

# Appendix M

## MS Access Tutorial

### Preview

After a database has been designed properly it must be implemented and the applications that make the database useful to the end users must be developed. Microsoft Access is a great tool for prototyping a database's implementation and its applications. Therefore, we will show you how to:

- Create the database
- Create and modify tables
- Enter data into the tables
- Import tables
- Define the relationship(s) between the tables
- Create queries
- Create forms
- Create menus linked with macros
- Create reports

### Data Files and Available Formats

**MS Access** **Oracle** **MS SQL** **My SQL**

**MS Access** **Oracle** **MS SQL** **My SQL**

Ch07\_SaleCo



*Data Files Available on [cengagebrain.com](https://cengagebrain.com)*



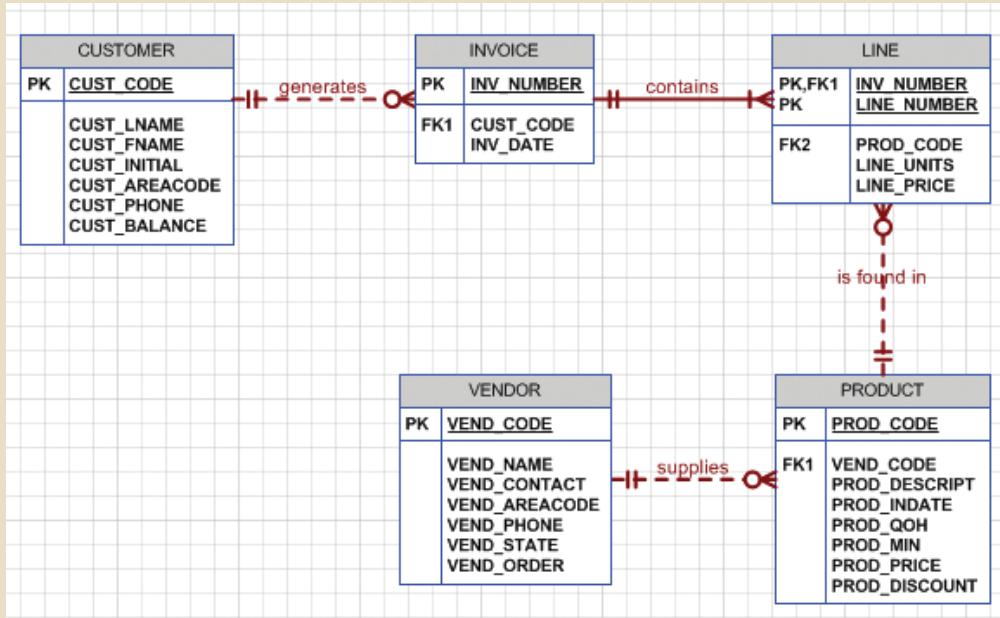
## Note

This tutorial assumes that you know how to perform basic operations in the Microsoft Windows environment.

## M-1 Create the Database

If you have studied our *Database Systems: Design, Implementation, and Management* text, you already know the important ground rule: The database design is the crucial first step in the journey that lets you successfully implement and manage a database. The database design shown in Figure M.1 in the form of an Entity Relationship Diagram (ERD) will be the basis for this tutorial.

**FIGURE M.1 THE SALECO DATABASE ERD**



## Note

If necessary, review the text's Chapters 4 through 6 to ensure that you have a sound basis for database design work. You can study Appendix A1, Designing Databases with Visio Professional 2010: A Tutorial, if you want to create the database design shown in Figure M.1. Appendix A2 shows the steps in Visio 2013.

Given the database design shown in Figure M.1, you are ready to implement it in Microsoft Access. The first step is to start the MS Access DBMS software, which is part of the Microsoft Office Professional suite. Use the same routine you use to open all other Microsoft office components to open Microsoft Access.

After Access opens choose the **Blank desktop database** option.

- When you select the **Blank desktop database** option shown in Figure M.2, Figure M.3 will appear. Note the default folder and notice that the default database name is **Database1.accdb**.
- Select the folder in which you want to store the database, rename the database file to **SaleCo**, and click **Create** as shown in Figure M.4. This option will create a new local Access database.

FIGURE M.2 CREATE A NEW DATABASE

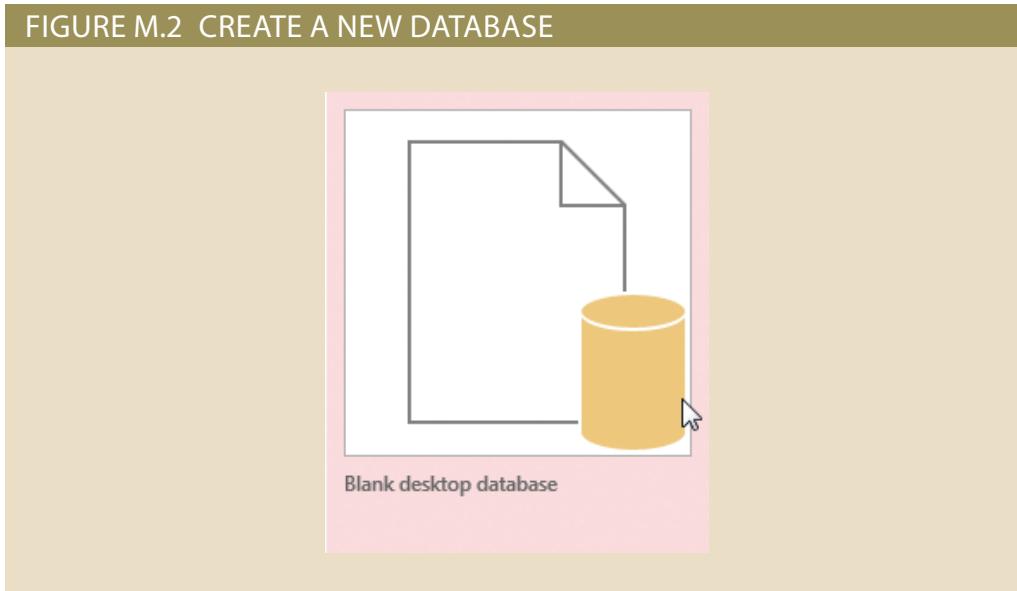


FIGURE M.3 DEFAULT FOLDER AND DATABASE NAME

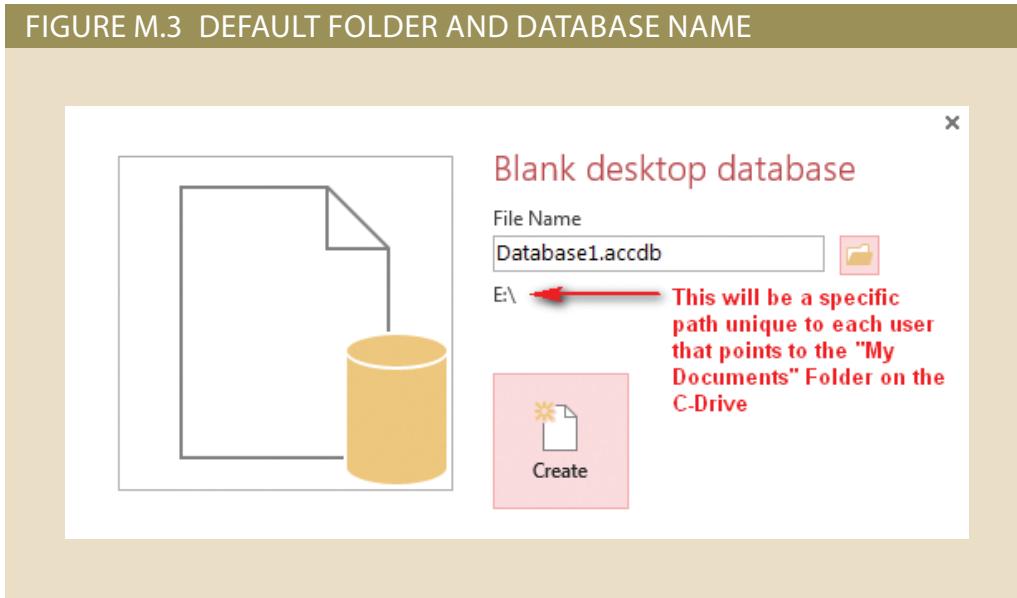
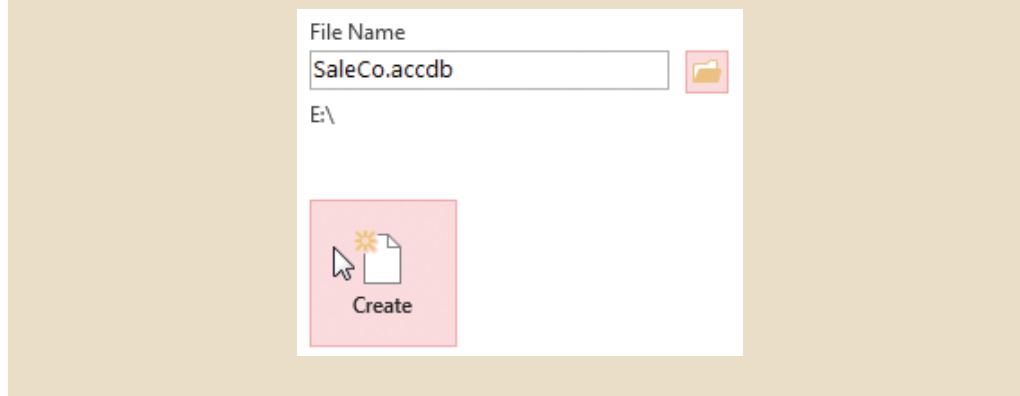
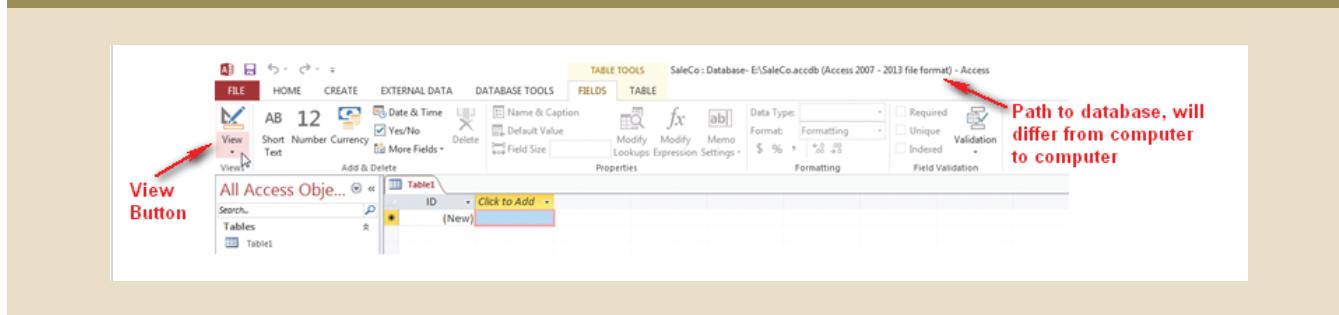


FIGURE M.4 SELECT FOLDER AND RENAME DATABASE



After following the steps outlined above, MS Access will open the **SaleCo** database as seen in Figure M.5. The next section teaches you how to create your first MS Access table.

FIGURE M.5 THE SALECO DATABASE

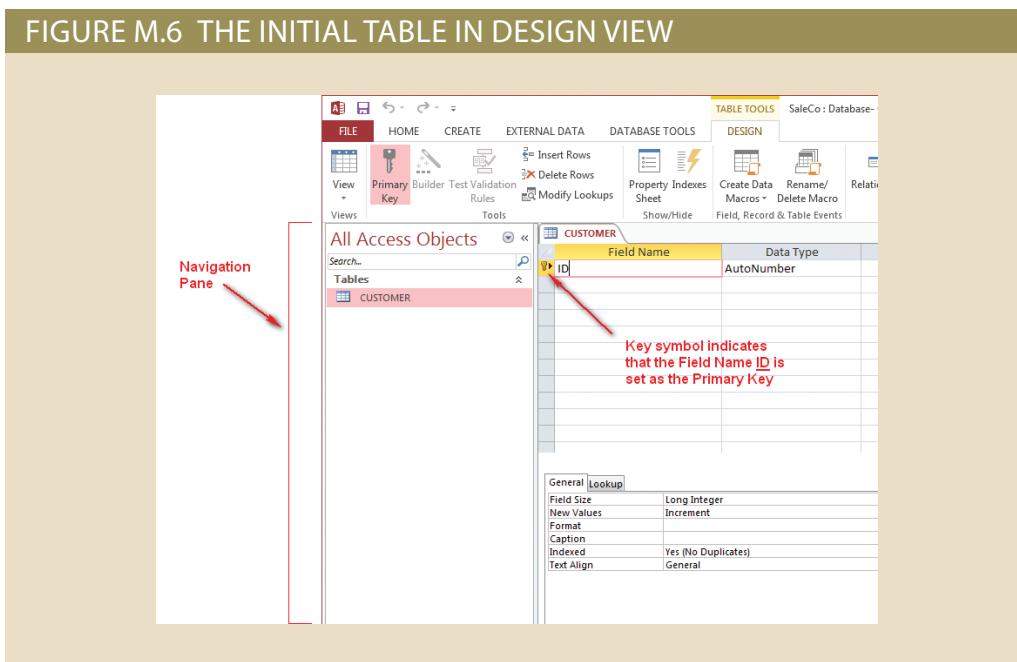


## M-2 Creating the Table Structures

A new table named Table1 appears when the **SaleCo** database is generated.

- Click the **View** button as shown in Figure M.5 to switch to **Design View**. This button switches back and forth from Design View (edit mode) to Datasheet View (entry mode).
- Type **CUSTOMER** in the Save As dialog box and click **OK**.
- The table changes from Table 1 to CUSTOMER. You are now in Design View, which allows you to name and define the fields of the table.
- Access automatically creates a field named ID with an AutoNumber data type as seen in Figure M.6.

FIGURE M.6 THE INITIAL TABLE IN DESIGN VIEW

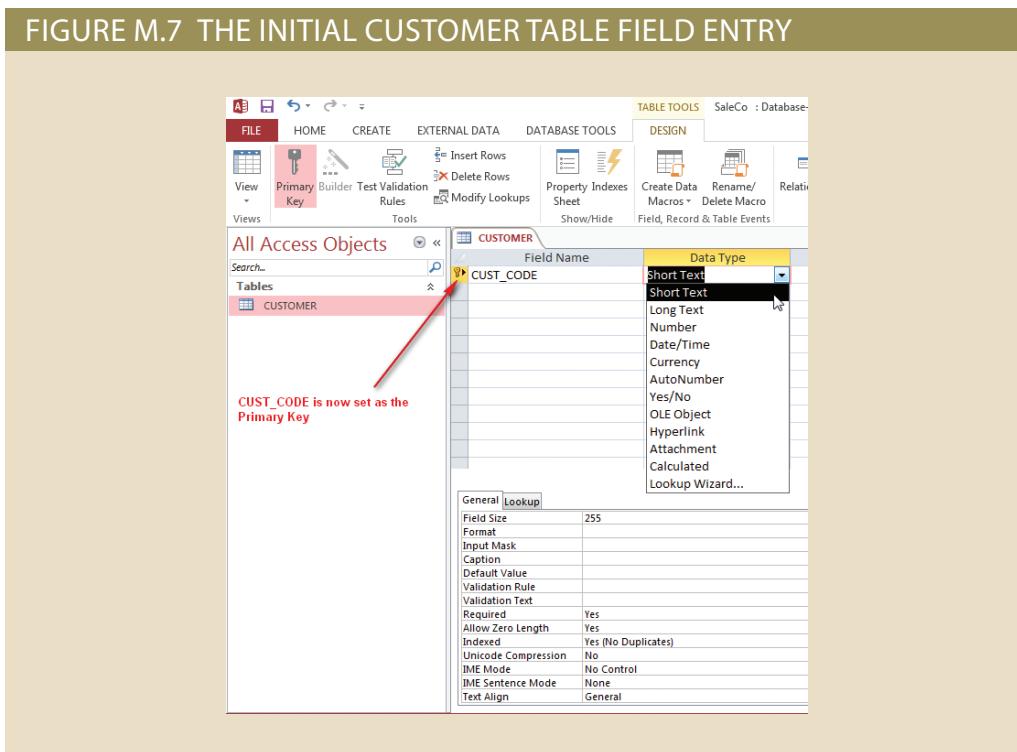


You are now ready to type in the first field name. (See Figure M.1 again to see what fields are included in the CUSTOMER table.)

Rename ID to **CUST\_CODE** and change the data type to **Short Text** as seen in Figure M.7.

The CUST\_CODE values will all consist of a five-character Short Text string, so the **Field Properties** box default **Field Size** of 255 should be reset to **5**. (Just select the “255” and type in the value “5” to get the job done.)

FIGURE M.7 THE INITIAL CUSTOMER TABLE FIELD ENTRY



- As you examine Figure M.7, note that the **Field Properties** box entry for the CUST\_CODE indicates that the **Indexed** default value is set to **Yes (No Duplicates)**.
- Note that this property enforces entity integrity, because no duplicate values will be permitted. This is automatically set when the primary key (PK) is assigned.

Add an optional description explaining what the field is as seen in Figure M.8.

FIGURE M.8 OPTIONAL DESCRIPTION

The screenshot shows the Microsoft Access 'Field Properties' dialog box for the 'CUS\_CODE' field. The 'Field Name' is 'CUS\_CODE', 'Data Type' is 'Short Text', and the 'Description (Optional)' is 'Customer Code, Primary Key for the CUSTOMER Table'.

Field Name	Data Type	Description (Optional)
CUS_CODE	Short Text	Customer Code, Primary Key for the CUSTOMER Table

Using the procedures you have just learned, go ahead and create the remaining attributes for the **SaleCo** database's CUSTOMER table. (Use the ERD in Figure M.1 as your guide.) The completed table is shown in Figure M.9.

- Note that in Figure M.9 the CUST\_BALANCE is a **Currency** data type and that the default value is set to 0. Therefore, the CUST\_BALANCE value will be shown as 0.00.
- Make sure that you use a **Short Text** format for all of the remaining attributes.
- Limit the field lengths for the following attributes as follows: CUST\_LNAME (**20**), CUST\_FNAME (**20**), CUST\_INITIAL (**1**), CUST\_AREACODE (**3**), and CUST\_PHONE(**8**).

Given the self-describing field names, there is little need to continue the entry of more detailed descriptions. For example, the CUST\_LNAME is clearly a customer's last name, so no further description is necessary.

FIGURE M.9 THE COMPLETED CUSTOMER TABLE

The screenshot shows the Microsoft Access 'Field Properties' dialog box for the 'CUST\_CODE' field in the 'CUSTOMER' table. The 'Field Name' is 'CUST\_CODE', 'Data Type' is 'Short Text', and the 'Description (Optional)' is 'Customer Code, Primary Key for the CUSTOMER Table'. A red arrow points to the 'Field Name' column with the text 'Enter these field names'. Another red arrow points to the 'Data Type' column with the text 'Enter these data types'. The 'Default Value' is set to '0'. A tooltip for 'Data Type' says: 'Type determines the kind of values can store in the field. Press F1 for help on data types.' The 'Format' dropdown is set to 'Currency'.

CUSTOMER		CUSTOMER table tab
Field Name	Data Type	Description (Optional)
CUST_CODE	Short Text	
CUST_LNAME	Short Text	
CUST_FNAME	Short Text	
CUST_INITIAL	Short Text	
CUST_AREACODE	Short Text	
CUST_PHONE	Short Text	
CUST_BALANCE	Currency	

General    **Lookup**

Format	Currency
Decimal Places	Auto
Input Mask	
Caption	
<b>Default Value</b>	0
Validation Rule	
Validation Text	
Required	No
Indexed	No
Text Align	General

**Enter these field names**

**Enter these data types**

Type determines the kind of values can store in the field. Press F1 for help on data types.

A table can have many fields, each one with a specific data type. A data type determines the type of values that a field can have. MS Access supports various data types; for a complete description of such data types, see Table M.1.

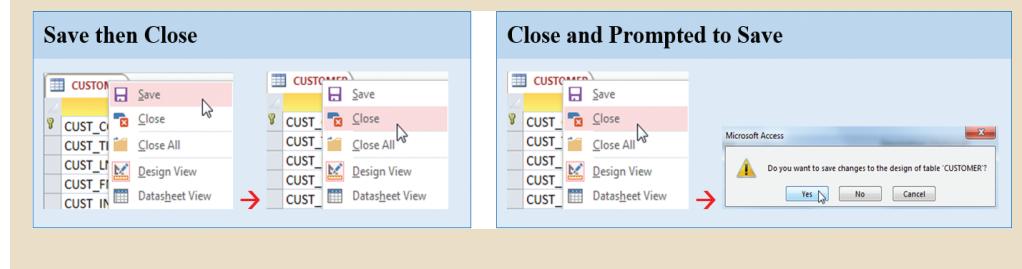
TABLE M.1

**DATA TYPES**

DATA TYPE	PROPERTIES
Short Text	Text or numbers not used in calculations with a maximum field size of 255
Long Text	Text with a field size greater than 255, commonly used for comments or notes
Number	Numeric data that can be used in calculations
Date/Time	Dates and times
Currency	Monetary values including symbols
AutoNumber	Sequential integers that are automatically generated by Access.
Yes/No	Either a Yes or No value (only 2 options)
OLE Object	Object Linking and Embedding (OLE), commonly used for Word documents and Excel files
Hyperlink	Link to web and email addresses
Attachment	External files, commonly used for image files, spreadsheets and documents
Calculated	A calculation resulting from other fields in the table
Lookup Wizard	Creates a lookup field, which displays a list of values you can choose from. Used to match the data types of other fields so that relationships can properly be created. The final data type is either Short Text or Number.

After defining all the CUSTOMER table fields, you are ready to save the table. Right-click the **CUSTOMER** table tab and click **Save** then right-click again and click **Close**. (Note: If you just close the table without saving first, Access will prompt you to save the table before closing. Click **Yes** to save changes. The two options are shown below in Figure M.10.)

FIGURE M.10 SAVE AND CLOSE THE CUSTOMER TABLE



As you saw in Figure M.6, to the left of the table tab there is an area known as the Navigation Pane. The Navigation Pane displays all the objects you have created in the database. There are four main object types in MS Access databases: Tables, Queries, Forms and Reports (see Table M.2). Double-clicking the object's icon in the Navigation Pane will open the specified object. The View button can then be used to change views.

TABLE M.2

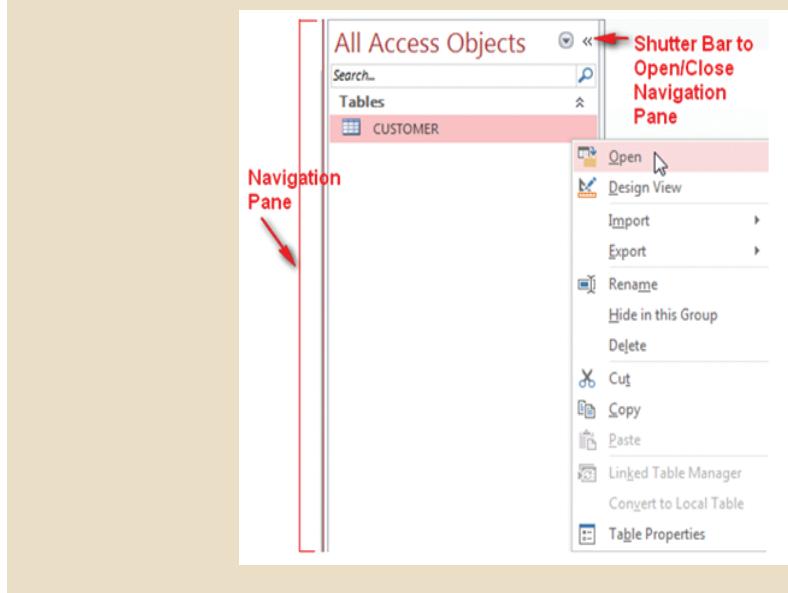
## OBJECTS AND THEIR FUNCTIONS

OBJECT	ICON	FUNCTION
Table		Stores all of the raw data in the database. Tables are linked with a common field (attribute) to reduce data.
Query		A specific request for data manipulation issued by the end user; a question or a task asked by the end user.
Form		Allows the end user to easily enter data.
Report		A printout of the data that can be customized to contain headers, footers, graphics, and calculations on groups.

## M-2a Data Entry

In this section you will learn how to enter and edit data in a table. In the Navigation Pane, right-click the **CUSTOMER** table and click the **Open** option shown in Figure M.11 to generate the table datasheet view shown in Figure M.12. (You can also double-click the CUSTOMER table.)

FIGURE M.11 OPENING THE CUSTOMER TABLE FOR DATA ENTRY



**FIGURE M.12 THE CUSTOMER DATASHEET VIEW**

CUST_CODE	CUST_LNAM	CUST_FNAM	CUST_INITI	CUST_AREA	CUST_PHON	CUST_BALAN
\$0.00						

Enter in the data shown in Figure M.13. Suppose you now try to enter the second record. After accepting the 0.00 CUST\_BALANCE value, you tap the **Enter** key to move the cursor to the second record. Now type in the same **10010** CUST\_CODE value you used for the first record as shown in Figure M.14. Then try to close the CUSTOMER table. Your reward will be the error message shown in Figure M.14, because you violated entity integrity requirements.

**FIGURE M.13 ENTERING THE FIRST CUSTOMER RECORD**

CUST_CODE	CUST_LNAM	CUST_FNAM	CUST_INITIA	CUST_AREA	CUST_PHON	CUST_BALAN
10010	Ramas	Alfred	A	615	844-2573	\$0.00

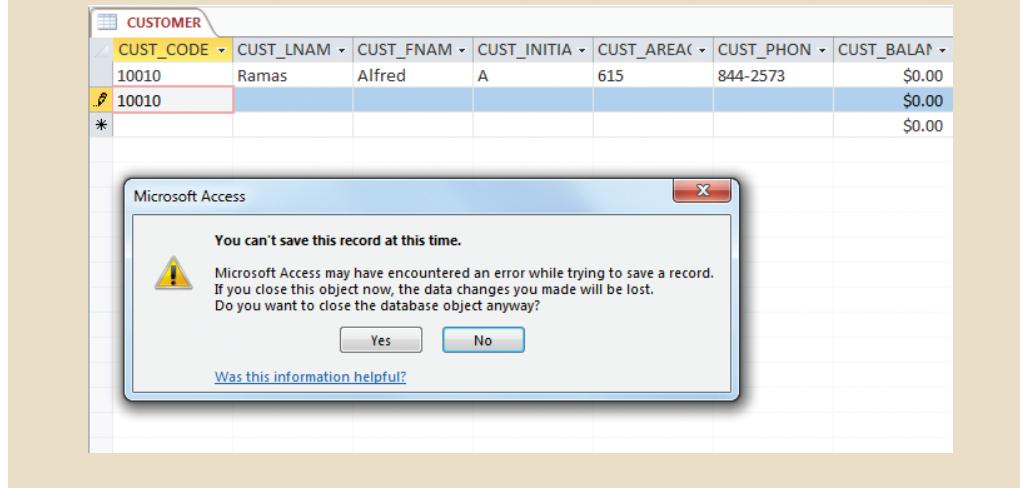
**FIGURE M.14 ENFORCEMENT OF ENTITY INTEGRITY**

CUST_CODE	CUST_LNAM	CUST_FNAM	CUST_INITIA	CUST_AREA	CUST_PHON	CUST_BALAN
10010	Ramas	Alfred	A	615	844-2573	\$0.00
10010						\$0.00

Microsoft Access  
The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship. Change the data in the field or fields that contain duplicate data, remove the index, or redefine the index to permit duplicate entries and try again.  
OK Help  
Was this information helpful?

Click **OK** to acknowledge the error message. This action will generate the result you see in Figure M.15.

**FIGURE M.15 RECORD NOT SAVED WARNING**



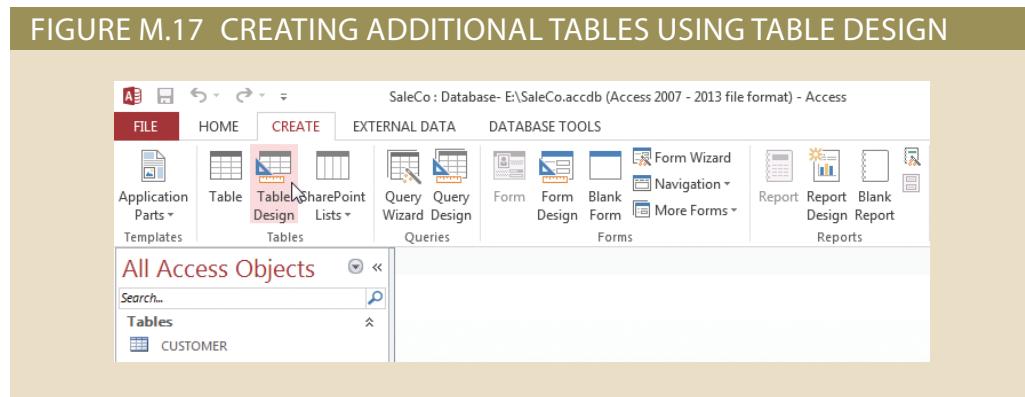
If you click the **Yes** button shown in Figure M.15, Access will return you to the Navigation Pane you saw in Figure M.11. In this case, the second record will not be saved and the table simply retains the first record “as is.” If you click the **No** button, the cursor returns to the data entry screen. This action lets you make the necessary correction by typing in a CUST\_CODE value other than 10010. Click **No** to return to Design View and finishing adding the remaining records.

You are now ready to enter the remaining records. When you are done, the CUSTOMER table contents should match Figure M.16. Save and close the table.

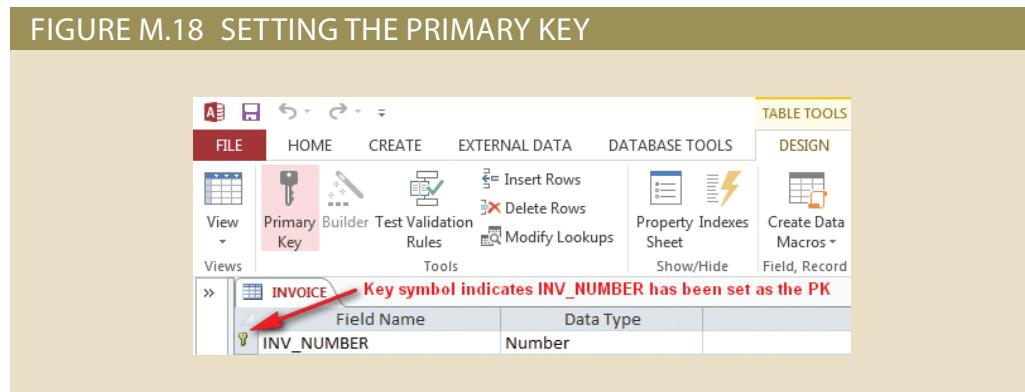
**FIGURE M.16 COMPLETED CUSTOMER TABLE ENTRIES**

CUST_CODE	CUST_LNAM	CUST_FNAM	CUST_INITIA	CUST_AREA	CUST_PHON	CUST_BALAN
10010	Ramas	Alfred	A	615	844-2573	\$0.00
10011	Dunne	Leona	K	713	894-1238	\$0.00
10012	Smith	Kathy	W	615	894-2285	\$162.52
10013	Olowksi	Paul	F	615	894-2180	\$536.75
10014	Orlando	Myron		615	222-1672	\$0.00
10015	O'Brien	Amy	B	713	442-3381	\$0.00
10016	Brown	James	G	615	297-1228	\$221.19
10017	Williams	George		615	290-2556	\$768.93
10018	Farris	Anne	G	713	382-7185	\$216.55
10019	Smith	Olette	K	615	297-3809	\$0.00

The next step is to create the INVOICE table. Click the **CREATE** tab and then **Table Design** as shown in Figure M.17.



In the Field Name column, enter the field name **INV\_NUMBER** and select the **Number** data type. Select the attribute **INV\_NUMBER** and click the **Primary Key** button located on the DESIGN tab as shown in Figure M.18. The narrow column to the left of the **Field Name** column now shows the key symbol to indicate that the INV\_NUMBER attribute is now the PK. You did not have to set this in the CUSTOMER table because Access automatically set the ID field as the primary key. We simply renamed the field. The PK stayed defined in the field. Right-click the **Table1** table tab and click **Save**. Enter **INVOICE** as the Table Name and click **OK**.

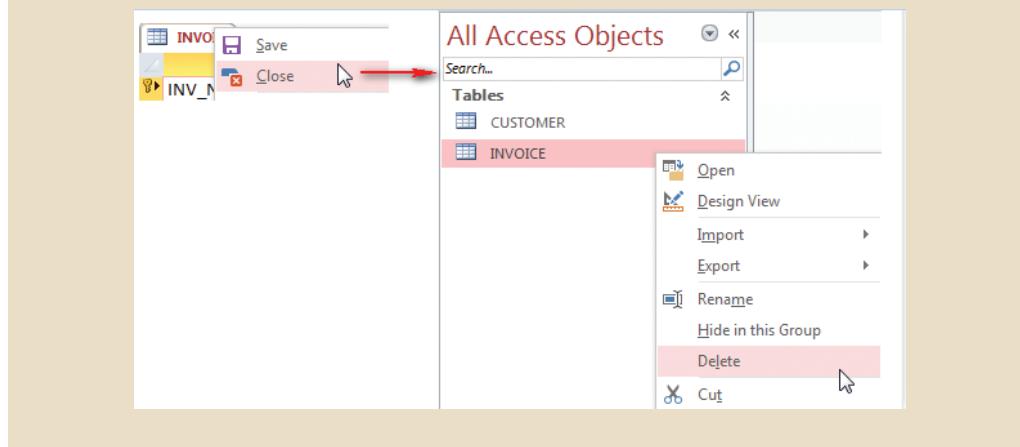


### Note

If you try to save or close the table without assigning a primary key, MS Access will prompt you with a message suggesting to define one. You can save the table without a primary key by clicking No.

