Sort and Query a Database

OUTCOMES

At the end of this chapter you will be able to:

OBJECTIVES

Mastering these objectives will enable you to:

PROJECT 2A
Sort and query a database.

1. Open an Existing Database (p. 117)
2. Create Table Relationships (p. 118)
3. Sort Records in a Table (p. 123)
4. Create a Query in Design View (p. 127)
5. Create a New Query from an Existing Query (p. 129)
6. Sort Query Results (p. 130)
7. Specify Criteria in a Query (p. 132)

PROJECT 2B
Create complex queries.

8. Specify Numeric Criteria in a Query (p. 138)
9. Use Compound Criteria (p. 143)
10. Create a Query Based on More Than One Table (p. 145)
11. Use Wildcards in a Query (p. 147)
12. Use Calculated Fields in a Query (p. 149)
13. Calculate Statistics and Group Data in a Query (p. 152)
14. Create a Crosstab Query (p. 155)

Brendan Fisher/Shutterstock

In This Chapter

In this chapter, you will sort Access database tables and create and modify queries. To convert data into meaningful information, you must manipulate your data in a way that you can answer questions. One question might be: *Which students have a grade point average of 3.0 or higher?* With such information, you could send information about scholarships or internships to selected students.

Questions can be answered by sorting the data in a table or by creating a query. Queries enable you to isolate specific data in database tables by limiting the fields that display and by setting conditions that limit the records to those that match specified conditions. You can also use a query to create a new field that is calculated by using one or more existing fields.

The projects in this chapter relate to **Capital Cities Community College**, which is located in the Washington D. C. metropolitan area. The college provides high-quality education and professional training to residents in the cities surrounding the nation's capital. Its four campuses serve over 50,000 students and offer more than 140 certificate programs and degrees at the associate's level. CapCCC has a highly acclaimed Distance Education program and an extensive Workforce Development program. The college makes positive contributions to the community through cultural and athletic programs and partnerships with businesses and non-profit organizations.

GO! with Microsoft® Access 2010 Comprehensive, by Shelley Gaskin and Nancy Graviett. Published by Prentice Hall.

Project 2A Instructors and Courses Database

myitlab Project 2A Training

Project Activities

In Activities 2.01 through 2.13, you will assist Carolyn Judkins, the Dean of the Business and Information Technology Division at the Jefferson Campus, in locating information about instructors and courses in the Division. Your results will look similar to Figure 2.1.

Project Files

For Project 2A, you will need the following file:

a02A_Instructors_Courses

You will save your database as:

Lastname_Firstname_2A_Instructors_Courses

Project Results

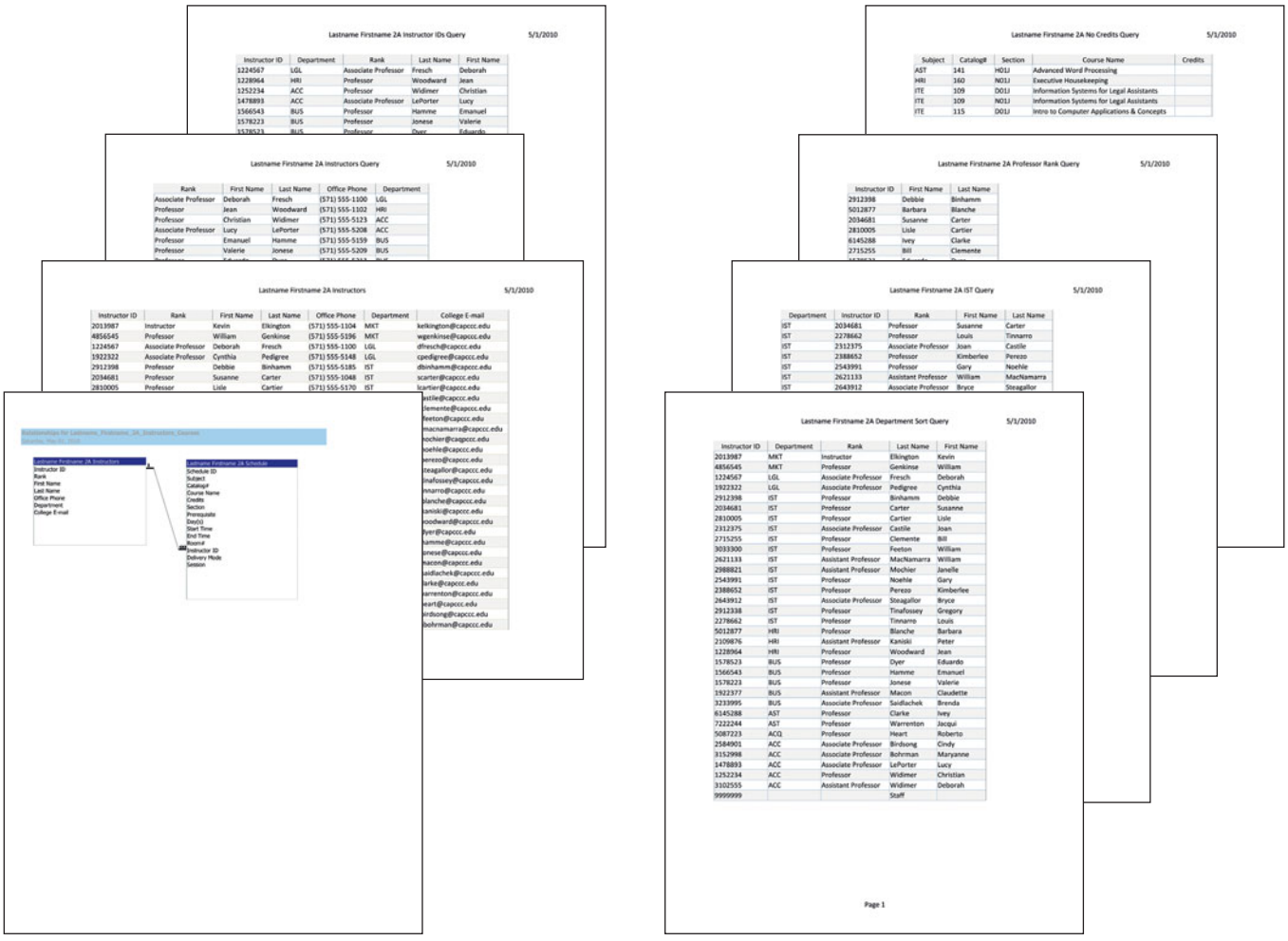


Figure 2.1
Project 2A Instructors and Courses

Objective 1 | Open an Existing Database

There will be instances in which you may want to work with a database and still keep the *original* version of the database. Like the other Microsoft Office 2010 applications, you can open a database file and save it with a new name.

Activity 2.01 | Opening and Renaming an Existing Database

1 Start **Access**. In **Backstage** view, click **Open**. Navigate to the student data files for this textbook, and then open the Access database **a02A_Instructors_Courses**.

