SESSION 2.2 VISUAL OVERVIEW

Click the Close button to close the Relationships window.

You click the Show Table button to open the Show Table dialog box. From there, you can choose a table to add to the Relationships window.

The Relationships window illustrates the relationships among a database's tables. Using this window, you can view or change existing relationships, define new relationships between tables, and rearrange the layout of the tables in the window.

The key symbol next to a field name indicates that the field is the table's primary key. For example, CustomerID is the primary key for the Customer table.

A one-to-many relationship exists between two tables when one record in the first table matches zero, one, or many records in the second table, and when one record in the second table matches at most one record in the first table. Here, the Customer and Contract tables have a one-to-many relationship because a customer can have many contracts, and each contract has only one customer. The two tables are still separate tables, but because they are joined, you can use the data in them as if they were one table.
TABLE RELATIONSHIPS

The "one" side of a one-to-many relationship is represented by the digit 1 at the end of the join line.

The "many" side of a one-to-many relationship is represented by the infinity symbol at the end of the join line.

When you add a table to the Relationships window, the fields in the table appear in a field list. Here, the window contains three field lists, one for each table: Customer, Contract, and Invoice.

The primary table is the "one" table in a one-to-many relationship. Here, Contract is the primary table because there is at most one contract for each invoice. The join line connects the common field used to create the relationship between two tables. Here, the common field ContractNum is used to create the one-to-many relationship between the Contract and Invoice tables. The related table is the "many" table in a one-to-many relationship. Here, Invoice is the related table because there can be more than one invoice set up for each contract.
Before you can begin to define the table relationships, as illustrated in the Session 2.2 Visual Overview, you need to finish creating the tables in the Belmont database.

**Adding Records to a New Table**

The Invoice table design is complete. Now, Oren would like you to add records to the table so it will contain the invoice data for Belmont Landscapes. You add records to a table in Datasheet view as you did in Tutorial 1, by typing the field values in the rows below the column headings for the fields. You’ll begin by entering the records shown in Figure 2-20.

<table>
<thead>
<tr>
<th>InvoiceNum</th>
<th>ContractNum</th>
<th>InvoiceDate</th>
<th>InvoiceItem</th>
<th>InvoiceAmt</th>
<th>InvoicePaid</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>3011</td>
<td>03/23/2013</td>
<td>Schematic Plan</td>
<td>$1,500.00</td>
<td>Yes</td>
</tr>
<tr>
<td>2031</td>
<td>3020</td>
<td>04/19/2013</td>
<td>Schematic Plan</td>
<td>$1,500.00</td>
<td>Yes</td>
</tr>
<tr>
<td>2073</td>
<td>3023</td>
<td>09/21/2015</td>
<td>Construction Observation</td>
<td>$10,000.00</td>
<td>No</td>
</tr>
<tr>
<td>2062</td>
<td>3026</td>
<td>09/12/2014</td>
<td>Permitting</td>
<td>$10,000.00</td>
<td>No</td>
</tr>
</tbody>
</table>

**To add the first record to the Invoice table:**

1. If you took a break after the previous session, make sure that the Belmont database is open, and the Navigation Pane is open.
2. In the Tables section of the Navigation Pane, double-click Invoice to open the Invoice table in Datasheet view.
3. Close the Navigation Pane, and then use the + pointer to resize each column so that the field names are completely visible.
4. In the Invoice Num column, type 2011, press the Tab key, type 3011 in the Contract Num column, and then press the Tab key.
   Next you need to enter the invoice date. Recall that you specified a custom date format, mm/dd/yyyy, for the InvoiceDate field. You do not need to type each digit; for example, you can type just "3" instead of "03" for the month, and you can type "13" instead of "2013" for the year. Access will display the full value according to the custom date format.
5. Type 03/23/13, press the Tab key, type Schematic Plan in the Invoice Item column, and then press the Tab key. Notice that Access displays the date "03/23/2013" in the Invoice Date column.
   Next you need to enter the invoice amount for the first record. This is a Currency field with the Currency format and two decimal places specified. Because of the field's set properties, you do not need to type the dollar sign, comma, or zeroes for the decimal places; Access will display these items automatically for you.
6. Type 1500 and then press the Tab key. Access displays the value as "$1,500.00."
   The last field in the table, InvoicePaid, is a Yes/No field. Notice the check box displayed in the column. By default, the value for any Yes/No field is "No;" therefore, the check box is initially empty. For Yes/No fields with check boxes, you press the Tab key to leave the check box unchecked, and you press the spacebar to insert a check mark in the check box. For the record you are entering in the Invoice table, the invoice has been paid, so you need to insert a check mark in the check box.
7. Press the spacebar to insert a check mark, and then press the Tab key. The values for the first record are entered. See Figure 2-21.
Now you can add the remaining three records. As you do, you’ll learn a shortcut for inserting the value from the same field in the previous record.