Right-click the **INVOICE** table tab and close the INVOICE table; right-click the **INVOICE** icon and select **Delete** as shown in Figure M.19. Then click **Yes**. We will import the remaining tables including the INVOICE table in the next step. This step was just to demonstrate how to create additional tables after the initial Access-generated table is modified.

FIGURE M.19 DELETING THE INVOICE TABLE

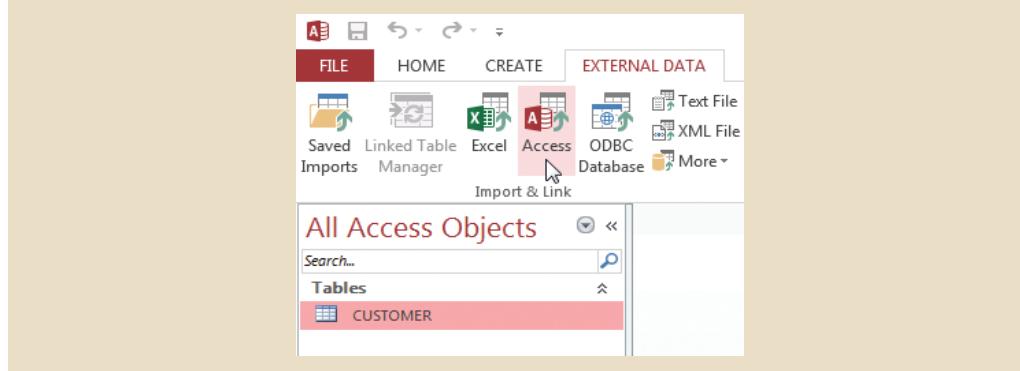


You are now ready to create the remaining tables (INVOICE, LINE, PRODUCT, and VENDOR) and their contents. However, rather than wasting your time on such repetitive tasks, we will show you how to import these items from a database you already have. That database, named **Ch07\_SaleCo**, is used in the text's Chapter 7, Introduction to Structured Query Language (SQL). This database is available online, so go ahead and download it now into a folder of your choice. You will learn how to import database components from the **Ch07\_SaleCo** database to your **SaleCo** database in the next section.

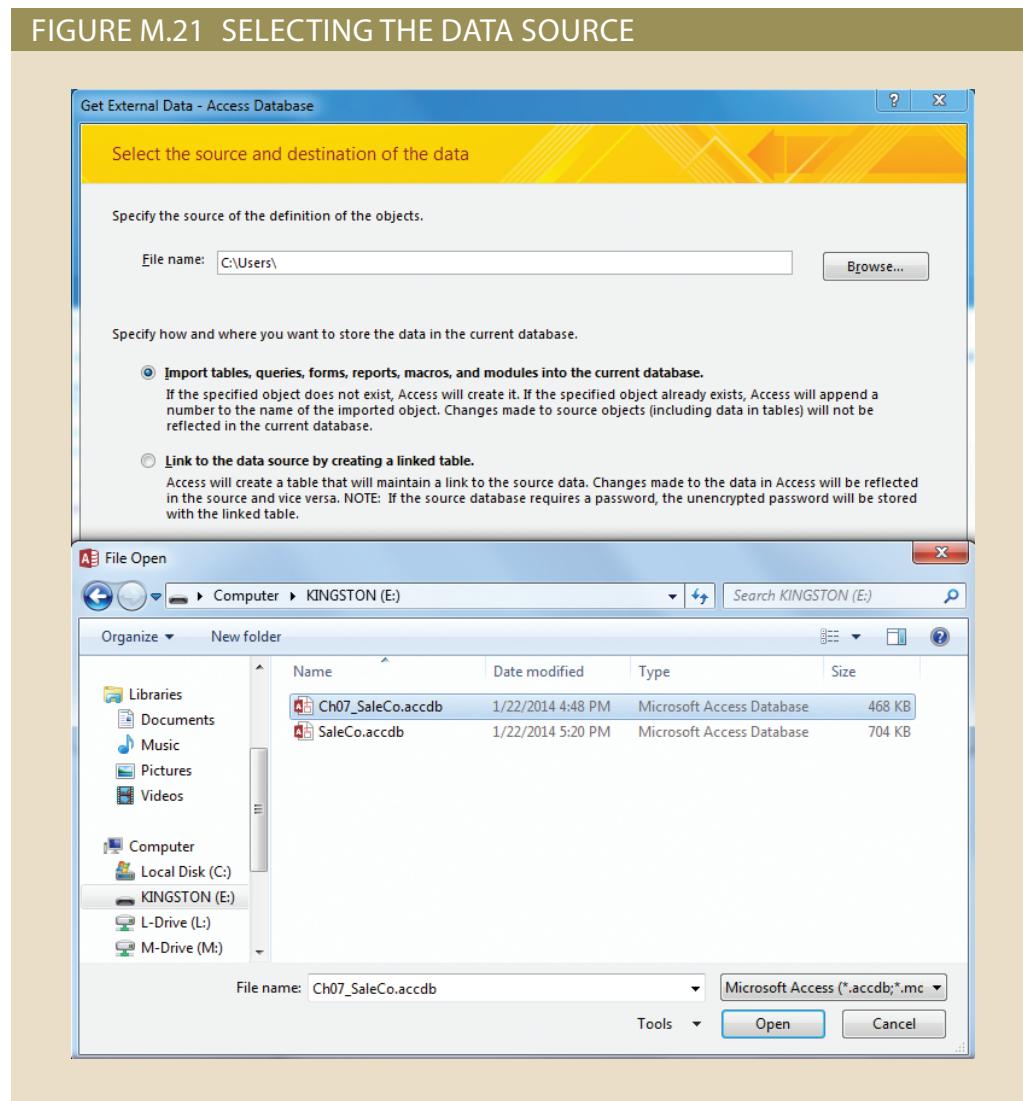
### M-3 Importing Objects from Other MS Access Databases

To import a table—or, for that matter, any other Access object—start by selecting the **EXTERNAL DATA** tab; then click the **Access** button, as shown in Figure M.20.

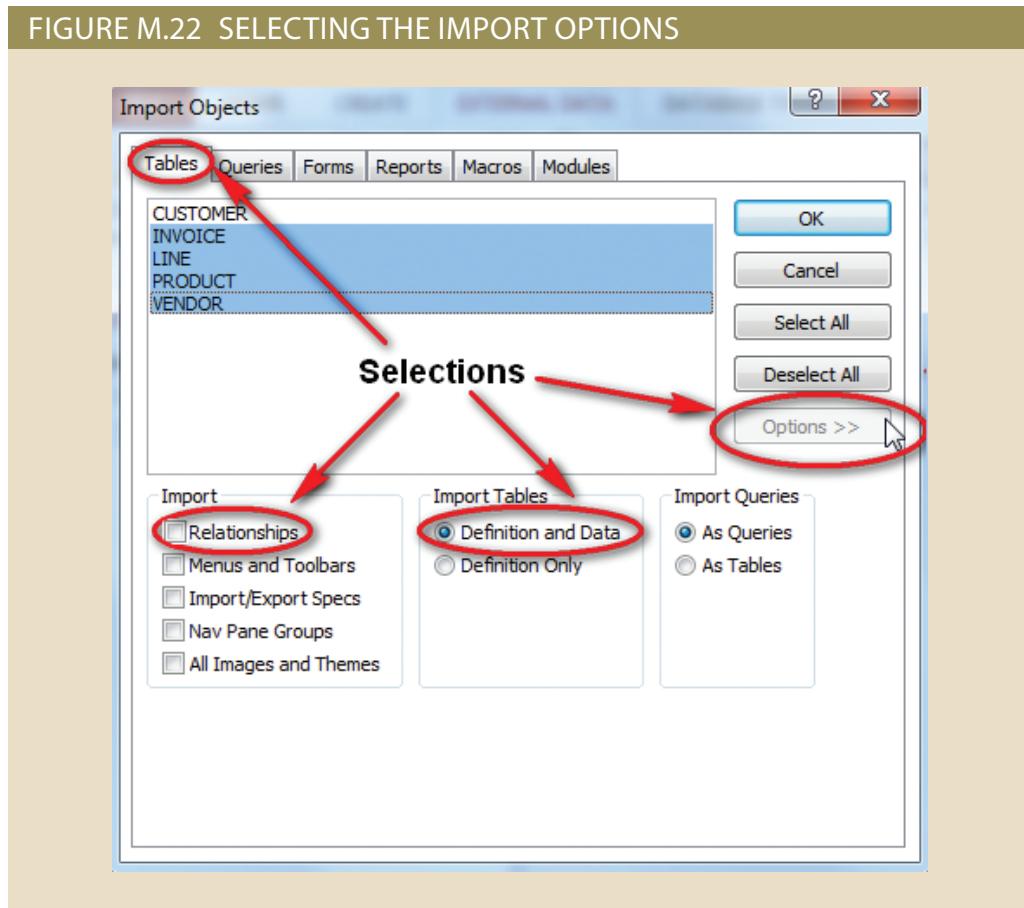
FIGURE M.20 STARTING THE IMPORT SEQUENCE



Next, select the data source. In this example, the **Ch07\_SaleCo** database will be used, so move to the folder on cengagebrain.com (or to your student data folder) that contains this database, select the **Ch07\_SaleCo** database as shown in Figure M.21, and click **Open**.



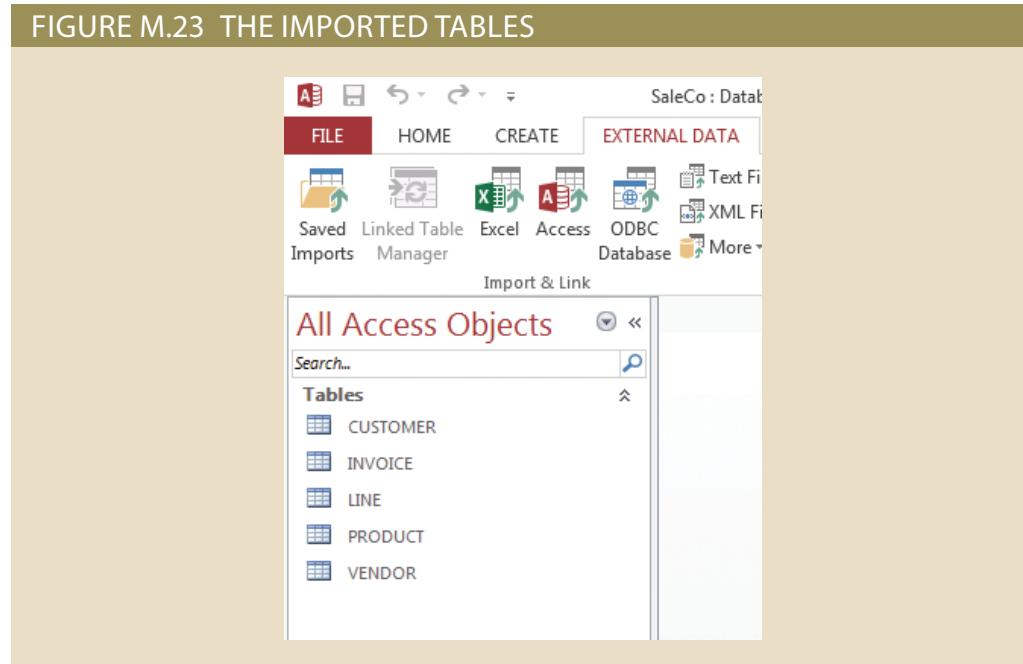
Now select what objects you want to import and in what format you want them imported. Figure M.22 shows what **Tables** are to be imported. Note that the **Options >>** have been selected to give you a chance to make the selections shown in Figure M.22.



As you examine Figure M.22, note the following selections:

- The INVOICE, LINE, PRODUCT and VENDOR tables have been marked.
- The **Relationships** option has been de-selected—no checkmark is shown in the box. (If the box contains a checkmark, click it to remove the checkmark. The reason for not including the relationships is that you will learn how to create such relationships later.)
- The **Definition and Data** option has been selected to ensure that the table structure and all the data contained in each selected table will be imported.

After you have made the selections shown in Figure M.22, click **OK** to import the tables. If you are presented with a Save Import Steps screen, click **Close**. The results of this action are shown in Figure M.23.



### M-3a Editing the Imported Tables

As you can tell by looking at Figure M.23, the imported tables are all there. For example, you can open the INVOICE table in Design View to inspect the table structure (see Figure M.24). Note the INV\_DATE Date/Time data type format.

FIGURE M.24 IMPORTED INVOICE TABLE STRUCTURE

The screenshot shows the Microsoft Access 'INVOICE' table structure. The 'INV\_DATE' field is selected, and its properties are being viewed. A red arrow points to the 'Format' dropdown in the 'Field Properties' dialog, which is set to 'Medium Date'. The dropdown menu lists various date formats, with 'Medium Date' currently selected.

Field Name	Data Type
INV_NUMBER	Number
CUST_CODE	Short Text
INV_DATE	Date/Time

To see the content of any of the tables, just double-click the table name in the Navigation Pane. For example, Figure M.25 shows the content of the INVOICE table in Datasheet View.

FIGURE M.25 THE INVOICE TABLE CONTENTS

The screenshot shows the Microsoft Access 'INVOICE' table in Datasheet View. The table has three columns: INV\_NUMBER, CUST\_CODE, and INV\_DATE. The data shows 10 rows of invoices, each with a unique ID, customer code, and a specific date.

INV_NUMBER	CUST_CODE	INV_DATE
1001	10014	16-Mar-18
1002	10011	16-Mar-18
1003	10012	16-Mar-18
1004	10011	17-Mar-18
1005	10018	17-Mar-18
1006	10014	17-Mar-18
1007	10015	17-Mar-18
1008	10011	17-Mar-18

While you have the INVOICE table open, you can change the spacing between the attribute columns. When you put the cursor on a boundary between the attribute columns, note

that the cursor changes to a two-sided arrow as shown in Figure M.26. You can drag the boundary to increase or decrease the column display width, or you can double-click any column boundary to let MS Access “Auto Fit” the column width. (Change the column display width to whatever size is required to show the column header or values.)

FIGURE M.26 CHANGING THE COLUMN SPACING

INV_NUMBE	CUST_CO	INV_DATE
1001	10014	16-Mar-18
1002	10011	16-Mar-18
1003	10012	16-Mar-18
1004	10011	17-Mar-18
1005	10018	17-Mar-18
1006	10014	17-Mar-18
1007	10015	17-Mar-18
1008	10011	17-Mar-18

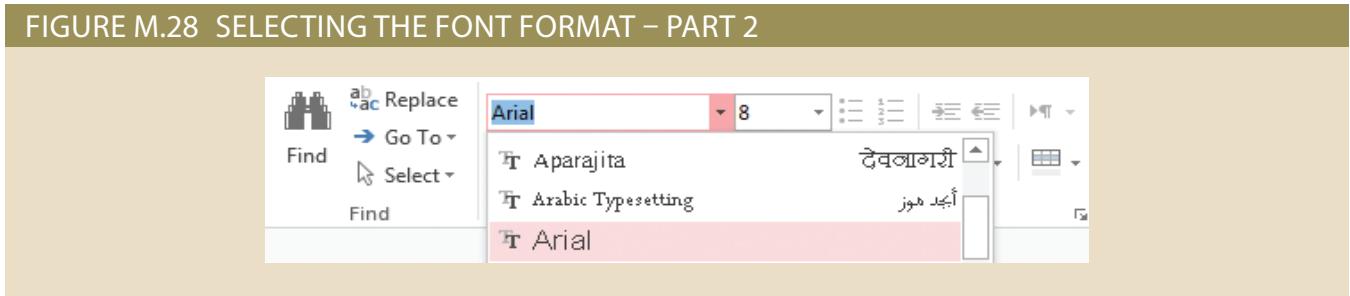
You can also change the presentation font by selecting the **Font** option in the Text Formatting group on the **HOME** tab, as shown in Figure M.27.

FIGURE M.27 SELECTING THE FONT FORMAT



Go ahead and experiment with different formatting options. A good rule to follow is this: If you don’t know what something is or what it does, just move your mouse over the option and a small help text (called a ScreenTip) will appear explaining the option. Note the selection of the font, style, and size in Figure M.28.

FIGURE M.28 SELECTING THE FONT FORMAT – PART 2



Now go ahead and review the table structures of the remaining tables. Use the database tables shown in Figure M.29 as your guide. Remember, you imported most of the tables from the **Ch07\_SaleCo** database, so the table values match those in that database **at this point**.

FIGURE M.29 THE SALECO DATABASE TABLES

The figure shows four tables from the SaleCo database displayed simultaneously in overlapping windows:

- CUSTOMER** (Top Left):
 

CUST_CODE	CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE	CUST_BALANCE
10010	Ramas	Alfred	A	615	844-2573	\$0.00
10011	Dunne	Leona	K	713	894-1238	\$0.00
10012	Smith	Kathy	W	615	894-2285	\$162.52
10013	Olowksi	Paul	F	615	894-2180	\$536.75
10014	Orlando	Myron		615	222-1672	\$0.00
10015	O'Brien	Amy	B	713	442-3381	\$0.00
10016	Brown	James	G	615	297-1228	\$221.19
10017	Williams	George		615	290-2556	\$768.93
10018	Farris	Anne	G	713	382-7185	\$216.55
10019	Smith	Olette	K	615	297-3809	\$0.00
- INVOICE** (Top Right):
 

INV_NUMBER	CUST_CODE	INV_DATE
1001 10014		18-Mar-18
1002 10011		18-Mar-18
1003 10012		18-Mar-18
1004 10011		17-Mar-18
1005 10018		17-Mar-18
1006 10014		17-Mar-18
1007 10015		17-Mar-18
1008 10011		17-Mar-18
- PRODUCT** (Bottom Left):
 

PROD_CODE	PROD_DESCRIP	PROD_INDATE	PROD_QOH	PROD_MIN	PROD_PRICE	PROD_DISCOUNT	VEND_CODE
111QER01	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	\$109.99	0.00	25595
13-Q2P2	7.25-in. pvr. saw blade	13-Dec-11	32	15	\$14.99	0.05	21344
14-Q1L3	9.00-in. pvr. saw blade	13-Nov-11	18	12	\$17.49	0.00	21344
1546-Q02	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	\$39.95	0.00	23119
1546-Q01	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	\$43.99	0.00	23119
1004	1 54778-2T	3	4.99				
1004	2 23109-HB	2	9.95				
1005	1 PVC230RT	12	5.97				
1006	1 SM-18277	3	6.99				
1006	2 23210-TY	1	109.92				
1006	3 23109-HB	1	9.95				
1006	4 89-WRF-Q	1	256.99				
1007	1 13-Q2P2	2	14.99				
1007	2 54778-2T	1	4.99				
1008	1 PVC230RT	5	5.87				
1008	2 WR3/T3	3	119.95				
1008	3 23109-HB	1	9.95				
- VENDOR** (Bottom Right):
 

VEND_CODE	VEND_NAME	VEND_CONTACT	VEND_AREACODE	VEND_PHONE	VEND_STATE	VEND_ORDER
21225 Bryson, Inc.	Smithson	615	223-3234	TN	Y	
21226 SuperLoc, Inc.	Flushing	904	216-8995	FL	N	
21231 D&E Supply	Singh	615	228-3245	TN	Y	
21344 Gomez Bros.	Ortega	615	889-2546	KY	N	
22567 Dome Supply	Smith	901	678-1419	GA	N	
23119 Randets Ltd.	Anderson	901	678-3998	GA	Y	
24004 Brackman Bros.	Browning	615	228-1410	TN	N	
24288 ORDVA, Inc.	Hakford	615	896-1234	TN	Y	
25443 B&K, Inc.	Smith	904	227-0093	FL	N	
25501 Damal Supplies	Smythe	615	890-3529	TN	N	
25595 Rubicon Systems	Orton	904	456-0092	FL	Y	

## Note

All of the previous figures have been shown in Tabbed Document view. Figure M.29 is shown in Overlapping Window view. To change to Overlapping Window view, use the Access Options dialog box as follows:

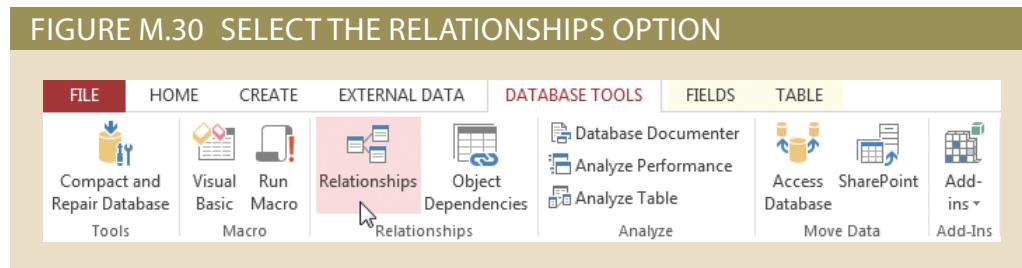
### FILE/Options / Current Database / Document Window Options / Overlapping Windows

You will be prompted with a message saying "You must close and reopen the current database for the specified option to take effect." Close the SaleCo Database and reopen it. Follow this same procedure to change back to Tabbed Documents.

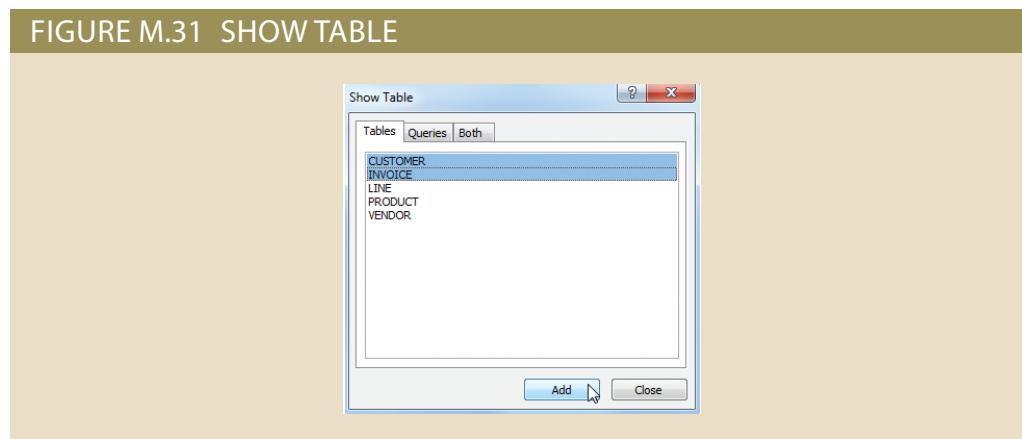
## M-4 Setting the Relationships between Tables

Thus far, you have learned to create, populate, and import tables. Actually, if you were to create a real-world relational database, you would first create the table structures and then create the relationships among them. The reason for doing so is simple—you want to make sure that referential integrity is enforced. (If you don't quite remember what referential integrity is and why it is important, review the text's Chapter 3, The Relational Database Model, Section 3-2, Keys.)

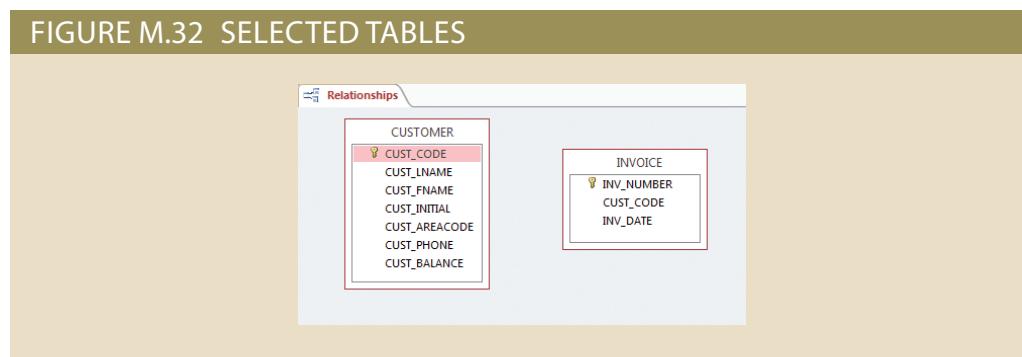
To create a relationship, start by selecting the **Relationships** button on the **DATABASE TOOLS** tab, as shown in Figure M.30.



After you click the Relationships button shown in Figure M.30, the **Show Table** dialog box will appear as seen in Figure M.31.

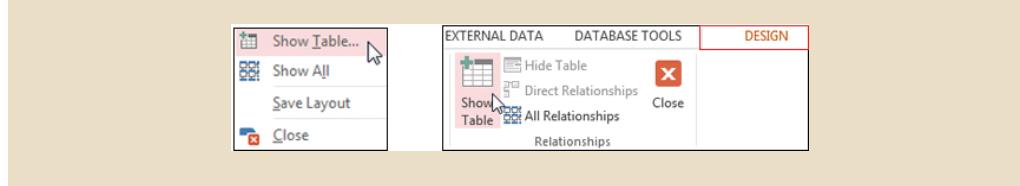


Now select the two tables as shown in Figure M.31 and then click the **Add** button. (You can also double-click each table to move it from the **Show Table** dialog box to the Relationships screen.) Click the **Close** button. This action will generate the screen shown in Figure M.32.



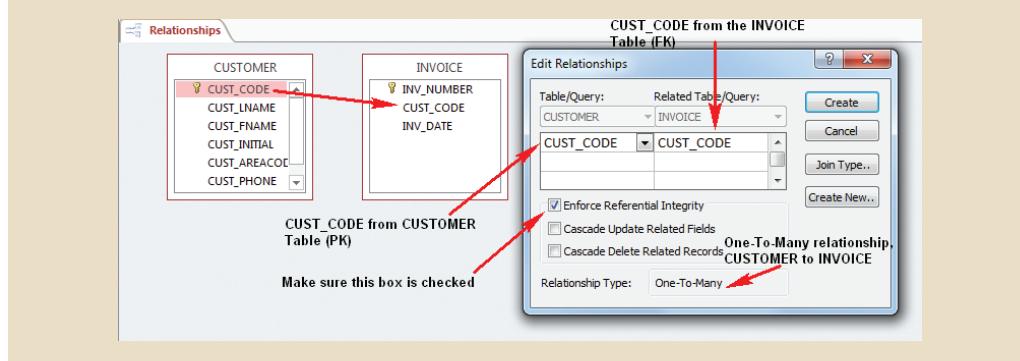
To add more tables, right-click anywhere on the blank relationships screen, as shown in Figure M.33, to show the context menu; then click **Show Table** or click the **Show Table** button located on the **DESIGN** tab. Both options will take you back to the screen shown in Figure M.31.

FIGURE M.33 ADD MORE TABLES



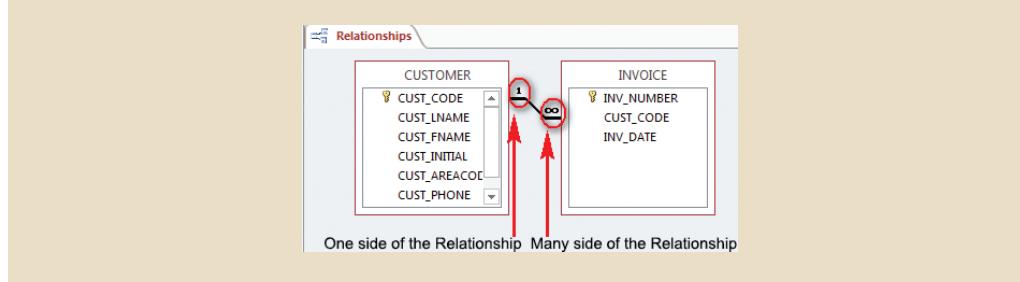
To create the relationship between the CUSTOMER and INVOICE tables, select **CUST\_CODE** in the CUSTOMER table and drag and drop it on CUST\_CODE in the INVOICE table. This action will produce the **Edit Relationships** dialog box you see in Figure M.34.

FIGURE M.34 CREATE RELATIONSHIP



Next, click to check the **Enforce Referential Integrity** checkbox option you see in Figure M.34's **Edit Relationships** dialog box and then click the **Create** button to create the relationship. This action will produce the results shown in Figure M.35. Note that the “1” side of the relationship is marked by the boldfaced 1, while the many (M) side of the relationship is marked by the infinity symbol  $\infty$ .

FIGURE M.35 THE RELATIONSHIP BETWEEN CUSTOMER AND INVOICE



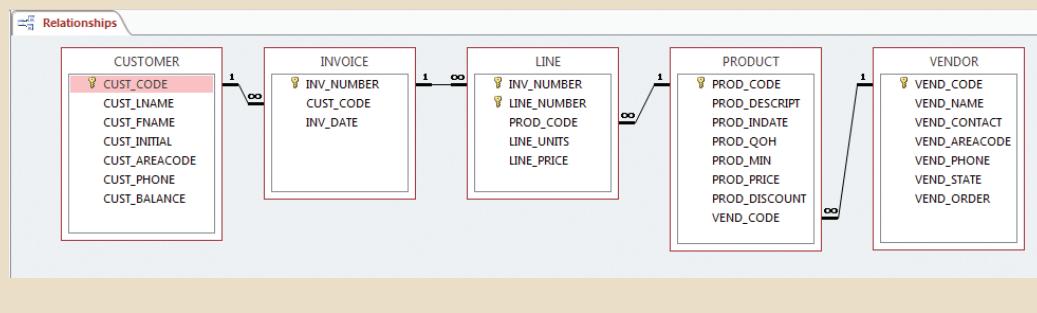


## Note

Always drag and drop from the 1 side to the M side in a one-to-many (1:M) relationship. If you are creating a relationship between tables in a 1:1 relationship, drag from the parent entity to the dependent entity.

Now go ahead and create the relationships between all the tables. When you are done, your results will resemble Figure M.36. (We have dragged and sized the tables to enhance the presentation.)

**FIGURE M.36 THE RELATIONSHIPS FOR ALL THE SALECO DATABASE TABLES**



After creating the relationships, click the **Close** button on the Ribbon and click **Yes** to save the changes.

## M-5 Repairing and Compacting the Database

A word of caution is in order. When you do a lot of work on your database, it tends to grow rapidly and fills up with all sorts of temporary objects or objects that may no longer exist. Sooner or later, this situation will degrade the performance of your database. Therefore, it is a good idea to clean up the debris frequently.

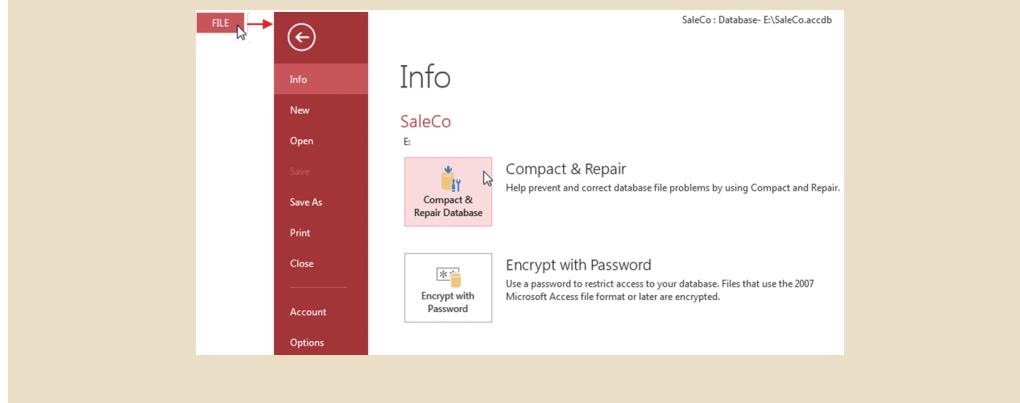
In the following sequence, note that the SaleCo database size is 704 KB at this point (see Figure M.37). Depending on the number of changes you made, your database size is likely to differ from the 704 KB you see here.

**FIGURE M.37 THE SALECO DATABASE SIZE BEFORE THE CLEANUP**

SaleCo.accdb 704 KB

The cleanup is accomplished by selecting the sequence shown in Figure M.38 – **FILE / Info / Compact & Repair Database**.

**FIGURE M.38 COMPACT & REPAIR THE DATABASE**



When the procedure is completed, look how much smaller the SaleCo database has become. Figure M.39 shows that the SaleCo database now uses only 504 KB.

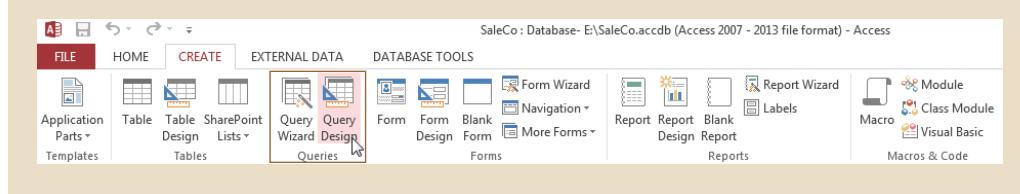
**FIGURE M.39 THE SALECO DATABASE SIZE AFTER THE CLEANUP**



## M-6 Queries

Queries are used to extract data from the database and to help transform data into information. To create a query you will use the query design feature of MS Access. To open this feature, click the **CREATE** tab on the Ribbon and then click **Query Design** as shown in Figure M.40.