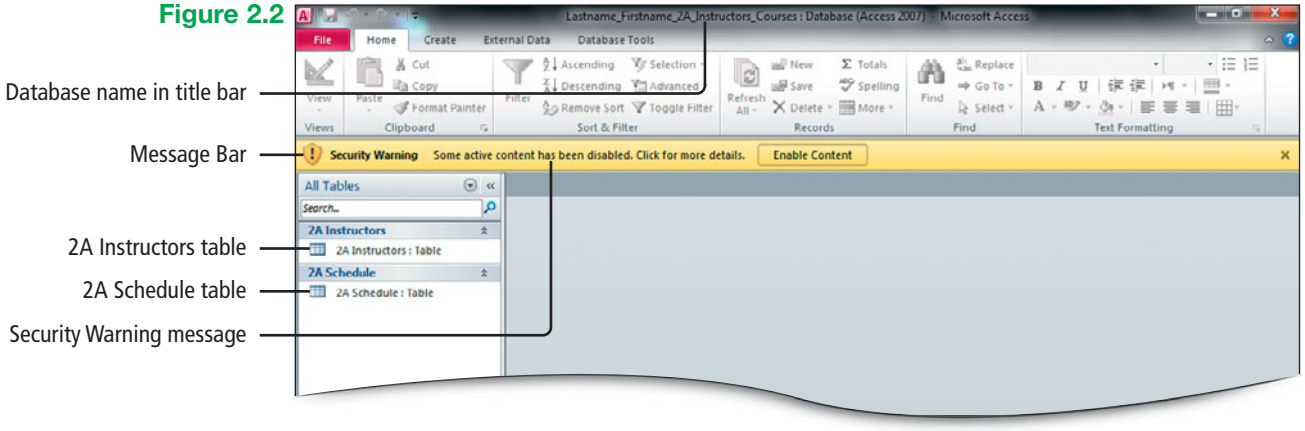
2 Click the **File tab** to return to **Backstage** view, and then click **Save Database As.** In the **Save As** dialog box, navigate to the location where you are saving your databases for this chapter. Create a new folder named **Access Chapter 2** and then click **Open**.

3 In the **File name** box, select the file name, to which *1* has been added at the end. Edit as necessary to name the database **Lastname_Firstname_2A_Instructors_Courses** and then press Enter.

Use this technique when you want to keep a copy of the original database file.

4 On the **Message Bar**, notice the **Security Warning**. In the **Navigation Pane**, notice that this database contains two table objects. Compare your screen with Figure 2.2.

Figure 2.2



Activity 2.02 | Resolving Security Alerts and Renaming Tables

The ***Message Bar*** is the area below the Ribbon that displays information such as security alerts when there is potentially unsafe, active content in an Office document that you open. Settings that determine the alerts that display on your Message Bar are set in the Access ***Trust Center***, which is an area of Access where you can view the security and privacy settings for your Access installation.

You may or may not be able to change the settings in the Trust Center, depending upon decisions made within your organization's computing environment. You can display the Trust Center from Options, which is available in Backstage view.

1 On the **Message Bar**, click the **Enable Content** button.

When working with the student files that accompany this textbook, repeat these actions each time you see this security warning. Databases for this textbook are safe to use on your computer.

2 In the **Navigation Pane**, right-click the **2A Instructors** table, and then click **Rename**. With the table name selected and using your own name, type **Lastname Firstname 2A Instructors** and then press Enter to rename the table. Using the same technique, **Rename** the **2A Schedule** table to **Lastname Firstname 2A Schedule**

Including your name in the table enables you and your instructor to easily identify your work, because Access includes the table name in the header of printed and PDF pages.

3 Point to the right edge of the **Navigation Pane** to display the ⇔ pointer. Drag to the right to widen the pane until both table names display fully.

Objective 2 | Create Table Relationships

Access databases are relational databases because the tables in the database can relate—actually connect—to other tables through common fields. Recall that common fields are fields that contain the same data in more than one table.

After you have a table for each subject in your database, you must provide a way to connect the data in the tables when you need meaningful information. To do this, create common fields in related tables, and then define table relationships. A ***relationship*** is an association that you establish between two tables based on common fields. After the relationship is established, you can create a query, a form, or a report that displays information from more than one table.

Activity 2.03 | Creating Table Relationships and Enforcing Referential Integrity

In this activity, you will create a relationship between two tables in the database.

1 Double-click your **2A Instructors** table to open it in the object window and examine its contents. Then open your **2A Schedule** table and examine its contents.

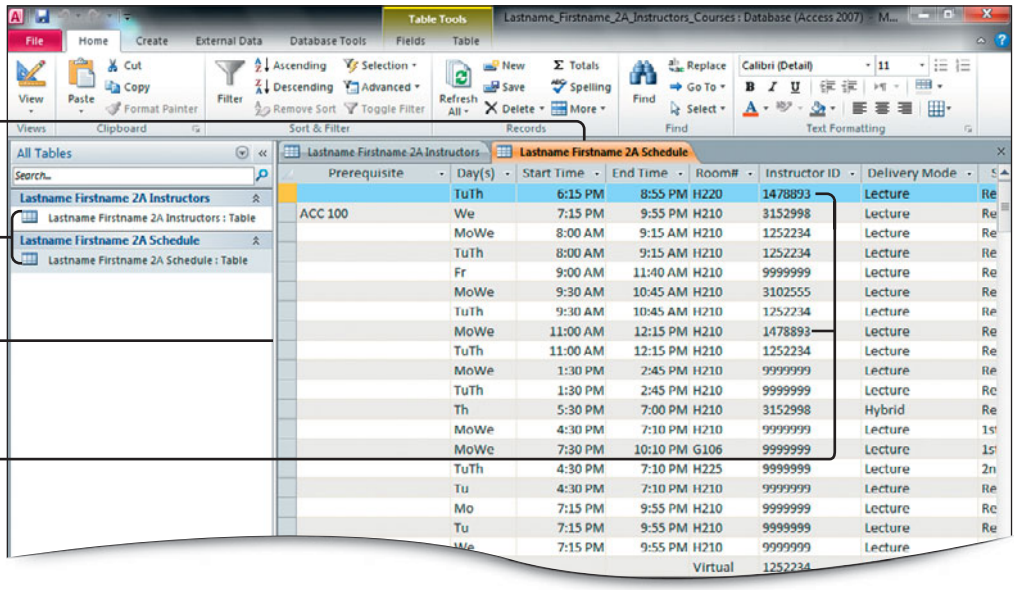
In the 2A Instructors table, *Instructor ID* is the primary key field, which ensures that each instructor will appear in the table only one time. No two instructors have the same Instructor ID.

In the 2A Schedule table, *Schedule ID* is the primary key field. Every scheduled course section during an academic term has a unique Schedule ID. The 2A Schedule table includes the *Instructor ID* field, which is the common field between the 2A Schedule table and the 2A Instructors table.

2 In the **2A Schedule** table, scroll to the right to display the Instructor ID field, and then compare your screen with Figure 2.3.

Because *one* instructor can teach *many* different courses, *one* Instructor ID number can be present *many* times in the 2A Schedule table. This relationship between each instructor and the courses is known as a ***one-to-many relationship***. This is the most common type of relationship in Access.

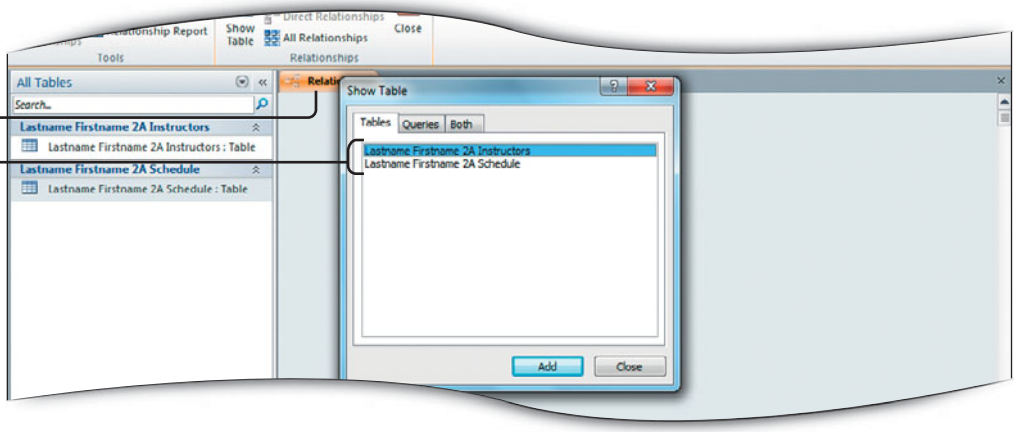
Figure 2.3



3 In the upper right corner of the object window, click **Close** ⊠ two times to close each table. Click the **Database Tools tab**, and then in the **Relationships group**, click the **Relationships** button. Compare your screen with Figure 2.4.

