100 **Part 1** *Database Fundamentals*

FIGURE AW-2-14

The Database Tools Command Tab
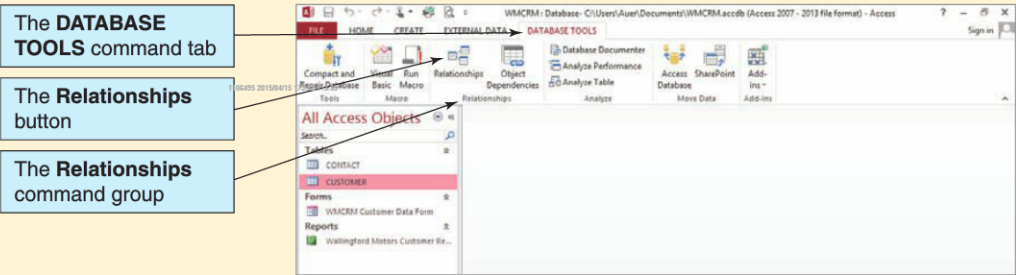
8. Click the **Enforce Referential Integrity** check box.
9. Click the **Create** button to create the relationship between CUSTOMER and CONTACT. The relationship between the tables now appears in the Relationships window, as shown in Figure AW-2-18.
10. To close the Relationships window, click the **Close** button in the upper-right corner of the document window. A Microsoft Access warning dialog box appears, asking whether you want to save changes to the layout of relationships. Click the **Yes** button to save the changes and close the window.

At this point, we need to add data on customer contacts to the CONTACT table. Using the CONTACT table in Datasheet view, as discussed earlier, we enter the data shown in Figure AW-2-1 into the CONTACT table. Again, note that there is *no* customer
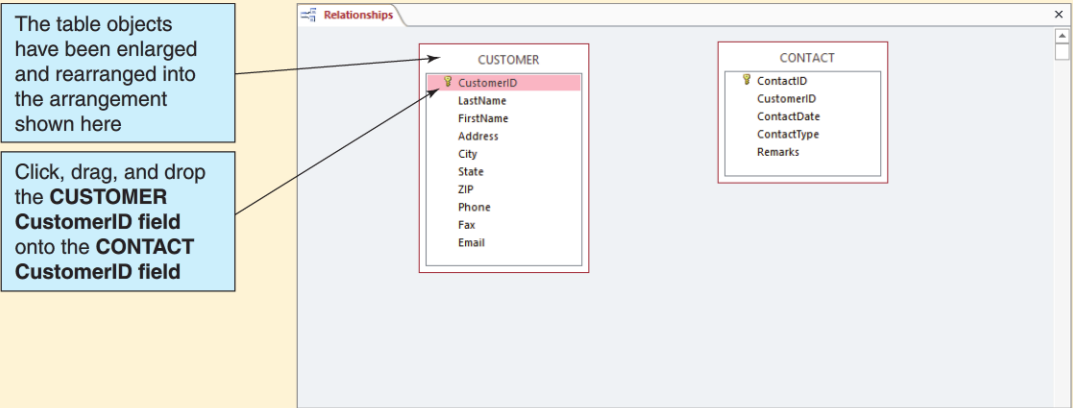
FIGURE AW-2-15
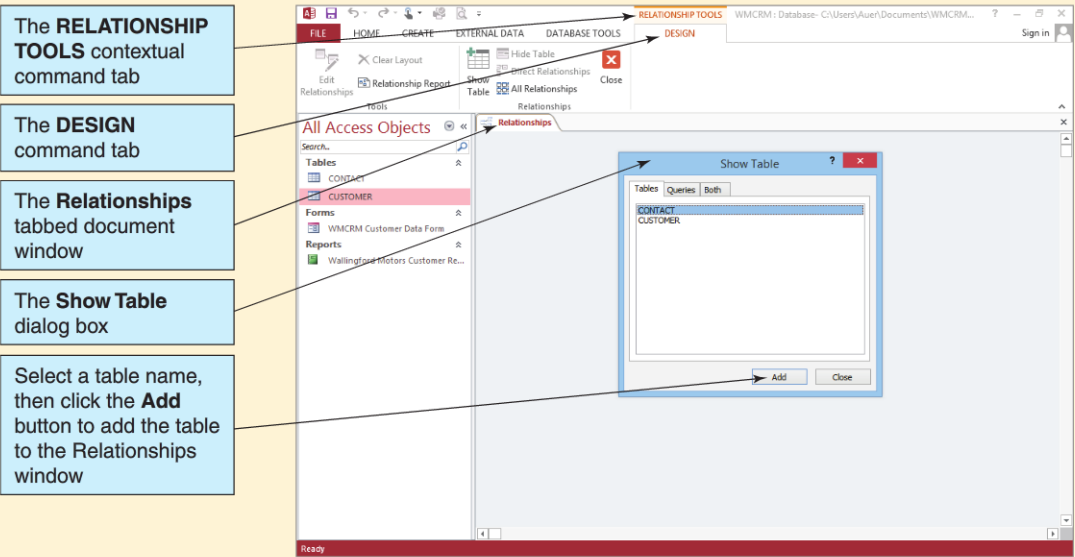
The Relationships Window