**FIGURE M.40 CREATING THE FIRST QUERY**



The **Show Table** dialog box appears as shown in Figure M.41. (Many of the wizards are useful, but you are better served by using the **Design View** option if you want to learn how to create and manipulate queries. All the queries in this tutorial will be created with the help of the MS Access Graphical User Interface, or GUI.)

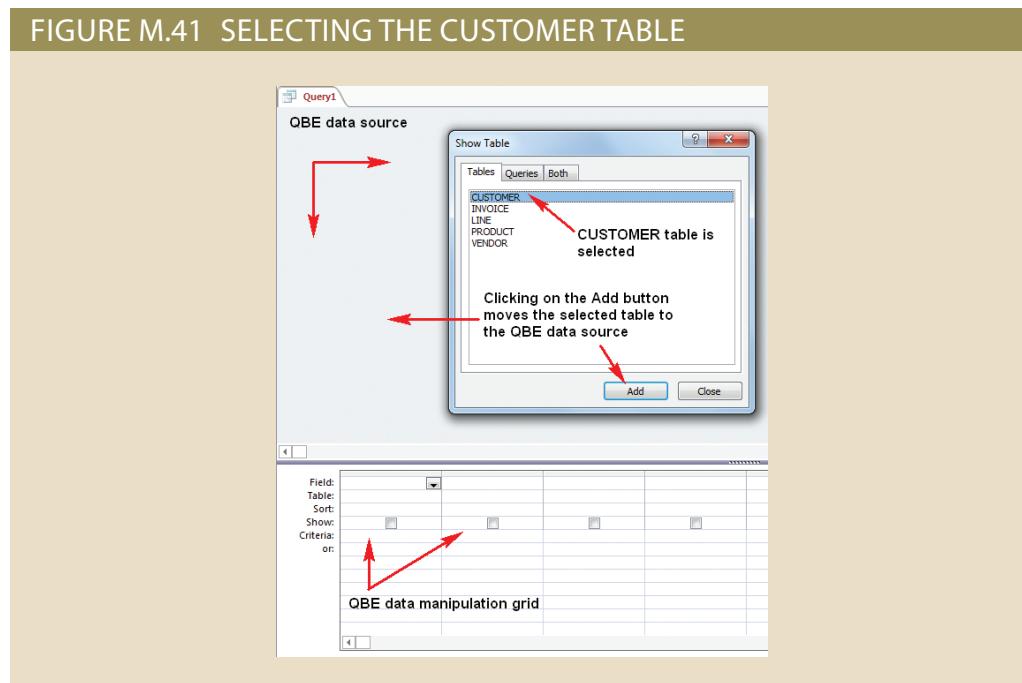
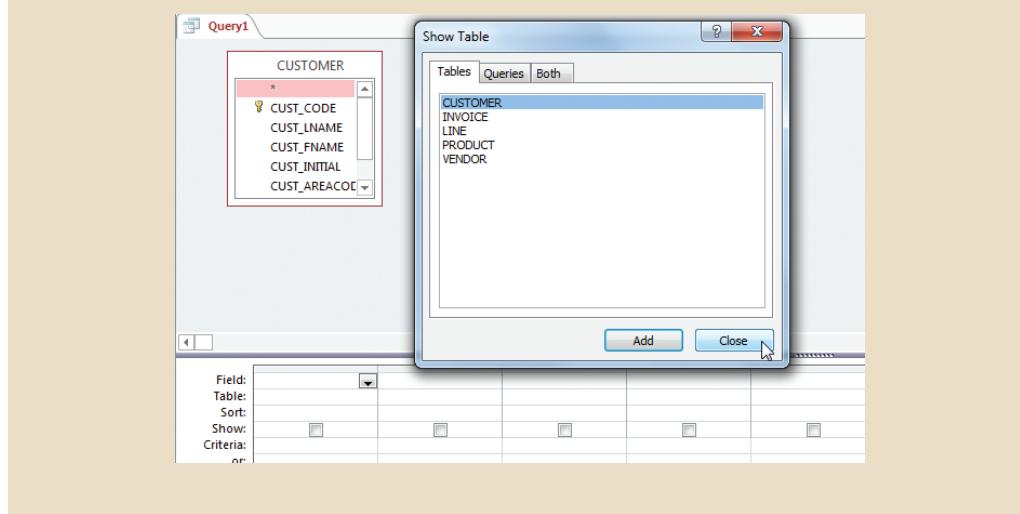


Figure M.41 shows three QBE (query by example) window components.

- The top (blank) portion of the window is the QBE window's ***data source*** display, which may be a one or more tables or queries.
- The bottom (grid) portion shows the available options in the QBE ***grid***. The grid represents the ***data manipulation*** portion of the QBE window. As its name suggest, the **Field:** option lets you place a table's—or query's—field on the output of this query. The **Table:** shows the field's origin. The **Sort:** option lets you sort selected field values in either ascending or descending order. The **Show:** option lets you select whether or not to display a selected field on the query output by checking or unchecking the box. The **Criteria:** option lets you add a condition to the query, so only the rows that meet the criteria are shown in the query output. And the **or:** option lets you add more conditions to the **Criteria:** constraint selection. (For example, you might place a restriction on the query output for a CUSTOMER table to show only customers named “Smith” **or** to also display customers whose current balance is over \$200.)

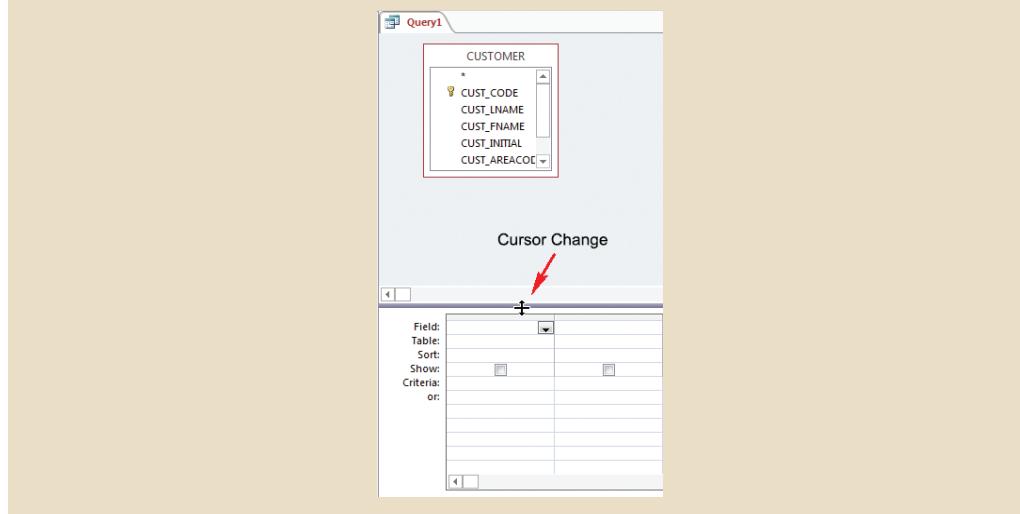
To place the CUSTOMER table on the data source portion of the screen, either select **CUSTOMER** as shown in Figure M.42 and then click the **Add** button or double-click the CUSTOMER table. When you have placed all the tables you need on the top (data source) portion of the screen, close the **Show Table** dialog box by clicking its **Close** button. (See Figure M.42.) In this example, only the CUSTOMER table has been selected as the data source.

FIGURE M.42 CLOSING THE SHOW TABLE WINDOW



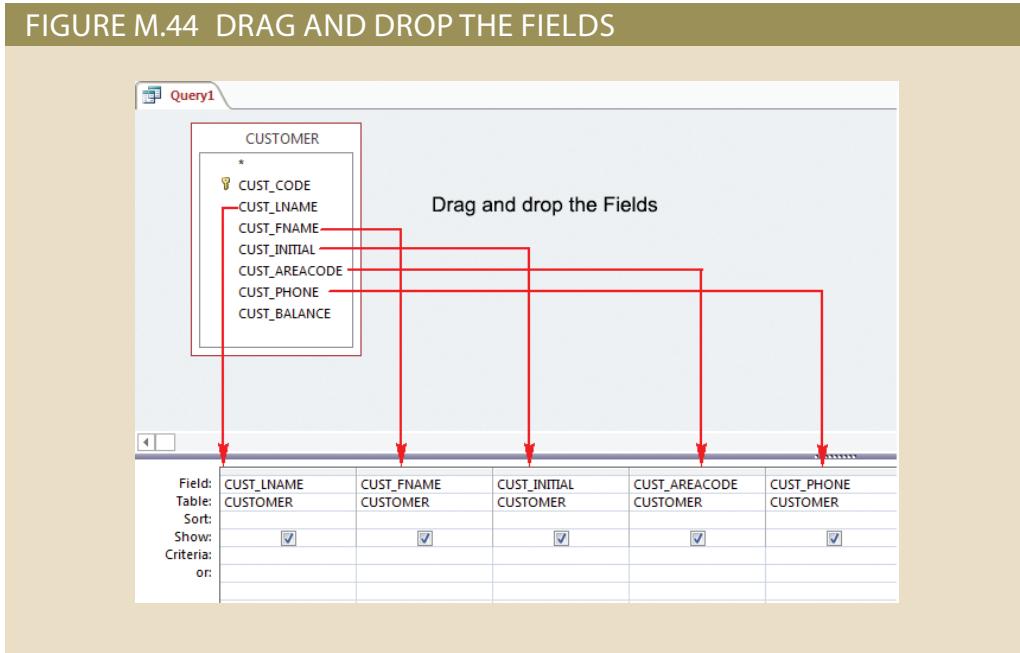
After you have selected the data source—the CUSTOMER table—and placed it on the data source portion of the QBE screen, size the QBE window's components by dragging their limits. Figure M.43 shows that the cursor changes when you place it on the boundary between the data source component that now contains the CUSTOMER table and the data manipulation grid component. When the cursor changes its shape to the two-sided arrow seen here, you can drag the boundary up or down.

FIGURE M.43 SIZE THE TABLE WINDOW



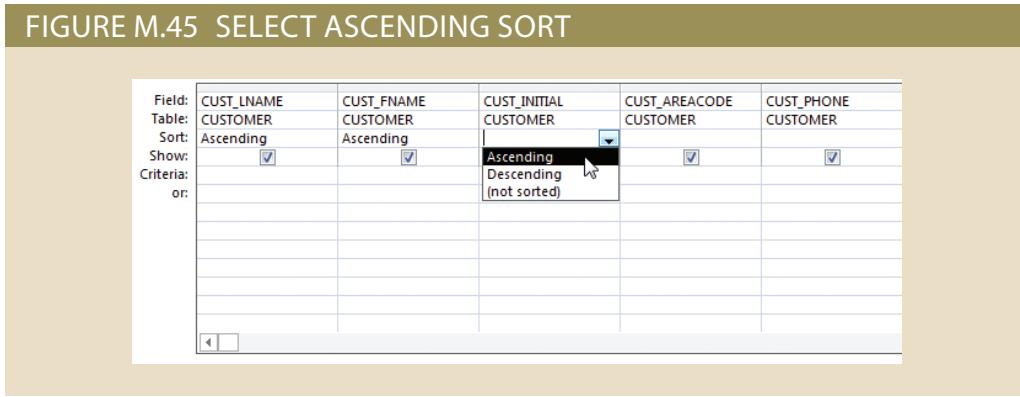
After enlarging the data component portion of the QBE window, drag the CUSTOMER table box lower corner to show all the fields. Then select the CUSTOMER table's fields and drag and drop them in the **Field:** spaces as shown in Figure M.44. You can drag and drop the individual fields from the CUSTOMER table to the field columns in the grid section or you can select multiple fields by CTRL-clicking them and dropping them on a single field space. (Access will automatically space them across the field columns.) Note that the field origin is automatically displayed in the Table: portion of the grid.

FIGURE M.44 DRAG AND DROP THE FIELDS



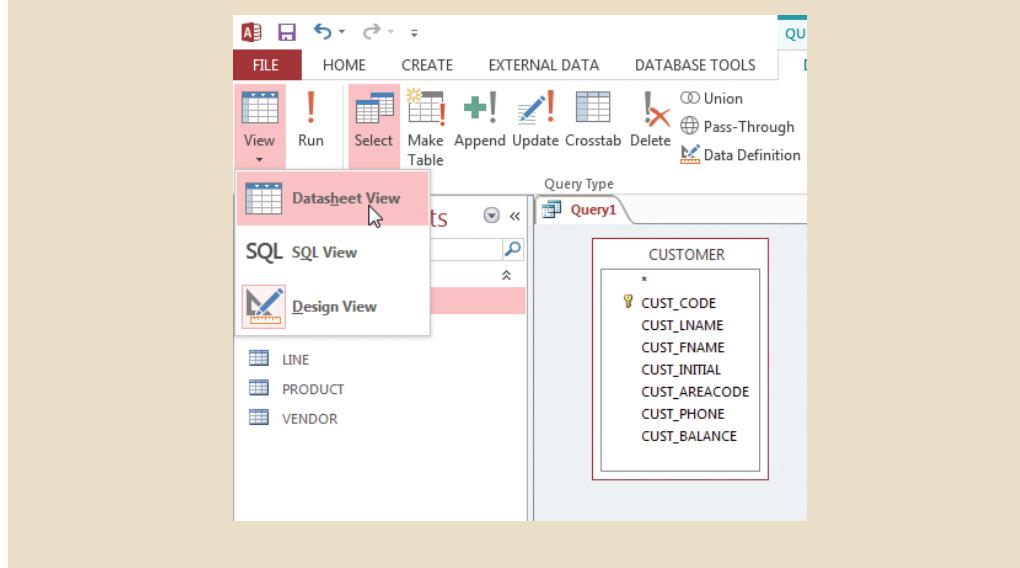
After you have selected the fields you want to use in the query, you can choose whether or not to sort the field values by clicking on the grid's **Sort:** option. Clicking on this option yields a drop-down list of sort options you see in Figure M.45. Click the option you want to use. Set the CUST\_LNAME, CUST\_FNAME, and CUST\_INITIAL fields to sort the fields in ascending order from left to right, as shown in Figure M.45. If you do not set a sort order, MS Access will display the records in random order.

FIGURE M.45 SELECT ASCENDING SORT



If you want to see the effect of the query actions you have taken thus far, change the **View** from the current Design View to **Datasheet View**, as shown in Figure M.46. Or you could click the **Run** button on the **DESIGN** tab on the Ribbon.

FIGURE M.46 SELECT DATASHEET VIEW

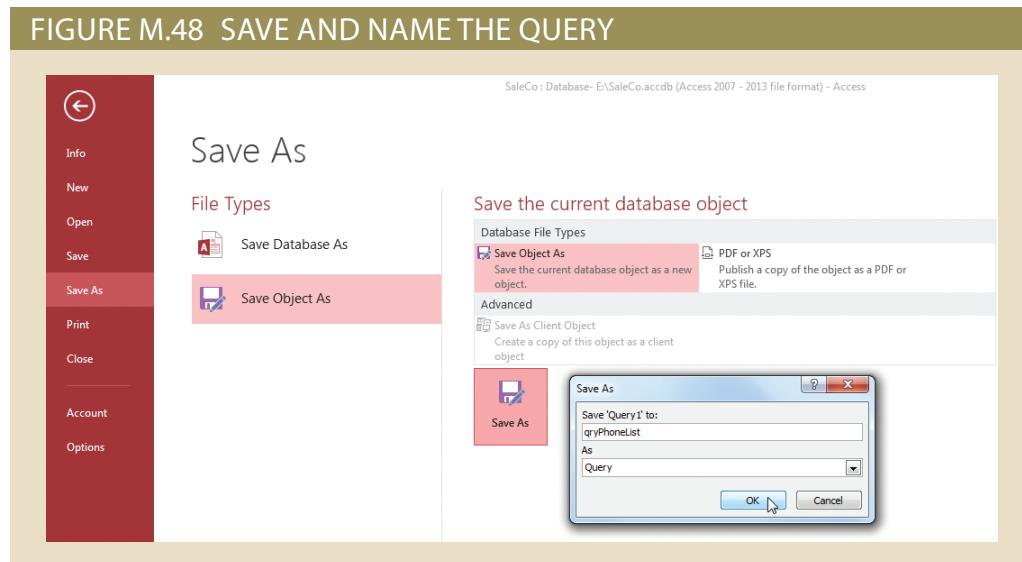


The query output is shown in **Datasheet View** as illustrated in Figure M.47.

FIGURE M.47 DATASHEET VIEW

CUST_LNAME	CUST_FNAME	CUST_INITIAL	CUST_AREACODE	CUST_PHONE
Brown	James	G	615	297-1228
Dunne	Leona	K	713	894-1238
Farriss	Anne	G	713	382-7185
O'Brien	Amy	B	713	442-3381
Ołowski	Paul	F	615	894-2180
Orlando	Myron		615	222-1672
Ramas	Alfred	A	615	844-2573
Smith	Kathy	W	615	894-2285
Smith	Olette	K	615	297-3809
Williams	George		615	290-2556

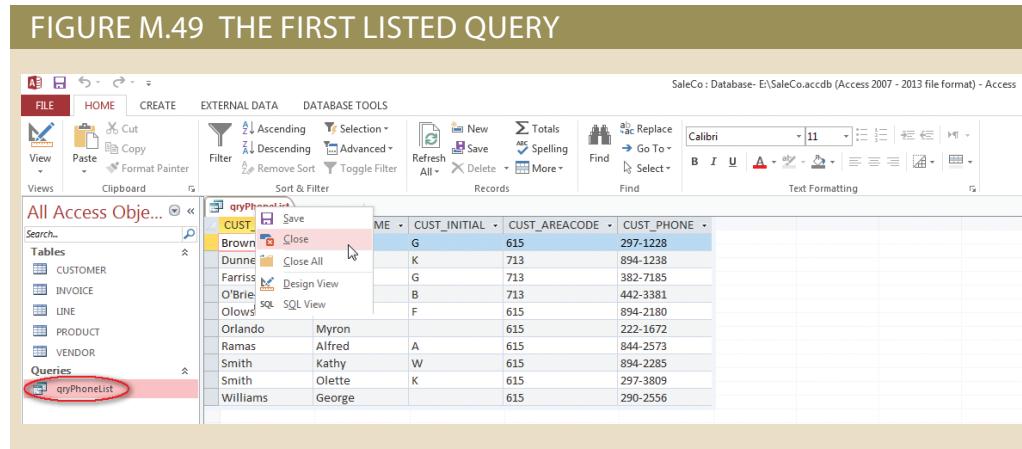
Before you do any additional work on the query, save it. Figure M.48 shows the selection of the **FILE / Save As / Save Object As / Save As** option.



The selection of the **FILE / Save As / Save Object As / Save As** option will produce the **Save As** dialog box you see in Figure M.48. The dialog box shows a default query name, in this case, **Query1**.

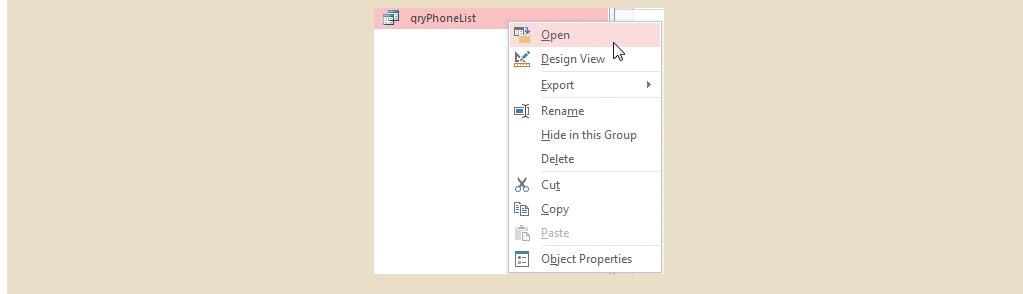
Always use a query name that is self-documenting. In this example, the query will be used as the basis for a phone list. Therefore, **PhoneList** is an appropriate name. However, you want to also show that this object is a query, so use the prefix **qry** to indicate that fact. Therefore, the appropriate name will be **qryPhoneList**. And you see that query name used in Figure M.48. (You will discover that this self-documenting naming convention is very desirable, because it enables you to easily keep track of multiple components in an application set. For example, you will learn how to create forms in Section M-7, and if you see two objects, **frmPhoneList** and **qryPhoneList**, you will know which object you are looking at, and it will be easy to see that the data source for the form is the query that has the same name.)

After you type the query name as shown in Figure M.48, click **OK** to save the query. Note that the saved query now shows up under the **Queries** header in the Navigation Pane as shown in Figure M.49. Right-click the **qryPhoneList** query tab and click **Close** as shown in Figure M.49.



Select the query **Open** option shown in Figure M.50. (You can also open the query by double-clicking the query name.)

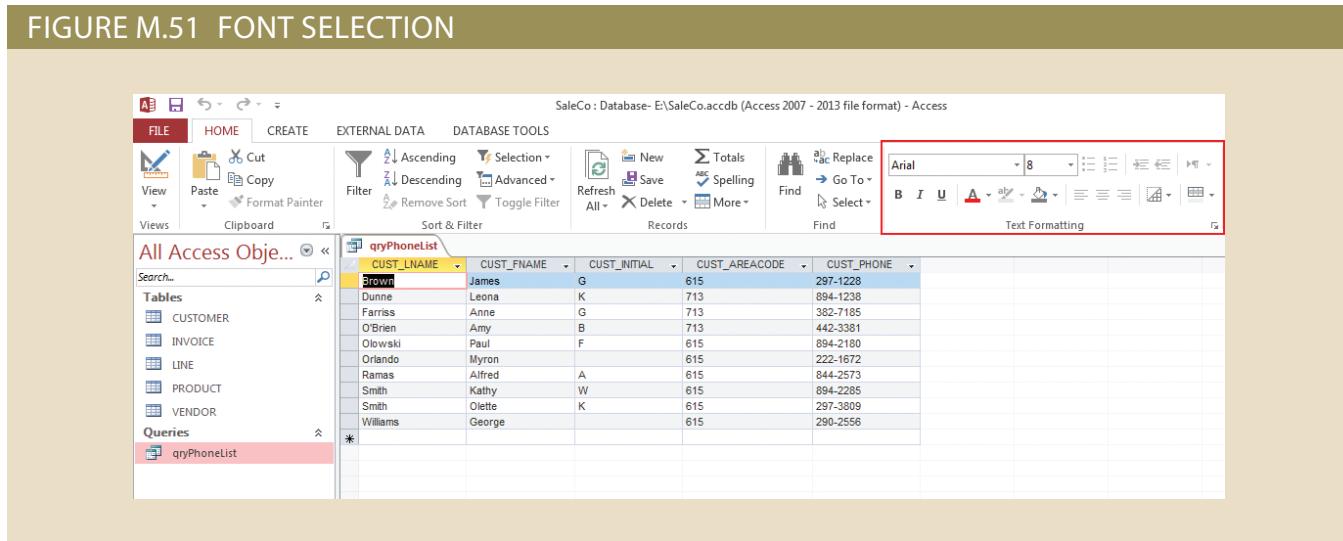
FIGURE M.50 OPEN THE QUERY



## M-6a Editing the Query Output

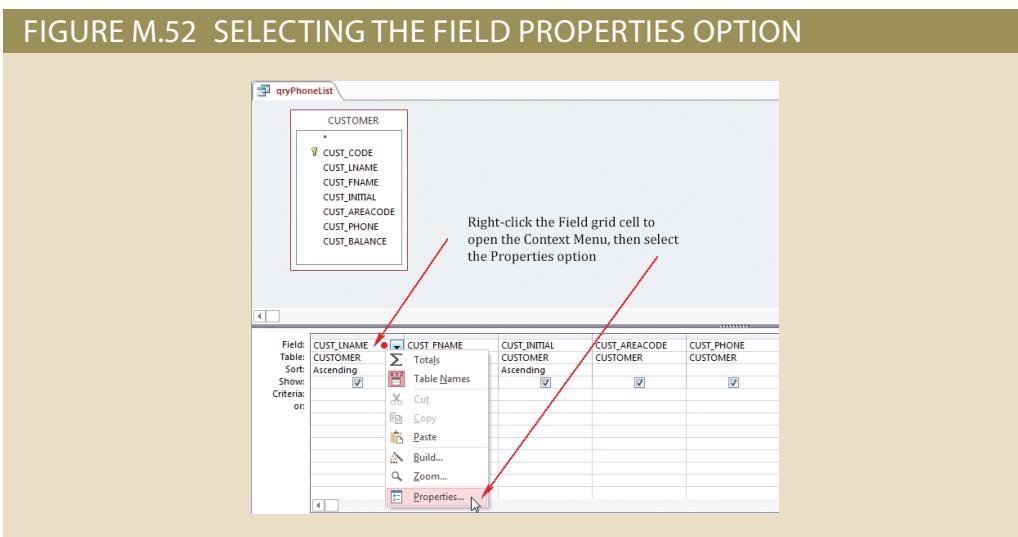
If you are not satisfied with the default font selection, font size, color, etc. when you open the query in its Datasheet View, you can easily modify those. Just click the **HOME** tab on the Ribbon to view the **Text Formatting** group shown in Figure M.51, where you can select the font, size, and style. Figure M.51 shows the selection of an Arial font, size of 8, and no style (bold, italics or underline).

FIGURE M.51 FONT SELECTION



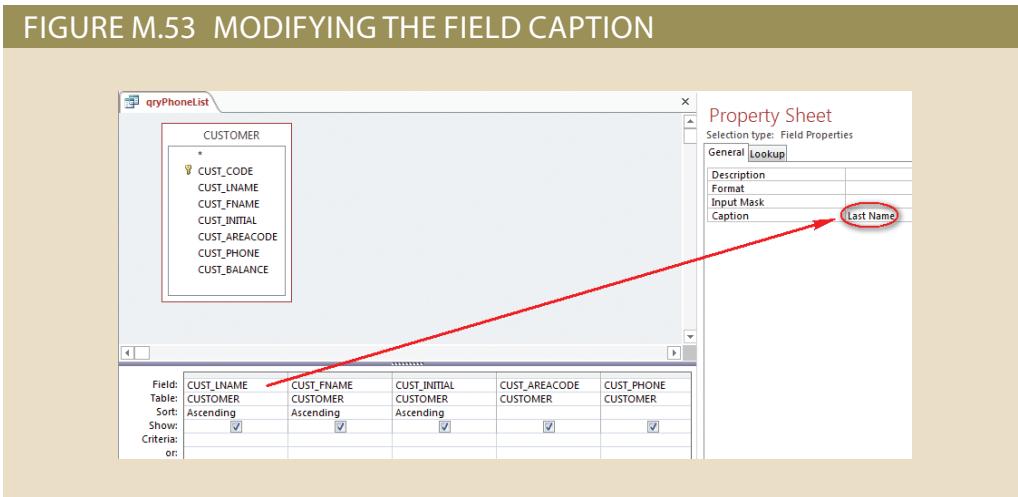
Because the default query output uses the field headers—such as **CUST\_LNAME**—used by the query data source table, you might want to make the presentation more “finished” looking by changing the query field headers. You can get that job done while you are in the query design mode. Note the procedure illustrated in Figure M.52. That is, right-click a selected field name grid cell to open the context menu, and then select **Properties** to generate the Property Sheet window you see in Figure M.53.

**FIGURE M.52 SELECTING THE FIELD PROPERTIES OPTION**



You can change the field header by selecting the **Caption** option in the Property Sheet window you see in Figure M.53. Then type in the field header you want to use. In this case, the field header will be **Last Name**. Changing the query header does *not* affect the attribute name (CUST\_LNAME) in the query data source—the CUSTOMER table.

**FIGURE M.53 MODIFYING THE FIELD CAPTION**



Repeat the editing for the remaining query fields and then open the query in its **Datasheet View** to see the editing results shown in Figure M.54.



### Note

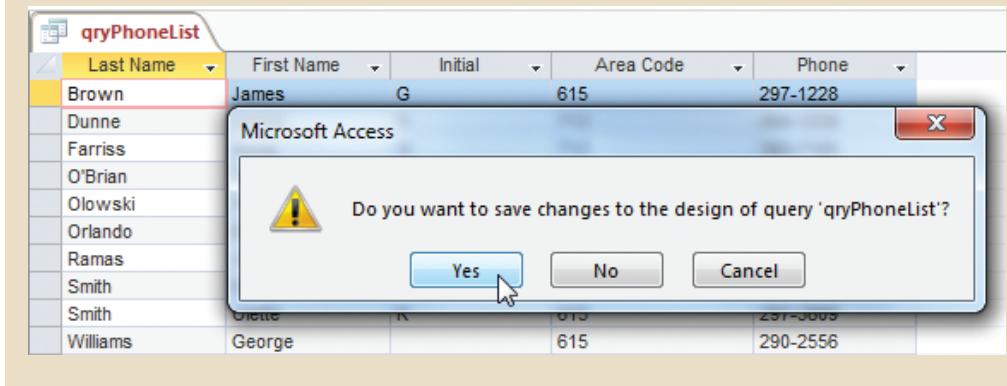
MS Access will automatically use the table field's captions, if they are defined. When defining a table, one of the field properties is the **Caption** property (see Figure M.9, under the General tab in the bottom section of the table definition window.) If you define captions for the table fields, those captions will be automatically be used for all queries, forms, and reports you create afterward.

FIGURE M.54 THE MODIFIED QUERY FIELD HEADERS

Last Name	First Name	Initial	Area Code	Phone
Brown	James	G	615	297-1228
Dunne	Leona	K	713	894-1238
Farriss	Anne	G	713	382-7185
O'Brien	Amy	B	713	442-3381
Ołowski	Paul	F	615	894-2180
Orlando	Myron		615	222-1672
Ramas	Alfred	A	615	844-2573
Smith	Kathy	W	615	894-2285
Smith	Olette	K	615	297-3809
Williams	George		615	290-2556

If you are satisfied with the results, remember to save the query again. (If you forget to save, Access will remind you as shown in Figure M.55.)

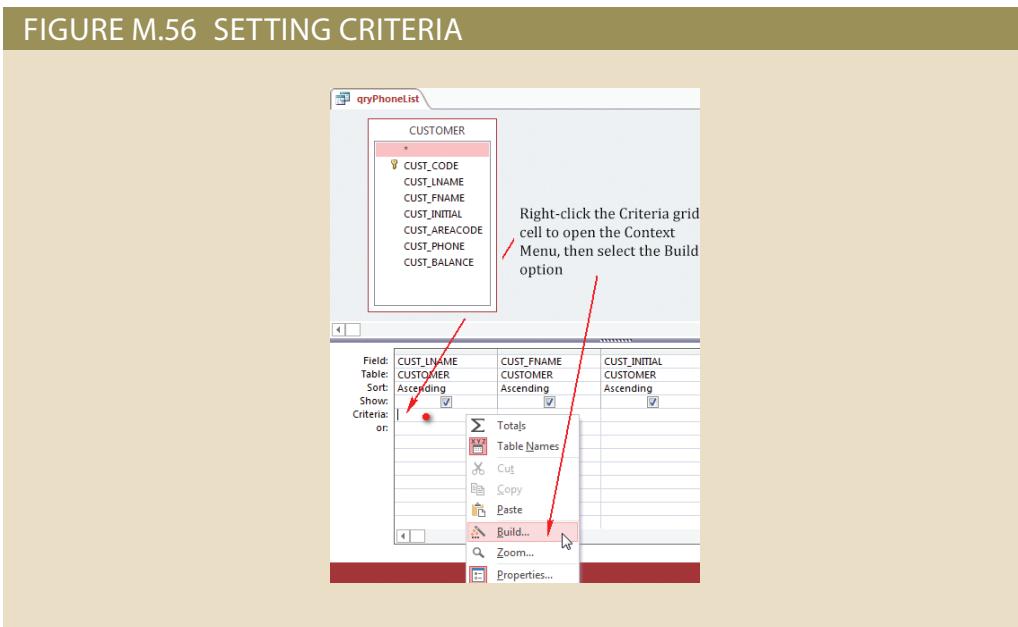
FIGURE M.55 REMINDER TO SAVE



## M-6b Parameter Queries

You can easily create a query in which you specify the **criteria** governing the query output. Note the procedure summarized in Figure M.56. In the following example, the objective will be to limit the phone list output to a specified customer's last name. (A query whose output is limited through specified criteria that restrict output for one or more parameters is also known as a **parameter query**.)

FIGURE M.56 SETTING CRITERIA



If you want to limit the query phone list output to customers whose last name is “Smith,” you can simply type “Smith” in the CUST\_LNAME criteria grid cell (marked by the red dot in Figure M.56). Unfortunately, that procedure means that the query must be changed each time a different last name limitation is required. You will have a much more flexible query if you let the end user specify the last name restriction through a dialog box. Such a dialog box is created automatically if you type

**Like “\*” & [Enter customer last name] & “\*”**

in the CUST\_LNAME criteria grid cell. (Review the LIKE operator and wildcard characters such as “\*” in Chapter 7, Introduction to Structured Query Language (SQL), Section 7-4d.)