To add the next three records to the Invoice table:

1. Refer to Figure 2-20 and enter the values in the second record’s Invoice Num, Contract Num, and Invoice Date columns.

Notice the value in the second record’s Invoice Item column, “Schematic Plan.” This value is the exact same value as in the first record. You can quickly insert the value from the same column in the previous record using the Ctrl + ‘ (apostrophe) keyboard shortcut.

2. In the Invoice Item column, press the Ctrl + ‘ keys. Access inserts the value “Schematic Plan” in the Invoice Item column for the second record.

3. Press the Tab key to move to the Invoice Amt column. Again, the value you need to enter in this column—$1,500.00—is the same as the value for this column in the previous record. So, you can use the keyboard shortcut again.

4. In the Invoice Amt column, press the Ctrl + ‘ keys. Access inserts the value $1,500.00 in the Invoice Amt column for the second record.

5. Press the Tab key to move to the Invoice Paid column, press the spacebar to insert a check mark in the check box, and then press the Tab key. The second record is entered in the Invoice table.

6. Refer to Figure 2-20 to enter the values for the third and fourth records, using the Ctrl + ‘ keys to enter the value in the fourth record’s Invoice Amt column. Also, for both records, the invoices have not been paid. Therefore, be sure to press the Tab key to leave the Invoice Paid column values unchecked (signifying “No”).

7. Resize the columns, as necessary, so that all field values are completely visible. Then click the Invoice Num column for the next new record. Your table should look like the one in Figure 2-22.
To finish entering records in the Invoice table, you’ll use a method that allows you to import the data.

**Importing Data from an Excel Worksheet**

Often, the data you want to add to an Access table exists in another file, such as a Word document or an Excel workbook. You can bring the data from other files into Access in different ways. For example, you can copy and paste the data from an open file, or you can import the data, which is a process that allows you to copy the data from a source without having to open the source file.

Oren had been using Excel to track invoice data for Belmont Landscapes and already created a worksheet, named “Invoices,” containing this data. You’ll import this Excel worksheet into your Invoice table to complete the entry of data in the table. To use the import method, the columns in the Excel worksheet must match the names and data types of the fields in the Access table.

**Caption Property Values and the Import Process**

When you want to import data from an Excel worksheet into an Access table, any Caption property values set for the fields in the table are not considered in the import process. For example, your Access table could have fields such as InvoiceDate and InvoiceAmt with Caption property values of Invoice Date and Invoice Amt, respectively. If the Excel worksheet you are importing has column headings such as Invoice Date and Invoice Amt, you might think that the data matches and you can proceed with the import. However, if the underlying field names in the Access table do not match the Excel worksheet column headings exactly, the import process will fail. It’s a good idea to double-check to make sure that the actual Access field names—and not just the column headings displayed in a table datasheet (as specified by the Caption property)—match the Excel worksheet column headings. If there are differences, you can change the column headings in the Excel worksheet to match the Access table field names before you import the data, ensuring that the process will work correctly.

The Invoices worksheet contains the following columns: InvoiceNum, ContractNum, InvoiceDate, InvoiceItem, InvoiceAmt, and InvoicePaid. These column headings match the field names in the Invoice table exactly, so you can import the data. Before you import data into a table, you need to close the table.
To import the Invoices worksheet into the Invoice table:

1. Click the Close ‘Invoice’ button on the object tab to close the Invoice table. A dialog box opens asking if you want to save the changes to the table layout. This dialog box opens because you resized the table columns.
2. Click the Yes button in the dialog box.
3. Click the External Data tab on the Ribbon.
4. In the Import & Link group on the External Data tab, click the Excel button. The Get External Data - Excel Spreadsheet dialog box opens. See Figure 2-23.

The dialog box provides options for importing the entire worksheet as a new table in the current database, adding the data from the worksheet to an existing table, or linking the data in the worksheet to the table. You need to add, or append, the worksheet data to the Invoice table.