The Show Table dialog box displays in the Relationships window. In the Show Table dialog box, the Tables tab displays all of the table objects in the database. Your two tables are listed.

Figure 2.4



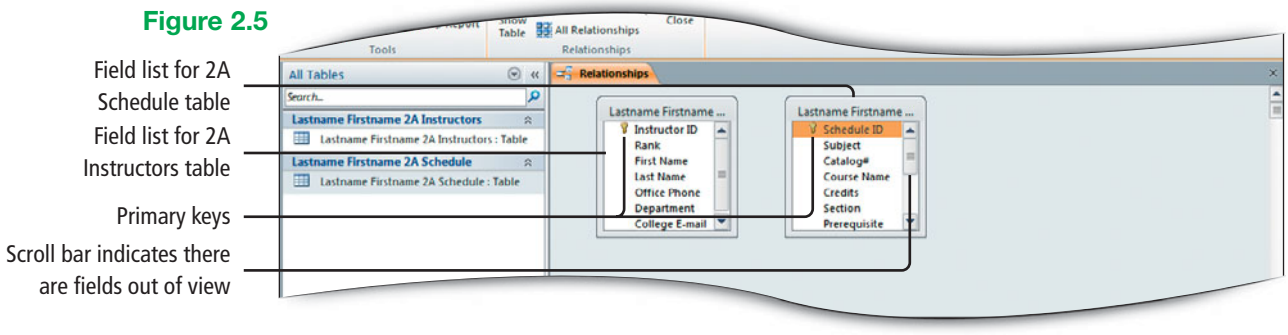
4 Point to the title bar of the **Show Table** dialog box, and then drag down and to the right slightly to move the **Show Table** dialog box away from the top of the **Relationships** window.

Moving the Show Table dialog box enables you to see the tables as they are added to the Relationships window.

5 In the **Show Table** dialog box, click your **2A Instructors** table, and then at the bottom of the dialog box, click **Add**. In the **Show Table** dialog box, double-click your **2A Schedule** table to add the table to the **Relationships** window. In the **Show Table** dialog box, click **Close**, and then compare your screen with Figure 2.5.

You can use either technique to add a table to the Relationships window. A ***field list***—a list of the field names in a table—for each of the two table objects displays, and each table's primary key is identified. Although this database currently has only two tables, larger databases can have many tables. Scroll bars in a field list indicate that there are fields that are not currently in view.

Figure 2.5



Alert! | Are There More Than Two Field Lists in the Relationships Window?

If you double-click a table more than one time, a duplicate field list displays in the Relationships window. To remove a field list from the Relationships window, right-click the title bar of the field list, and then click Hide Table. Alternatively, click anywhere in the field list, and then on the Design tab, in the Relationships group, click the Hide Table button.

6 In the **2A Schedule** field list—the field list on the right—point to the title bar to display the pointer. Drag the field list to the right until there is about 2 inches between the field lists.

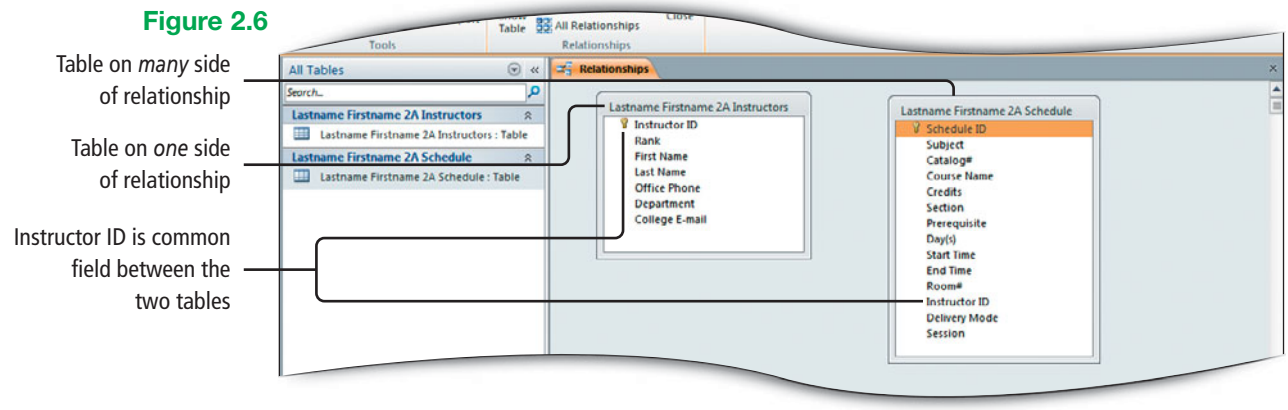
7 In the **2A Instructors** field list—the field list on the left—point to the lower right corner of the field list to display the pointer, and then drag down and to the right to increase the height and width of the field list until the entire name of the table in the title bar displays and all of the field names display.

This action enables you to see all of the available fields and removes the vertical scroll bar.

8 By using the same technique and the pointer, resize the **2A Schedule** field list so that all of the field names and the table name display as shown in Figure 2.6.

Recall that *one* instructor can teach *many* scheduled courses. This arrangement of the tables on your screen displays the *one table* on the left side and the *many table* on the right side. Recall also that the primary key in each table is the field that uniquely identifies the record in each table. In the 2A Instructors table, each instructor is uniquely identified by the Instructor ID. In the 2A Schedule table, each scheduled course section is uniquely identified by the Schedule ID.

Figure 2.6



Note | The Field That Is Highlighted Does Not Matter

After you rearrange the two field lists in the Relationships window, the highlighted field indicates the active field list, which is the list you moved last. This is of no consequence for completing the activity.

Another Way

On the Design tab, in the Tools group, click the Edit Relationships button. In the Edit Relationships dialog box, click Create New, and then in the Create New dialog box, designate the tables and fields that will create the relationship.

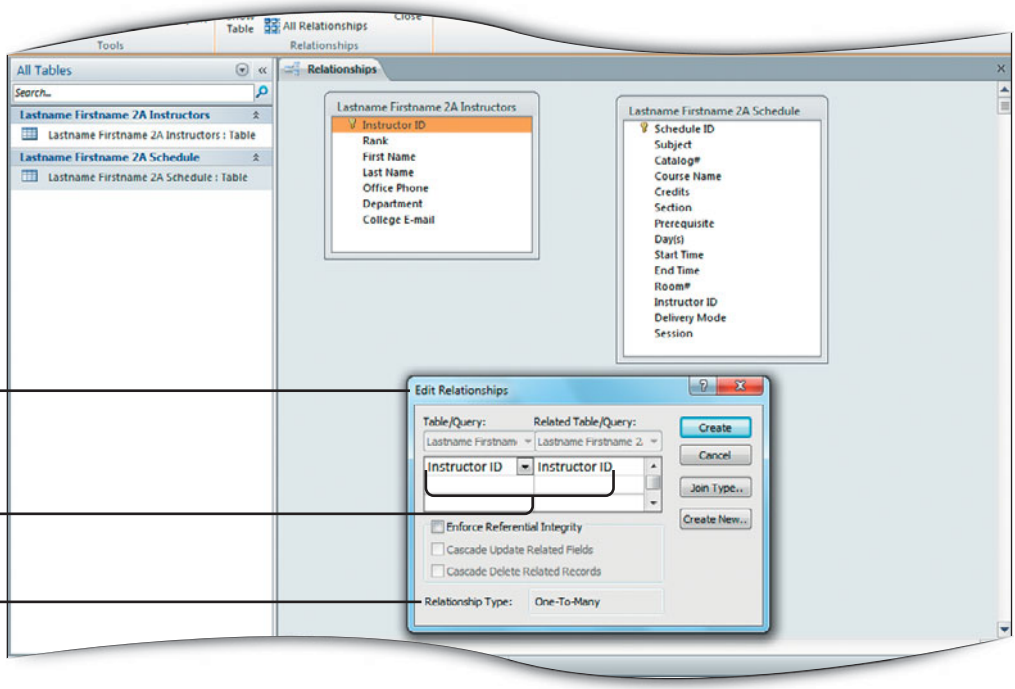
9 In the **2A Instructors** field list, point to **Instructor ID**, hold down the left mouse button, and then drag down and to the right into the **2A Schedule** field list until the pointer's arrow is on top of **Instructor ID**. Then release the mouse button to display the **Edit Relationships** dialog box.