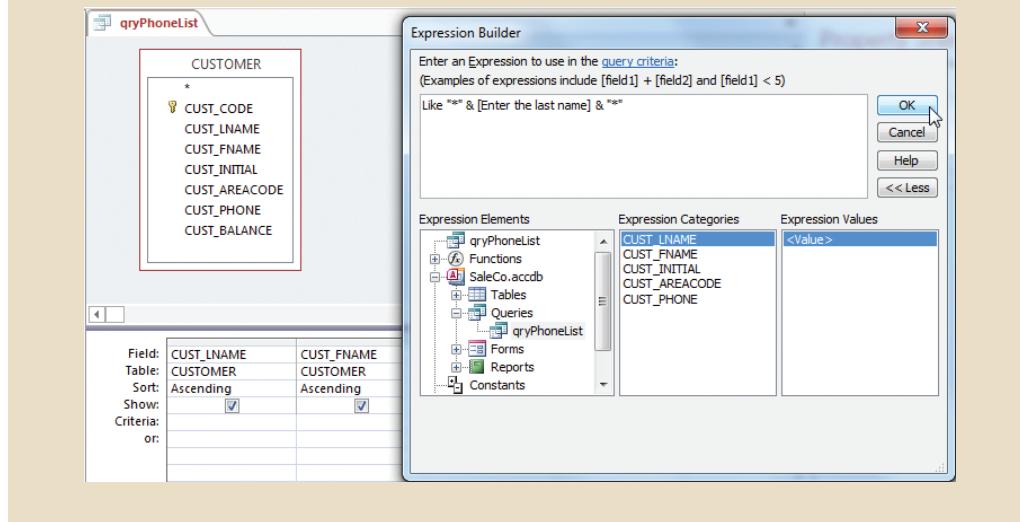


### Note

Although you can type the simple criteria restriction directly into the grid, you should become used to the **Expression Builder**. This tool is especially useful when you later try to enter more complex criteria or even simple criteria with multiple components. You may have a difficult time typing the sometimes long character strings without making errors; it will be a lot easier to select items from a list than to do the typing. Therefore, we will use the Expression Builder in most examples.

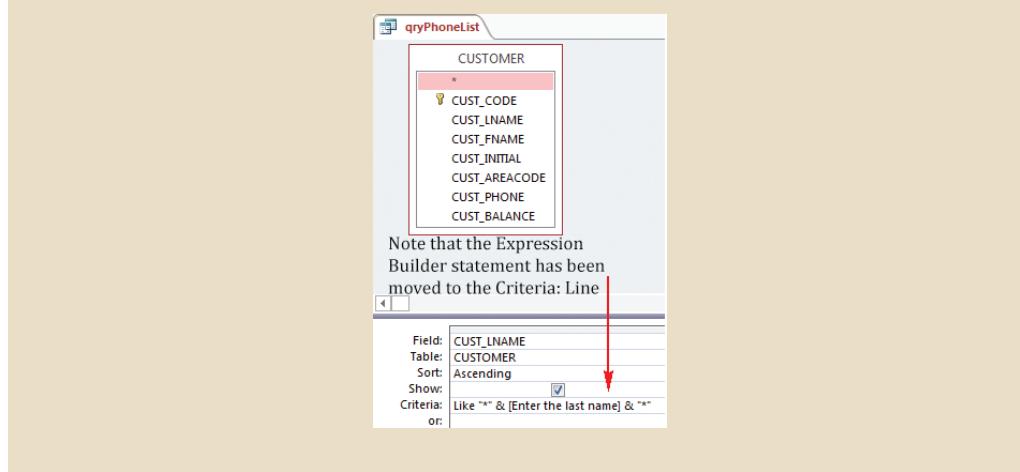
If you click the **Build...** option shown in Figure M.56, you will see the **Expression Builder** window. You can then type the entire expression as shown in Figure M.57.

FIGURE M.57 THE EXPRESSION BUILDER



When you have completed the expression shown in Figure M.57, click **OK** to close the Expression Builder and to transfer the expression to the QBE grid as shown in Figure M.58. (If you want to see the entire expression in the grid space, drag the grid column limit to widen it, as was done in Figure M.58.)

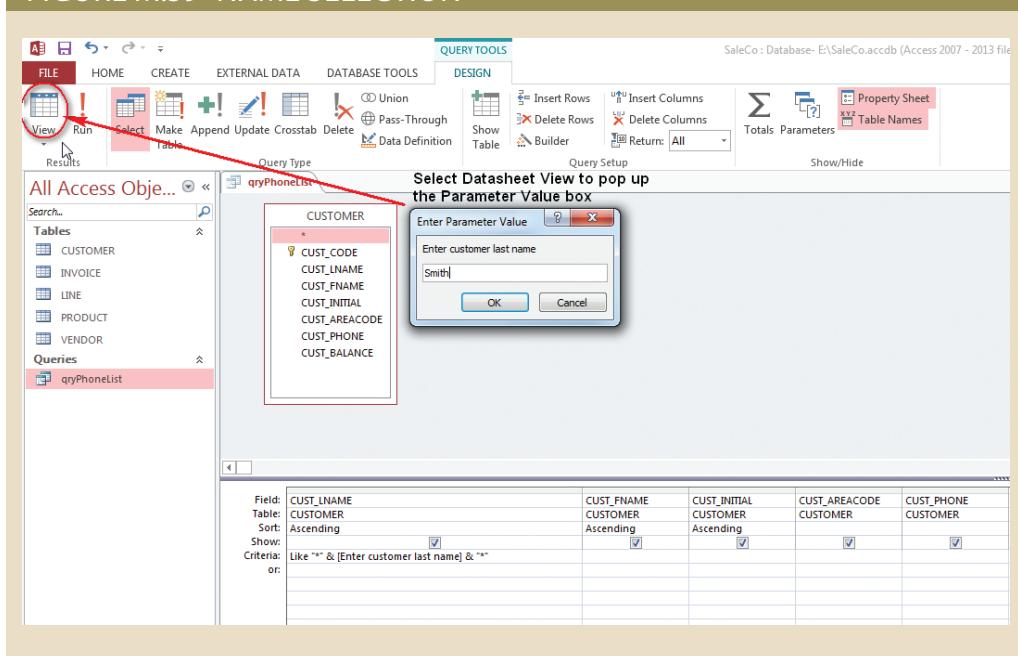
FIGURE M.58 THE COMPLETED CRITERIA LINE



Next, open the query in its **Datasheet View**. The expression you see in Figure M.58 will trigger the input request you see in Figure M.59. Type the last name **Smith** to generate the output shown in Figure M.60. Incidentally, the parameter search is not case-sensitive. Therefore, it does not matter whether you type **SMITH**, **smith**, **Smith**, or any other combination of lower- and uppercase letters. Also, keep in mind that the use of the **\*** wildcard character in combination with **Like** will yield the results shown in Table M.3:

**TABLE M.3****INPUT CRITERIA WITH “LIKE” AND “\*”**

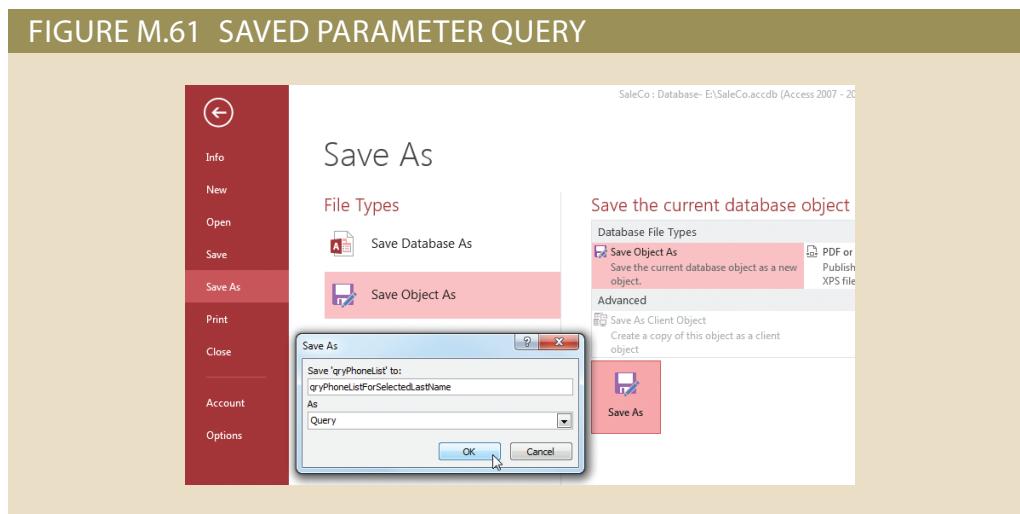
INPUT	OUTPUT
None. (You just tap the <b>Enter</b> key, instead of typing a character and then tapping the <b>Enter</b> key.)	All records.
The letter <b>s</b> .	All records corresponding to a customer whose last name includes the letter <b>s</b> . For example, Ramas <b>s</b> , Williams, Olowski, Farris <b>s</b> , and Smith would all be included.
The letters <b>br</b> .	All records corresponding to a customer whose last name includes the letters <b>br</b> . For example, customer O'Brian and Brown would be included.

**FIGURE M.59 NAME SELECTION****FIGURE M.60 PHONE NUMBERS FOR SELECTED NAME**

Last Name	First Name	Initial	Area Code	Phone
Smith	Kathy	W	615	894-2285
Smith	Olette	K	615	297-3809

Use the **Save As** option to save a new version of this query. Save the **qryPhoneList** query you have just modified as **qryPhoneListForSelectedLastName** as seen in Figure M.61.

FIGURE M.61 SAVED PARAMETER QUERY



You can use the same technique to limit output by any selected criteria for any field. For example, Figure M.62 shows a query that limits its output by VEND\_CODE values that are null using the PRODUCT table. (In short, the output will show all products that do not have a known vendor.) The output is shown in Figure M.63.

FIGURE M.62 PRODUCTS WITHOUT VENDORS

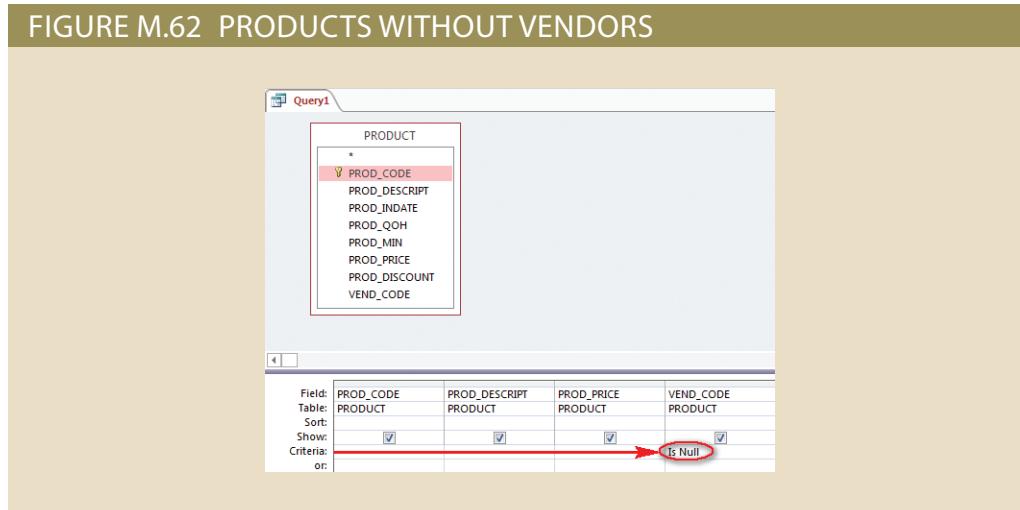


FIGURE M.63 PRODUCTS WITHOUT VENDORS OUTPUT

PROD_CODE	PROD_DESCRPT	PROD_PRICE	VEND_CODE
Table:	Table:	Table:	Table:
23114-AA	Sledge hammer, 12 lb.	\$14.40	
PVC23DRT	PVC pipe, 3.5-in., 8-ft	\$5.87	

Note that the query was saved as **qryProductsWithoutVendors**.

## M-6c Multiple Table Queries

You can include any number of tables or queries in your queries. For example, note that the query in Figure M.64 includes three tables: CUSTOMER, INVOICE and LINE. You do not have to relate the tables, because that was done earlier when you setup the relationships (see Section M-4).

Examine the LINE table and look at INV\_NUMBER 1004. The Customer ordered 3 products for \$4.99 and 2 products for \$9.95. The Invoice Amount before tax is \$34.87. We are going to skip right to adding the tax to the Invoice Amount to achieve the Invoice total for all invoices as seen in Figure M.64a.

- Select the **Totals** button at the top of the screen (shown by the summation symbol). This will add a Total row to each of the fields, and by default they will all be set to **Group By**.
- Right-click the empty field to the right on INV\_DATE and select **Build**. Build the Expression named INV\_TOTAL as seen in Figure M.64. The expression is as follows:  $\text{Sum}(([LINE\_UNITS]*[LINE\_PRICE]) + (([LINE\_UNITS]*[LINE\_PRICE])*0.08))$
- Set the INV\_TOTAL Property Sheet to **Format: Currency**.
- INV\_TOTAL is now a calculated field derived from the price of the product \* the quantity of product which would result in the (Invoice Amount) + (the tax on the Invoice Amount). Note that the default name for the Expression will be Expr1, change this to INV\_TOTAL.
- Make sure in the **Total** row that **Group By** is selected for CUST\_CODE, CUST\_LNAME, CUST\_FNAME, CUST\_INITIAL, INV\_NUMBER and INV\_DATE. INV\_TOTAL should be selected as an **Expression**.

FIGURE M.64 MULTIPLE TABLE QUERY

Save this query as **qryCustomerInvoices** and then open this query in **Datasheet View** to see its output shown in Figure M.64a.

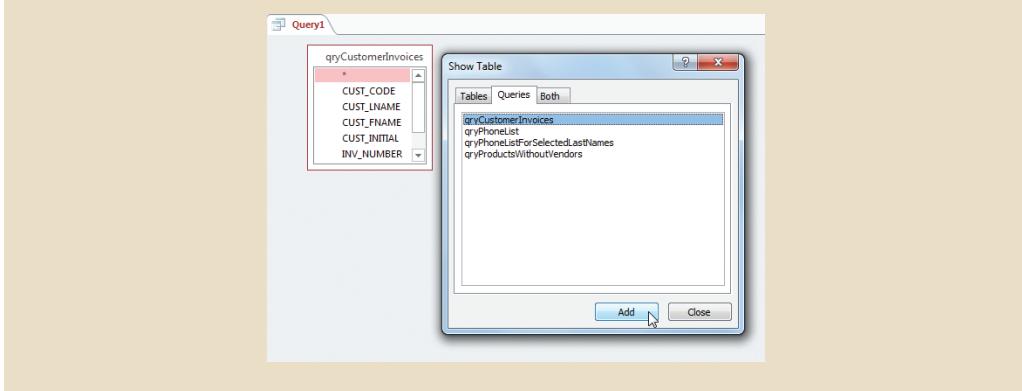
FIGURE M.64A THE QRYCUSTOMERINVOICES OUTPUT

CUST_CODE	CUST_LNAME	CUST_FNAME	CUST_INITIAL	INV_NUMBER	INV_DATE	INV_TOTAL
10011	Dunne	Leona	K	1002	16-Mar-18	\$10.78
10011	Dunne	Leona	K	1004	17-Mar-18	\$37.66
10011	Dunne	Leona	K	1008	17-Mar-18	\$431.08
10012	Smith	Kathy	W	1003	16-Mar-18	\$166.16
10014	Orlando	Myron		1001	16-Mar-18	\$26.94
10014	Orlando	Myron		1006	17-Mar-18	\$429.66
10015	O'Brien	Amy	B	1007	17-Mar-18	\$37.77
10018	Farriss	Anne	G	1005	17-Mar-18	\$76.08

## M-6d Querying a Query

Suppose you want to know the total sales for each of the customers. If you run the **qryCustomerInvoices** query, you will see all the invoices for each of the customers. For example, if you look at Figure M.64a, you see that customer 10011, Leona Dunne, has three invoices for \$10.78, \$37.66, and \$431.08, respectively. The sum of these invoice totals will be \$479.52. How do write a query that will generate the sum of all the invoice totals for each of the customers? The answer turns out to be simple: As you can see in Figure M.65, you can write a query that uses the **qryCustomerInvoices** query output as its data source. (Note that the **Show Table** dialog box shows that the **Queries** tab was selected.)

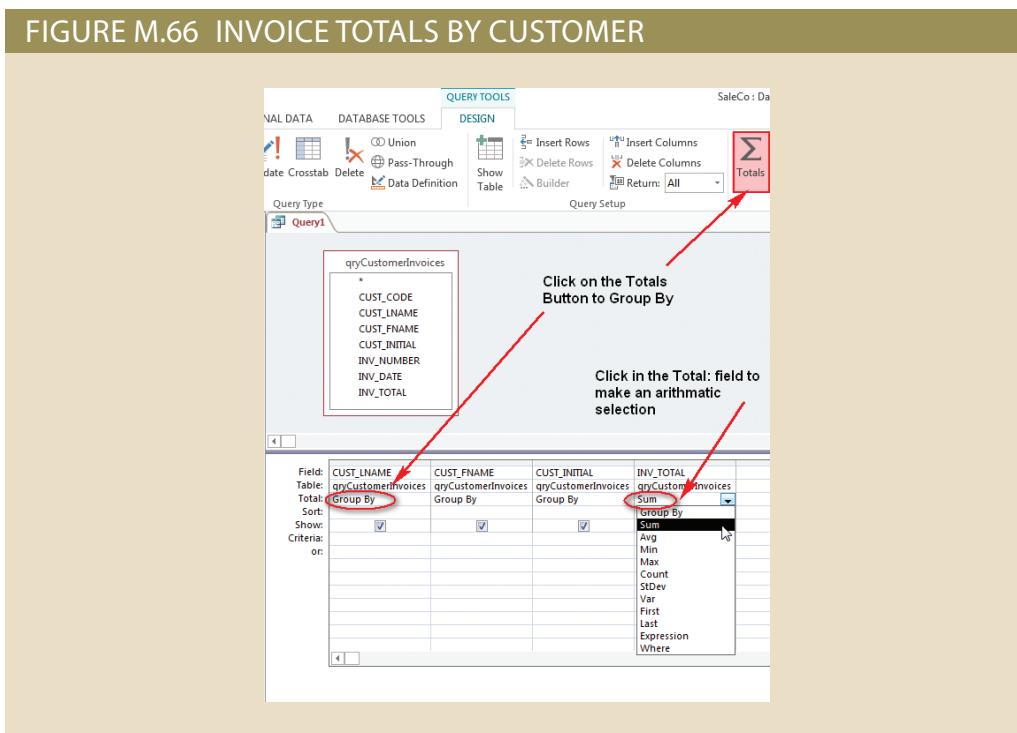
FIGURE M.65 QUERYING A QUERY



After defining the data source, you must define what fields you want to list in your query. Figure M.66 shows the remaining actions required to complete the query design:

- Select **CUST\_LNAME**, **CUST\_FNAME**, **CUST\_INITIAL**, and **INV\_TOTAL** fields.
- Click the **Totals** button on the Ribbon. This adds the **Total** row in the query grid with **Group By** option selected by default.
- In the **INV\_TOTAL** field, click the **Total** row down arrow and select the **Sum** option from the list.
- Set the Properties for **INV\_TOTAL** to **Format: Currency**.

FIGURE M.66 INVOICE TOTALS BY CUSTOMER



Now check the completed query in its **Datasheet View** to generate the results shown in Figure M.67. Note that the \$479.52 sum for customer Leona Dunne is correct. (Customer Leona Dunne's customer number is 10011.)

FIGURE M.67 CUSTOMER INVOICE TOTALS

qryCustomerInvoiceTotals			
CUST_LNAME	CUST_FNAME	CUST_INITIAL	SumOfINV_TOTAL
Dunne	Leona	K	\$479.52
Farris	Anne	G	\$76.08
O'Brian	Amy	B	\$37.77
Orlando	Myron		\$456.59
Smith	Kathy	W	\$166.16

Save the query as **qryCustomerInvoiceTotals** to complete the query design process.

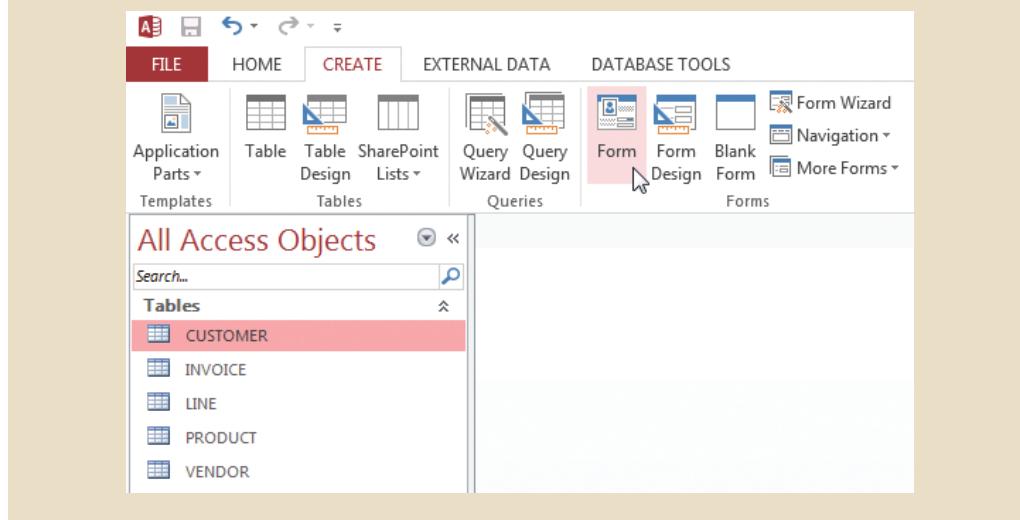
## M-7 Forms

While queries let you get data and/or information from the database, forms let you control the presentation format much better. In addition, forms will enable you to control data input and to present the results from multiple queries and/or tables. Forms can also be used to tie the application components together through menus and other devices. In short, forms are the way in which the end user is best connected to the database and they provide that “professional” look to your data management efforts.

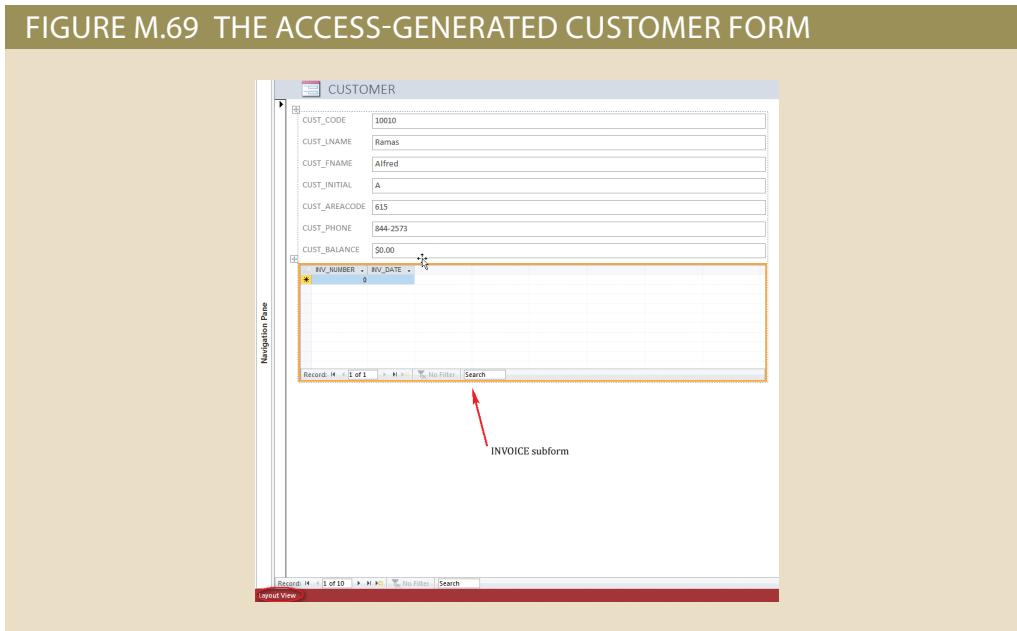
Forms can be based on tables and/or queries. The simplest and most efficient way to create a form is to follow the steps outlined below (also shown in Figure M.68). The steps to quickly create a form are the following:

- Select the **CREATE** tab on the Ribbon.
- Select the **CUSTOMER** table then select the **Form** button to create the form.
- The result is the CUSTOMER form shown in Figure M.69.

**FIGURE M.68 STARTING A NEW FORM**



**FIGURE M.69 THE ACCESS-GENERATED CUSTOMER FORM**



- As you can see by looking at Figure M.69, the form opens up in **Layout View** format. (Given the work you have done with the query development, you should be familiar with the various views.)
- Since the relationship between the CUSTOMER and the INVOICE table has already been established, the form automatically generates an INVOICE subform. Select the **INVOICE** subform and press **Delete**. The selection is confirmed by the selection markers (the orange box around the subform perimeter).
- Save and name the form **frmCUSTOMER** before continuing the form design process.



### Note

Here is another example of self-documentation: The **frm** prefix indicates that the object is a form and the capital letters used in the **CUSTOMER** portion indicate that the query data source was a table named CUSTOMER. If the form's data source had been a query named **qryCustomer**, the form name would have indicated the data source through the use of lowercase letters, using "caps" only to separate the components.

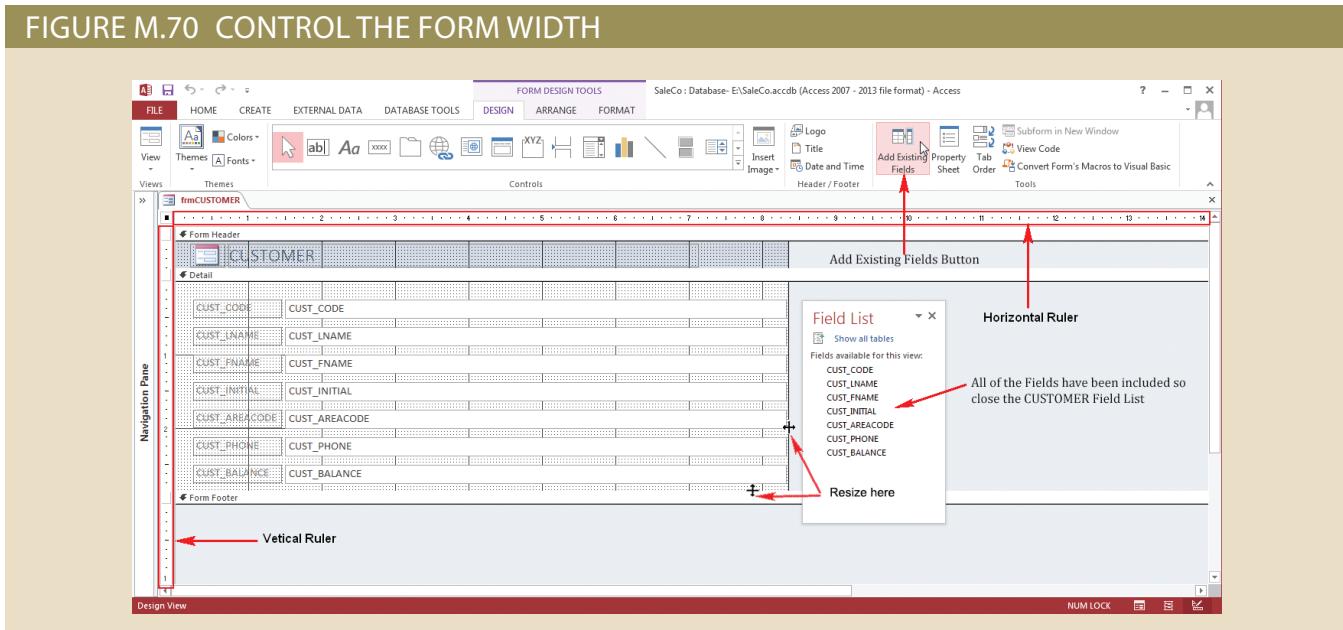
*Making it easy to keep track of the many components in a set of database applications is a mark of professionalism. Nobody wins if nobody can figure out what the database objects are or what they do. Documentation conventions should be made in clear in the formal application documentation.*

## M-7a Editing the Form

While the **frmCUSTOMER** form presents the data properly, it is only a starting point. The form will be enhanced for functionality and appearance reasons (for example, better spacing, customizing labels, text boxes, and so on). It is important that the forms are well designed and functionally accurate to meet the end-user needs. To customize the form you use the **Design View**.

The first time you open the form in Design View format, you may see the Field List window shown in Figure M.70. All the CUSTOMER table's fields have already been included in the form, so go ahead and close this Field List. (If you later want to open this Field List box again, click the **Add Existing Fields** button shown in Figure M.70.)

**FIGURE M.70 CONTROL THE FORM WIDTH**



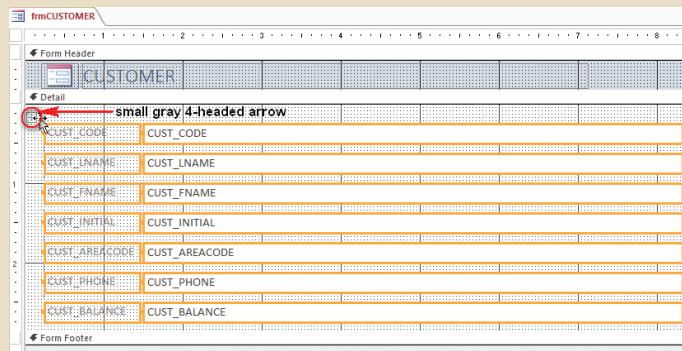
The first thing to notice is that a form has three main areas (see Figure M.70):

- Header area: section on the top of the form containing the title of the form.
- Detail area: contains the form record data—this is where the table data will appear.
- Footer area: section on the bottom of the form that contains text common to all records.

If you want to widen the form, put the cursor on the form's edge to change the cursor to the format shown in Figure M.70, and then drag the form limit to wherever you want it to be. You can control the vertical size the same way. Just put the cursor on the **top edge** of the **Form Footer** and drag up or down to suit your needs. If you put the cursor on the **bottom edge** of the **Form Footer**, you will be able to create a footer area and control its width. (Incidentally, the horizontal and vertical rulers can be used later to help you line up selected output components or to mark multiple selection on the form.)

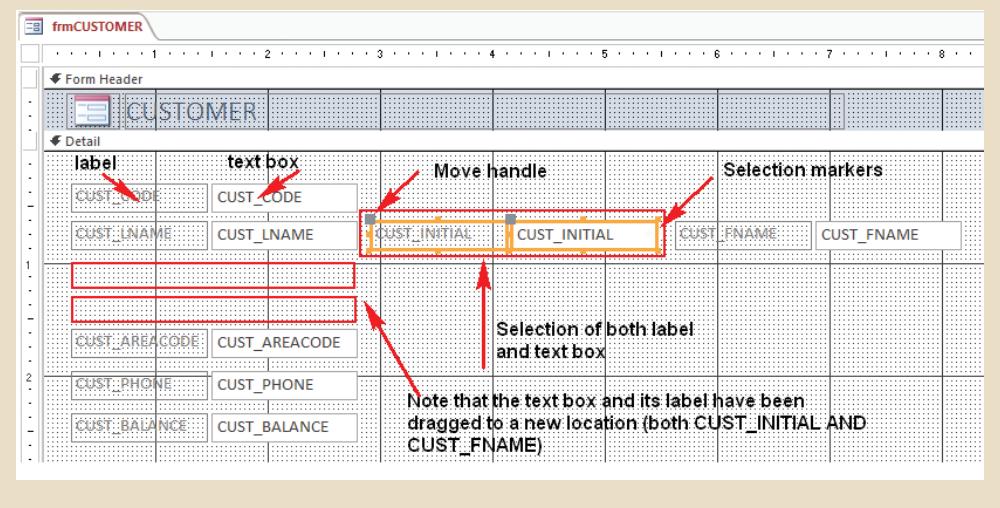
The table fields appear in the Detail area of the form, one record at a time. By default, when using the Form button on the Ribbon the fields are grouped together as a single group. This default layout arrangement for the fields is known as "Stacked." This means that all of the fields are contained in a single layout and resized or moved as one group. To remove this layout, select the one of the fields in the Detail area, and a small gray 4-headed arrow icon will appear in the top left corner. Click this **4-headed arrow** and all of the fields will automatically be selected (see Figure M.71); then select the **ARRANGE** tab and click **Remove Layout**. This allows you to now move and resize the fields individually. Figure M.71 shows how to select all the attributes when they are in the stacked arrangement.

FIGURE M.71 REMOVING THE LAYOUT



Now, you can rearrange the fields and labels in your form. Use Figure M.72 as a guide to rearrange your form fields. Notice that the controls in Figure M.72 have been resized and lined up.

FIGURE M.72 DESIGN COMPONENTS



As you examine Figure M.72, keep the following points in mind as well as the shapes from Table M.4:

- The move handle is the small dark square box situated on the top left of each control.
- Selecting the gray move handle of the text box portion selects only the text box.
- Selecting the gray move handle of the label box selects only the label box.
- To move or resize both the text box and label you will need to select them both, you can click anywhere in the center of the label or the center of the textbox to select both of them. Holding down the CTRL key will allow you to select them both as well or make multiple selections. Figure M.73 also shows other ways to select multiple components.
- To align several fields so that they are along a straight edge, highlight the attributes you wish to align; then on the **FORM DESIGN TOOLS** tab, select the **ARRANGE** tab, select the **Align** button, and then choose the option you want.