FIGURE AW-2-16

The Table Objects in the Relationships Window

with CustomerID of 2—this is because we deleted and reentered the data for Jessica Christman in Chapter 1's section of "The Access Workbench." Also note that because referential integrity is enabled, we *cannot* enter a CustomerID that does not already exist in the CUSTOMER table. The CONTACT table with the data inserted looks as shown in Figure AW-2-19. Be sure to close the table after the data have been entered.

FIGURE AW-2-17

The Edit Relationships Dialog Box
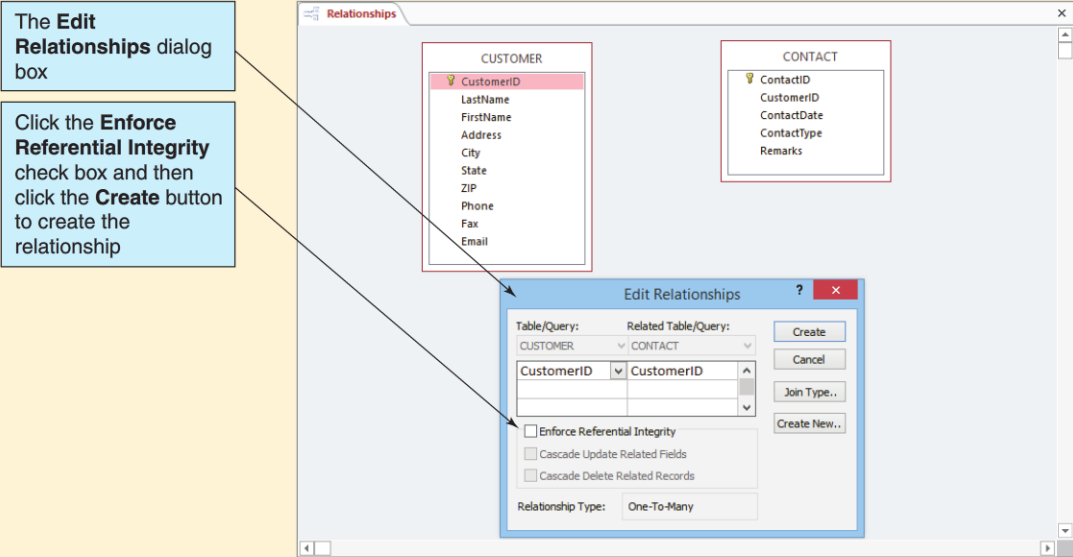
(Continued)