5. Click the Browse button. The File Open dialog box opens. The Excel workbook file is named “Invoices” and is located in the Access1\Tutorial folder provided with your Data Files.
6. Navigate to the Access1\Tutorial folder, where your starting Data Files are stored, and then double-click the Invoices Excel file. You return to the dialog box.
7. Click the Append a copy of the records to the table option button. The box to the right of this option becomes active. Next, you need to select the table to which you want to add the data.
8. Click the arrow on the box, and then click Invoice.
9. Click the OK button. The first Import Spreadsheet Wizard dialog box opens. See Figure 2-24.
The dialog box confirms that the first row of the worksheet you are importing contains column headings. The bottom section of the dialog box displays some of the data contained in the worksheet.

10. Click the **Next** button. The second, and final, Import Spreadsheet Wizard dialog box opens. Notice that the Import to Table box shows that the data from the spreadsheet will be imported into the Invoice table.

11. Click the **Finish** button. A dialog box opens asking if you want to save the import steps. If you needed to repeat this same import procedure many times, it would be a good idea to save the steps for the procedure. However, you don't need to save these steps because you will be importing the data only one time. Once the data is in the Invoice table, Oren will no longer use Excel to track invoice data.

12. Click the **Close** button in the dialog box to close it without saving the steps.

The data from the Invoices worksheet has been added to the Invoice table. Next, you'll open the table to view the new records.

**To open the Invoice table and view the imported data:**

1. Open the Navigation Pane, and then double-click **Invoice** in the Tables section to open the table in Datasheet view.

2. Resize the Invoice Item column so that all field values are fully displayed, being sure to scroll down through the entire datasheet and repeat the resizing as necessary. When you resize a column by double-clicking the pointer on the column dividing line, you are sizing the column to its **best fit**—that is, so the column is just wide enough to display the longest visible value in the column, including the field name.
3. Press the **Ctrl+Home** keys to scroll to the top of the datasheet. Notice that the table now contains a total of 176 records—four records you entered plus 172 records imported from the Invoices worksheet. The records are displayed in primary key order by the values in the Invoice Num column. See Figure 2-25.

4. Save and close the Invoice table, and then close the Navigation Pane.

Two of the tables—Contract and Invoice—are now complete. According to Oren's plan for the Belmont database, you need to create a third table, named "Customer," to track data about Belmont Landscapes' residential and commercial customers. You'll use a different method to create this table.

### Creating a Table by Importing an Existing Table Structure

If another Access database contains a table—or even just the design, or structure, of a table—that you want to include in your database, you can easily import the table and any records it contains or import only the table structure into your database.

Oren documented the design for the new Customer table by listing each field's name; data type; and size, description, and caption (if applicable), as shown in Figure 2-26. Note that each field in the Customer table will be a Text field, and the CustomerID field will be the table's primary key.
Sarah already created an Access database containing a Customer table design. She never entered any records into the table because she wasn't sure if the table design was complete or correct. After reviewing the table design, both Sarah and Oren agree that it contains some of the fields Oren wants to track, but that some changes are needed. So, you can import the table structure in Sarah's database to create the Customer table in the Belmont database, and then modify the imported table to produce the final table structure Oren wants.

To create the Customer table by importing the structure of another table:

1. Make sure the External Data tab is the active tab on the Ribbon.
2. In the Import & Link group, click the Access button. The Get External Data - Access Database dialog box opens. This dialog box is similar to the one you used earlier when importing the Excel spreadsheet.
3. Click the Browse button. The File Open dialog box opens. The Access database file from which you need to import the table structure is named “Sarah” and is located in the Access1\Tutorial folder provided with your Data Files.
4. Navigate to the Access1\Tutorial folder, where your starting Data Files are stored, and then double-click the Sarah database file. You return to the dialog box.
5. Make sure the Import tables, queries, forms, reports, macros, and modules into the current database option button is selected, and then click the OK button. The Import Objects dialog box opens. The dialog box contains tabs for importing all the different types of Access database objects—tables, queries, forms, and so on. The Tables tab is the current tab.
6. Click the Options button in the dialog box to see all the options for importing tables. See Figure 2-27.
Note the Import Tables section of the dialog box, which contains options for importing the definition and data—that is, the structure of the table and any records contained in the table—or the definition only. You need to import only the structure of the Customer table Sarah created.

7. On the Tables tab, click **Customer** to select this table.

8. In the Import Tables section of the dialog box, click the **Definition Only** option button, and then click the **OK** button. Access creates the Customer table in the Belmont database using the structure of the Customer table in the Sarah database, and opens a dialog box asking if you want to save the import steps.