As you drag, a small graphic displays to indicate that you are dragging a field from one field list to another. A table relationship works by matching data in two fields—the common field. In these two tables, the common field has the same name—*Instructor ID*. Common fields are not required to have the same names; however, they must have the same data type and field size.

10 Point to the title bar of the **Edit Relationships** dialog box, and then drag the dialog box below the two field lists as shown in Figure 2.7.

Both tables include the Instructor ID field—the common field between the two tables. By dragging, you create the *one-to-many* relationship. In the 2A Instructors table, Instructor ID is the primary key. In the 2A Schedule table, Instructor ID is referred to as the ***foreign key*** field. The foreign key is the field in the related table used to connect to the primary key in another table. The field on the *one* side of the relationship is typically the primary key.

Figure 2.7



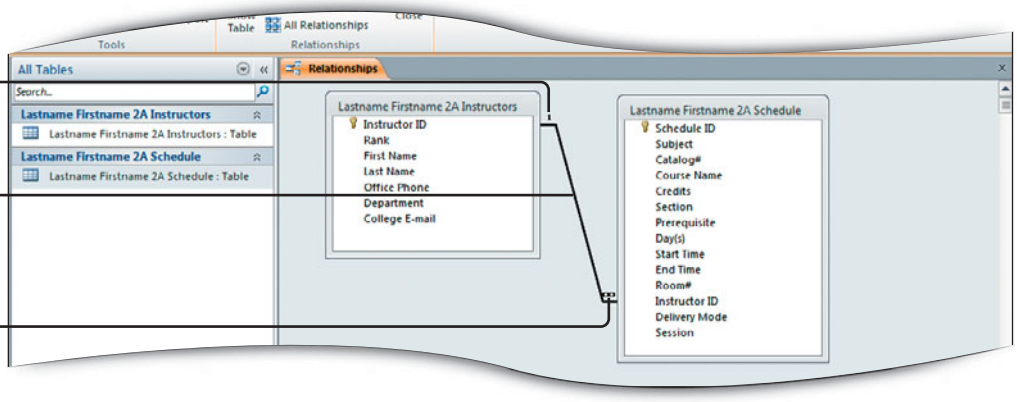
11 In the **Edit Relationships** dialog box, click to select the **Enforce Referential Integrity** check box.

Referential integrity is a set of rules that Access uses to ensure that the data between related tables is valid. Enforcing referential integrity ensures that an instructor cannot be added to the 2A Schedules table if the Instructor ID is *not* included in the 2A Instructors table. Similarly, enforcing referential integrity ensures that you cannot delete an instructor from the 2A Instructors table if there is a course listed in the 2A Schedule table for that instructor.

12 In the **Edit Relationships** dialog box, click the **Create** button, and then compare your screen with Figure 2.8.

A ***join line***—the line joining two tables—displays between the two tables. On the join line, *1* indicates the *one* side of the relationship, and the infinity symbol (∞) indicates the *many* side of the relationship. These symbols display when referential integrity is enforced.

Figure 2.8



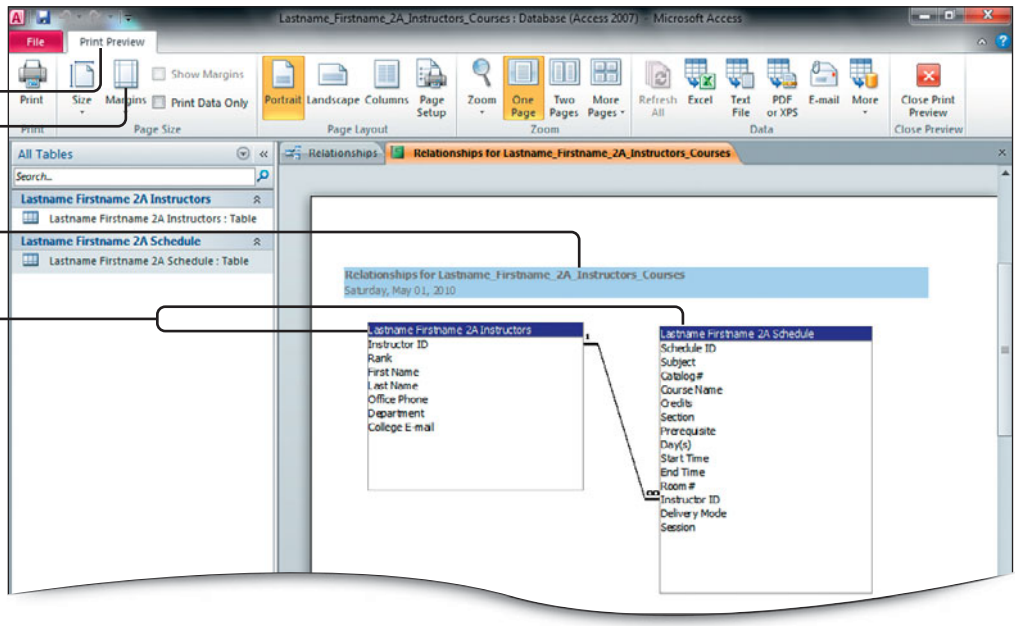
Activity 2.04 | Printing a Relationship Report and Displaying Subdatasheet Records

The Relationships window provides a map of how your database tables are related, and you can print this information as a report.

1 With the **Relationships** window open, on the **Design tab**, in the **Tools group**, click the **Relationship Report** button to create the report and display it in Print Preview.

2 On the **Print Preview tab**, in the **Page Size group**, click the **Margins** button, and then click **Normal**. Compare your screen with Figure 2.9. If instructed to do so, create a paper or electronic printout of this relationship report.

Figure 2.9



3 On the **Quick Access Toolbar**, click the **Save** button to save the report. In the **Save As** dialog box, click **OK** to accept the default name.

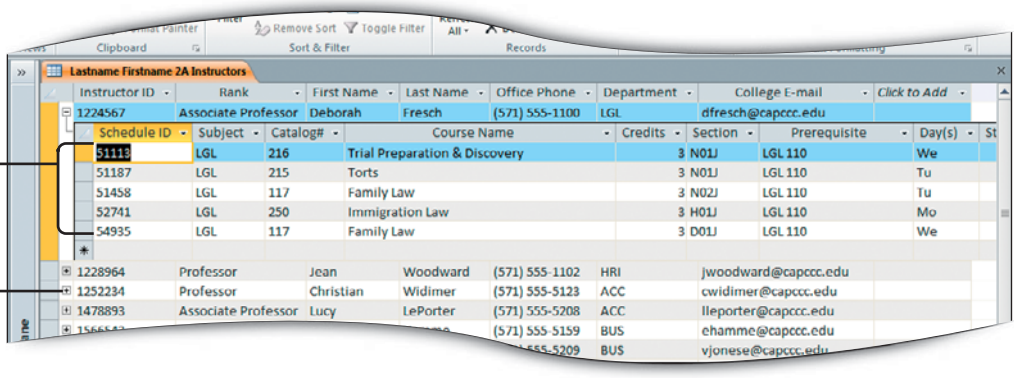
The report name displays in the Navigation Pane under *Unrelated Objects*. Because the report is just a map of the relationships, and not a report containing actual records, it is not associated with any of the tables.