FIGURE AW-2-18

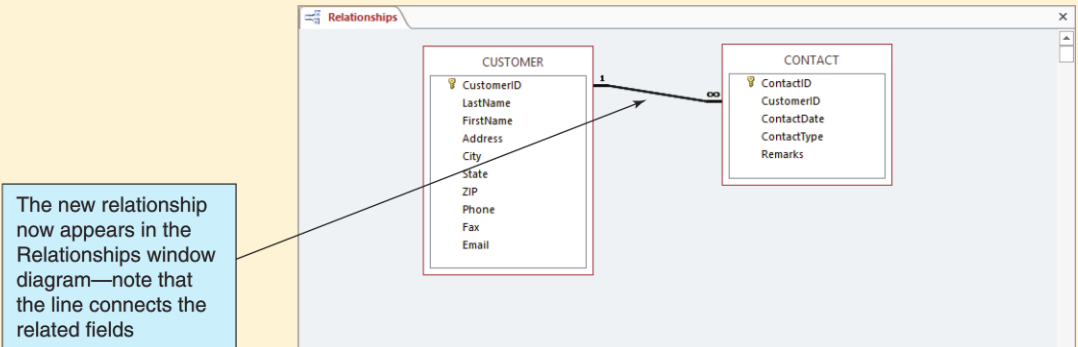**The Completed Relationship**

The new relationship now appears in the Relationships window diagram—note that the line connects the related fields

Using a Form That Includes Two Tables

In Chapter 1's section of "The Access Workbench," we created a data entry form for the CUSTOMER table. Now we will create a Microsoft Access form that will let us work with the combined data from both tables.

Creating a Form for Both the CUSTOMER and CONTACT Tables

1. Click the **Create** command tab.
2. Click the **Form Wizard** button in the Forms command group. The Form Wizard appears.
3. Select the **CUSTOMER** table in the Tables/Queries drop-down list. To add all the columns, click the **right-facing double-chevron** button. Do *not* click the **Next** button yet.

FIGURE AW-2-19

Data in the CONTACT Table

ContactID	CustomerID	ContactDate	ContactType	Remarks
1	1	7/7/2014	Phone	General interest in a Gaea.
2	1	7/7/2014	Email	Sent general information.
3	1	7/12/2014	Phone	Set up an appointment.
4	1	7/14/2014	Meeting	Bought a HiStandard.
5	3	7/19/2014	Phone	Interested in a SUHi, set up an appointment.
6	1	7/21/2014	Email	Sent a standard follow-up message.
7	4	7/27/2014	Phone	Interested in a HiStandard, set up an appointment.
8	3	7/27/2014	Meeting	Bought a SUHi.
9	4	8/2/2014	Meeting	Talked up to a HiLuxury. Customer bought one.
10	3	8/3/2014	Email	Sent a standard follow-up message.
11	4	8/10/2014	Email	Sent a standard follow-up message.
12	5	8/15/2014	Phone	General interest in a Gaea.
(New)	0			

Record: 1 of 12   No Filter   Search

FIGURE AW-2-20

The Completed Form for CUSTOMER and CONTACT Data

4. Select the **CONTACT** table in the Tables/Queries drop-down list. Individually select and add the **ContactDate, ContactType**, and **Remarks** columns to the Selected Fields list by using the **right-facing single-chevron** button. Now click the **Next** button.

   ■ **NOTE:** You have just created a set of columns from two tables that you want to appear on one form.

5. When asked "How do you want to view your data?" use the default **by CUSTOMER** selection because we want to see all contacts for each customer. Also use the selected **Forms with subforms** option to treat the CONTACT data as a subform within the CUSTOMER form. Click the **Next** button.

6. When asked "What layout would you like for your subform?" click the **Next** button to use the default Datasheet layout.

7. When asked "What titles do you want for your form?" type the form title **WMCRM Customer Contacts Form** into the Form: text box and the form title **Contact Data** into the Subform: text box. Click the **Finish** button. The completed form appears.

8. Click the **Shutter Bar Open/Close** button to minimize the Navigation Pane. The completed form is displayed as shown in Figure AW-2-20.

9. Click the **Shutter Bar Open/Close button** to expand the Navigation Pane.

10. Close the form window.

Creating a Report That Includes Data from Two Tables

In this section, we will create a report that includes data from two or more tables. This Microsoft Access report will let us use the combined data from both the CUSTOMER and CONTACT tables.

