first table and dragging it to the related field in the second table
   C. Referential integrity keeps records in related fields valid and accurate
   D. For most people, table relationships are an easy to understand, straight-forward concept
3. You have created a relationship between a Customers table and a related
Customer Orders table and have established referential integrity in this
relationship, without specifying the cascade delete related records option.
Can you delete a record in the Customer table if it has related records in
the Customer Orders table (Yes or No?)

4. If the Cascade Delete Related Records referential integrity option is
selected, when you delete a record in the main table, Access will
automatically delete any matching records in the related table. (True or
False?)

Homework

1. Open the Homework database.
2. Open the Customers table in Design View. Set the SSN field as the table’s primary
key (if it isn’t already). Save your changes and close the Customers table.
3. Using the Relationship window, establish a relationship between the Customers table
and the Claims table, using the SSN field to join the two tables.
4. Enforce referential integrity for the Customers/Claims relationship.
5. Enable Cascade Update Related Fields and Cascade Delete Related Records for the
Customers/Claims relationship.
7. Close the Homework database.

Quiz Answers

1. False. With the exception of AutoNumber and Number fields, related fields must
always have the same data type in order to be used to join two tables.
2. D. If only this were true! The truth is, understanding how to create and work with
table relationships is a concept that most users struggle with initially.
3. No. Referential Integrity would prevent you from deleting a record in a primary table
if it has records in a related tables (unless the Delete Cascade Related Records option
is selected.)
4. True. The Delete Cascade Related Records option automatically deletes records in
any related tables.
Chapter Two: Working with Forms

Chapter Objectives:

• Create subforms to display information from a one-to-many relationship

Chapter Task: Create and work with subforms

A form created in Access is similar to the ordinary paper forms that you fill out with a pen or pencil—only you don’t have to worry about trying to read bad penmanship. In Access, forms provide an easy way to enter and view data in a table. Here are just a few examples of how forms make working with data easier.

• Easier to View and Use: Instead of scrolling back and forth in a table’s Datasheet, a form lets you focus on one record at a time.

• See Data any Way You Want: You can design forms to present information any way you like.

• Combine Data from Linked Tables: One form can display data from several related tables or queries—and your users will never know that they are working with 2 sources!

And that’s just for starters. No doubt about it—forms make your database easier to use. Just like a Windows dialog box (which is really what a form is), Access forms can include fill-in-the-blank fields, check boxes, drop-down lists, and more.

This chapter explains everything you have ever wanted to know about forms—and maybe a few things you didn’t want to know.

Prerequisites

• How to use menus, toolbars, dialog boxes, and shortcut keystrokes.

• How to open and modify database objects.

• How to add and edit database records.
Lesson 2-1: Creating a Subform

A subform is a form within a form. The primary form is called the main form, and the form within the form is called the subform. Subforms are especially useful when you want to show data from tables or queries with a one-to-many relationship. For example, a Customer form might have a subform that displays each customer’s Orders. Subforms are a great way to display information from a one-to-many table relationship.

The main form and subform are linked so that the subform displays only records that are related to the current record in the main form. For example, when the main form displays a particular customer, the subform displays only orders for that customer.

1. **Open the frmEmployees form in Design View.**
   Usually you will want to have the Control Wizard assist you when you add a subform.

2. **Make sure the Control Wizard button on the Toolbox is pressed in.**
   The Control Wizard is on whenever its button is depressed.

   **NOTE:** If the Toolbox isn’t displayed, click the Toolbox button on the Form Design toolbar.
   Before you add a subform, make sure that you have enough room for it on the main form.

3. **Resize the main form window as needed, so that you have enough room to add the subform.**
   OK! You’re ready to add the subform! Here’s how:
4. Click the **Subform/Subreport button** on the Toolbox.
   The mouse pointer changes to a \( + \), indicating that you can click and drag the subform onto the main form.

5. **Place the \( + \) pointer just below the DOB field. Click and drag the \( + \) pointer down and to the right, until the subform covers most of the bottom half of the main form.**

   The Subform/Subreport Wizard appears and asks if you want to use an existing form for your subform or if you want to build a new form, using tables or queries. In this exercise we will have the Wizard build us a new form using tables and queries to use as our subform.

6. **Click Next.**
   The next step of the Wizard appears. Here you have to select the table or query and fields that you want to display in your subform. We want our subform to display the tours that each employee has sold, so we will select the qryTours query as the source for our subform.

7. **Select Query: qryCustomerTours query from the Tables and Queries combo box.**
   Now you need to select the fields you want to appear in the subform. You must select the related field used to join the main form and subform. This related field must appear on both the main form (where it is called the *parent field*) and on the subform (where it is called the *child field*). In this exercise we will use the EmployeeID that will be used to link the two forms.