4 In the object window, **Close** the report, and then **Close** the **Relationships** window.

5 From the **Navigation Pane**, open your **2A Instructors** table, and then **Close** the **Navigation Pane**. For the first record—*Instructor ID 1224567*—on the left side of the record, click the **plus sign** (+), and then compare your screen with Figure 2.10.

Plus signs to the left of a record in a table indicate that *related* records exist in another table. Clicking the plus sign displays the related records in a ***subdatasheet***. In the first record, for *Deborah Fresch*, you can see that related records exist in the 2A Schedule table—she is teaching five LGL courses that are listed in the schedule. The plus sign displays because you created a relationship between the two tables using the Instructor ID field—the common field.

Figure 2.10



6 For the first record, click the **minus sign** (-) to collapse the subdatasheet.

More Knowledge | Other Types of Relationships: One-to-One and Many-to-Many

There are other relationships you can create using the same process in the Relationships window. The type of relationship is determined by the placement of the primary key field. A one-to-one relationship exists between two tables when a record in one table is related to a single record in a second table. In this case, both tables use the same field as the primary key. This is most often used when data is placed in a separate table because access to the information is restricted.

You can also create a many-to-many relationship between tables, where many records in one table can be related to many records in another table. For example, many students can enroll in many courses. To create a many-to-many relationship, you must create a third table that contains the primary key fields from both tables. These primary key fields are then joined to their related fields in the other tables. In effect, you create multiple one-to-one relationships.

Objective 3 | Sort Records in a Table

Sorting is the process of arranging data in a specific order based on the value in a field. For example, you can sort the names in your address book alphabetically by each person's last name, or you can sort your DVD collection by the date of purchase. Initially, records in an Access table display in the order they are entered into the table. When a primary key is established, the records display in order based on the primary key field.

Activity 2.05 | Sorting Records in a Table in Ascending or Descending Order

In the following activity, you will determine the departments of the faculty in the Business and Information Technology Division by sorting the data. You can sort data in either ***ascending order*** or ***descending order***. Ascending order sorts text alphabetically (A to Z) and sorts numbers from the lowest number to the highest number. Descending order sorts text in reverse alphabetical order (Z to A) and sorts numbers from the highest number to the lowest number.

Another Way

On the Home tab, in the Sort & Filter group, click the Ascending button.

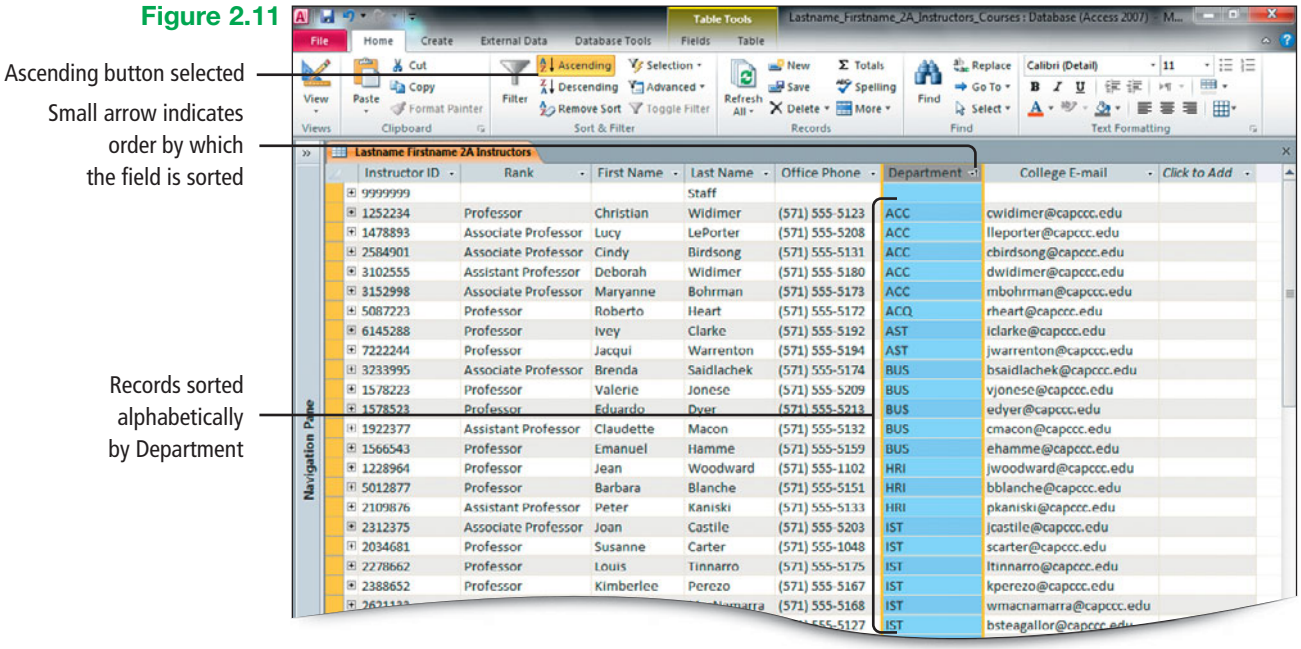
1. Notice that the records in the **2A Instructors** table are sorted in ascending order by **Instructor ID**, which is the primary key field.

2. In the field names row, click the **Department arrow**, click **Sort A to Z**, and then compare your screen with Figure 2.11.

To sort records in a table, click the arrow to the right of the field name in the column on which you want to sort, and then choose the sort order. After a field is sorted, a small arrow in the field name box indicates its sort order. The small arrow in the field name points up, indicating an ascending sort; and in the Ribbon, the Ascending button is selected.

The records display in alphabetical order by Department. Because the department names are now grouped together, you can quickly scroll the length of the table to see the instructors in each department. The first record in the table has no data in the Department field because the Instructor ID number *9999999* is reserved for Staff, a designation that is used until a scheduled course has been assigned to a specific instructor.

Figure 2.11



3. On the **Home tab**, in the **Sort & Filter group**, click the **Remove Sort** button to clear the sort and return the records to the default sort order, which is by the primary key field—*Instructor ID*.

4. Click the **Last Name arrow**, and then click **Sort Z to A**.

The records in the table are sorted by last name in reverse alphabetical order. The small arrow in the Field name box points down, indicating a descending sort. On the Ribbon, the Descending button is selected.

5. In the **Sort & Filter group**, click the **Remove Sort** button.

Activity 2.06 | Sorting Records in a Table on Multiple Fields

To sort a table on two or more fields, first identify the fields that will act as the ***outermost sort field*** and the ***innermost sort field***. The outermost sort field is the first level of sorting, and the innermost sort field is the second level of sorting. For example, you might want to sort first by the Last Name field, which would be the outermost sort field, and then by the First Name field, which would be the innermost sort field. After you identify your outermost and innermost sort fields, sort the innermost field first, and then sort the outermost field.

In this activity, you will sort the records in descending order by the department name. Within each department name, you will sort the records in ascending order by last name.

