Microsoft®

Establishing a Relationship in Access 2002
Student Edition

The Richard Stockton College of New Jersey
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.

© 2002 by CustomGuide, Inc. 1502 Nicollet Avenue South, Suite 1; Minneapolis, MN 55403

This material is copyrighted and all rights are reserved by CustomGuide, Inc. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of CustomGuide, Inc.

We make a sincere effort to ensure the accuracy of the material described herein; however, CustomGuide makes no warranty, expressed or implied, with respect to the quality, correctness, reliability, accuracy, or freedom from error of this document or the products it describes. Data used in examples and sample data files are intended to be fictional. Any resemblance to real persons or companies is entirely coincidental.

The names of software products referred to in this manual are claimed as trademarks of their respective companies. CustomGuide is a registered trademark of CustomGuide, Inc.
<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>16</td>
</tr>
</tbody>
</table>
Chapter One:
Creating Relational Databases

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!

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 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 of 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 to 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, a lot of practice, and a strong understanding of tables and fields.

**Figure 1-1**
This database tracks customers and their orders in two separate but related tables.

**Figure 1-2**
Sketch out a diagram of your database, including its tables and how they relate to each other.
Microsoft Access can create either type of database—flat file or relational. However, most Access databases tend to be of the relational type.

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 reenter a customer’s name and address each time that customer places 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. Make 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, Item 3, and so on, you should probably break each item up into its own table.

- **Identify and Add the Fields Common to Each Table**
  In Figure 1-1 the Customers 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 Customers 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 two sides of the relationship by drawing a “1” on the “one” side of the relationship line and an “∞” (infinity symbol) on the “many” side of the relationship line.

All this writing and planning may seem like a lot of work, but they’re both critical steps in creating a sound database. Carpenters 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 Relationships window, shown in Figure 1-6. The Relationships 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 try to link two tables is caused by fields with different data types and/or sizes.

- **The primary key in one table is usually linked with a matching field in the other table.**
  Notice that in Figure 1-6 the tblTours table’s primary key, TourID, links to the TourID field in the tblCustomerTours table.
Chapter One: Creating Relational Databases

• Fields related to an AutoNumber primary key field must be Number fields with the Long Integer Field Size.

Now 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 tblCustomers, tblCustomerTours, and tblTours 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 tblCustomerTours table and tblTours table 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 TourID field from the tblTours table to the TourID 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: Okay, so dragging and dropping isn’t quite that easy. Access is very picky about where you point, click, drag, and drop. You need to be very accurate and drag the pointer right next to the field you’re linking to.

   The Edit 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 Customers table. We’ll discuss referential integrity more later in the chapter.

8. Click Create to create the relationship between the tblTours and tblCustomerTours tables.
   The Edit Relationships dialog box closes and a line appears between the two tables’ TourID 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. Referential integrity would not allow you to change the Social Security number in one record without changing 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.
• 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 Records option).

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

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

1. Drag the CustomerID field from the tblCustomers table to the CustomerID field in the tblCustomerTours table.

   The Edit Relationships dialog box appears. We want to enforce referential integrity so that you can’t enter a CustomerID in the tblCustomerTours table unless that CustomerID number already exists in the tblCustomers 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 Edit Relationships dialog box:

   • **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 box and the Cascade Delete Related Records box.

   Now you can save the changes to the relationship.

4. Click Create to create the relationship between the tblCustomers and tblCustomerTours tables.

   Access creates the relationship between the two tables and enforces referential integrity between them. Notice that the join line between the tblCustomers table and the tblCustomerTours 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).

5. Close the Relationships window and click Yes to save the changes you made.

   In the next lesson you will get to test the results of your new referential integrity settings.

---

**Quick Reference**

To Enforce Referential Integrity:

1. In the Database window, click the Relationships button on the toolbar.

   Or...

   Right-click any blank area in the Database window and select Relationships.

2. Double-click the join line for the relationship you want to work with.

3. Check the Enforce Referential Integrity box.

4. 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.

5. If you want Access to automatically delete orphan records in the related table, check the Cascade Delete Related Records box.

6. Click OK.
Lesson 1-4: Testing Referential Integrity and Printing and Deleting Table Relationships

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 exist in a related table (unless you select the Cascade Delete Related Records option).

Think you have a better understanding of 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 record in the tblCustomers 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 enabled 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 tblCustomers 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 have a super-human memory), you won’t know which CustomerID to enter. (By the way, this is why you will want to use a lot of forms and queries in relational databases—to hide all the technical inner workings 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>EmployeeID</th>
<th>TourID</th>
<th>Number of Tickets</th>
<th>Date</th>
<th>First Class</th>
<th>Smoker</th>
<th>Ship Via</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4/2/00</td>
<td>No</td>
<td>No</td>
<td>FedEx</td>
<td>450</td>
</tr>
</tbody>
</table>

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 <Tab> or clicking in any other record when you have finished entering the record’s information.

Because a CustomerID 1 doesn’t exist in the tblCustomers table, a dialog box appears, as shown in Figure 1-10, 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 look at the other two topics in this lesson—printing and deleting table relationships.

10. Click the Relationships button on the toolbar.

Sometimes you may want a printed hard copy of the Relationships window.

11. Select File → Print Relationships from the menu.

A report appears in Print Preview. You don’t actually need to print anything for now, so…

12. Close the report without saving changes.

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>
</table>
| One to One     | Each record in a table relates to one record in another table. This is the simplest type of relationship, but it doesn't occur very often because it's usually easier to store such information in one table instead of two.  
Example: Each customer has one credit report. |
| One to Many    | Each record in a table relates to one or more records in another table. This is the most common type of relationship.  
Example: Each customer has one or more invoices. |
| Many to Many   | One or more records in a table relate to one or more records in another table. Many-to-many relationships are very confusing and thus 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 junction table.  
Example: Each sales representative sells several products, and each product is sold by several sales representatives. |
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: In 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. Modify the report that appears in Print Preview, if necessary, and then click the Print button on the toolbar.

- To Delete a Table Relationship: Open the Relationships window, click the join line that connects the tables, and press <Delete>. Click Yes to confirm the deletion.
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, straightforward 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 the Cascade Delete Related Records option. Can you delete a record in the Customers 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?)

5. How can you display the relationships in a database?
   A. Select View → Relationships from the menu.
   B. Click the Relationships button in the Database window.
   C. Click the Relationships button on the toolbar.
   D. Select Edit → Relationships from the menu.

6. How can you add a table to the Relationships window?
   A. Select Edit → Add Table from the menu.
   B. Click the Show Table button on the toolbar.
   C. Select the table from the Table list on the toolbar.
   D. Select Tools → Add Table from the menu.

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 Insurance Claims table, using the SSN field to join the two tables.
4. Enforce referential integrity for the Customers/Insurance 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 table (unless the Cascade Delete Related Records option is selected).

4. True. The Cascade Delete Related Records option automatically deletes records in any related tables.

5. C. Click the Relationships button on the toolbar to display the relationships in a database.

6. B. You can add a table to the Relationships window by clicking the Show Table button on the toolbar.