FIGURE M.73 MULTIPLE COMPONENT SELECTION

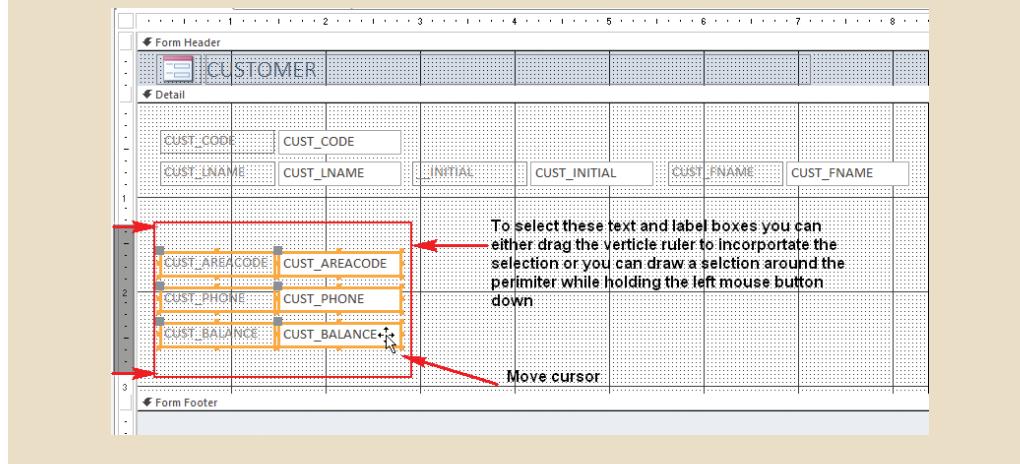


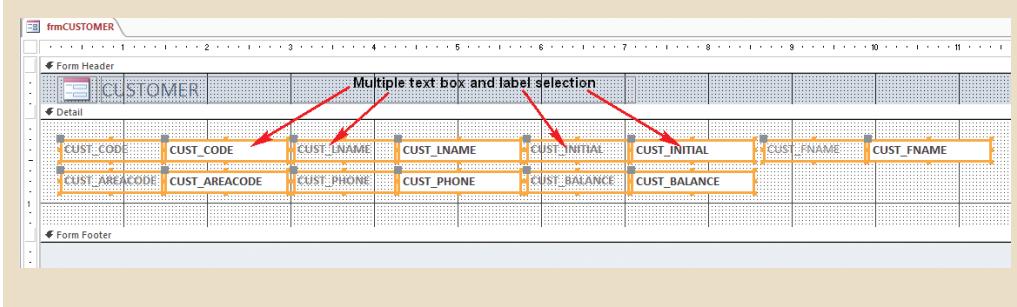
TABLE M.4

## MOUSE POINTER SHAPES

SHAPE	WHEN WILL IT SHOW UP?	WHAT DOES IT DO?
	When you point to any control that is not selected on the form	A single click will select a control.
	When the upper left hand corner of a control is pointed to in design view or the middle of a control in layout view. In design view the move handles will appear for you to select the control.	When the control(s) is selected this mouse pointer will drag the control(s).
	When you point to any sizing handle of a control in form or design view	Dragging with this mouse pointer resizes the control.
	When you point to the edge of any form in Design View	Drags the extents of a form to make the form bigger or smaller.

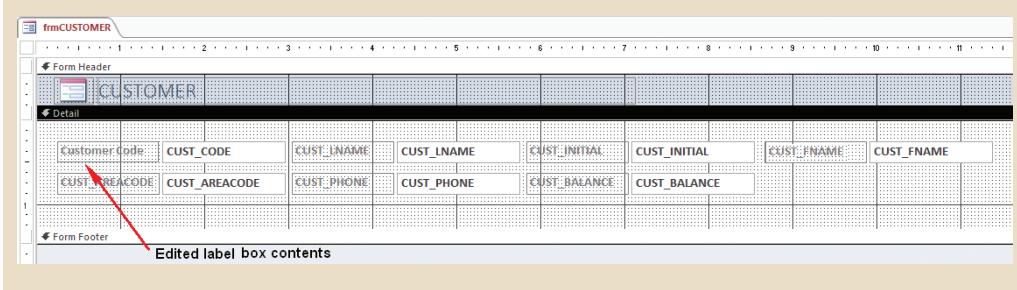
Practice moving the form components around until they match Figure M.74. Then use the same techniques that you learned earlier to change text-formatting properties and to change the font to **Bold** for both labels and text boxes, as shown in Figure M.74.

FIGURE M.74 FONT SET TO BOLD



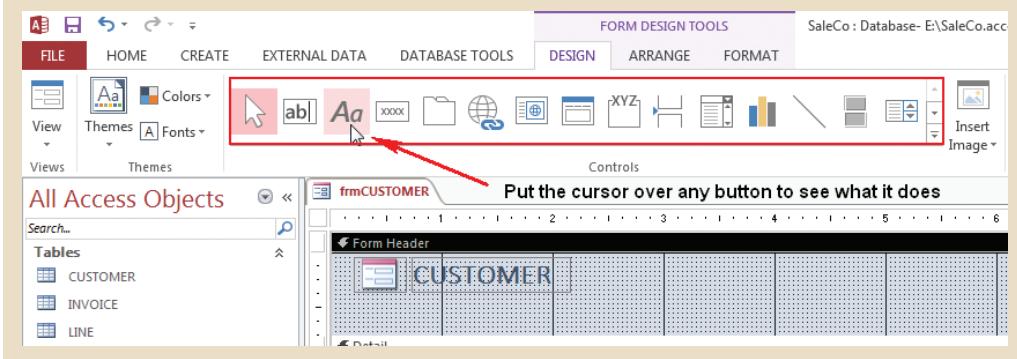
You can also change the label box contents by selecting the label box and then clicking on that selected label box to put it in edit mode. You can then edit the label text. Note that Figure M.75 shows that the CUST\_CODE label was changed to **Customer Code**. (Remember that the label box width can be changed by dragging its limits.)

FIGURE M.75 EDITING THE LABEL BOXES



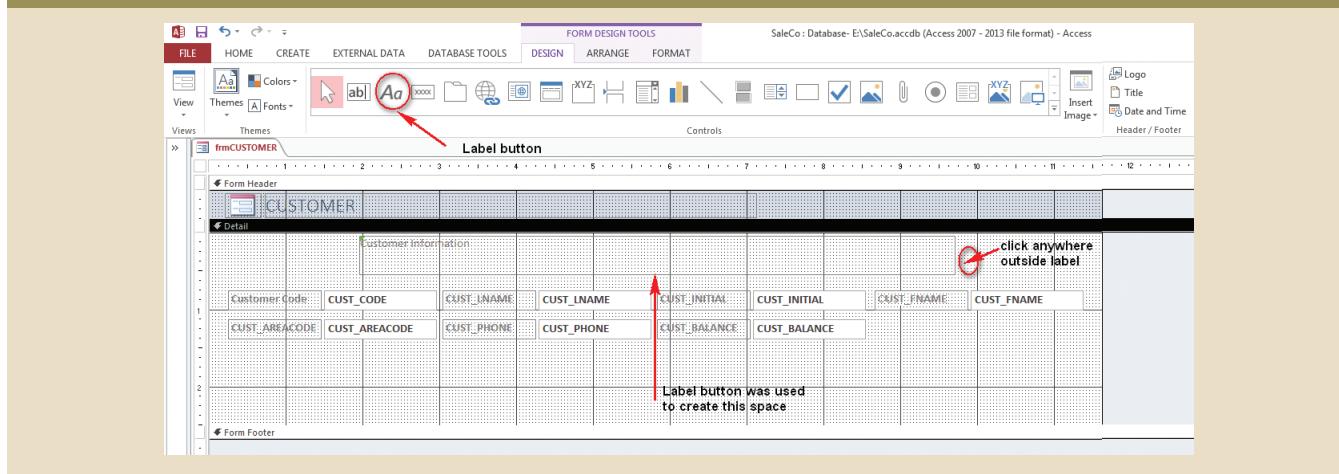
You have a large number of form design tools available. For example, you can add additional text, create boxes to delineate form components, and so on. Click the DESIGN tab to view the **Controls** options you see in Figure M.76.

FIGURE M.76 CONTROLS OPTIONS



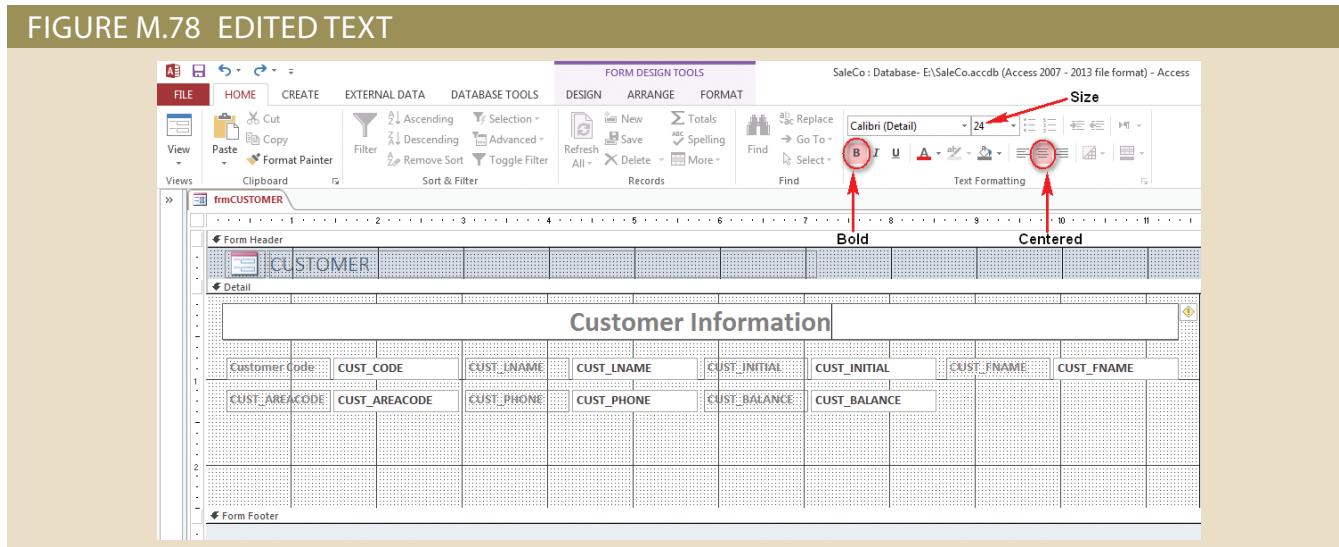
For example, you can add a label to the form. In this case, you will add a label in the Detail area, above the controls in your form. The first step is to create space for the label by selecting all the existing controls in the Detail area and moving them down about half an inch, so the bottom of the first row of controls rest on the one-inch mark. Next, click the **Label** button in the **Controls** group on the **DESIGN** tab (see Figure M.76). Then click and drag on the form Detail area to insert the label control as shown in Figure M.77. If you drag the cursor anywhere on the form, you will create a box in which you can type whatever is needed. In this example, the typed text is **Customer Information**. Click anywhere outside the label to return the cursor to its normal function as shown in Figure M.77.

FIGURE M.77 USING THE LABEL TOOL



You can edit the text as needed. Note that Figure M.78 shows that the font size was changed, the font was bolded, the text was centered, and the label box was resized.

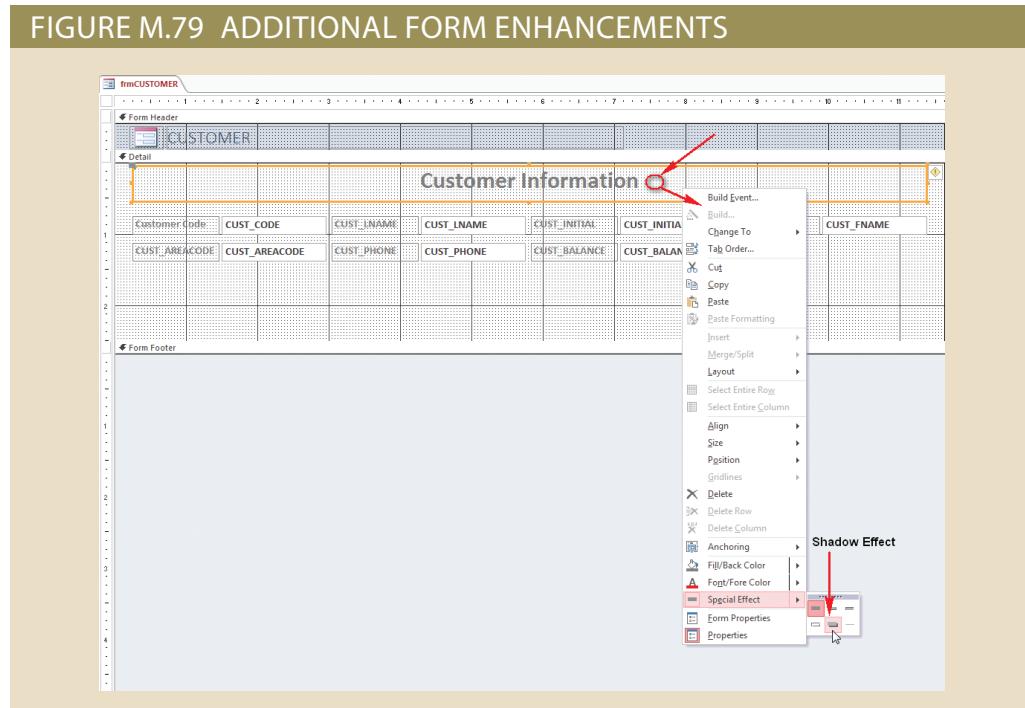
FIGURE M.78 EDITED TEXT



The form's looks can be improved by giving the labels and/or text boxes borders a certain special effect and by adding color. You can use the Rectangle tool to create logical groupings of information presented on the form. For example, let's add the Shadow Special Effect to the Customer Information label box. To accomplish this do the following:

- Select the **Customer Information** label as shown in Figure M.79. (The selection is confirmed by the selection markers—the orange box around the label perimeter.)
- Right-click the selection, the context menu appears.
- Select the **Special Effect** option, and then select the **Shadow** effect (second row, second column) option as shown in Figure M.79.

FIGURE M.79 ADDITIONAL FORM ENHANCEMENTS

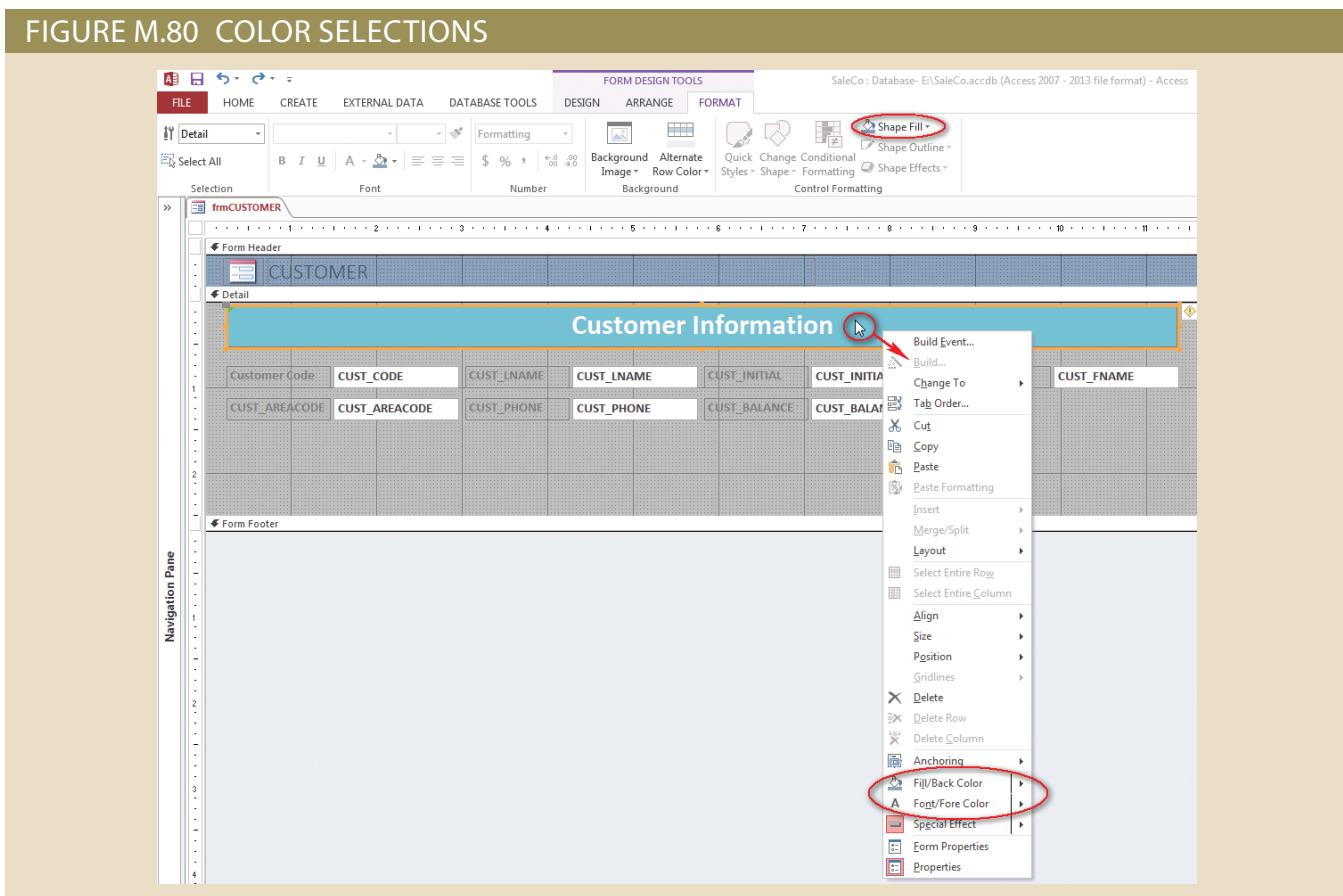


Let's take a look at just a few more form design options. For example, if you look at Figure M.80, you will see that there are quite a few color options available. To match the figure you see here, you can take the following actions:

- To make the form Detail area gray, right-click any empty portion of the Detail section of the form. In the context menu that appears, click the **Fill/Back color** option and click on a gray color square.
- To make the form Header area blue, right-click an empty portion of the header section of the form. On the context menu click the **Fill/Back color** option and click on a blue color square.
- To make the **Customer Information** label background aqua, click the label to select it. Right-click it to open the context menu, click the **Fill/Back color** option and click on an aqua color square.
- To change the font color in the label to white, make sure that the label is still selected, right-click to open a context menu, click the **Font/Fore color** option, and click on the white color square.

Note that Access keeps track of all the recently selected colors to make it easy to later match color selections of other form components.

FIGURE M.80 COLOR SELECTIONS



It is often useful to group logically similar fields together as a visual unit. You can use the Rectangle tool to draw a rectangle around such a group. To get the job done, go to Controls group under the DESIGN tab and select the **Rectangle** button. If you don't see the Rectangle button in the group, scroll to the right of the group to see the additional controls. Then drag a rectangle around the fields you want to group. The rectangle is initially clear and just shows its outline. However, you can right-click the outline of the rectangle and use the **Fill/Back color** option on the context menu to give the rectangle a color. Figure M.81 shows that the selected fill color was light gray. Unfortunately, the default setting on the Rectangle tool places the rectangle in front of the other controls, thus shading them out. However, you can use the **Position/Send to Back** option from the context menu to send the now opaque light gray rectangle to the back, thus making the fields visible again.

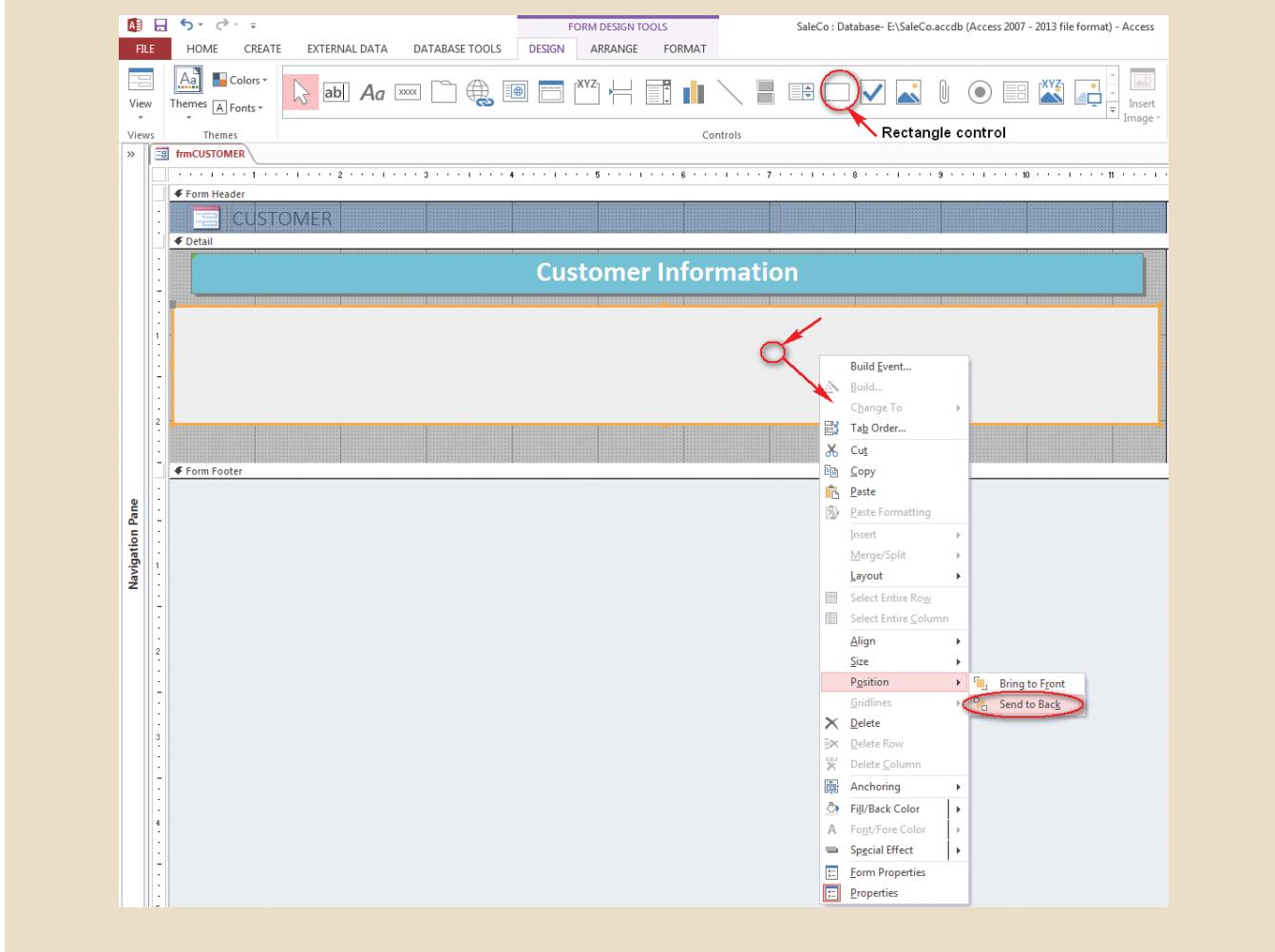


### Note

You could also use the **Bring to Front** or **Send to Back** option from **ARRANGE** tab under the **FORM DESIGN TOOLS** menu to position the control where you want.

In addition, you can change the background color of any control by using the **Shape Fill** option on the **FORMAT** tab.

FIGURE M.81 USING THE RECTANGLE TOOL

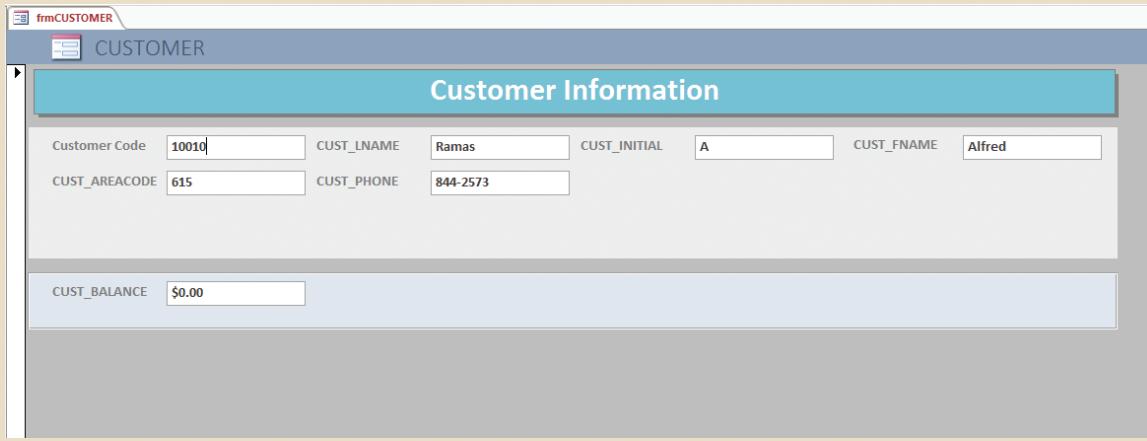


Finally, use the techniques you learned here to make a few more changes:

- Move the **CUST\_BALANCE** field, add a rectangle around it, and change its color.
- Drag the **Customer Information** label limits to line up with the two rectangles on the CUSTOMER form.

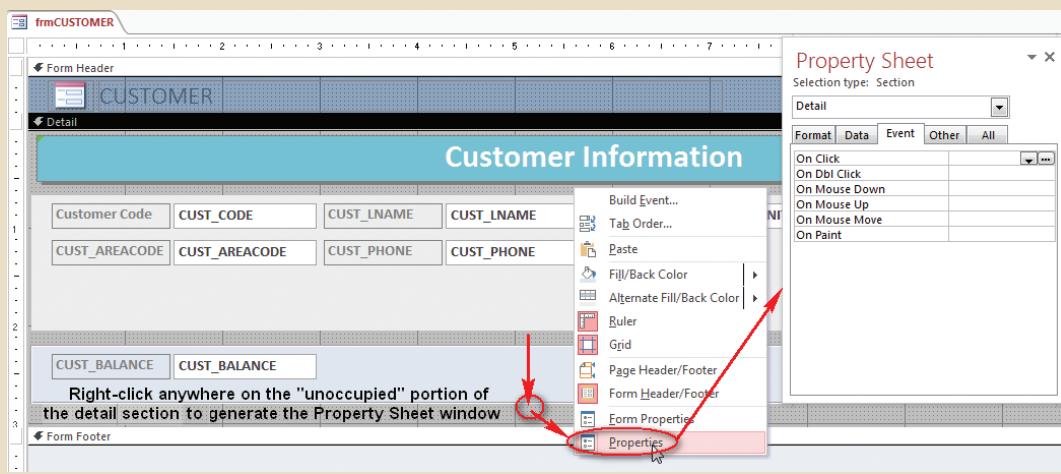
When you are done, open the form in **Form View**. Your form should match Figure M.82.

FIGURE M.82 THE FORM IN FORM VIEW



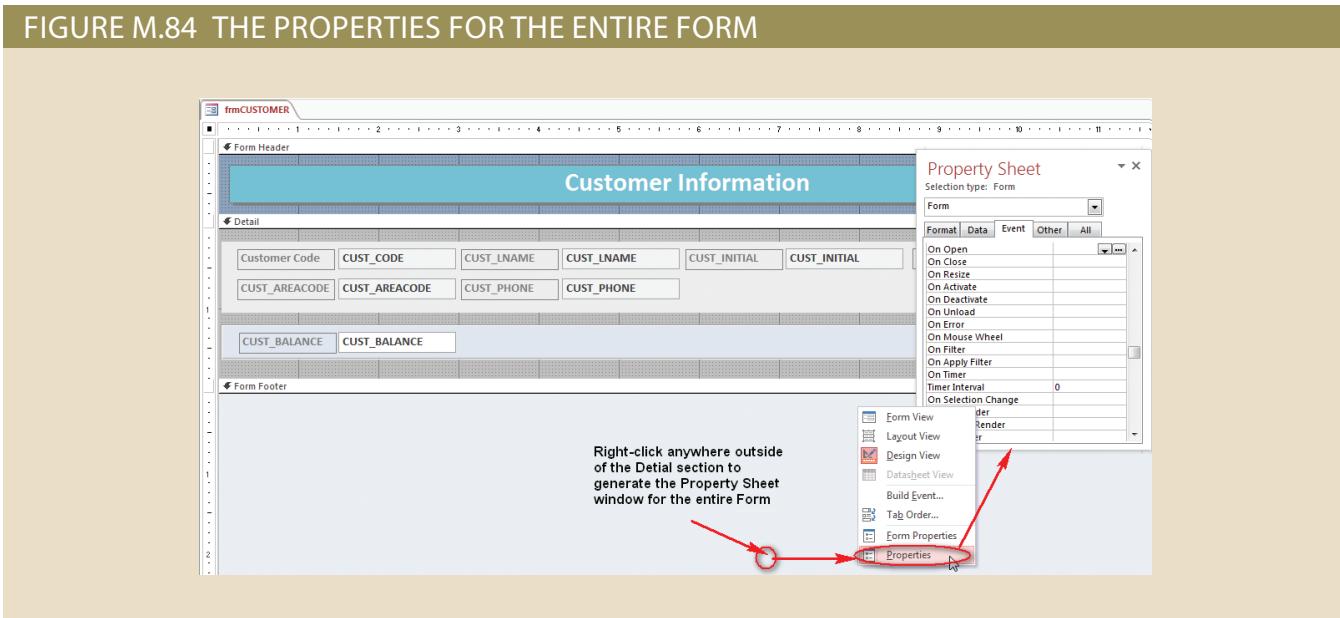
Go back to form **Design View**. Notice that right-clicking on the unoccupied portion of the Detail section of the form in Design View will generate the context menu for the Detail section (see Figure M.83). Once the Properties Sheet is open, you can generate the properties for any Access object by simply clicking on the object. Go ahead and perform this action a few times until it becomes routine.

FIGURE M.83 GENERATING THE PROPERTY SHEET



You can use the context menu to open the Property Sheet window for any Access object. For example, if you want to see the properties of just the CUST\_CODE text box, click the CUST\_CODE **text** box to select it and then right-click and select Properties to access the Property Sheet window. If you want to see the properties of the Customer Code: **label** box, click the CUST\_CODE label to select it and then right-click it and select Properties to access the Property Sheet window. Figure M.84 shows the Property Sheet for the entire form.

FIGURE M.84 THE PROPERTIES FOR THE ENTIRE FORM

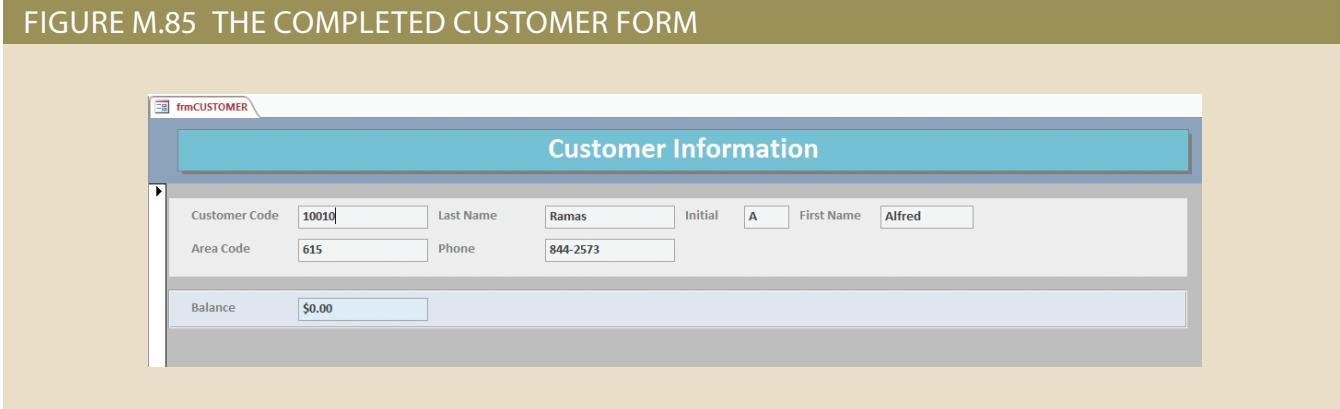


The form is essentially completed. However, you may want to make a few additional changes:

- Put the **Customer Information** label in the Header section of the form. Delete the current content of the Header section. Then, drag and drop the label box to the header. You can create the header space by simply dragging the form's Detail area border down.
- Change the labels on the Detail section, and resize the rectangles, labels, and text boxes to match Figure M.85.
- Finally, you can change the white background for the text boxes to gray using the fill color option shown earlier.
- Save the form and close it.

Figure M.85 shows the final **frmCUSTOMER** form in Form View.

FIGURE M.85 THE COMPLETED CUSTOMER FORM



## M-7b Forms Based on Queries

Forms may be used to present data and/or information from multiple sources. Queries are particularly good as a basis for form design, because they can access multiple tables directly.