   **NOTE:** It’s very important that the underlying tables or queries of the main form and subform have a related field, and that the related field appears somewhere on both forms. See the “Understanding Table Relationships” in the “Working with Tables and Fields” chapter if you’re having trouble understanding this concept.

8. **Select the EmployeeID field and click the \( > \) button to add the field to the Selected fields list.**
   Now that you’ve added the most important field that will link the two forms you can add the remaining fields that you want to appear on the subform.

9. **Add the Date, TourID, CustomerID, and Number of Tickets fields to the Selected fields list.**
   If you’re this far you should be a pro at adding and removing fields using the Wizard. Just make sure that you add the fields in the order specified in Step 8.

10. **Click Next.**
    The next step in the Wizard is to define the fields that link the main form and the subform. The Subform Wizard is often smart enough to recognize the field and use it to link the two forms—as it is in this exercise. If not, you will have to click the Define my own option and select the two related fields.

11. **Click Finish.**
    Access creates the subform and adds it to the main form.

12. **Save your changes and to the frmEmployees form and display it in Form View.**
    The frmEmployees form displays information on each employee and the new subform displays the tours that employee booked.

Subforms created with the Subform/Subreport wizard are usually a little rough around the edges and will require a little clean up work on your part. We’ll learn how to modify and work with a subform in the next lesson.
Lesson 2-2: Modifying and Working with Subforms

Subforms rarely come out the way you want them to the first time: They may be too small or too large, and must be resized so that the main form and subform fit together nicely. If you’re using an existing form as a subform you may need to change the subform layout. Subforms can be displayed using one of three different formats:

- **Datasheet**: Displays multiple records in a table, using one line per record. Tables and queries display their data in Datasheet layout.
- **Single Form**: Displays one record at a time on a form.
- **Continuous Form**: Displays multiple records on a form. The main difference between Datasheets and continuous forms is that you can design and customize continuous forms.

1. **Make sure that you have the frmEmployees form you modified in the previous lesson open in Form view.**
   
   When you add records on a main form or subform Access stores the appropriate data in each table. Usually you won’t even realize that you are working with several tables! The subform you added in the previous lesson has enlarged the main form so that it is large enough to display the subform contents. You will need to resize the form window to display the form properly.

2. **Select Window → Size to Fit Form from the menu.**
   
   Access resizes the window to fit the form. Let’s see how the new subform works.

3. **Click the Next Record Navigation button on the main form.**
   
   Notice that the subform displays tours booked by the current employee. Notice that the main form and sub form each have their own set of navigation buttons that you can use to add and move between records. Try adding a new record to the subform.
4. **Click the New record button on the subform record navigation bar.**
   The record indicator jumps to the blank row at the end of the subform datasheet and the blinking insertion point ( | ) appears in the EmployeeID field. Notice that Access has automatically filled in the EmployeeID field with the main form’s EmployeeID.

   **NOTE:** If you have set referential integrity between two or more related fields in a subform’s underlying table or query you will have to obey those referential integrity rules in order to add or edit a record in the subform. For example, you can’t enter a number in the TourID field unless that number exists in the Tours table.

5. **Enter a new record in the subform using the following information:**
   EmployeeID  Date  TourID  CustomerID  Number of Tickets
   (Current)  9/2/01  China  Ali Chien  2

   It’s easy to switch between the main form and subform: simply click a field or control in the form you want to move to or press <Ctrl> + <Shift> + <Tab>.

6. **Click any field in the main form to move to the main form.**

7. **Switch to Design View.**
   First get rid of that annoying subform label.

8. **Select the label for the subform and press <Delete>.**
   In Access 2000 you can directly modify subforms from the main form (previously you had to work with subforms in a separate window.)

9. **In the subform, resize the text fields and corresponding text labels for the TourID, CustomerID, and Number of Tickets fields.** Then move the resized text fields and corresponding text labels closer together.
   Compare your subform to the one in Figure 2-15. Now let’s resize the subform.

10. **Resize the subform window as needed, then click and drag the right edge of the subform to the left to eliminate the wasted, empty space.**
    Here’s how to change the layout of a subform:

11. **Double-click the subform’s form selector in the upper-left corner of the subform.**
    The Properties dialog box appears.

12. **Click the Format tab, click in the Default View box, click the ^ button and select Datasheet.** Close the Properties dialog box.
    Now you need to resize the subform control and the main form.

13. **Resize the subform control so that it fits nicely in the main form, and then drag the right edge of the main form to resize it as well.**
    Let’s see how our form looks.