9. Click the **Close** button to close the dialog box without saving the import steps.

10. Open the Navigation Pane and note that the Customer table is listed in the Tables section.

11. Double-click **Customer** to open the table, and then close the Navigation Pane. The Customer table opens in Datasheet view. The table contains no records. See Figure 2-28.
The table structure you imported contains some of the fields Oren wants, but not all (see Figure 2-26); it also contains some fields Oren does not want in the Customer table. You can add the missing fields using the Data Type gallery.

**Adding Fields to a Table Using the Data Type Gallery**

The **Data Type gallery**, available in the Add & Delete group on the Fields tab, allows you to add a group of related fields to a table at the same time, rather than adding each field to the table individually. The group of fields you add is called a **Quick Start selection**. For example, the Address Quick Start selection adds a collection of fields related to an address, such as Address, City, State, and so on, to the table at one time.

Next, you'll use the Data Type gallery to add the missing fields to the Customer table.

**To add fields to the Customer table using the Data Type gallery:**

1. Click the **Fields** tab on the Ribbon. Note the More Fields button in the Add & Delete group. Clicking this button displays the Data Type gallery.

   Before inserting fields from the Data Type gallery, you need to place the insertion point in the field to the right of where you want to insert the new fields. According to Oren’s design, the Address field should come after the Phone field, so you need to make the next field, FaxNumber, the active field.

2. Click the first row in the **FaxNumber** field to make it the active field.

3. In the Add & Delete group, click the More Fields button. The Data Type gallery opens and displays options for different types of fields you can add to your table.

4. Scroll the gallery down so the Quick Start section is visible. See Figure 2-29.

![Customer table with the Data Type gallery displayed](image-url)
The Quick Start section provides options that will add multiple, related fields to the table at one time. The new fields will be inserted to the left of the current field.

5. In the Quick Start section, click **Address**. Access adds five fields to the table: Address, City, State Province, ZIP Postal, and Country Region. See Figure 2-30.

![Image of Customer table after adding fields from the Data Type gallery](image)

### Modifying the Imported Table

Refer back to Oren’s design for the Customer table (Figure 2-26). To finalize the table design, you need to modify the imported table by deleting fields, renaming fields, changing field data types, and moving some fields. You’ll begin by deleting fields.

### Deleting Fields from a Table Structure

After you’ve created a table, you might need to delete one or more fields. When you delete a field, you also delete all the values for that field from the table. So, before you delete a field you should make sure that you want to do so and that you choose the correct field to delete. You can delete fields from either Datasheet view or Design view.

#### Deleting a Field from a Table Structure

- In Datasheet view, click the column heading for the field you want to delete.
- In the Add & Delete group on the Fields tab, click the Delete button.
- In Design view, click the Field Name box for the field you want to delete.
- In the Tools group on the Design tab, click the Delete Rows button.

The Address Quick Start selection added a field named “Country Region” to the Customer table. Oren doesn’t need a field to store country data because all Belmont Landscapes customers are located in the United States. You’ll begin to modify the Customer table structure by deleting the Country Region field.

#### To delete the Country Region field from the table in Datasheet view:

1. Click the first row in the **Country Region** field (if necessary).
2. In the Add & Delete group on the Fields tab, click the **Delete** button. The Country Region field is removed and the first field, CustomerID, is now the active field.
You can also delete fields from a table structure in Design view. You’ll switch to Design view to delete the other unnecessary fields.

To delete the fields in Design view:

1. In the Views group on the Fields tab, click the View button. The Customer table opens in Design view. See Figure 2-31.

2. Click the FaxNumber Field Name box to make it the current field.

3. In the Tools group on the Design tab, click the Delete Rows button. The FaxNumber field is removed from the Customer table structure.

   You’ll delete the County, Web Page, and Notes fields next. Instead of deleting these fields individually, you’ll select and delete them at the same time.

4. Click and hold down the mouse button on the row selector for the County field, and then drag the mouse to select the Web Page and Notes fields.

5. Release the mouse button. The rows for the three fields are outlined in an orange box, meaning all three fields are selected.

6. In the Tools group, click the Delete Rows button. See Figure 2-32.
Renaming Fields in Design View

To match Oren's design for the Customer table, you need to rename the StateProvince and ZIPPostal fields. In Tutorial 1, you renamed the default primary key field, ID, in Datasheet view. You can also rename fields in Design view by simply editing the names in the Table Design grid.