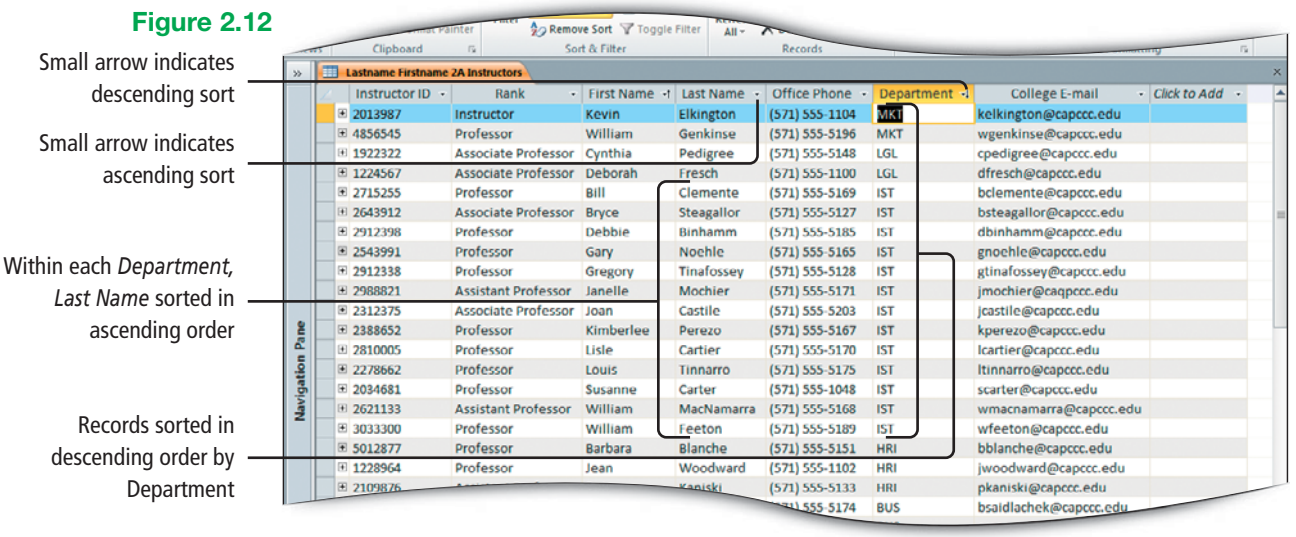
1 In the **Last Name** field, click any record. In the **Sort & Filter group**, click the **Ascending** button.

The records are sorted in ascending alphabetical order by Last Name—the innermost sort field.

2 Point anywhere in the **Department** field, and then right-click. From the shortcut menu, click **Sort Z to A**. Compare your screen with Figure 2.12.

The records are sorted in descending alphabetical order first by Department—the *outermost* sort field—and then within a specific Department grouping, the sort continues in ascending alphabetical order by Last Name—the *innermost* sort field. The records are sorted on multiple fields using both ascending and descending order.

Figure 2.12



3 Display **Backstage** view, click **Print**, and then click **Print Preview**. In the **Page Layout** group, click the **Landscape** button. In the **Zoom group**, click the **Two Pages** button, and notice that the table will print on two pages.

4 On the **Print Preview tab**, in the **Print group**, click the **Print** button. Under **Print Range**, click the **Pages** option button. In the **From** box, type **1** and then in the **To** box, type **1** to print only the first page. If directed to submit a paper copy, click OK or create an electronic copy as instructed. To create a PDF of only the first page, in the Data group, click PDF or XPS, click the Options button, and then indicate *Page 1 to 1*. In the **Close Preview group**, click the **Close Print Preview** button.

5 In the object window, **Close** the table. In the message box, click **Yes** to save the changes to the sort order.

6 **Open** the **Navigation Pane**, and then open the **2A Instructors** table. Notice the table was saved with the sort order you specified.

7 In the **Sort & Filter group**, click the **Remove Sort** button. **Close** the table, and in the message box, click **Yes** to save the table with the sort removed. **Close** the **Navigation Pane**.

Generally, tables are not stored with the data sorted. Instead, queries are created that sort the data; and then reports are created to display the sorted data.

Objective 4 | Create a Query in Design View

Recall that a ***select query*** is a database object that retrieves (selects) specific data from one or more tables and then displays the specified data in Datasheet view. A query answers a question such as *Which instructors teach courses in the IST department?* Unless a query has already been set up to ask this question, you must create a new query.

Database users rarely need to see all of the records in all of the tables. That is why a query is so useful; it creates a ***subset*** of records—a portion of the total records—according to your specifications and then displays only those records.

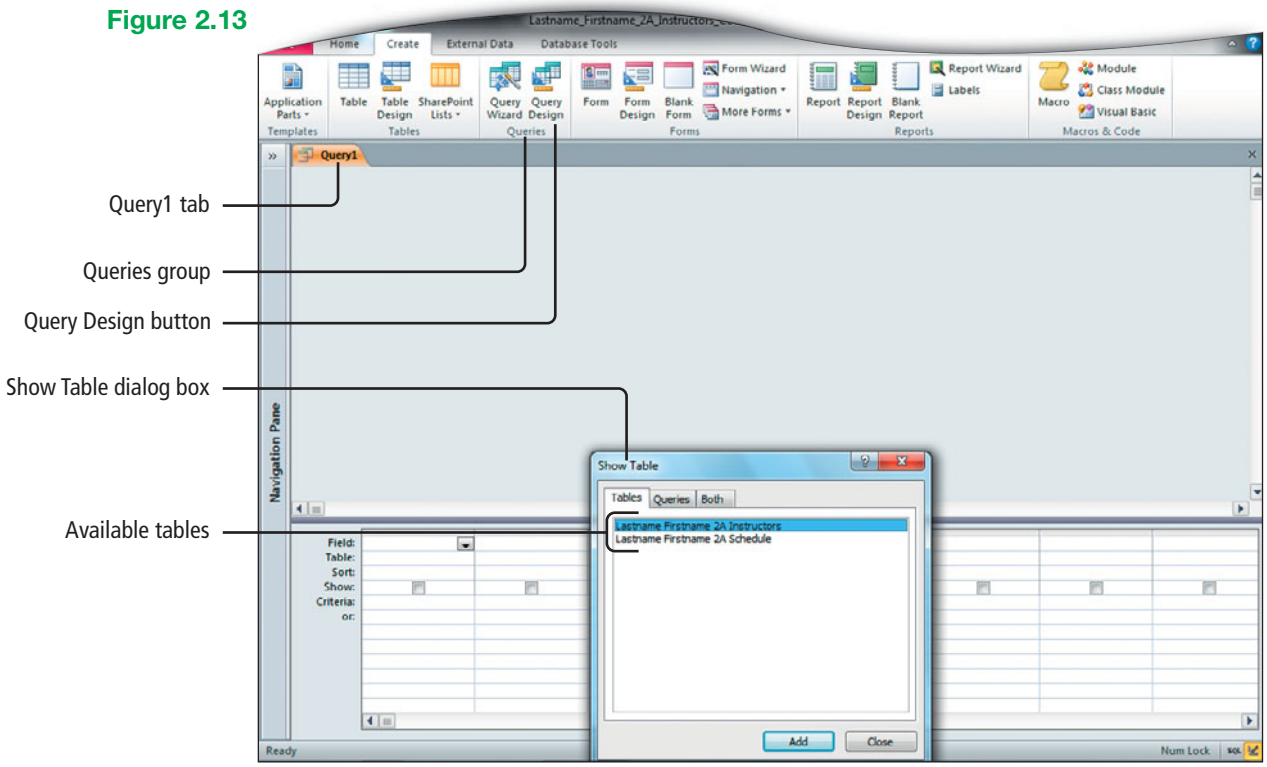
Activity 2.07 | Creating a New Select Query in Design View

Previously, you created a query using the Query Wizard. To create complex queries, use Query Design view. The table or tables from which a query selects its data is referred to as the ***data source***.

1 On the Ribbon, click the **Create tab**, and then in the **Queries group**, click the **Query Design** button. Compare your screen with Figure 2.13.

A new query opens in Design view and the Show Table dialog box displays, which lists both tables in the database.