14. **Display the frmEmployees form in Form View.**
    The subform is displayed in continuous form layout instead of Datasheet layout.

15. **Save your changes to the form and exit Microsoft Access.**
    Whew! We just covered a lot of ground in this chapter. Try moving on to the Chapter Review to see how much you remember.
Chapter Two Review

Lesson Summary

Creating a Subform

- **To Create a Subform:** Display the form in **Design View**, click the **Toolbox button** on the toolbar if necessary and make sure the **Control Wizard button** on the Toolbox is selected. Resize the form if necessary, then click the **Subform/ Subreport button** on the Toolbox and click and drag where you want the subform to appear in the form. Click **Next** and specify the table or query you want to use for the subform and select the fields you want to appear in the subform. Click **Next**, specify the parent and child fields that link the main form and subform if necessary and click **Finish**.

Modifying and Working with Subforms

- **To Modify a Subform:** Display the main form in **Design View** and double-click the subform. Edit the subform as needed. When you’re finished close the subform and save your changes.
Chapter Three: Working with Queries

Chapter Objectives:
- Understand the various types of queries and their purposes
- Create queries based on more than one table

Chapter Task: Create queries that analyze and manipulate multiple tables

Queries are the stars of Microsoft Access. Queries make sense out of all thousands of jumbled records and display exactly what you need to know. Queries discover things like what the average price of tea in China is or which customers bought the most parakeet food from your company. Queries can even make widespread changes to the records in your database without wearing out your mouse and keyboard! For example, a delete query can automatically delete a whole bunch of records that meet your criteria.

In this chapter you will learn how to harness the power of queries. First you will learn about all the different types of queries: simple select queries, parameter queries that prompt you for more information, crosstab queries that summarize records in an easy understand format, action queries that actually modify the records in your database.

All this power comes with a price tag: many people find that queries are one of the more difficult database objects, and learning how to fully utilize queries isn’t something you can learn in an afternoon. By the time you finish this chapter, however, you will have a good start towards understanding and mastering queries.

Prerequisites
- How to use menus, toolbars, dialog boxes, and shortcut keystrokes.
- How to open and modify database objects.
- A basic understanding of queries: how to specify criteria and sort information.
Lesson 3-1: A Quick Review

Before we start tackling the functions and types of queries, let’s take a step back and review. This lesson is more of a “cheat sheet” than an exercise. Hopefully it will help you remember what you already know about queries.

Table 3-1: Common Criteria Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>=&quot;MN&quot;</td>
<td>Finds records equal to MN</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>&lt;&gt;&quot;MN&quot;</td>
<td>Finds records not equal to MN</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;10</td>
<td>Finds records less than 10</td>
</tr>
<tr>
<td>&lt;=</td>
<td>&lt;=10</td>
<td>Finds records less than or equal to 10</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;10</td>
<td>Finds records greater than 10</td>
</tr>
<tr>
<td>&gt;=</td>
<td>&gt;=10</td>
<td>Finds records greater than or equal to 10</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>BETWEEN 1/1/99 AND 12/31/99</td>
<td>Finds records between 1/1/99 AND 12/31/99</td>
</tr>
<tr>
<td>LIKE</td>
<td>LIKE &quot;S*&quot;</td>
<td>Finds text beginning with the letter S. You can use LIKE with wildcards such as *</td>
</tr>
<tr>
<td>NOT</td>
<td>NOT &quot;MN&quot;</td>
<td>Finds records not equal to MN</td>
</tr>
<tr>
<td>IS NULL</td>
<td>IS NULL</td>
<td>Finds records whose fields are empty</td>
</tr>
<tr>
<td>IS NOT NULL</td>
<td>IS NOT NULL</td>
<td>Finds records whose fields contain values.</td>
</tr>
</tbody>
</table>
Lesson 3-2: Creating a Multiple Table Query

In Access you will often need to look at and analyze information that comes from not one but several different tables. Since Access is a relational database, it's easy to establish a relationship between two or more tables and to look at the information that goes together.

Just like it sounds, a multiple-table query blends together information from two or more related tables. Working with a multiple-table query usually isn't much different than a single-table query. You tell Access which tables you want to use in your query and specify the fields and criteria you want to see. The main difference between a multiple-table query and a single-table query is that with multiple-table queries, Access creates a link between related tables. In query design view, this link (called a join) appears as a line that connects two more tables.

When you create a multiple-table query Access will usually link or join the tables automatically. Sometimes, however you will have to manually join two tables in the Query design window. You manually join two tables by dragging from one field list to the other matching field in the field list for the other table. If the tables don’t have any fields in common, you must add another table to act as a bridge between them. This lesson will give you some experience creating a multiple-table query.