To rename the fields in Design view:

1. Click to position the insertion point to the right of the word StateProvince in the eighth row's Field Name box, and then press the Backspace key eight times to delete the word "Province." The name of the eighth field is now State.

   You can also select an entire field name and then type new text to replace it.

2. In the ninth row's Field Name box, drag to select the text ZIPPostal, and then type Zip. The text you type replaces the original text. See Figure 2-33.

Figure 2-33

Customer table after renaming fields

Changing the Data Type for a Field in Design View

According to Oren's plan, all of the fields in the Customer table should be Text fields. The table structure you imported specifies the Number data type for the Phone field. In Tutorial 1, you used an option in Datasheet view to change a field's data type. You can also change the data type for a field in Design view.

To change the data type of the Phone field in Design view:

1. Click the right side of the Data Type box for the Phone field to display the list of data types.

2. Click Text in the list. The Phone field is now a Text field. Note that, by default, the Field Size property is set to 255. According to Oren's plan, the Phone field should have a Field Size property of 14. You'll make this change next.

3. Press the F6 key to move to and select the default Field Size property, and then type 14.
Each of the remaining fields you added using the Address Quick Start selection—Address, City, State, and Zip—also has the default field size of 255. You need to change the Field Size property for these fields to match Oren’s design. You’ll also delete any Caption property values for these fields because the field names match how Oren wants them displayed.

To change the Field Size and Caption properties for the fields:
1. Click the **Address** Field Name box to make it the current field.
2. Press the F6 key to move to and select the default Field Size property, and then type 35.
   Note that the Caption property setting for this field is the same as the field name. This field doesn’t need a caption, so you can delete this value.
3. Press the Tab key three times to move to and select the word Address in the Caption box, and then press the Delete key. The Caption property value is removed.
4. Repeat Steps 1 through 3 to change the Field Size property for the City field to 25 and to delete its Caption property value.
5. Change the Field Size property for the State field to 2 and delete its Caption property value.
6. Change the Field Size property for the Zip field to 10 and delete its Caption property value.
7. Click the **Save** button on the Quick Access Toolbar to save your changes to the Customer table.

Finally, Oren would like you to set the Description and Caption properties for the CustomerID, Last, and First fields. You’ll make these changes now.

To enter the Description and Caption property values:
1. Click the **CustomerID** Description box, and then type **Primary key**.
2. In the Field Properties pane, click the **Caption** box.
   After you leave the Description box, a Property Update Options button appears below the Description box for the CustomerID field. When you change a field’s property in Design view, you can use this button to update the corresponding property on forms and reports that include the modified field. For example, if the Belmont database included a form that contained the CustomerID field, you could choose to propagate, or update, the modified Description property in the form by clicking the Property Update Options button, and then choosing the option to make the update everywhere the field is used. The text on the Property Update Options button varies depending on the task; in this case, if you click the button, the option is “Update Status Bar Text everywhere CustomerID is used.”
   Because the Belmont database does not include any forms or reports that are based on the Customer table, you do not need to update the properties, so you can ignore the button for now.
3. In the Caption box for the CustomerID field, type **Customer ID**.
4. Click the **Description** box for the Last field, and then type **Contact’s last name**.
5. Click the **Caption** box, and then type **Last Name**.
6. Click the **Description** box for the First field, and then type **Contact's first name**.
7. Click the **Caption** box, and then type **First Name**. See Figure 2-34.

![Figure 2-34](image)

**Customer table after entering descriptions and captions**

- Description property values entered
- Data type changed to Text
- Caption property entered

8. Click the **Save** button on the Quick Access Toolbar to save your changes to the Customer table.
9. In the Views group on the Design tab, click the **View** button to display the table in Datasheet view.
10. Use the **+** pointer to resize each column so that more fields are visible in the datasheet. Then click in the first row for the **Customer ID** column. See Figure 2-35.

![Figure 2-35](image)

**Modified Customer table in Datasheet view**

After viewing the Customer table datasheet, Oren decides that he would like the First field to appear before the Last field. Earlier in this tutorial, when you created the Invoice table, you learned how to change the order of fields in Design view. Although you can move fields in Datasheet view by dragging a field's column heading to a new location, doing so rearranges only the display of the table's fields; the table structure is not changed. To move a field permanently, you must display the table in Design view.
To move the Last field to follow the First field:

1. In the Views group on the Home tab, click the View button. The Customer table opens in Design view.
2. Point to the row selector for the Last field so the pointer changes to a shape.
3. Click the row selector to select the entire row for the Last field.
4. Place the pointer on the row selector for the Last field, click the pointer down to the line below the row selector for the First field.
5. Release the mouse button. The Last field now appears below the First field.
6. Click the Save button on the Quick Access Toolbar to save the table, and then display the table in Datasheet view.