Figure 2.13



2 In the **Show Table** dialog box, double-click **2A Instructors**, and then **Close** the **Show Table** dialog box.

A field list for the 2A Instructors table displays in the upper area of the Query window. The Instructor ID field is the primary key field in this table. The Query window has two parts: the ***table area*** (upper area), which displays the field lists for tables that are used in the query, and the ***design grid*** (lower area), which displays the design of the query.

Alert! | Is There More Than One Field List in the Query Window?

If you double-click a table more than one time, a duplicate field list displays in the Query window. To remove a field list from the Query window, right-click the title bar of the field list, and then click Remove Table.

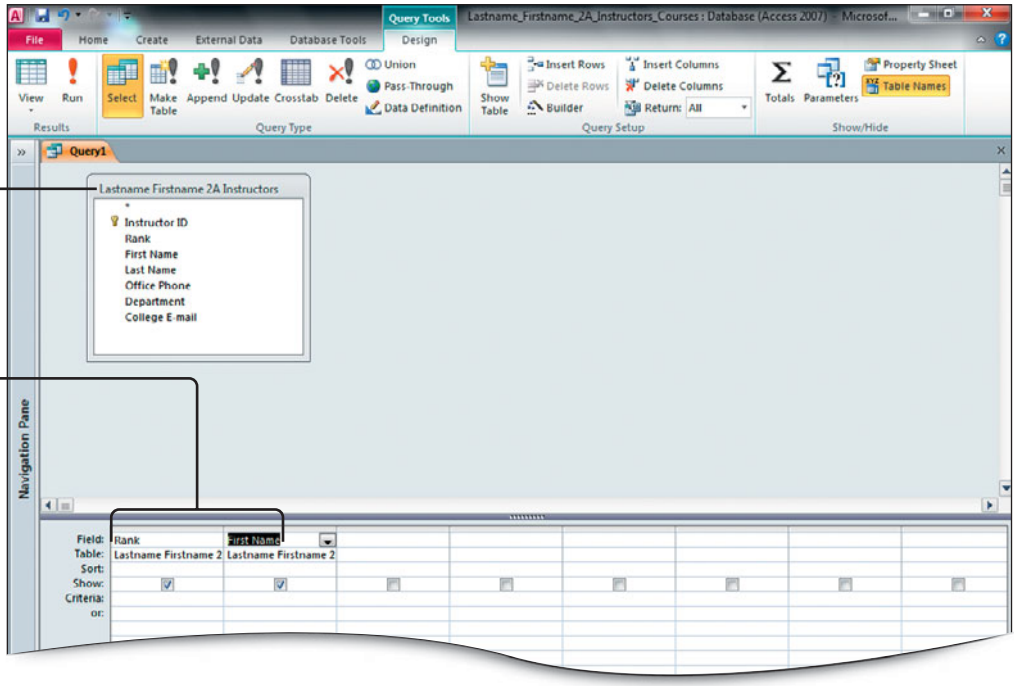
3 Point to the lower right corner of the field list to display the pointer, and then drag down and to the right to expand the field list, displaying all of the field names and the table name. In the **2A Instructors** field list, double-click **Rank**, and then look at the design grid.

The Rank field name displays in the design grid in the Field row. You limit the fields that display when the query is run by placing only the desired field names in the design grid.

4 In the **2A Instructors** field list, point to **First Name**, hold down the left mouse button, and then drag down into the design grid until the pointer displays in the **Field** row in the second column. Release the mouse button, and then compare your screen with Figure 2.14.

This is a second way to add field names to the design grid. As you drag the field, a small rectangular shape attaches to the mouse pointer. When you release the mouse button, the field name displays in the Field row.

Figure 2.14



5 In design grid, in the **Field** row, click in the third column, and then click the **arrow** that displays. From the list, click **Last Name** to add the field to the design grid, which is a third way to add a field to the design grid.

6 Using one of the techniques you just practiced, add the **Office Phone** field to the fourth column and the **Department** field to the fifth column in the design grid.

Activity 2.08 | Running, Saving, Printing, and Closing a Query

After you create a query, you ***run*** it to display the results. When you run a query, Access looks at the records in the table (or tables) you have included in the query, finds the records that match the specified conditions (if any), and displays only those records in a datasheet. Only the fields that you have added to the design grid display in the query results. The query always runs using the current table or tables, presenting the most up-to-date information.

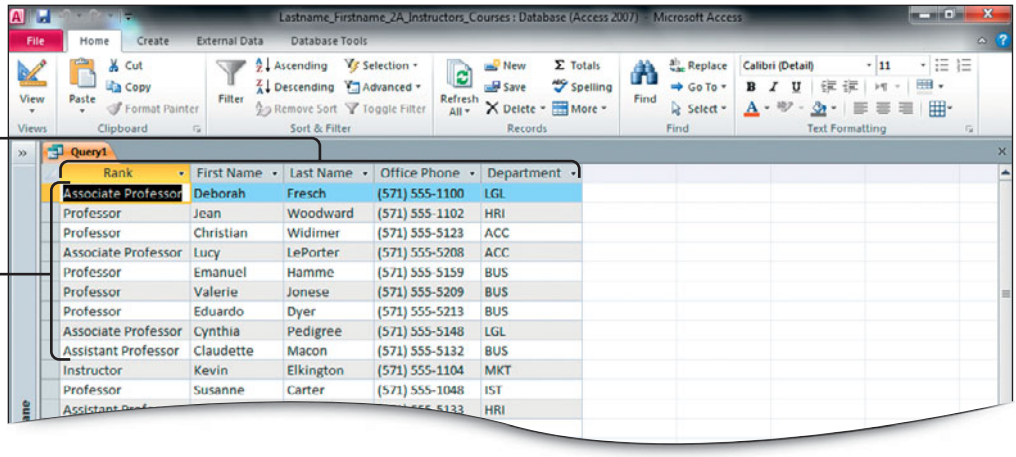
Another Way

On the Design tab, in the Results group, click the View button to automatically start the Run command.

1 On the **Design tab**, in the **Results group**, click the **Run** button, and then compare your screen with Figure 2.15.

This query answers the question, *What is the Rank, First Name, Last Name, Office Phone number, and Department of all of the instructors in the 2A Instructors table?* A query is a subset of the records in one or more tables, arranged in Datasheet view, using the fields and conditions that you specify. The five fields that you specified in the design grid display in columns, and the records from the 2A Instructors table display in rows.

Figure 2.15



2 On the **Quick Access Toolbar**, click the **Save** button. In the **Save As** dialog box, type **Lastname Firstname 2A Instructors Query** and then click **OK**.

Save your queries if you are likely to ask the same question again; doing so will save you the effort of creating the query again to answer the same question.

Alert! | Does a Message Display After Entering a Query Name?

Query names are limited to 64 characters. For all projects, if you have a long last name or first name that results in your query name exceeding the 64-character limit, ask your instructor how you should abbreviate your name.

3 Display **Backstage** view, click **Print**, and then click **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

Queries answer questions and gather information from the data in the tables. Queries are typically created as a basis for a report, but query results can be printed like any other table of data.

4 **Close** the query. **Open** the **Navigation Pane**, and then notice that the **2A Instructors Query** object displays under the **2A Instructors** table object.

The new query name displays in the Navigation Pane under the table with which it is related—the 2A Instructors table. Only the design of the query is saved. The records still reside in the table object. Each time you open the query, Access runs it again and displays the results based on the data stored in the related table(s). Thus, the results of a query always reflect the latest information in the related table(s).

Objective 5 | Create a New Query from an Existing Query

You can create a new query from scratch or you can open an existing query, save it with new name, and modify the design to suit your needs. Using an existing query saves you time if your new query uses all or some of the same fields and conditions in an existing query.

Activity 2.09 | Creating a New Query from an Existing Query

1 From the **Navigation Pane**, open your **2A Instructors Query** by either double-clicking the name or by right-clicking and clicking Open.