1. Open the Lesson 7 database.
   You should already know how to create a query by now…
Quick Reference

To Create a Multi-Table Query in Design View:
1. Click the Queries icon in the Objects bar then double-click the Create query in design view icon.

2. Click the Show Tables button if necessary. Select the table or query you want to add and click OK.

3. Repeat Step 2 as necessary for additional tables or queries. Click Close when you’re finished.

4. If Access doesn’t automatically join the tables click the related field in the first table and drag it to the related field in the second table. Repeat as necessary to connect all the tables.

5. Double-click each field you want to include form the field list.

6. In the design grid, enter any desired search criteria for the field in the Criteria box.

7. Click the Sort box list arrow for the field and select a sort order.

8. Close the query window.

9. Click Yes to save the query, enter a query name, and then click OK.

10. Click the Date field’s Sort Row and select Ascending.

11. Save the Query as qryFirstQuarterTours.

12. Click the Run button on the toolbar.

13. Close the Query.

2. Click the Queries icon in the Objects bar then double-click the Create query in design view icon.

   The Query design window and Show Tables dialog box both appear. You have to select the tables and/or queries you want to use.

3. Click the tblEmployees table and click Add.

   A field list for the tblEmployees table appears in the top half of the Query design window. You also want to add the tblCustomerTours table and the tblTours table to the query.

4. Add the tblCustomerTours table and tblTours table to the query.

   When you have finished adding the tables and/or queries to your query you can close the Show Tables dialog box.

5. Click Close.

   If the two tables are related, Access will automatically connect their common fields with a join line. For example, Access automatically joins the tblTours table and tblCustomerTours tables because they are already related. If the tables aren’t related you will have to manually join the tables by dragging from one field list to the other matching field in the field list from the other table.

   You need to connect the tblEmployees table with the tblCustomerTours table.

6. Click on EmployeeID in the tblEmployees field list and drag and drop it onto Employee in the tblCustomerTours field list.

   NOTE: You can remove a join from a query by clicking the join line (careful—there’s not much there to click!) and pressing <Delete>.

   Next you need to specify the fields you want to appear in the query results. You can add fields to the query design grid in two ways:
   - By double-clicking the field on the field list.
   - By clicking and dragging the field down to the design grid yourself.

   Because field lists don’t have much room to display their contents you may have to scroll up or down the list in order to find the field you want.

7. Double-click the LastName and FirstName fields in the tblEmployee field list.

   Access adds the LastName and FirstName fields from the tblEmployees table to the design grid. Next add the fields for the Tour and tblCustomerTours tables.

8. Double-click the TourName field in the tblTour field list and the Date and Cost fields in the tblCustomerTours field list.

   Next you need to specify any criteria for the query. For this exercise you want to see tours from the first quarter of the year.

9. Click the Date column’s Criteria row and type Between 1/1/00 and 3/31/00.

   You want to sort your query by date, so…

10. Click the Date field’s Sort Row and select Ascending.

    That’s it! You’ve just created a multiple-table query.

11. Save the Query as qryFirstQuarterTours.

    OK—let’s run our new query!

12. Click the Run button on the toolbar.

    Access displays the results of the query. The results show the names of the employees who sold tours and the names, dates, and costs of the tours.

13. Close the Query.
Chapter Three Review

Lesson Summary

A Quick Review

- **To Create a Query in Design View:** From the database window, click the **Queries tab**, click **New**, select **Design View** from the list and click **OK**. Select the table or query you want to use and click **OK**. Repeat as necessary for additional tables or queries you want to add to the query and click **Close** when you're finished. Double-click the fields that you want to appear in the query or click and drag the fields onto the design grid. In the design grid, enter any desired search criteria for the field in the **Criteria box** and/or click the **Sort box list arrow** for the field and select a sort order. Click the **Save button** on the toolbar and give a name to the query.

Creating a Multiple Table Query

- **To Create a Query in Design View:** From the database window, click the **Queries tab**, click **New**, select **Design View** from the list and click **OK**. Select the table or query you want to use and click **OK**. Repeat as necessary for additional tables or queries that you want to add to the query and click **Close** when you're finished. If Access doesn't automatically join the tables click the related field in the first table and drag it to the related field in the second table. Repeat as necessary to connect all the tables. Double-click the fields that you want to appear in the query or click and drag the fields onto the design grid. In the design grid, enter any desired search criteria for the field in the **Criteria box** and/or click the **Sort box list arrow** for the field and select a sort order. Click the **Save button** on the toolbar and give a name to the query.