Trouble? If the fields in the datasheet do not appear in the same order as they did in Design view, close the Customer table, reopen it in Datasheet view, and then close the Navigation Pane. If a field that you deleted (FaxNumber, County, Web Page, or Notes) still appears in the table, close the table, open it in Design view, delete the field(s), save and close the table, open the table in Datasheet view, and then close the Navigation Pane.

With the Customer table design set, you can now enter records in it. You’ll begin by entering two records, and then you’ll use a different method to add the remaining records. Note: Be sure to enter your first name and last name where indicated.

To add two records to the Customer table:

1. Enter the following values in the columns in the first record (these values are for a residential customer with no company name):
   
   Customer ID = 11001
   Company = [do not enter a value; leave blank]
   First Name = [student’s first name]
   Last Name = [student’s last name]
   Phone = 616-866-3901
   Address = 49 Blackstone Dr
   City = Rockford
   State = MI
   Zip = 49341
   Email = student2@milocal123.com

2. Enter the following values in the columns in the second record, for a commercial customer:
   
   Customer ID = 11012
   Company = Grand Rapids Engineering Dept.
   First Name = Anthony
   Last Name = Rodriguez
   Phone = 616-454-9801
   Address = 225 Summer St
   City = Grand Rapids
   State = MI
   Zip = 49503
   Email = arod24@grad11.gov

3. Close the Customer table.
Before Belmont Landscapes decided to store data using Access, Sarah managed the company's customer data in a different system. She exported that data into a text file and now asks you to import it into the new Customer table. You can import the data contained in this text file to add the remaining records to the Customer table.

**Adding Data to a Table by Importing a Text File**

There are many ways to import data into an Access database. So far, you've learned how to add data to an Access table by importing an Excel spreadsheet, and you've created a new table by importing the structure of an existing table. You can also import data contained in text files.

To complete the entry of records in the Customer table, you'll import the data contained in Sarah's text file. The file is named "Customer" and is located in the Access1\Tutorial folder provided with your Data Files.

**To import the data contained in the Customer text file:**

1. Click the **External Data** tab on the Ribbon.
2. In the Import & Link group, click the **Text File** button. The Get External Data - Text File dialog box opens. This dialog box is similar to the one you used earlier when importing the Excel spreadsheet and the Access table structure.
3. Click the **Browse** button. The File Open dialog box opens.
4. Navigate to the **Access1\Tutorial** folder, where your starting Data Files are stored, and then double-click the **Customer** file. You return to the dialog box.
5. Click the **Append a copy of the records to the table** option button. The box to the right of this option becomes active. Next, you need to select the table to which you want to add the data.
6. Click the **arrow** on the box, and then click **Customer**.
7. Click the **OK** button. The first Import Text Wizard dialog box opens. The dialog box indicates that the data to be imported is in a delimited format. A delimited text file is one in which fields of data are separated by a character such as a comma or a tab. In this case, the dialog box shows that data is separated by the comma character in the text file.
8. Make sure the **Delimited** option button is selected in the dialog box, and then click the **Next** button. The second Import Text Wizard dialog box opens. See Figure 2-36.
This dialog box asks you to confirm the delimiter character that separates the fields in the text file you're importing. Access detects that the comma character is used in the Customer text file and selects this option. The bottom area of the dialog box provides a preview of the data you're importing.

9. Make sure the **Comma** option button is selected, and then click the **Next** button. The third, and final, Import Text Wizard dialog box opens. Notice that the Import to Table box shows that the data will be imported into the Customer table.

10. Click the **Finish** button. A dialog box opens asking if you want to save the import steps. You'll only import the customer data once, so you can close the dialog box without saving the import steps.

11. Click the **Close** button in the dialog box to close it without saving the import steps.

Oren asks you to open the Customer table in Datasheet view so he can see the results of importing the text file.

**To view the Customer table datasheet:**

1. Open the Navigation Pane, and then double-click **Customer** to open the Customer table in Datasheet view. The Customer table contains a total of 40 records.

2. Close the Navigation Pane.
3. Resize all the columns to their best fit, scrolling the table datasheet as necessary. When finished, scroll back to display the first fields in the table. See Figure 2-37.