- Right-click **qryCustomerInvoices** and select **Copy**. Then press **Ctrl+V** to paste the query. Name the new query **qryInvoicesByCustomer**.
- Open it in **Design View**; remove CUST\_LNAME, CUST\_FNAME, and CUST\_INITIAL fields from the query design as shown in Figure M.86. To remove the fields, hover over the top of the CUST\_LNAME field; a small black arrow will appear. Once the arrow appears, click and slide the mouse over to select all three fields. Once selected they should all be highlighted black. Click the **Delete** button to delete the selected fields.
- Now we are going to build the new field called **Customer Name** using the Expression Builder as shown in Figure M.86. Note that the new field uses an expression that concatenates the customer's first name, middle initial, and last name. However, because some customers may not have an initial, a logical function must be used. The Immediate If (**IIf**) function uses the following format:

**IIf(expression, action if true, action if false)**

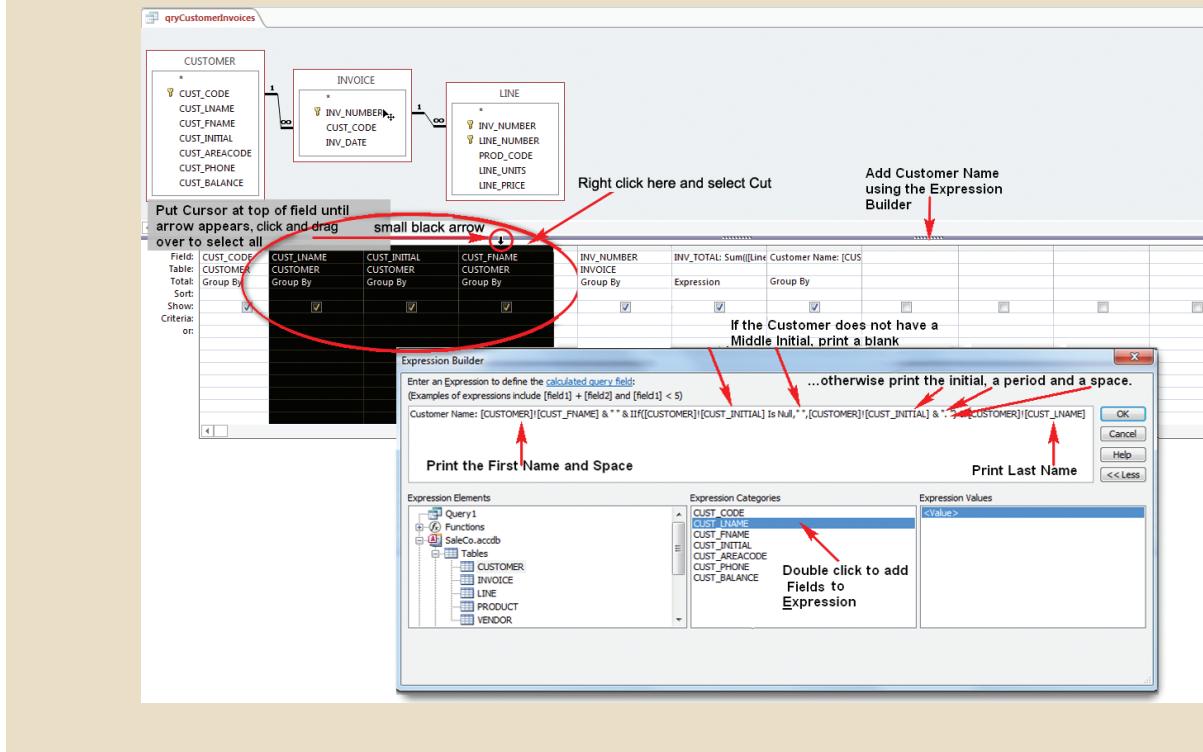
Note that the two possible actions are separated by commas.

- If there is no middle initial – that is, the **CUST\_INITIAL Is Null** – a blank space must be printed after the customer's first name. If the customer has a middle initial — that is, the customer initial is not null -- the initial must be printed, followed by a period and space. In either case, the customer's last name must be printed, so the CUST\_LNAME is not included in the logical **IIf** statement.
- To build this query, right-click the empty field to the right of the INV\_TOTAL field and select **Build**. The Expression Builder window appears. The left side of the window contains the Expression Elements column. Here you can click the plus “+” sign to expand and drill down the elements available for you. For example, you can click the plus sign to the left of the database name to get the Tables, Queries, Forms, and Reports. Keep expanding the tables to see all available tables and then all available fields on a table.
- From the Expression Elements list, expand the tables, then expand the CUSTOMER table and double-click **CUST\_FNAME**, then **CUST\_INITIAL**, **CUST\_INITIAL** once more, and then **CUST\_LNAME** to add each attribute needed for the expression.
- Note that when you add the attributes «**Expr**» appears between them, this is not needed so just delete them when they show up.
- Finish the expression by typing the **&**, the blank spaces enclosed by quotes, and the **IIf** function components as shown in Figure M.86. When you are done with the Expression Builder, click **OK** to save the expression. The expression should read as follows:

**Customer Name: [CUSTOMER]![CUST\_FNAME] & " " &  
IIf([CUSTOMER]![CUST\_INITIAL] Is Null," ",[CUSTOMER]![CUST\_INITIAL]  
& "?")& [CUSTOMER]![CUST\_LNAME]**

- Leave the INV\_TOTAL as an Expression and change **Customer Name** to **Group By**.
- Finally rearrange the fields so they are ordered as CUST\_CODE, INV\_NUMBER, Customer Name, INV\_DATE, and INV\_TOTAL, as shown in Figure M.87. To move the fields simply hover over the top of the field until the black arrow shows up, click, click again, and drag.

FIGURE M.86 THE DESIGN VIEW OF THE INVOICE QUERY



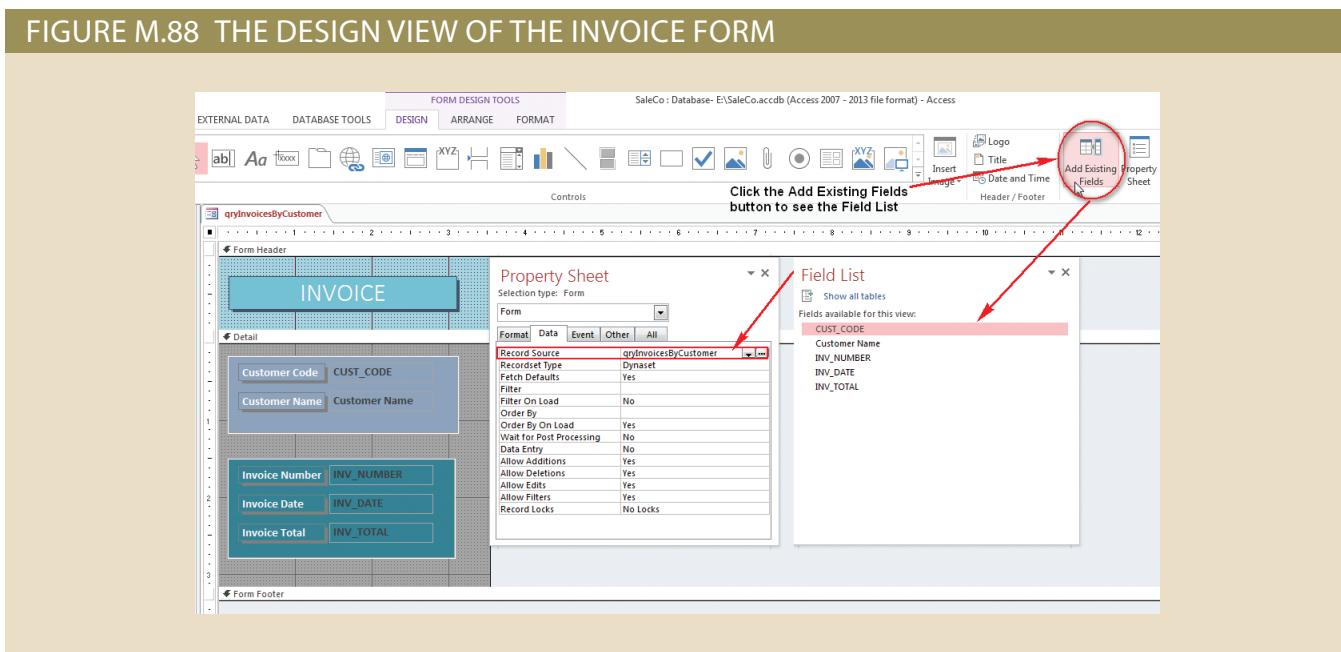
Save the query and then open it in **Datasheet View** as shown in Figure M.87.

FIGURE M.87 THE DATASHEET VIEW OF THE INVOICE QUERY

CUST_CODE	INV_NUMBER	Customer Name	INV_DATE	INV_TOTAL
10011	1002	Leona K. Dunne	16-Mar-18	\$10.78
10011	1004	Leona K. Dunne	17-Mar-18	\$37.66
10011	1008	Leona K. Dunne	17-Mar-18	\$431.08
10012	1003	Kathy W. Smith	16-Mar-18	\$166.16
10014	1001	Myron Orlando	16-Mar-18	\$26.94
10014	1006	Myron Orlando	17-Mar-18	\$429.66
10015	1007	Amy B. O'Brien	17-Mar-18	\$37.77
10018	1005	Anne G. Farriss	17-Mar-18	\$76.08

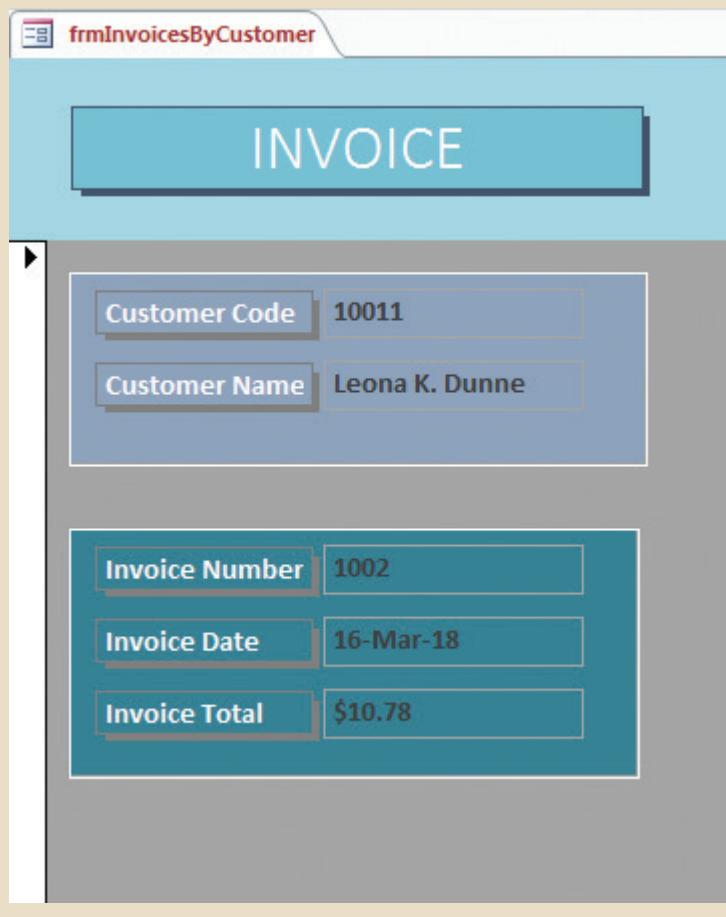
Next, begin the form design just as you did in the previous section. However, this time you use the **qryInvoicesByCustomer** query as the data source. Use the formatting techniques you learned in the previous section to produce the form you see in Figure M.88.

FIGURE M.88 THE DESIGN VIEW OF THE INVOICE FORM



Save the form as **frmInvoicesByCustomer**. Figure M.89 shows the final form view relative to its **Forms** list.

FIGURE M.89 THE FORM VIEW OF THE INVOICE FORM



## M-7c Forms with Subforms (Using the Form Wizard)

Forms contribute much to the ability to display data and/or information in a meaningful and visually appealing way. However, from an information management point of view it would be very desirable to see the interaction of related data. Fortunately, information from the LINE, INVOICE, and CUSTOMER tables can be combined by showing the customer and invoice data with the related invoice lines on a single screen through a form/subform combination. In this example, the subform contains the LINE data, while the main form contains the “parent” CUSTOMER and INVOICE data. (In a 1:M relationship, the “M” side would be found in the subform and the “1” side would be seen in the main form.) Go back and study the relationships and notice that the CUSTOMER table is related to the INVOICE table via CUST\_CODE and the INVOICE table is related to the LINE table via INV\_NUMBER. These relationships must be established for the form/subform to work properly.

In this section you will learn how to generate one form as a subform to another using the Form Wizard. You will also learn some additional techniques that will turn out to be very useful when you try to control input via a form.

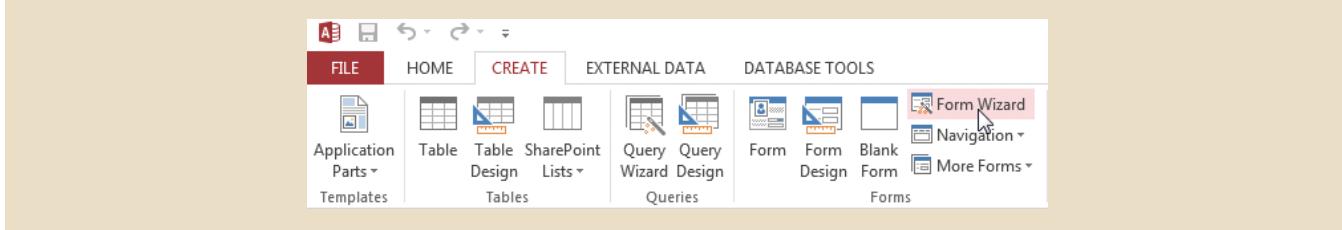


### Note

Queries and Reports also have wizards that you can use to quickly generate what you are trying to represent. They all present you with a series of prompts for options that you will need to select in order to customize your query, form, or report.

Let's go ahead and build the **Invoice** form with a line subform using the wizard. Start by clicking the **Form Wizard** button on the **CREATE** tab as shown in Figure M.90.

**FIGURE M.90 SELECTING THE FORM WIZARD OPTION**



The Form Wizard will take you through a series of four prompts

1. Which fields do you want on your form? (This is shown in Figure M.91.)

Add the following fields in the following order as shown in Figure M.91:

- Select the **INVOICE** table then select the **INV\_NUMBER** and **INV\_DATE** fields.
- Select the **CUSTOMER** table then select the **CUST\_CODE**, **CUST\_LNAME**, **CUST\_FNAME**, **CUST\_INITIAL**, **CUST\_AREACODE**, and **CUST\_PHONE** fields.
- Select the **LINE** table then select the **LINE\_NUMBER**, **PROD\_CODE**, **LINE\_UNITS**, and **LINE\_PRICE** fields.
- Click the **Next** button.

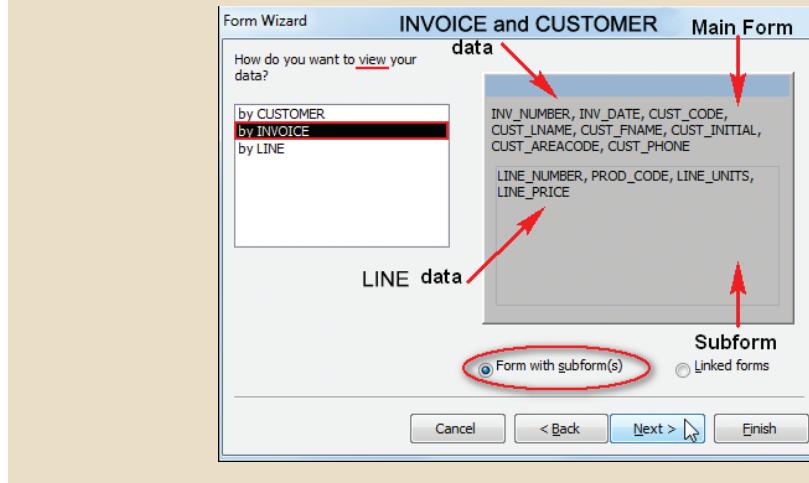
FIGURE M.91 ADDING FIELDS TO THE FORM WIZARD



2. How do you want to view your data?

Select **by INVOICE**, and choose the **Form with subform(s)** option as shown in Figure M.92. This tells the wizard that you want the main form to contain the INVOICE and CUSTOMER data while the subform will contain the LINE data. Due to the fact that we added INVOICE, CUSTOMER, and LINE in that particular order, the Form Wizard knows by default that the main form will contain the INVOICE and CUSTOMER data and the subform will contain the LINE items related to that particular invoice number. If we had added them in a different order then Figure M.92 would look different, so the order is important when using the Form Wizard. Therefore each invoice number with the related customer information is shown in one record with the line items pertaining to that one invoice number. If a customer has multiple invoices then the customer will be shown in multiple records with the corresponding line items. Click **Next**.

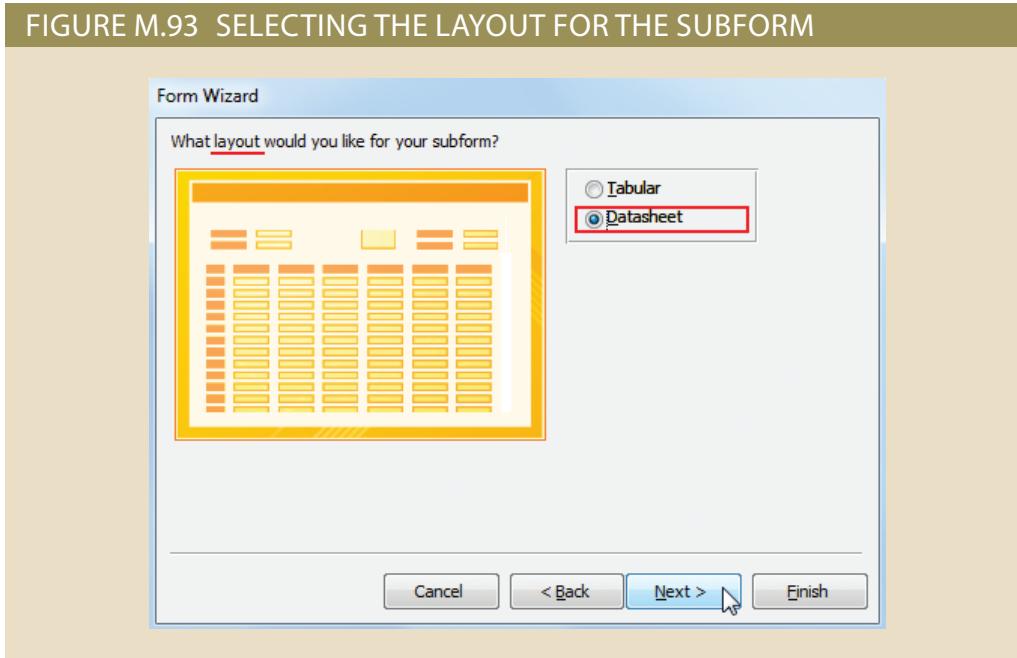
FIGURE M.92 SELECTING THE VIEW FORMAT FOR THE FORM



3. What layout would you like for your subform?

Choose the **Datasheet** option as shown in Figure M.93. This will show the subform in Datasheet View which shows all the line items related to the invoice (it looks similar to a query). Click **Next**.

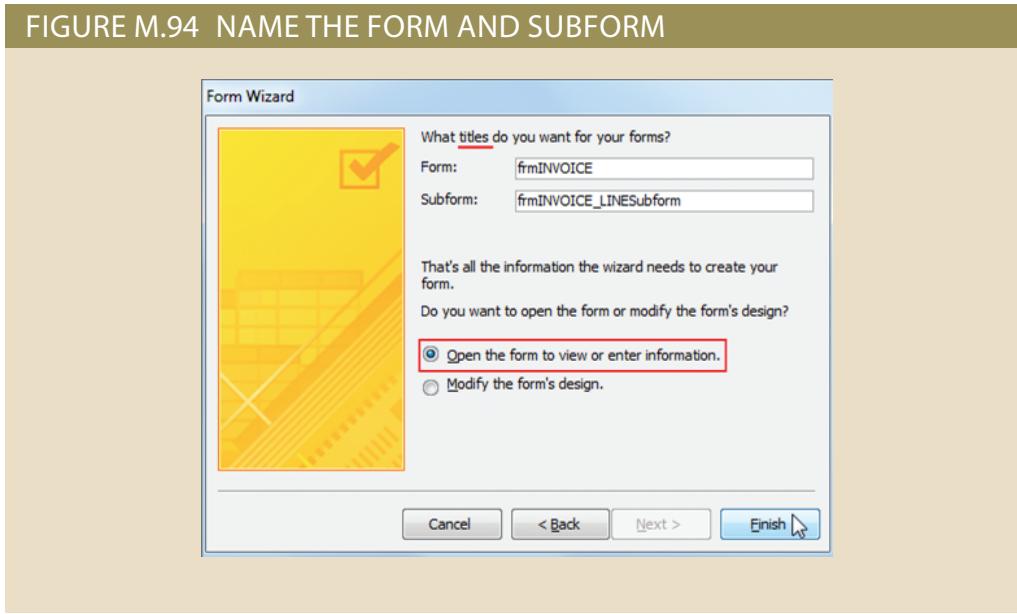
**FIGURE M.93 SELECTING THE LAYOUT FOR THE SUBFORM**



4. What titles do you want for your forms?

Use the proper naming conventions and name the form **frmINVOICE** and the subform **frmINVOICE\_LINESubform** as shown in Figure M.94. Select the **Open the form to view or enter information** option. Click **Finish**.

**FIGURE M.94 NAME THE FORM AND SUBFORM**

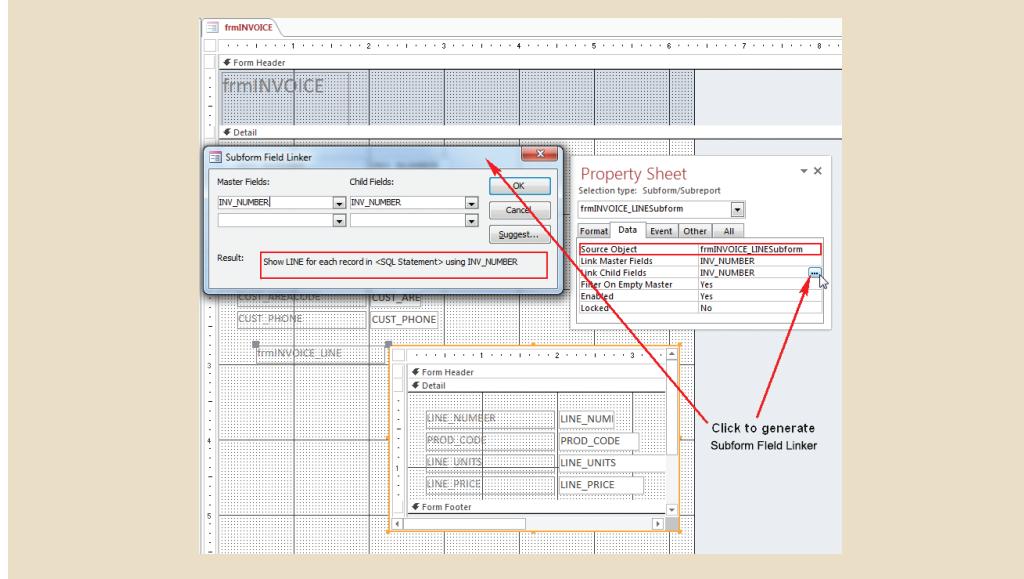


As soon as you finish the Form Wizard prompts, the form and subform appear in Form View. Switch to **Design View** and you will see the generated label for both the form and subform indicating which is which.

Select the subform (an orange border will surround the subform). Right-click on the orange border and select **Properties**. The Property Sheet window for the subform will open. Under the **Data** tab, you will see the **Link Master Field** and **Link Child Fields** properties. These properties tell MS Access how your main form and subform are related (see Figure M.95). In this case, the default values are correct. However, you can change these

settings anytime. For example, click the three dots on the right ([...]) of **Link Child Fields** to produce the Subform Field Linker dialog box you see in Figure M.95. Note that the (correct) link is made through the INV\_NUMBER, because the INV\_NUMBER is the foreign key (FK) in the LINE table that links the LINE table to the INVOICE table. Because the relationship has already been established and was established correctly, this selection is correct, so just click **OK** to accept it. You are done; that's all there is to it. Close the Property Sheet.

FIGURE M.95 SETTING THE LINK



Next we are going to resize the subform so that we can add a new field. This is the same procedure we used to resize the other two forms; you are just performing it on the subform now. The selection box is the size that the subform will appear in relation to the form and within that selection you can resize the form to fit the new selection size. With the subform still selected, drag the Detail area down just below the 2" mark on the vertical ruler (this is the vertical ruler for the subform, not the entire form). Next drag the entire selection box down just past the 2-1/2" mark. Drag the subform footer down about  $\frac{1}{2}$ " as well. You may also need to increase the extents of the original form. Figure M.96 shows the appropriate pointers to use to achieve the proper size. The size of your subform should look similar to Figure M.96.

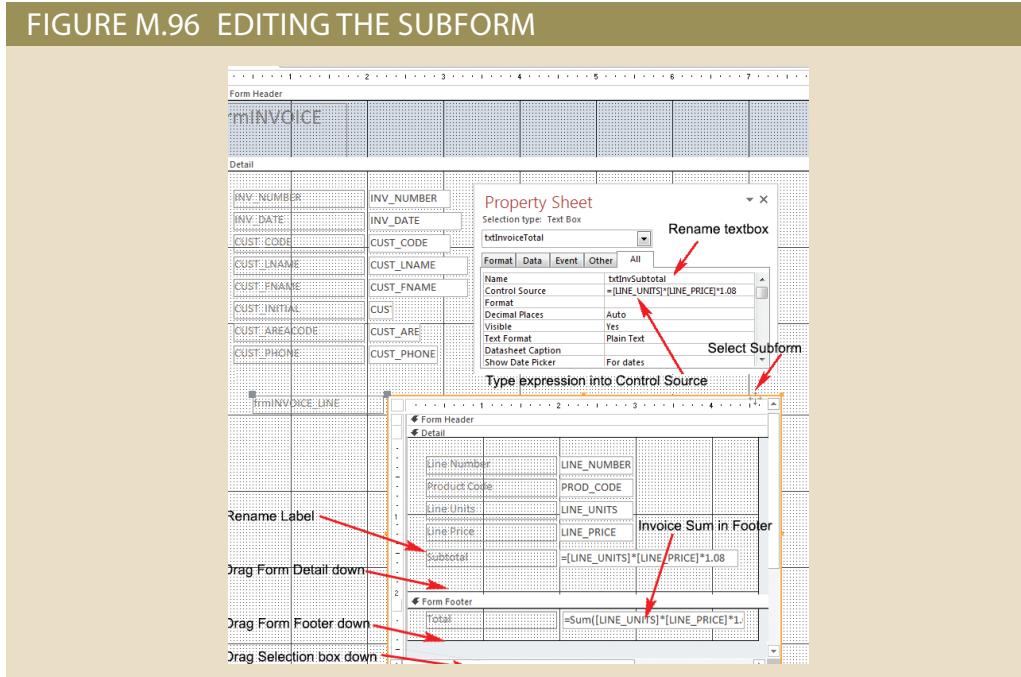
Now we will learn how to use text boxes to add calculations to the form. This is similar to the Expression Builder in query design but instead you will access the Expression Builder through the Control Source on the Property Sheet. This process is demonstrated in Figure M.96.

- Click the **DESIGN** tab, click the **Text Box** button in the **Controls** group, and then click under the LINE\_PRICE located inside the subform and draw a new text box. You will see its automatically generated label. Select the label and click the **Property Sheet** button. Click the **All** tab and rename the label to **lblSubtotal** and type **Subtotal** in the Caption box.
- Select the text box. The Property Sheet shows the properties of the text box. Click the **All** tab, and rename the text box **txtInvSubtotal**.
- Click the **Control Source** property. Click the three dots to the right of **Control Source** and build or type the expression **=[LINE\_UNITS]\*[LINE\_PRICE]\*1.08**. If you know

the names of the fields and what you are trying to calculate you can simply type it into **Control Source** or the text box itself.

- Create a text box in the footer of the subform. Use the proper naming conventions to give the label a good name and change the caption to **Total**. Name the text box **txtInvTotal**. Datasheet View does not show subform footers in form view so this text box will be hidden. It is important that it goes in the footer of the subform because the subform contains the LINE\_UNITS and LINE\_PRICE fields. If you were to put this text box in the footer of the main form an error would occur because it will not recognize where the fields are coming from.
- The txtInvTotal text box will compute the total value of the invoice. In the Control Source for this control, enter **=SUM([LINE\_UNITS]\*[LINE\_PRICE]\*1.08)**. This will add up all the invoice lines for the invoice lines displayed on the subform. Because the subform displays the lines for one and only one given invoice, the txtInvTotal control represents the total for that given invoice.
- In the main form, create a text box in the bottom of the Main Form just below the subform. Type **=[frmINVOICE\_LINESubform].[Form]![txtInvTotal]** into the **Control Source**. This text box will essentially call the value that is stored in the txtInvTotal text box that is located on the INVOICE\_LINE subform. It will reflect the Total Invoice for each individual record. Therefore, the label for this text box will use the caption: Invoice Total (see Figure M.97).
- Go to the property sheet for the entire form and click the **Data** tab. Under **Record Source** click the three dots on the right. This will bring up the query that generates the output of the form. Change the **Sort to Ascending** for **INV\_NUMBER**. Close and accept the changes.
- Format the form and subform fields the same way you formatted the previous ones.
- Notice that the subform in Datasheet View does not need to be formatted. However, the fields must be presented in a given order. If you need to change the order of fields, switch to Layout View, and change the order of the fields in the subform by selecting and dragging the fields to their proper place. Click the Save button on the Quick Access section of the title bar. This will save the form and the subform field layouts.

**FIGURE M.96 EDITING THE SUBFORM**



Switch to Layout View or Design View. Delete the frmINVOICE\_LINE label and resize the selection box of the subform so that all of the fields are visible. Resize the columns to display the entire headers; modify the layout of the main form fields and labels to match Figure M.97. Note also that there is no INV\_NUMBER field in the subform. Why? Is there a need to have the INV\_NUMBER field on the subform? The answer is No. When using a main form/subform layout, MS Access automatically updates the linking field value in the subform.



## Note

You can only resize the field widths on a datasheet subform in Layout View.

**FIGURE M.97 THE FORM-SUBFORM IN LAYOUT VIEW**

The screenshot shows the frmINVOICE form in Layout View. The main form contains fields for Invoice Number (1001), Invoice Date (16-Mar-14), Customer Code (10014), Last Name (Orlando), Initial (M), First Name (Myron), Area Code (615), and Phone (222-1672). Below these is a subform titled "1st of 2 Invoice Lines" which displays two rows of data:

Line Number	Product Code	Line Units	Line Price	Subtotal
1	1 13-02/P2	1	\$14.99	\$16.19
2	2 23109-HB	1	\$9.95	\$10.75
0		0	\$0.00	\$0.00

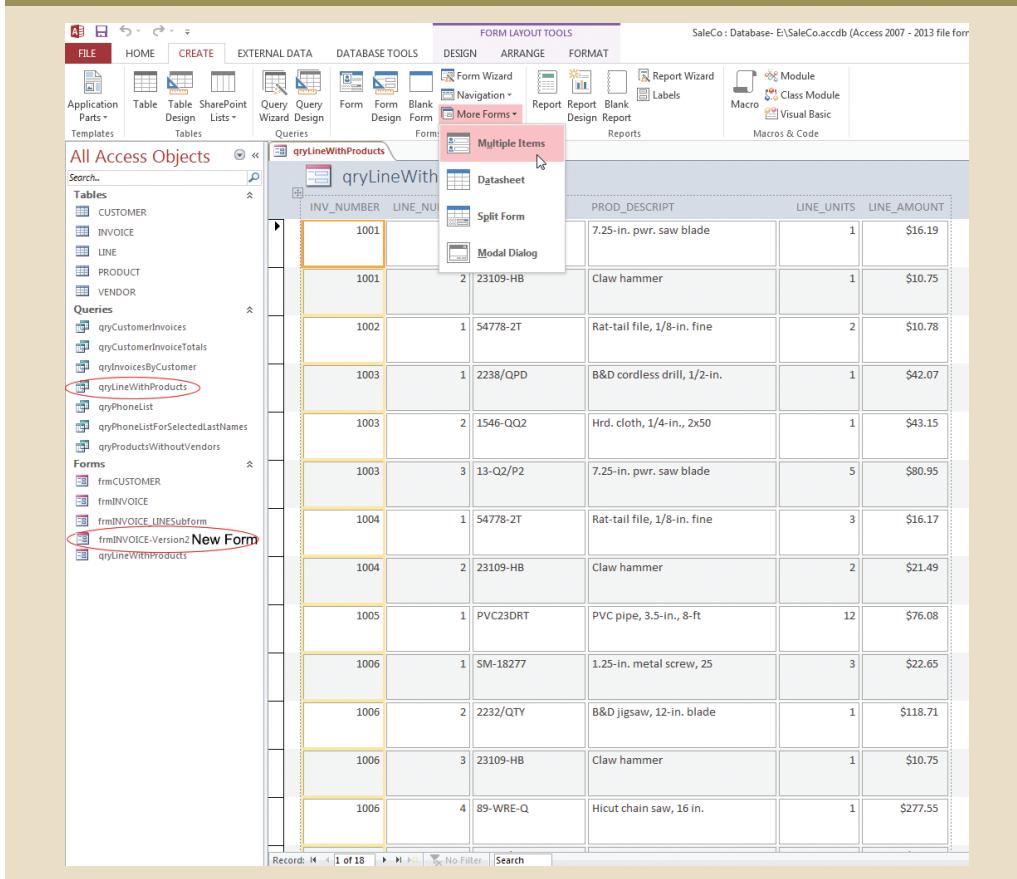
A tooltip "Double click to auto-size column header" points to the header row. Another tooltip "Resize selection box" points to the subform's selection box. The main form's record navigation bar shows "Record: 1 of 2". The subform's record navigation bar shows "Record: 1 of 2". The bottom status bar displays "Invoice Total \$26.94" and "1st invoice has 2 Lines".