(Continued)

Creating a Report for Both the CUSTOMER and CONTACT Tables

1. Click the **Create** tab.
2. Click the **Report Wizard** button to display the Report Wizard.
3. Select the **CUSTOMER** table in the Tables/Queries drop-down list. One by one, click **LastName, FirstName, Phone, Fax**, and **Email** to select each one, and then click the **right-facing single-chevron** button to add each column to the Selected Fields list. Do *not* click the Next button yet.
4. Select the **CONTACT** table in the Tables/Queries drop-down list. Individually select and add the **ContactDate, ContactType**, and **Remarks** columns to the Selected Fields list by clicking the **right-facing single-chevron** button. Now click the **Next** button.
5. When asked "How do you want to view your data?" click the **Next** button to use the default **by CUSTOMER** selection (in order to see all contacts for each customer).
6. When asked "Do you want to add any grouping levels?" click the **Next** button to use the default nongrouped column listing.
7. We are now asked "What sort order do you want for detail records?" This is the sort order for the CONTACT information. The most useful sorting order is by date, in ascending order. Click the **sort field 1** drop-down list arrow and select **ContactDate**. Leave the sort order button set to **Ascending**. Click the **Next** button.
8. We are now asked "How would you like to lay out your report?" We will use the default setting of stepped layout, but click the **Landscape orientation** radio button to change the report orientation to landscape. Then click the **Next** button.
9. When asked "What title do you want for your report?" edit the report title to read **Wallingford Motors Customer Contacts Report**. Leave the **Preview the report** radio button selected. Click the **Finish** button. The completed report is displayed in Print Preview mode.
10. Click the **Close Print Preview** button to close Print Preview.
11. Click the **Home** command tab.
12. Click the **Shutter Bar Open/Close** button to minimize the Navigation Pane. The completed report is displayed as shown in Figure AW-2-21.

FIGURE AW-2-21

The Wallingford Motors Customer Contact Report

View gallery arrow button—use this button to access Design view if needed

13. Although this may not be the best layout for the report, the Microsoft Access Form Wizard has created a usable report with all columns correctly sized to display the information (if there are any columns that are not correctly displayed use the Layout view in the view gallery to make minor adjustments—this tool can be used to make basic adjustments by simply clicking the report section you want to change). We will discuss how to use report Design view to modify reports in Chapter 5's section of "The Access Workbench."
14. Click the **Shutter Bar Open/Close** button to expand the Navigation Pane.
15. Click the document window's **Close** button to close the report window.

Closing the Database and Exiting Microsoft Access

We have finished the work we need to do in this chapter's "The Access Workbench." As usual, we finish by closing the database and Microsoft Access.

Closing the WMCRM Database and Exiting Microsoft Access

1. To close the WMCRM database and exit Microsoft Access 2013, click the **Close** button in the upper-right corner of the Microsoft Access 2013 window.

SUMMARY

The relational model is the most important standard in database processing today. It was first published by E. F. Codd in 1970. Today, it is used for the design and implementation of almost every commercial database.

An entity is something of importance to a user that needs to be represented in a database. A relation is a two-dimensional table that has the characteristics listed in Figure 2-1. In this book, and in the database world in general, the term *table* is used synonymously with the term *relation*. Three sets of terminology are used for relational structures. The terms *table, row*, and *column* are used most commonly, but *file, record*, and *field* are sometimes used in traditional data processing. Theorists also use the terms *relation, tuple*, and *attribute* for the same three constructs. Sometimes these terms are mixed and matched. Strictly speaking, a relation may not have duplicate rows; however, sometimes this condition is relaxed because eliminating duplicates can be a time-consuming process.

A key is one or more columns of a relation that is used to identify a row. A unique key identifies a single row; a nonunique key identifies several rows. A composite key is a key that has two or more attributes. A relation has one primary key, which must be a unique key. A relation may also have additional unique keys, called candidate keys. A primary key is used to represent the table in relationships, and many DBMS products use values of the primary key to organize table storage. In addition, an index normally is constructed to provide fast access via primary key values. An ideal primary key is short, numeric, and never changes.

A surrogate key is a unique numeric value that is appended to a relation to serve as the primary key. Surrogate key values have no meaning to the user and are normally hidden on forms, query results, and reports.

A foreign key is an attribute that is placed in a relation to represent a relationship. A foreign key is the primary key of a table that is different from (foreign to) the table in which it is placed. Primary and foreign keys may have different names, but they must use the same data types and sets of values. A referential integrity constraint specifies that the values of a foreign key be present in the primary key.

A null value occurs when no value has been given to an attribute. The problem with a null value is that its meaning is ambiguous. It can mean that no value is appropriate, that