### Figure 2-37
Customer table after importing data from the text file

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Company</th>
<th>First Name</th>
<th>Last Name</th>
<th>Phone</th>
<th>Address</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Save and close the Customer table, and then open the Navigation Pane.

The Belmont database now contains three tables—Contract, Customer, and Invoice—and the tables contain all the necessary records. Your final task is to complete the database design by defining the necessary relationships between its tables.

## Defining Table Relationships

One of the most powerful features of a relational database management system is its ability to define relationships between tables. You use a common field to relate one table to another. The process of relating tables is often called performing a join. When you join tables that have a common field, you can extract data from them as if they were one larger table. For example, you can join the Customer and Contract tables by using the CustomerID field in both tables as the common field. Then you can use a query, form, or report to extract selected data from each table, even though the data is contained in two separate tables, as shown in Figure 2-38. In the CustomerContracts query shown in Figure 2-38, the CustomerID, Company, First, and Last fields are from the Customer table, and the ContractNum and ContractAmt fields are from the Contract table. The joining of records is based on the common field of CustomerID. The Customer and Contract tables have a type of relationship called a one-to-many relationship.
One-to-Many Relationships

As shown earlier in the Session 2.2 Visual Overview, a one-to-many relationship exists between two tables when one record in the first table matches zero, one, or many records in the second table, and when one record in the second table matches at most one record in the first table. For example, as shown in Figure 2-38, customer 11045 has two contracts in the Contract table. Other customers have one or more contracts. Every contract has a single matching customer.

Access refers to the two tables that form a relationship as the primary table and the related table. The primary table is the "one" table in a one-to-many relationship; in Figure 2-38, the Customer table is the primary table because there is only one customer for each contract. The related table is the "many" table; in Figure 2-38, the Contract table is the related table because a customer can have zero, one, or many contracts.
Problem Solving: Avoiding Inconsistent Data

Because related data is stored in two tables, inconsistencies between the tables can occur. Referring to Figure 2-38, consider the following three scenarios:

- Oren adds a record to the Contract table for a new customer, Taylor McNulty, using CustomerID 12050. Oren did not first add the new customer’s information to the Customer table, so this contract does not have a matching record in the Customer table. The data is inconsistent, and the contract record is considered to be an orphaned record.
- In another situation, Oren changes the CustomerID in the Customer table for Kalamazoo Neighborhood Development from 11045 to 12090. Because there is no longer a customer with the CustomerID 11045 in the Customer table, this change creates two orphaned records in the Contract table, and the database is inconsistent.
- In a third scenario, Oren deletes the record for Kalamazoo Neighborhood Development, customer 11045, from the Customer table because this customer no longer does business with Belmont Landscapes. The database is again inconsistent; two records for customer 11045 in the Contract table have no matching record in the Customer table.

You can avoid these types of problems and avoid having inconsistent data in your database by specifying referential integrity (discussed next) between tables when you define their relationships.

Referential Integrity

Referential integrity is a set of rules that Access enforces to maintain consistency between related tables when you update data in a database. Specifically, the referential integrity rules are as follows:

- When you add a record to a related table, a matching record must already exist in the primary table, thereby preventing the possibility of orphaned records.
- If you attempt to change the value of the primary key in the primary table, Access prevents this change if matching records exist in a related table. However, if you choose the Cascade Update Related Fields option, Access permits the change in value to the primary key and changes the appropriate foreign key values in the related table, thereby eliminating the possibility of inconsistent data.
- When you attempt to delete a record in the primary table, Access prevents the deletion if matching records exist in a related table. However, if you choose the Cascade Delete Related Records option, Access deletes the record in the primary table and also deletes all records in related tables that have matching foreign key values.

Understanding the Cascade Delete Related Records Option

Although there are advantages to using the Cascade Delete Related Records option for enforcing referential integrity, its use does present risks as well. You should rarely select the Cascade Delete Related Records option because setting this option might cause you to inadvertently delete records you did not intend to delete. It is best to use other methods for deleting records that give you more control over the deletion process.

Defining a Relationship Between Two Tables

When two tables have a common field, you can define a relationship between them in the Relationships window (see the Session 2.2 Visual Overview). Next, you need to define a one-to-many relationship between the Customer and Contract tables, with
Customer as the primary table and Contract as the related table, and with CustomerID as the common field (the primary key in the Customer table and a foreign key in the Contract table). You’ll also define a one-to-many relationship between the Contract and Invoice tables, with Contract as the primary table and Invoice as the related table, and with ContractNum as the common field (the primary key in the Contract table and a foreign key in the Invoice table).