Thus far, you saw how forms could be used as subforms on other forms using data from tables. However, you can also create subforms based on queries. If you already have several calculated or concatenated fields in your query, then it would be easier to use the query for the subform instead of the table.

First make a copy of **frmINVOICE** and name it **frmINVOICE-Version2** so that you can replace the subform that was generated from a table with the new one you are going to make from a query. Import the **qryLineWithProducts** query from the **Ch07\_SaleCo** database stored in the student folder using the same technique described in Section M-3.

Select **qryLineWithProducts** from the Navigation Pane, click the **CREATE** tab, and under the Forms group, click **More Forms** and select **Multiple Items**. You will see the output shown in Figure M.98. The **Multiple Items** form produces a form with a tabular output. We have seen the stacked output which was produced by default when we created the frmCUSTOMER and the datasheet output which we chose for our frmINVOICE\_LINESubform.

FIGURE M.98 FORM WITH TABULAR OUTPUT



Naturally, you can modify the wizard-produced form, using the design techniques you learned earlier in this section. Save the form as **frmLineWithProducts-TabularView**.

Next, go to **frmINVOICE-Version2** and substitute the **frmINVOICE\_LINESubform** with the new form by simply changing the Source Object to **frmLineWithProducts-TabularView** as shown in Figure M.99.

You will need to recreate the text box in the subform footer. Copy over the **txtInvTotal** text box and label from **frmINVOICE\_LINESubform**. Then paste it in the new subform footer. The formula is the same and it should work just fine. Subform footers appear in tabular view so the text box and label need to have the Visible property set to **No** as shown in Figure M.99.

Format the form as you did the previous forms, and display the form in **Form View** as shown in Figure M.100. Compare this output with the one shown in Figure M.97. The first figure does not contain the Product Description because we did not use the **PRODUCT** table in the initial Form Wizard.

FIGURE M.99 THE EDITED FORM WITH SUBFORM

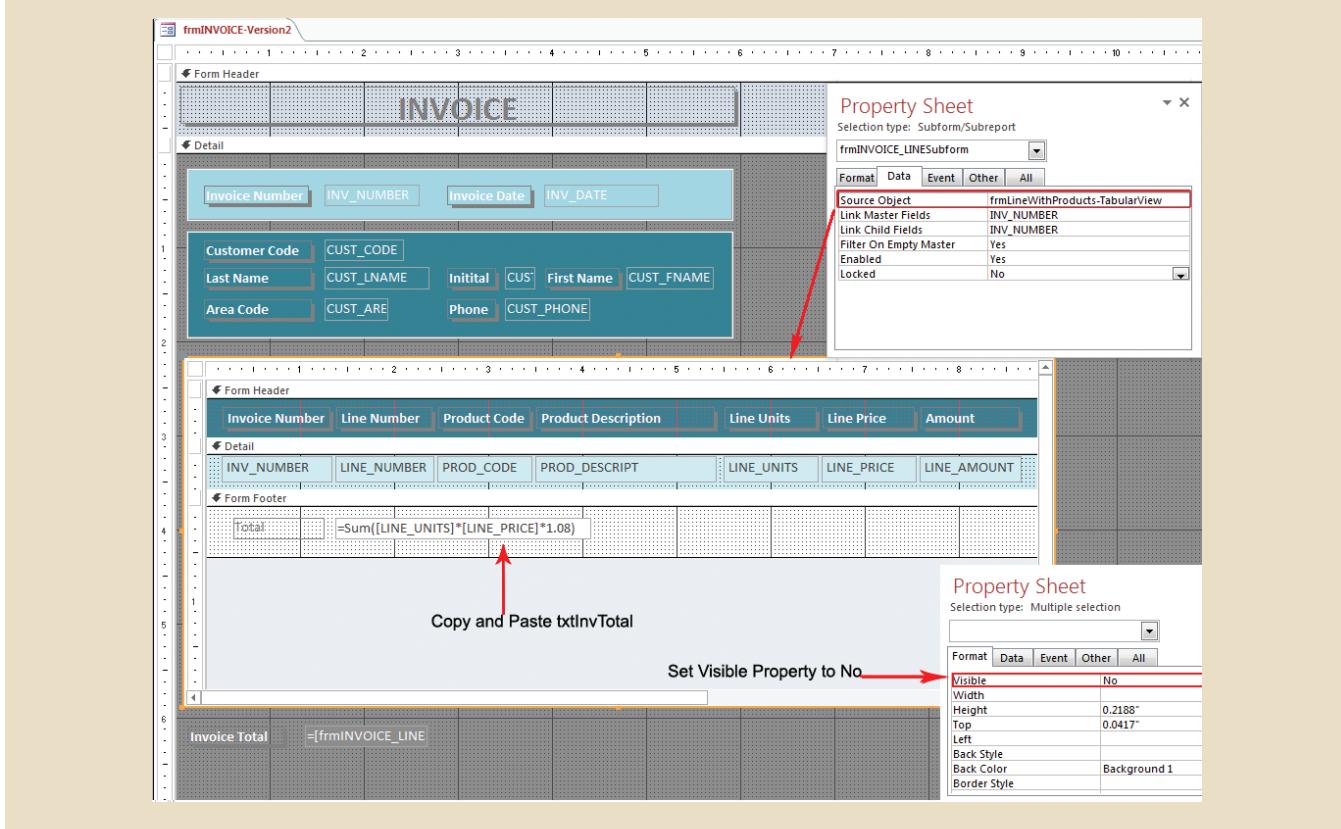
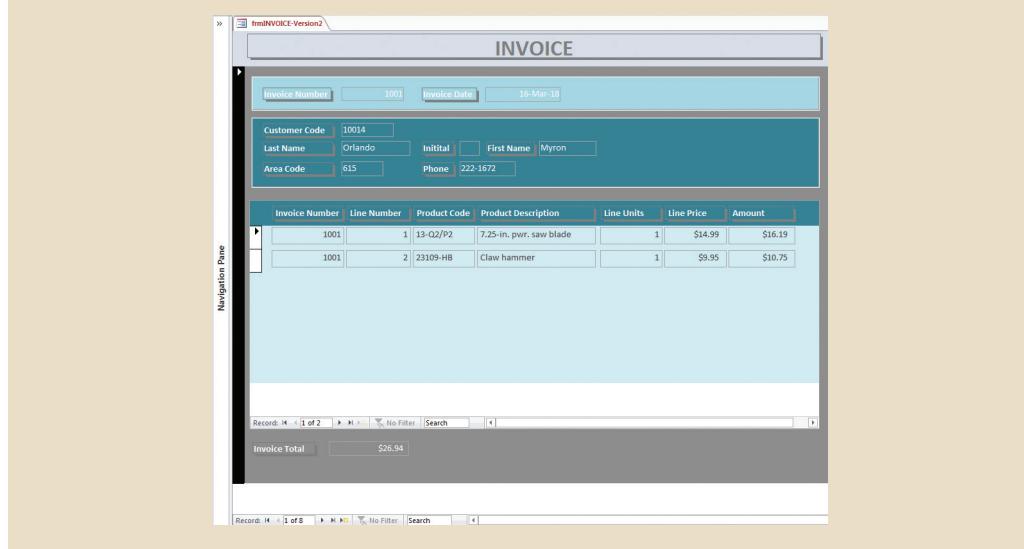


FIGURE M.100 EDITED FORM WITH SUBFORM OUTPUT



## M-7d Controlling Input with Combo Boxes

In a real-world environment, the end user is likely to use a scanner to enter product codes and prices. However, when you are prototyping database applications, you probably don't use such "final product" options. So how do you control end user input when it involves something as convoluted as a product code? The answer is simple—just design the form that uses such inputs to include **combo boxes**. A combo box is simply a text box that lists the available input options from which the end user may choose. To create a simple combo box you must define five properties:

- **Row Source** indicates how the list of values in the drop down list is formed and how the list of values that will populate the combo box drop down list is generated. Generally, the list is populated by a query or table rows, but could also be a comma separated list of values. This property works together with the **Row Source Type** property. Therefore, here you indicate either a query or a table name or a list of values.
- **Bound Column** indicates which of the columns in the combo box drop down list contains the value that will be stored in the table's field (indicated in the Control Source property). Remember, this control represents a field in a table; when the user selects a row from the combo box drop down list, one of the column values in the selected row will be stored in this field. This is indicated by the position of the column in the drop down query, for example: 1 will be the first column, 2 will be the second, etc. The Bound Column defaults to the first column.
- **Column Count** indicates how many columns are in the drop down list. This property works together with the Column Widths property explained next. By default only the first column values are displayed on the combo box drop down list.
- **Column Widths** specify the width of each of the columns you indicated in the column count property. You must specify the width (in inches) for each column separated by commas. Here you can make any column "invisible," by indicating a width of 0".
- **List Width** represents the total width of the drop down list. Normally, the sum of all column widths. This width could be wider than the width of the combo box control in the form.



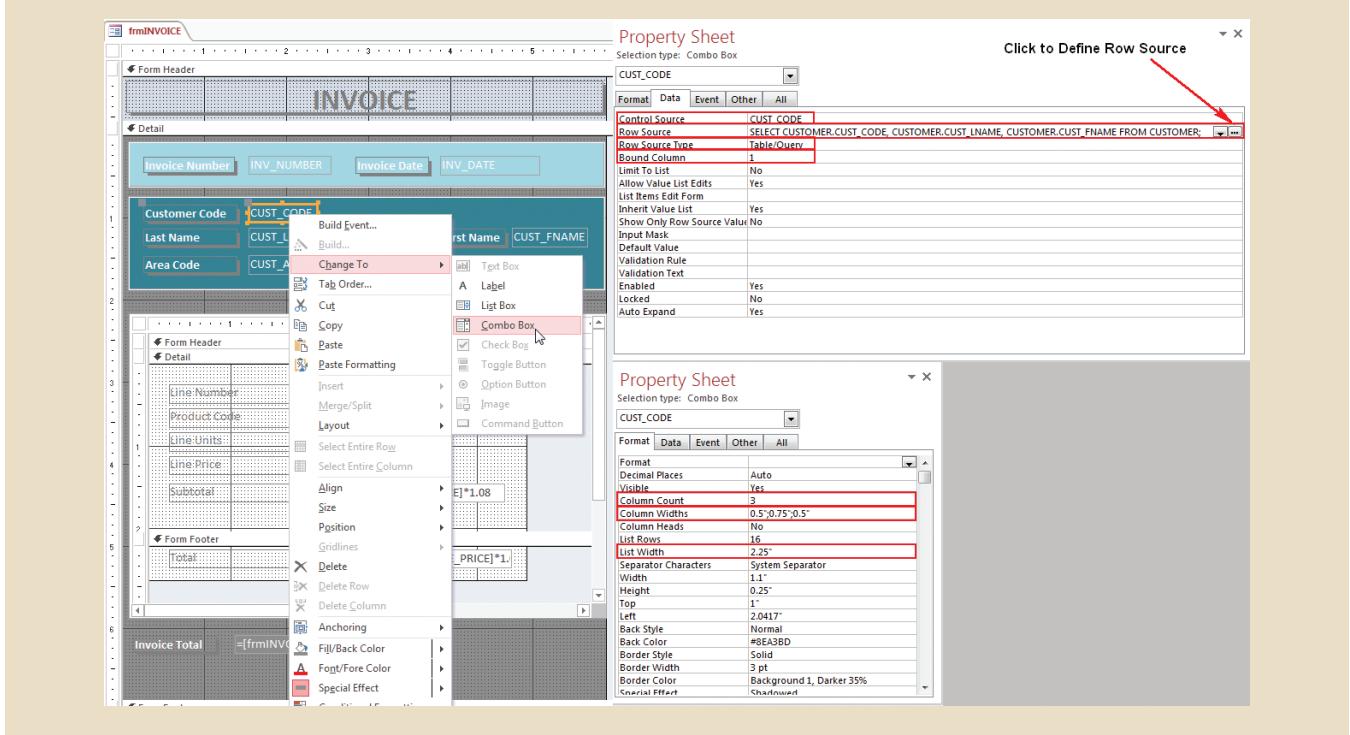
## Note

The combo box control will always show a value in the text box. This value is generally the first column of the row source query that is visible and has a width greater than 0. For example, when you click a combo box drop down arrow, the drop down list will show a list of rows, generally from a query or table. The total list width is determined by the List Width property and it could be wider than the combo box text box itself. Once the user selects one row from the drop down list, the first visible column (determined by the Column Widths property) is what is displayed on the combo box text box. However, remember, what is stored in the combo box control source is the value in the column indicated by the Bound Column property.

Microsoft Access makes it easy to convert a text box to a combo box. To illustrate how it is done, open the **frmINVOICE** in Design View (see Figure M.101) and follow the procedure below:

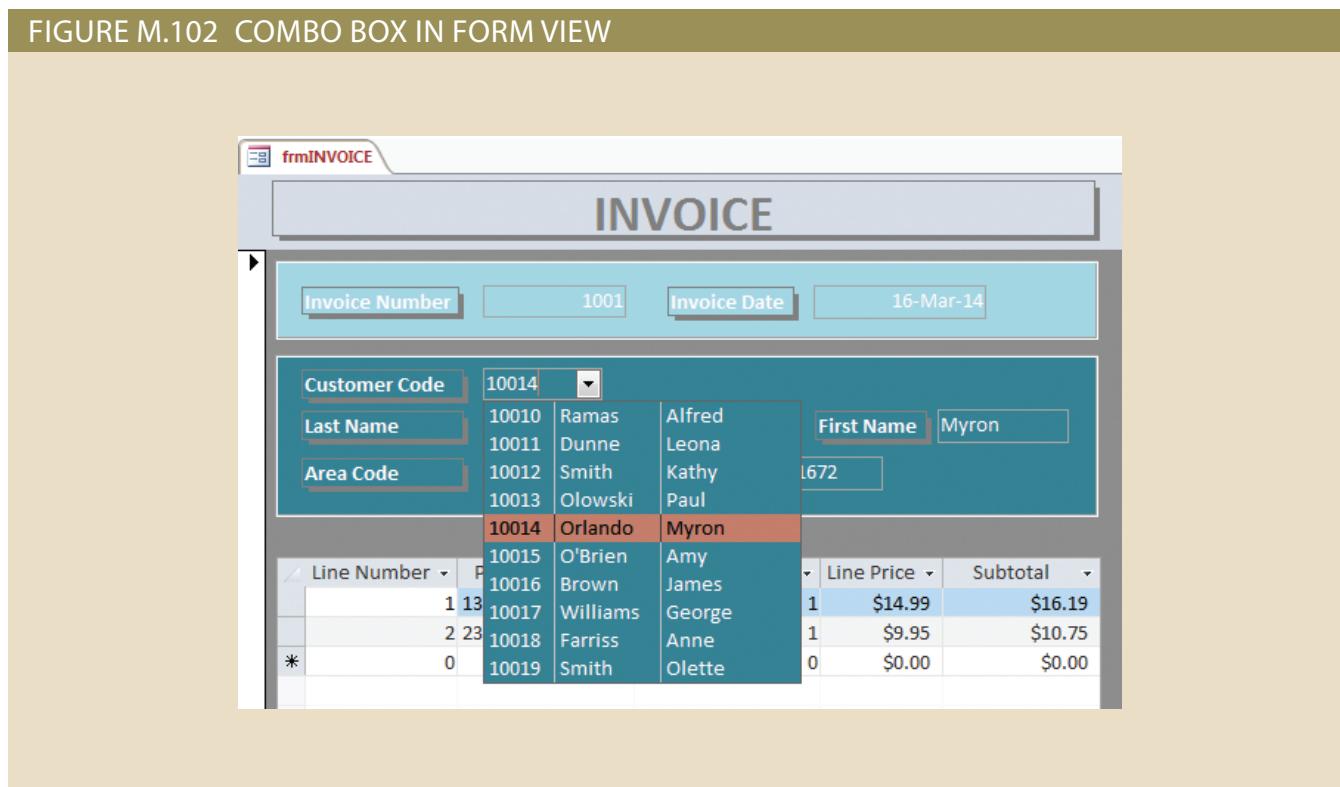
- Right-click the CUST\_CODE text box control and select **Change To... Combo Box**.
- Open the Property Sheet for the new Combo Box and select the **Data** tab.
- Click the **Row Source** property, then click the three dots to the right to generate a query design window.
- Add the CUSTOMER table to the QBE data source.
- Add CUST\_CODE, CUST\_LNAME, CUST, FNAME to the QBE data manipulation grid.
- Close the Query and click **Yes** to save changes.
- From the Form Design View select the Property Sheet for the CUST\_CODE Combo Box.
- Note that under the **Data** tab the **Bound Column** property is set to 1. The Bound Column indicates which field from the Row Source query/table will be stored in the Control Source field. Because the Control Source field is CUST\_CODE in the INVOICE table, we need to store only valid customer codes. The CUST\_CODE field in the Row Source query is the first field; therefore, the **Bound Column** is column 1.
- Select the **Format** tab and change the **Column Count** to 3. (This allows for all three fields to be displayed in the drop down list. By default, the column count is set to 1, which will only display the CUST\_CODE column.)
- Type **0.5;0.75;0.5** into the **Column Widths** property. This sets the width for fields 1, 2, and 3.
- Type **2.25** into the **List Width** Property. (This sets the width for the entire drop down list, generally it needs to be .5" larger than the sum of the 3 fields.)

FIGURE M.101 CREATING A COMBO BOX



Save the form. Then open the **frmINVOICE** in its Form View. If you now click the combo box drop-down arrow, you will see the available entries in Figure M.102. Note that you can select a customer based on their code, first, or last name.

FIGURE M.102 COMBO BOX IN FORM VIEW

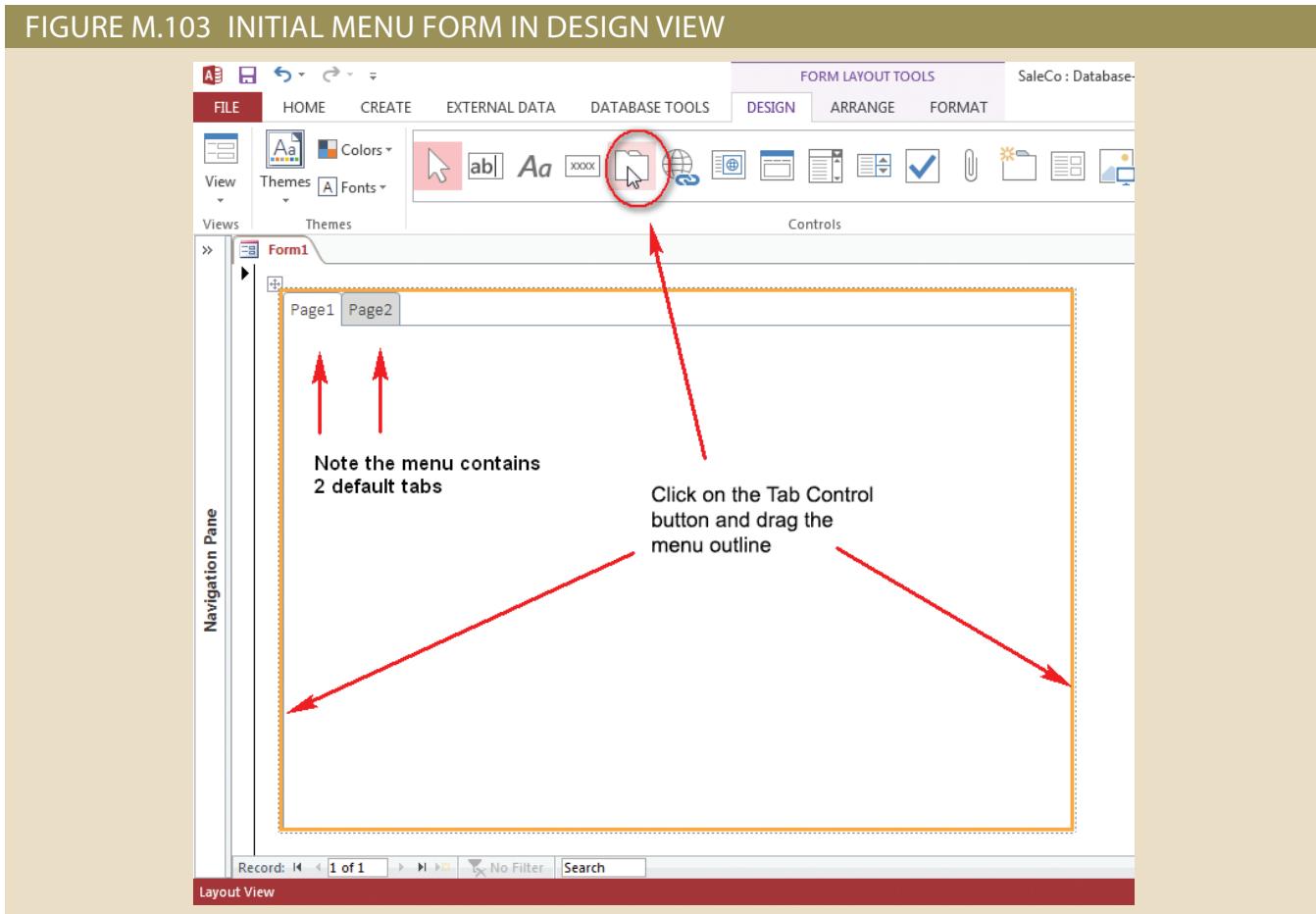


## M-7e Menus

If you want to make queries, forms, and reports easily available, a menu is a good way to get the job done. A menu is a form containing buttons to other objects (forms, queries or reports) and it is not bound to any table or query. Creating menus is easy—just click the **CREATE** tab, and then click the **Blank Form** button under the **Forms** group. You will see an empty form in **Layout View** in which to create the menu.

Next, under the **FORM LAYOUT TOOLS**, select **DESIGN**, and under **Controls**, select the **Tab Control** button shown in Figure M.103—note that the cursor changes shape when you do that—and draw the menu outline by dragging the cursor outline on the form. Figure M.103 shows the result of dragging the menu cursor to create a large Tab control on the menu form. Note also that the initial use of the Tab control automatically generates two tabbed menu pages, **Page1** and **Page2**. Right-click on an empty space on **Page1** and select **Insert Page**. This will generate **Page3**. Use the Property Sheet to rename **Page1** to **Queries**, **Page2** to **Forms**, and **Page3** to **Reports**.

FIGURE M.103 INITIAL MENU FORM IN DESIGN VIEW



You are now ready to place a few command buttons on the menu pages. Such buttons will be used to let the end user open a particular object—in this case, a query, a form, or a report. (You will learn about report generation in Section M-8.) Clicking such a button will generate a command that will execute the selected option.



### Note

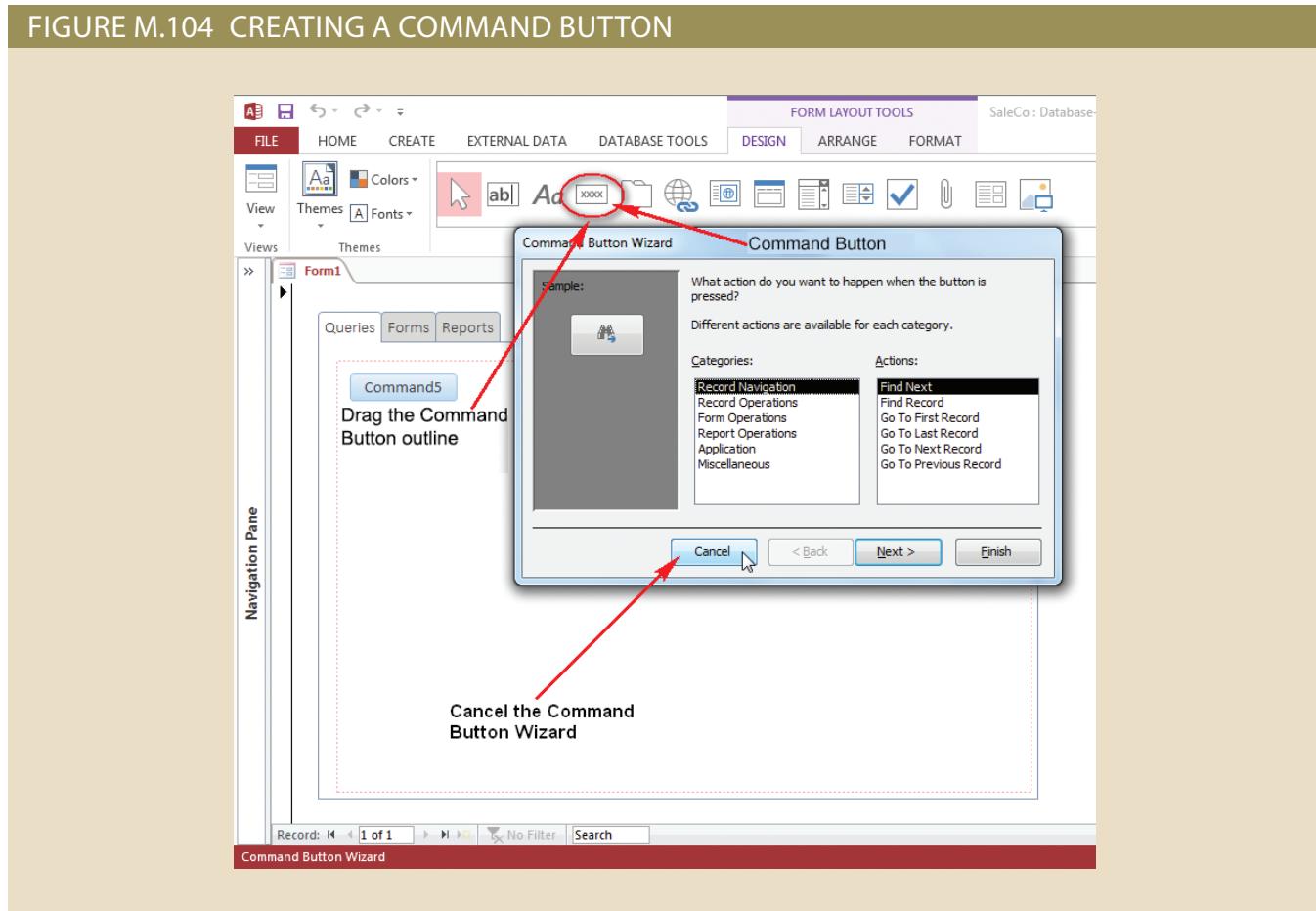
If you do not pay attention to the page you are supposed to be on, you may wind up placing command buttons on all pages simultaneously or you may place a queries command button on a forms or reports page. Therefore, if you intend to produce one or more command buttons on the Queries page, make sure that you select the tab for that page when you are in Design View or Layout View.

Figure M.104 shows how a command button is created on the Queries page. Make sure that the Queries page of the menu form has been selected. Then do the following:

- Click the **Command** button control on the Ribbon.
- Drag the Command button outline onto the page.
- The Command Button Wizard appears.
- Because here you are just creating some buttons, click **Cancel**.

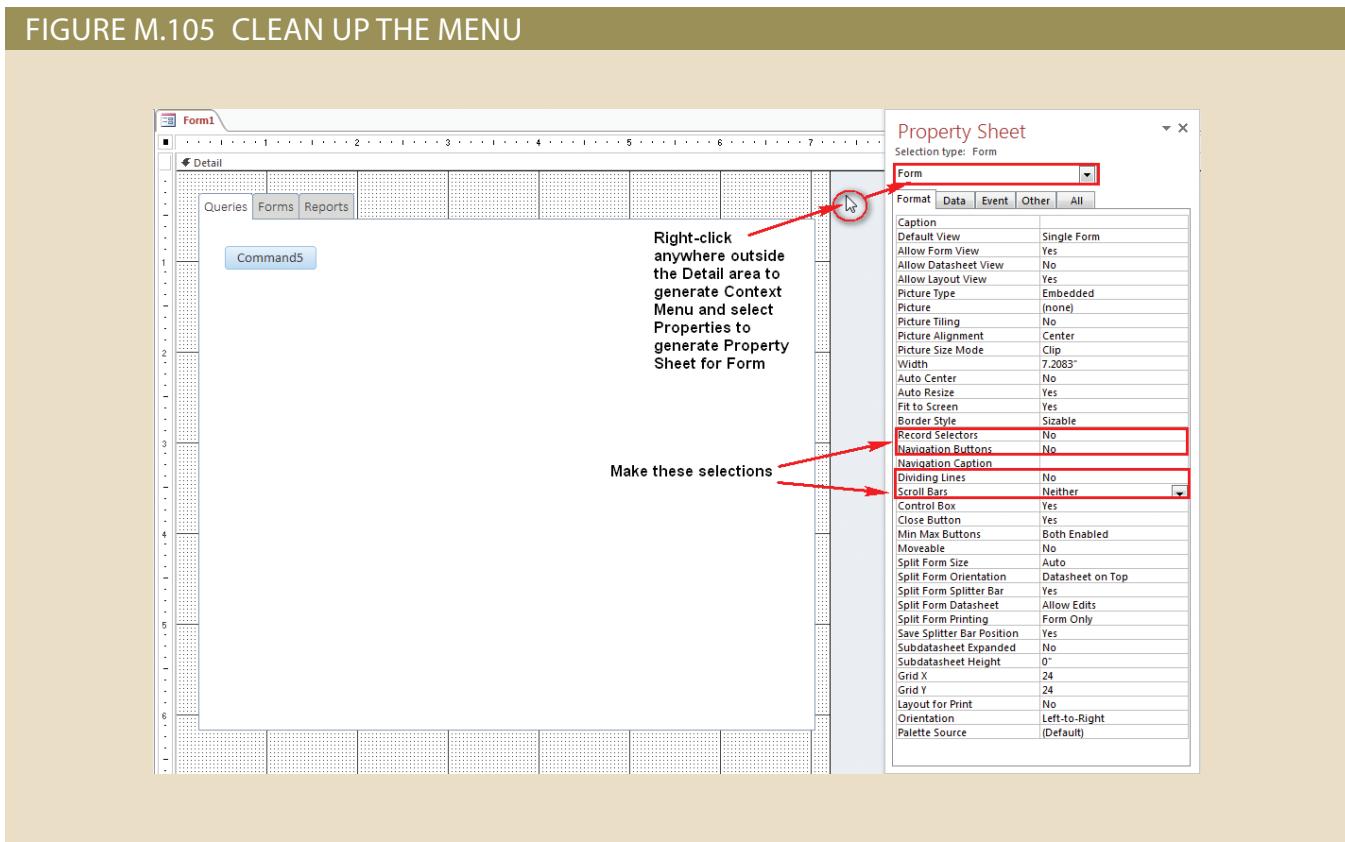
As you can see from Figure M.104, the Command Button Wizard is very easy and self-explanatory. For example, you can follow the prompts and create a button to open a form by clicking Form Operations, Open Form, and then select the form you want to open. It's that easy! To open a query, just click Miscellaneous, then click Run Query, and select the query you want to open. (You will learn how to create and use macros in Section M-9.)

FIGURE M.104 CREATING A COMMAND BUTTON



As you can see in Figure M.104, the menu has a few unwanted properties such as record selector border. Switch to **Design View** to “clean” the menu form by performing the actions indicated in Figure M.105. Save the menu form as **frmMainMenu**.

FIGURE M.105 CLEAN UP THE MENU

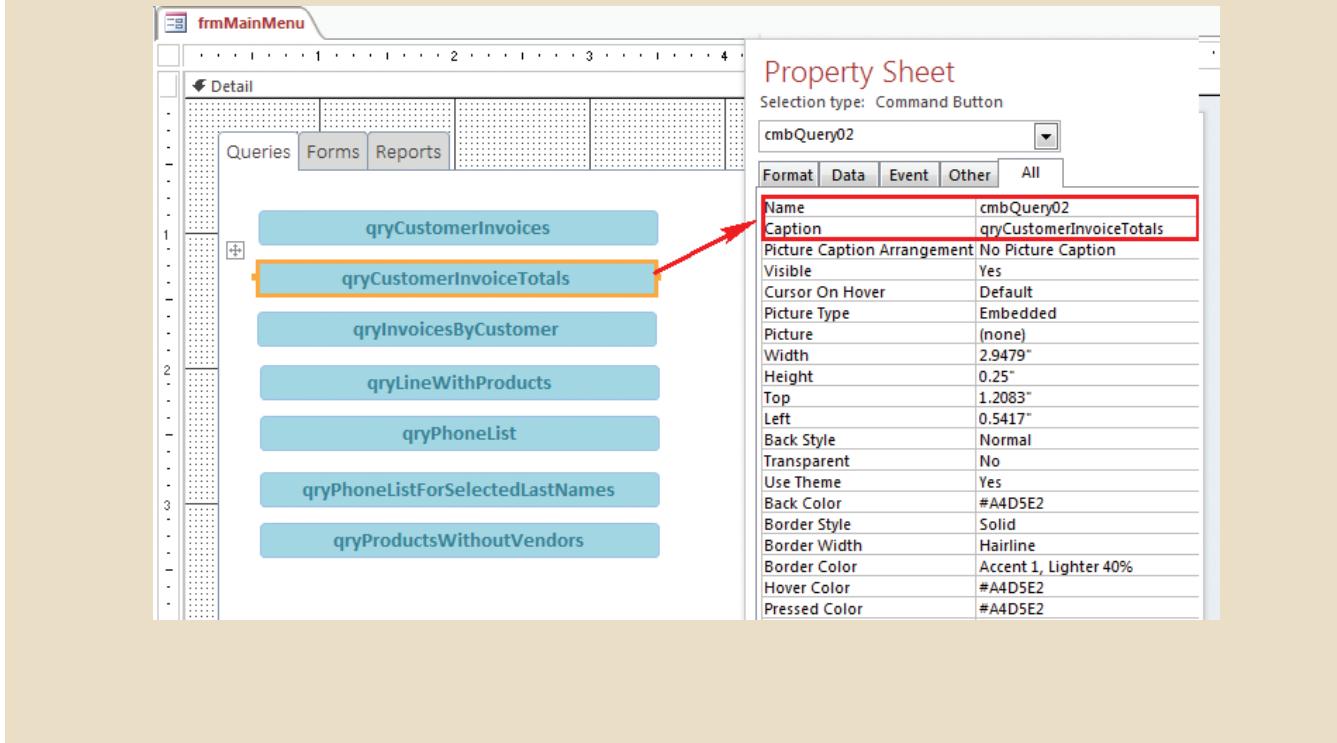


To edit the first command button's name and caption, stay in Design View, right-click the command button, and open the Property Sheet for that button. Change the name to **cmbQuery01** and the caption to **qryCustomerInvoices**.

Change the command button's text to boldface and change the size and location of the command button by dragging its limits. Note that this command button occurs on the Queries page only; the remaining two pages are still blank. (Go ahead and click from tab to tab to see the results.)

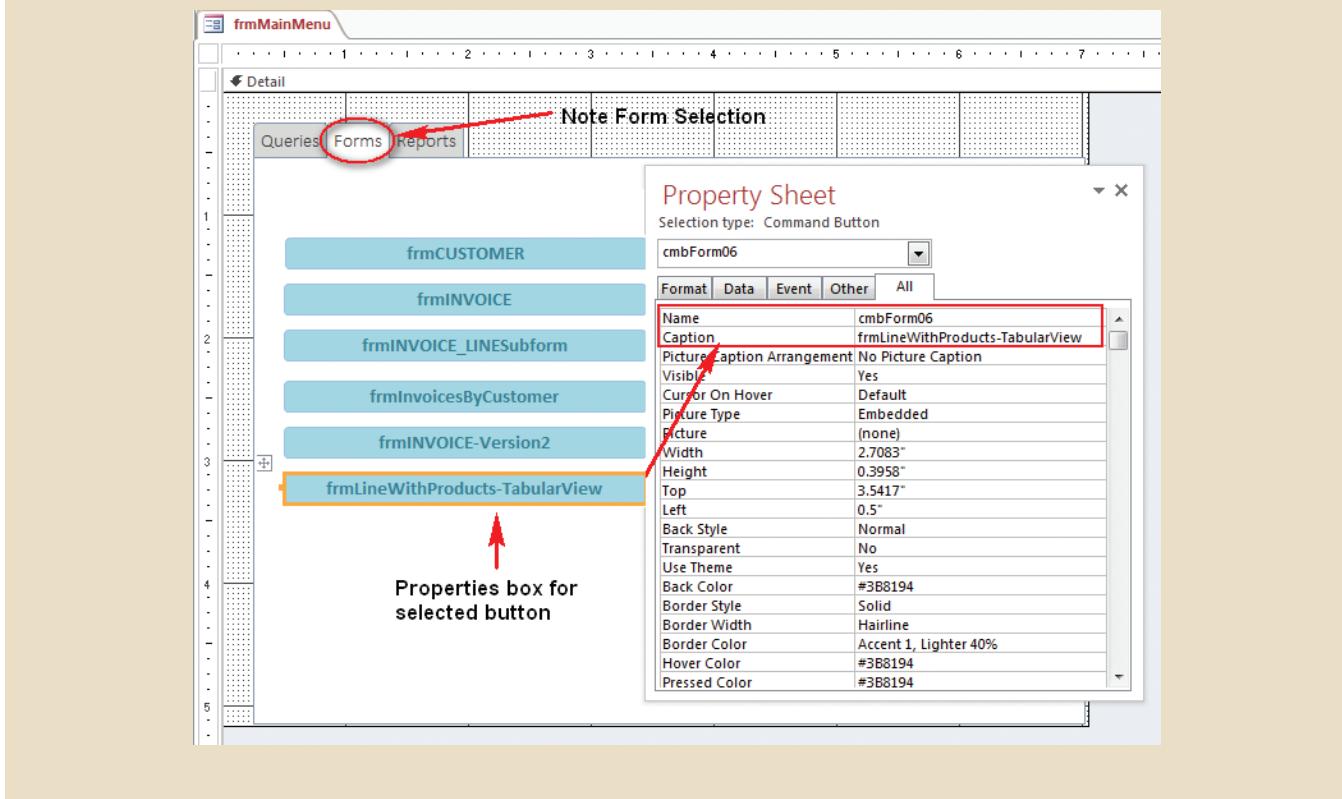
Let's create few more buttons for the **Queries** page. The easiest way to do this is to use a procedure you should know if you are familiar with Windows. That is, select the object in Design View and then use the copy/paste routine to make and place copies. Make sure that that the orange selection box is on the Queries page and not the **entire tab control** (or that the Property Sheet indicates the Queries page). This ensures that the buttons will only be copied to the Queries page. Drag the buttons to their intended positions. Note that the copy/paste routine ensures that all the buttons have the same properties. When you have made and placed the button copies, edit each to match the results in Figure M.106. Don't forget to save your efforts from time to time. You can go to the **FORMAT** tab on the Ribbon and use the existing options to customize the look of the buttons further. Continue to name the buttons as shown in Figure M.106.

FIGURE M.106 MULTIPLE QUERIES PAGE COMMAND BUTTONS



Next, copy the buttons you see here and paste them into the **Forms** page; then edit the buttons to match the forms that will be included. When you are done, your page in Design View should match Figure M.107.

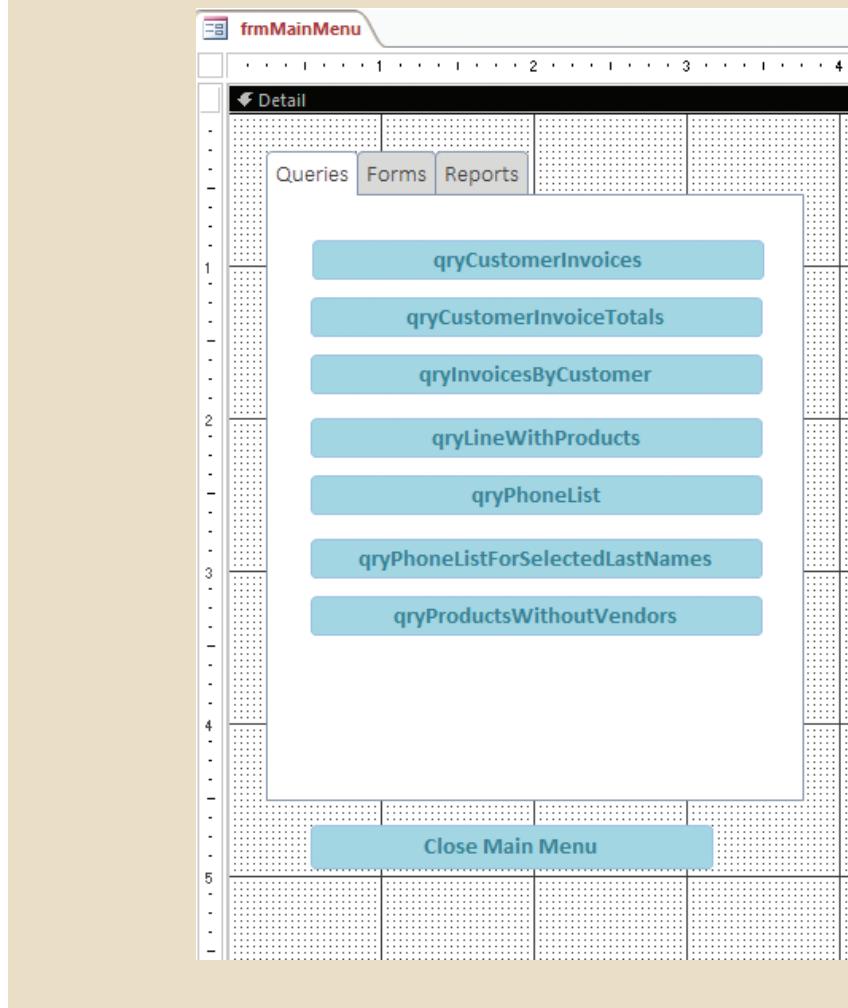
FIGURE M.107 FORM PAGE COMMAND BUTTONS



You have not yet created any reports, so leave the **Reports** page blank. (You will learn how to create reports in Section M-8.)

Next, let's create a **Close Main Menu** command button. Figure M.108 shows that a single button has been created outside of the tabs and therefore, this will make the button accessible from all of the tabs. In this case, when the **Command Button Wizard** appears, you would select **Form Operations**, **Close Form**. Notice also that the form has been resized. Save and close the form.

FIGURE M.108 CLOSE MAIN MENU BUTTON



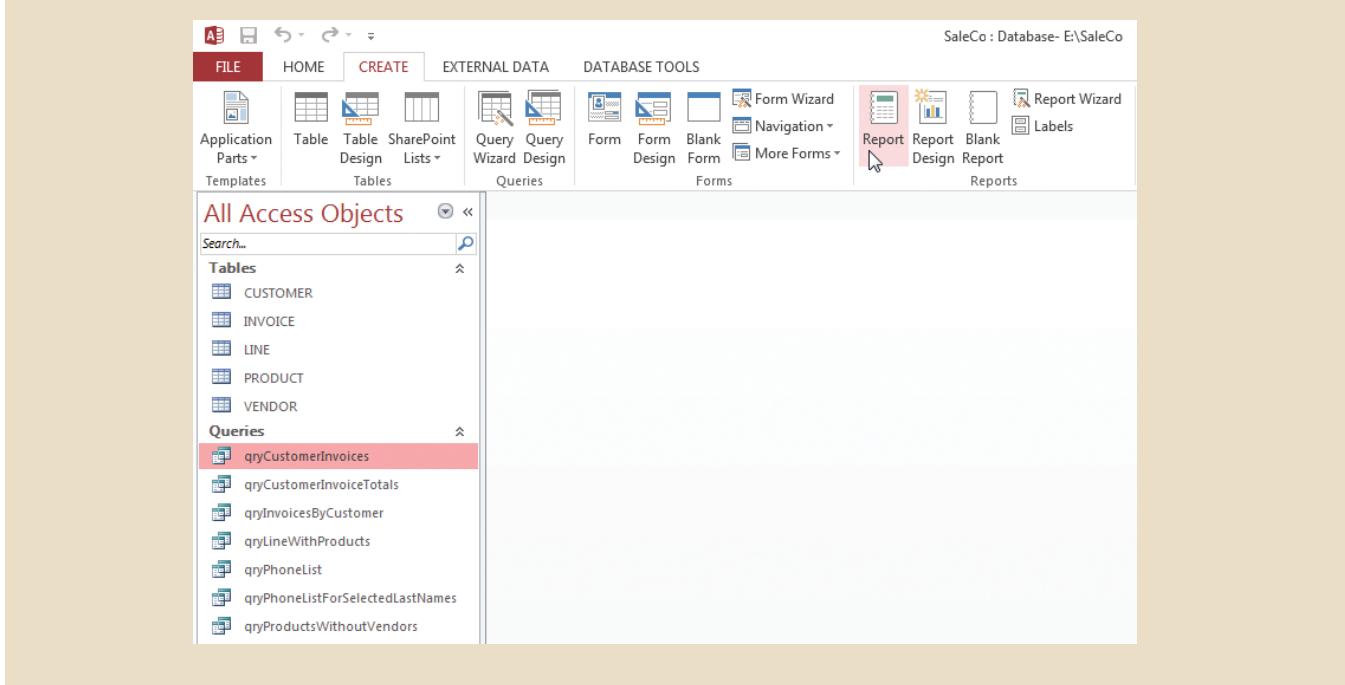
## M-8 Reports

Although reports often contain the same information as forms, they do have several advantages. First, it is much easier to show multiple-record information in reports than in forms. Second, given their layout, reports are much easier to print than forms. In addition, if you have a lot of numeric data to present, the Access report format enables you to produce subtotals and grand totals as the report is generated.

Reports can be based on tables and/or queries. The simplest and most efficient way to create a report is to follow the steps outlined below (also shown in Figure M.109). The steps to quickly create a form are the following:

- Select the data source for the report in this case, the query named **qryCustomerInvoices**.
- Click **CREATE** on the Ribbon, then click the **Report** button.

FIGURE M.109 DESIGNING A NEW REPORT



As soon as you click the **Report** button shown in Figure M.109, Access will automatically generate the report and show it in Layout View. You can switch to Report View by clicking **VIEW** and selecting **Report View** on the Ribbon. See Figure M.110.

FIGURE M.110 NEW REPORT OUTPUT

CUST_CODE	CUST_LNAME	CUST_FNAME	CUST_INITIAL	INV_NUMBER	INV_DATE	INV_TOTAL
10011	Dunne	Leona	K	1002	16-Mar-18	\$10.78
10011	Dunne	Leona	K	1004	17-Mar-18	\$37.66
10011	Dunne	Leona	K	1008	17-Mar-18	\$431.08
10012	Smith	Kathy	W	1003	16-Mar-18	\$166.16
10014	Orlando	Myron		1001	16-Mar-18	\$26.94
10014	Orlando	Myron		1006	17-Mar-18	\$429.66
10015	O'Brien	Amy	B	1007	17-Mar-18	\$37.77
10018	Farriss	Anne	G	1005	17-Mar-18	\$76.08

8

Page 1 of 1

Although the report output shown in Figure M.110 is already quite usable, you could add functionality and better formatting using the now-familiar design tools. However, before you start editing the report, save the report as **rptCustomerInvoices**.

To make this report more useful, we will modify the report to include invoice subtotals per customer. In this case, to do subtotals by customer, we need to group the report by customer code and then show invoice subtotals *for each customer*. To achieve this goal follow the steps described next.

- Switch the report to **Layout View**.
- The REPORT LAYOUT TOOLS menu appears on the Ribbon.
- Select and delete the **Record Count** text box under the CUST\_CODE column.
- Click the first **CUST\_CODE** value on the report area. The entire column of customer code values is highlighted with an orange border.
- Right-click the column value and select **Group On CUST\_CODE** from the context menu. See Figure M.111. This option will create a grouping by CUST\_CODE and will add a header area at the beginning of each group.
- Move the **CUST\_LNAME**, **CUST\_INITIAL**, and **CUST\_FNAME** fields from the detail area to the CUST\_CODE group header area.

Next, we need to add the subtotals for the invoices:

- Click the **first INV\_TOTAL value** on the report area. The entire column of invoice total values is highlighted with an orange border.
- Right-click the column value, select **Total INV\_TOTAL** from the context menu and then **SUM**. This option automatically adds a footer area with subtotal line after the last invoice line for each customer.

Finally, we are going to add a sort order by INV\_NUMBER (within each customer grouping):

- Click the first INV\_NUMBER value on the report area. The entire column of invoice number values is highlighted with an orange border.
- Right-click the column value, and select the option **Sort Smallest to Largest** from the context menu.

FIGURE M.111 ADDITION OF A GROUP HEADER

The screenshot shows a Microsoft Access report titled "qryCustomerInvoices". The report has a header section with the title and a timestamp. Below the header is a table with the following columns: CUST\_CODE, CUST\_LNAME, CUST\_FNAME, CUST\_INITIAL, INV\_NUMBER, INV\_DATE, and INV\_TOTAL. The data is grouped by CUST\_CODE. A context menu is open over the first row of data, specifically over the value "10011" in the CUST\_CODE column. The menu path "Group On CUST\_CODE" is highlighted. Other options in the menu include Cut, Copy, Paste, Paste Formatting, Insert, Merge/Split, Layout, Select Entire Row, Select Entire Column, Total CUST\_CODE, Sort A to Z, Sort Z to A, Clear filter from CUST\_CODE, Text Filters, Equals "10011", Does Not Equal "10011", Contains "10011", Does Not Contain "10011", Delete, Delete Row, Delete Column, Change To, Position, Gridlines, Anchoring, Report Properties, and Properties. At the bottom of the report area, there are buttons for "Add a group" and "Add a sort". The page number "Page 1 of 1" is visible at the bottom right.

Now that you have added grouping and totals to the report, the updated report now has CUST\_CODE header and footer areas. To complete the report, go to **Design View** and do the following:

- Add a **Customer Total**: label to the subtotal in the CUST\_CODE footer and a **Grand Total**: label to the Report Footer area. See Figure M.112.
- Move the page number components from the Page Footer area to the Report Header area. See Figure M.112.

Save the report.

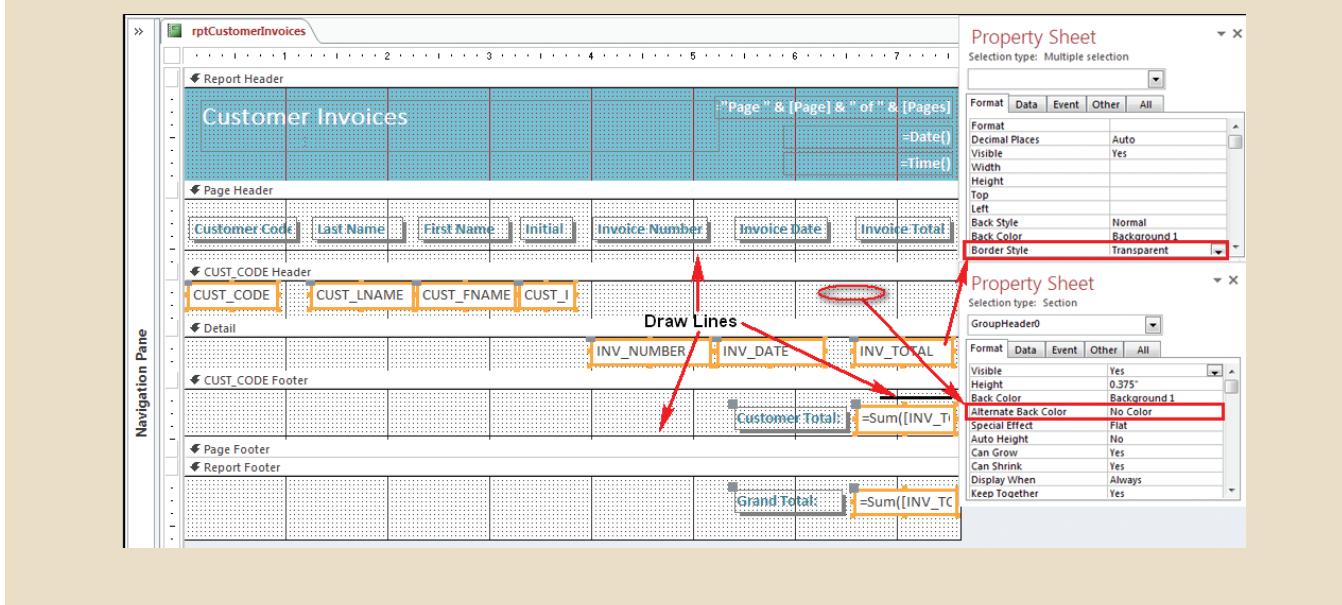
FIGURE M.112 SUBTOTALS AND TOTALS

Customer Code	Last Name	First Name	Initial	Invoice Number	Invoice Date	Invoice Total
10011			K	1002	16-Mar-18	\$10.78
				1004	17-Mar-18	\$37.66
				1008	17-Mar-18	\$431.08
					<b>Customer Total:</b>	<b>\$479.52</b>
10012			W	1003	16-Mar-18	\$166.16
					<b>Customer Total:</b>	<b>\$166.16</b>
10014				1001	16-Mar-18	\$26.94
				1006	17-Mar-18	\$429.66
					<b>Customer Total:</b>	<b>\$456.59</b>
10015				1007	17-Mar-18	\$37.77
					<b>Customer Total:</b>	<b>\$37.77</b>
10018			G	1005	17-Mar-18	\$76.08
					<b>Customer Total:</b>	<b>\$76.08</b>
					<b>Grand Total:</b>	<b>\$1,216.11</b>

As you examine Figure M.112, note the creation of the (computed) customer total text box and especially note its computation through the SUM function. The **same function is used to compute the grand total**. Aside from the fact that the two new text boxes have different labels, the important difference between the two text boxes (customer total and grand total) is their **location**. The SUM function in the text box located in the Report Footer area will add up all the invoice totals for all customers. The SUM function in the text box located at the CUST\_CODE Footer area will add up all the invoice totals but only for each individual group of customers. That is, it will add up all the invoices for customer 10011, then 10012, etc.

Next, use Figure M.113 as a guide to reduce spacing in the Page Footer area to change the labels and formatting. Note that the customer code header, detail, and customer code footer have different colored backgrounds. You can select these portions in Design View and change the **Alternate Back Color** property to **No Color**. Note that the text boxes also have a border around them. Select the text boxes and change the **Border Style** to **Transparent**. Rename the fields and change the font colors to match Figure M.113.

FIGURE M.113 ADDITIONAL EDITS



The line tool on the **DESIGN** tab can be used to create lines to visually separate certain parts of the report as shown in Figure M.113. Change the thickness of the line in the CUST\_CODE footer by using the following path: **FORMAT** tab on the Ribbon and then **Shape Outline, Line Thickness, 2 pt**. Figure M.114 shows the report output in Report View.

FIGURE M.114 COMPLETED REPORT OUTPUT

**rptCustomerInvoices**

**Customer Invoices**

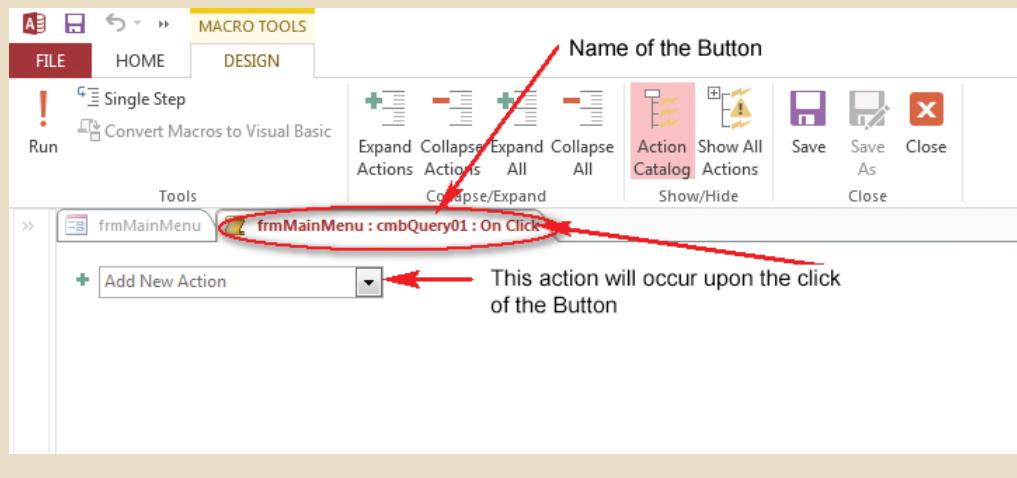
Page 1 of 1  
Friday, October 13, 2017  
11:33:07 AM

Customer Code	Last Name	First Name	Initial	Invoice Number	Invoice Date	Invoice Total
10011	Dunne	Leona	K			
				1002	16-Mar-18	\$10.78
				1004	17-Mar-18	\$37.66
				1008	17-Mar-18	\$431.08
				<b>Customer Total:</b>		<b>\$479.52</b>
10012	Smith	Kathy	W			
				1003	16-Mar-18	\$166.16
				<b>Customer Total:</b>		<b>\$166.16</b>
10014	Orlando	Myron				
				1001	16-Mar-18	\$26.94
				1006	17-Mar-18	\$429.66
				<b>Customer Total:</b>		<b>\$456.59</b>
10015	O'Brien	Amy	B			
				1007	17-Mar-18	\$37.77
				<b>Customer Total:</b>		<b>\$37.77</b>
10018	Farris	Anne	G			
				1005	17-Mar-18	\$76.08
				<b>Customer Total:</b>		<b>\$76.08</b>
				<b>Grand Total:</b>		<b>\$1,216.11</b>

## M-9 Macros

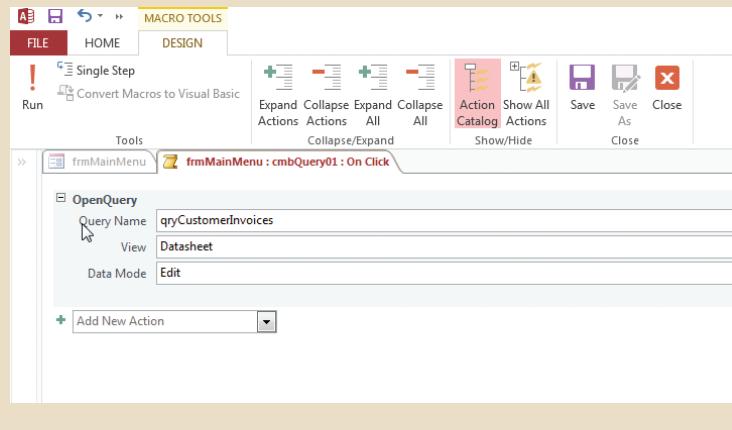
Macros are code sets that let you perform a wide variety of actions that range from opening and closing forms, queries, and reports to calculating and inserting values on a form. Open **frmMainMenu** in Design View. Right-click the **qryCustomerInvoices** button, select **Build Event**, and then select **Macro Builder** and click **OK**. This will take you to the screen shown in Figure M.115.

FIGURE M.115 DESIGN VIEW OF A MACRO



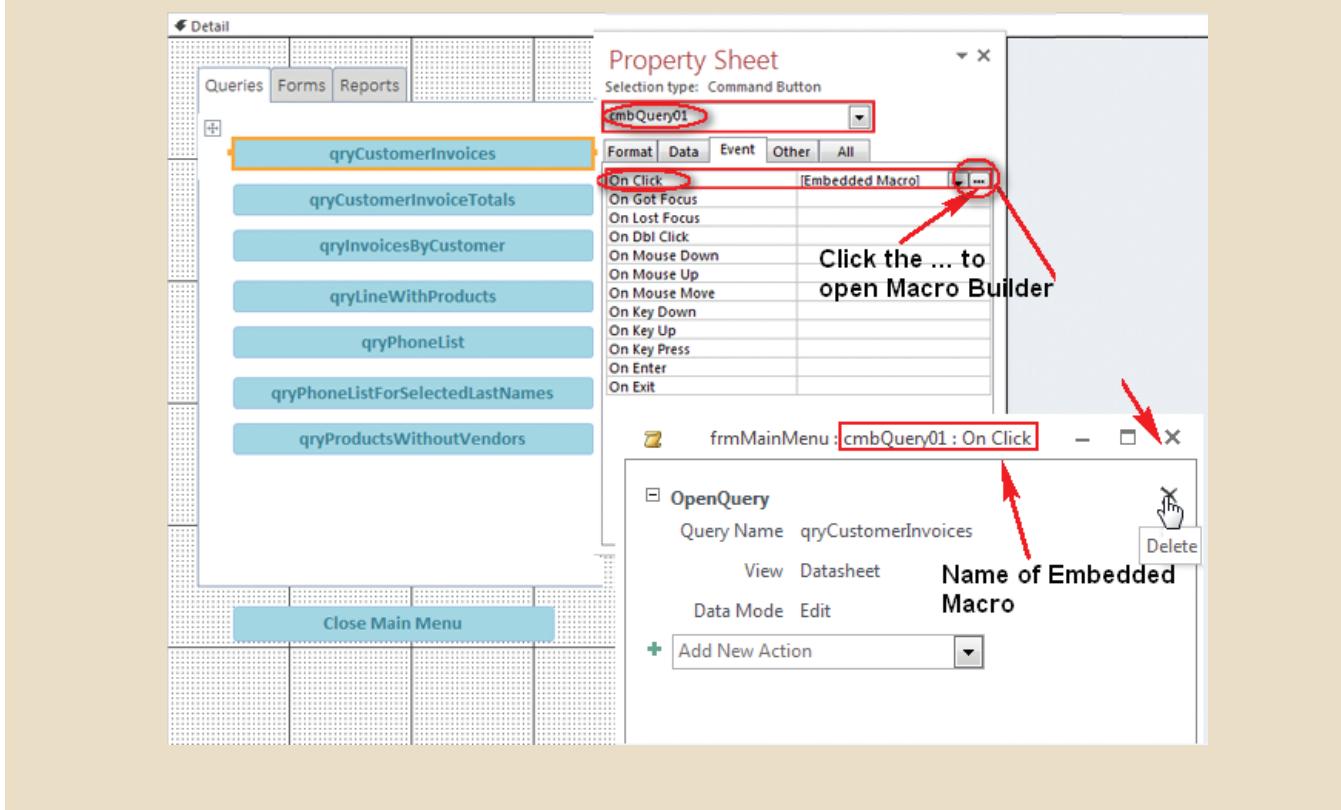
Select the **Add New Action** drop-down and select **OpenQuery**. Type in the query name that you wish for the button to open, or select it from the drop-down list. For this button it is **qryCustomerInvoices**. Leave the defaults that are set for View (Datasheet) and Data Mode (Edit) as they are. This is shown in Figure M.116. Exit the macro and click **Yes** to save changes. You do not have to give the macro a name because it is an embedded macro that is associated with the cmdQuery01 button.

FIGURE M.116 ADDING AN ACTION TO THE MACRO



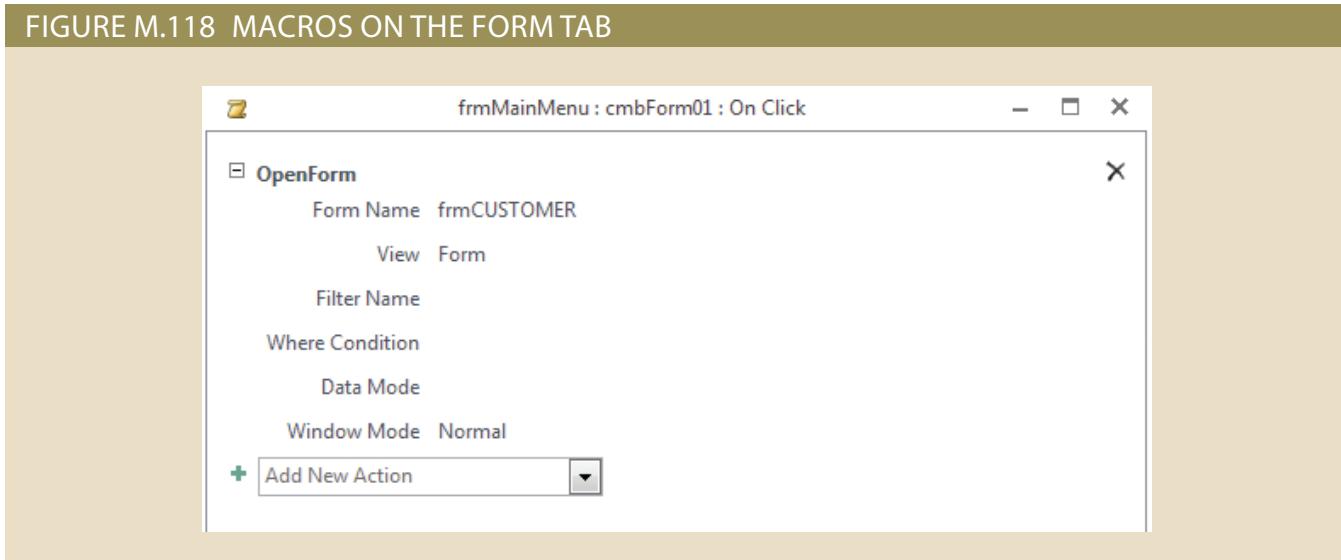
If you wish to edit the macro, open the Property Sheet for the selected button, click the **Event** tab, and then click the ellipses (...) button for the **On Click** property. This will take you to the Macro Builder. To delete an action from the Macro Builder, hover your mouse over the far right corner of Macro Builder screen near the action you want to delete and select the **X** that shows up in the corner. This will remove the current action. Figure M.117 demonstrates how to delete actions from a macro. You can practice deleting the macro but make sure not to save upon closing (we don't want to actually delete the macro).

FIGURE M.117 EDITING THE MACRO



Switch to the **Forms** tab so that you can build the macros for each of the buttons on the form. Select **Add New Action** then **OpenForm**. Select the appropriate form name for each button and leave all the other default settings as they are. This is shown in Figure M.118.

FIGURE M.118 MACROS ON THE FORM TAB



Create the rest of the macros for all of the buttons on the Form tab. Go ahead and create one last button on the Report tab with an associated macro to open the report we created in Section M-8.

Return to the Main Menu form in Form View and test all of your buttons.

## M-9 Conclusion

Only a few examples are shown in this tutorial. The objective is not to develop full-blown applications, but to show you some examples of what can be done in the Microsoft Access environment. Once you have seen those examples, you have a foundation on which to build greater expertise. Keep in mind that Access is a superb prototyping tool, but it is not capable of serving the full database and information needs of even medium-sized organizations, let alone large ones. Products such as Microsoft's SQL Server, IBM's DB2, or Oracle are better candidates for such environments. Nevertheless, given its ability to let you develop superb prototypes, Access has earned a place of honor in the ranks of database professionals.