**To define the one-to-many relationship between the Customer and Contract tables:**

1. Click the **Database Tools** tab on the Ribbon.
2. In the Relationships group on the Database Tools tab, click the **Relationships** button. The Show Table dialog box opens. See Figure 2-39.

![Show Table dialog box](image)

You must add each table participating in a relationship to the Relationships window. Because the Customer table is the primary table in the relationship, you’ll add it first.

3. Click **Customer**, and then click the **Add** button. The Customer table’s field list is added to the Relationships window.
4. Click **Contract**, and then click the **Add** button. The Contract table’s field list is added to the Relationships window.
5. Click the **Close** button in the Show Table dialog box to close it.

So that you can view all the fields and complete field names, you’ll resize the Customer table field list.

6. Use the arrow pointer to drag the bottom of the Customer table field list to lengthen it until the vertical scroll bar disappears and all the fields are visible.

To form the relationship between the two tables, you drag the common field of CustomerID from the primary table to the related table. Then Access opens the Edit Relationships dialog box, in which you select the relationship options for the two tables.

7. Click **CustomerID** in the Customer field list, and then drag it to **CustomerID** in the Contract field list. When you release the mouse button, the Edit Relationships dialog box opens. See Figure 2-40.

**TIP**

You can also double-click a table in the Show Table dialog box to add it to the Relationships window.
The primary table, related table, common field, and relationship type (One-To-Many) appear in the dialog box. After you click the Enforce Referential Integrity check box, the two cascade options become available. If you select the Cascade Update Related Fields option, Access will update the appropriate foreign key values in the related table when you change a primary key value in the primary table. You will not select the Cascade Delete Related Records option because doing so could cause you to delete records that you do not want to delete; this option is rarely selected.

8. Click the Enforce Referential Integrity check box, and then click the Cascade Update Related Fields check box.

9. Click the Create button to define the one-to-many relationship between the two tables and to close the dialog box. The completed relationship appears in the Relationships window, with the join line connecting the common field of CustomerID in each table. See Figure 2-41.

Now you need to define the one-to-many relationship between the Contract and Invoice tables. In this relationship, Contract is the primary ("one") table because there is at most one contract for each invoice. Invoice is the related ("many") table because there are zero, one, or many invoices set up for each contract, depending on how many project phases are involved for each contract.
To define the relationship between the Contract and Invoice tables:

1. In the Relationships group on the Design tab, click the Show Table button. The Show Table dialog box opens.

2. Click Invoice on the Tables tab, click the Add button, and then click the Close button to close the Show Table dialog box. The Invoice table's field list appears in the Relationships window to the right of the Contract table's field list.

Because the Contract table is the primary table in this relationship, you need to drag the ContractNum field from the Contract field list to the Invoice field list.

3. Click and drag the ContractNum field in the Contract field list to the ContractNum field in the Invoice field list. When you release the mouse button, the Edit Relationships dialog box opens.

4. Click the Enforce Referential Integrity check box, and then click the Cascade Update Related Fields check box.

5. Click the Create button to define the one-to-many relationship between the two tables and to close the dialog box. The completed relationship appears in the Relationships window. See Figure 2-42.

Both relationships defined

With both relationships defined, you have connected the data among the three tables in the Belmont database.

6. Click the Save button on the Quick Access Toolbar to save the layout in the Relationships window.

7. In the Relationships group on the Design tab, click the Close button to close the Relationships window.

8. Click the File tab on the Ribbon to display Backstage view.

9. Make sure the Info tab is selected in the navigation bar, and then click the Compact & Repair Database button. Access compacts and repairs the Belmont database.

10. Click the File tab, and then click Close Database to close the Belmont database.
Session 2.2 Quick Check

1. To insert a check mark in an empty check box for a Yes/No field, you press the ____________.

2. What is the keyboard shortcut for inserting the value from the same field in the previous record into the current record?

3. ____________ data is a process that allows you to copy the data from a source without having to open the source file.

4. The ____________ gallery allows you to add a group of related fields to a table at the same time, rather than adding each field to the table individually.

5. What is the effect of deleting a field from a table structure?

6. A(n) ____________ text file is one in which fields of data are separated by a character such as a comma or a tab.

7. The ____________ is the “one” table in a one-to-many relationship, and the ____________ is the “many” table in the relationship.

8. ____________ is a set of rules that Access enforces to maintain consistency between related tables when you update data in a database.