The query runs, opens in Datasheet view, and displays the records from the 2A Instructors table as specified in the query design grid.

2 Display **Backstage** view, and then click **Save Object As**. In the **Save As** dialog box, type **Lastname Firstname 2A Instructor IDs Query** and then click **OK**. Click the **Home tab**, and then in the **Views group**, click the **View** button to switch to **Design** view.

A new query, based on a copy of the 2A Instructors Query, is created and displays in the object window and in the Navigation Pane under its data source—the 2A Instructors table.

3 **Close** the **Navigation Pane**. In the design grid, point to the thin gray selection bar above the **Office Phone** field name until the pointer displays. Click to select the **Office Phone** column, and then press Del.

This action deletes the field from the query design only—it has no effect on the field in the underlying 2A Instructors table. The Department field moves to the left. Similarly, you can select multiple fields and delete them at one time.

4 From the gray selection bar, select the **First Name** column. In the selected column, point to the selection bar to display the pointer, and then drag to the right until a dark vertical line displays on the right side of the **Last Name** column. Release the mouse button to position the **First Name** field in the third column.

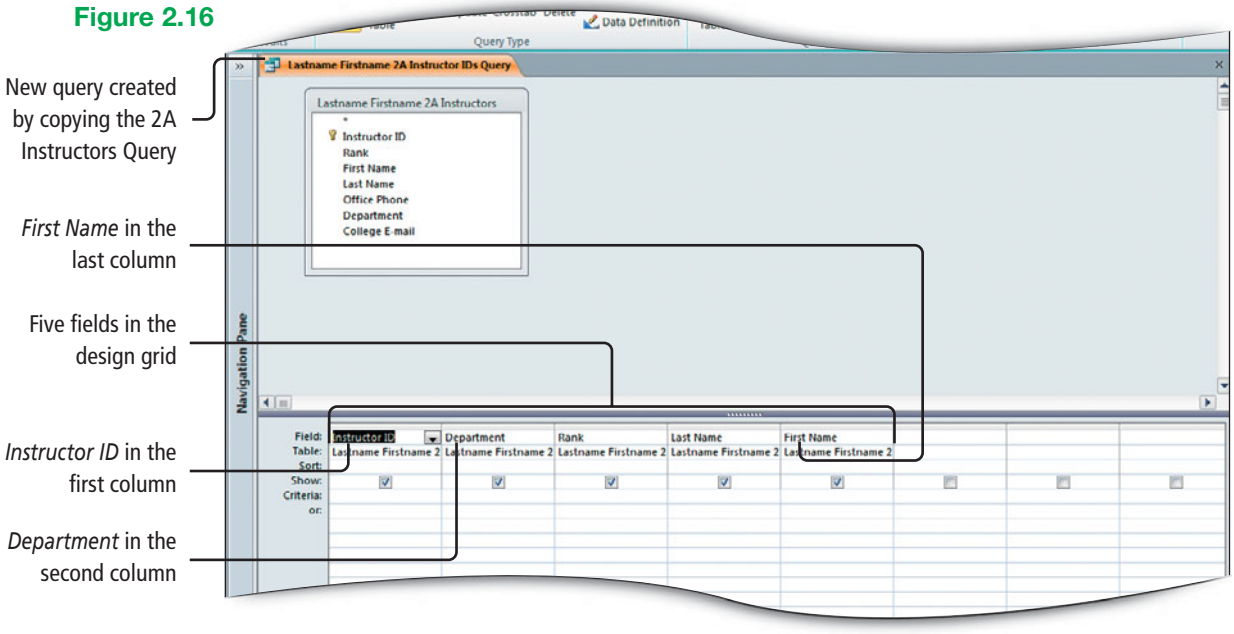
To rearrange fields in the query design, select the field to move, and then drag it to a new position in the design grid.

5 Using the technique you just practiced, move the **Department** field to the left of the **Rank** field.

6 From the field list, drag the **Instructor ID** field down to the first column in the design grid until the pointer displays, and then release the mouse button. Compare your screen with Figure 2.16.

The Instructor ID field displays in the first column, and the remaining four fields move to the right. Use this method to insert a field to the left of a field already displayed in the design grid.

Figure 2.16



7 On the **Design tab**, in the **Results group**, click the **Run** button.

This query answers the question, *What is the Instructor ID, Department, Rank, Last Name, and First Name for every instructor in the 2A Instructors table?* The results of the query are a subset of the records contained in the 2A Instructors table. The records are sorted by the primary key field—Instructor ID.

8 From **Backstage** view, display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

9 **Close** the query, and in the message box, click **Yes** to save the changes to the design—deleting a field, moving two fields, and adding a field. **Open** the **Navigation Pane**.

The query is saved and closed. The new query name displays in the Navigation Pane under the related table. Recall that when you save a query, only the *design* of the query is saved; the records reside in the related table object or objects.

Objective 6 | Sort Query Results

You can sort the results of a query in ascending or descending order in either Datasheet view or Design view. Use Design view if your query results should display in a specified sort order, or if you intend to use the sorted results in a report.

Activity 2.10 | Sorting Query Results

In this activity, you will save an existing query with a new name, and then sort the query results by using the Sort row in Design view.

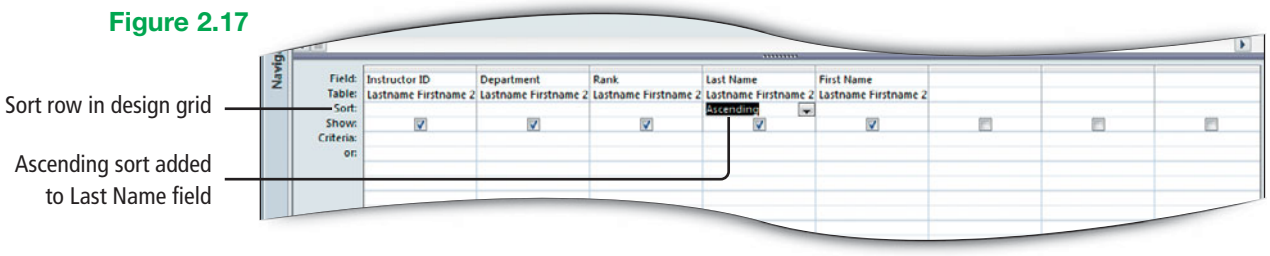
1 On the **Navigation Pane**, click your **2A Instructor IDs Query**. Display **Backstage** view, and then click **Save Object As**. In the **Save As** dialog box, type **Lastname Firstname 2A Department Sort Query** and then click **OK**. Click the **Home tab**, and then drag the right edge of the **Navigation Pane** to the right to increase its width so that the names of the new query and the relationship report display fully.

Access creates a new query, based on a copy of your 2A Instructors ID Query; that is, the new query includes the same fields in the same order as the query on which it is based.

2 In the **Navigation Pane**, right-click your **2A Department Sort Query**, and then click **Design View**. **Close** « the **Navigation Pane**.

3 In the design grid, in the **Sort** row, click in the **Last Name** field to display the insertion point and an arrow. Click the **Sort arrow**, and then in the list, click **Ascending**. Compare your screen with Figure 2.17.

Figure 2.17



4 On the **Design tab** in the **Results group**, click the **Run** button.

In the query result, the records are sorted in ascending alphabetical order by the Last Name field, and two instructors have the same last name of *Widimer*.

5 On the **Home tab** in the **Views group**, click the **View** button to switch to **Design** view.

6 In the **Sort** row, click in the **First Name** field, click the **Sort arrow**, and then click **Ascending**. **Run** the query.

In the query result, the records are sorted first by the Last Name field. If instructors have the same last name, then Access sorts those records by the First Name field. The two instructors with the last name of *Widimer* are sorted by their first names.

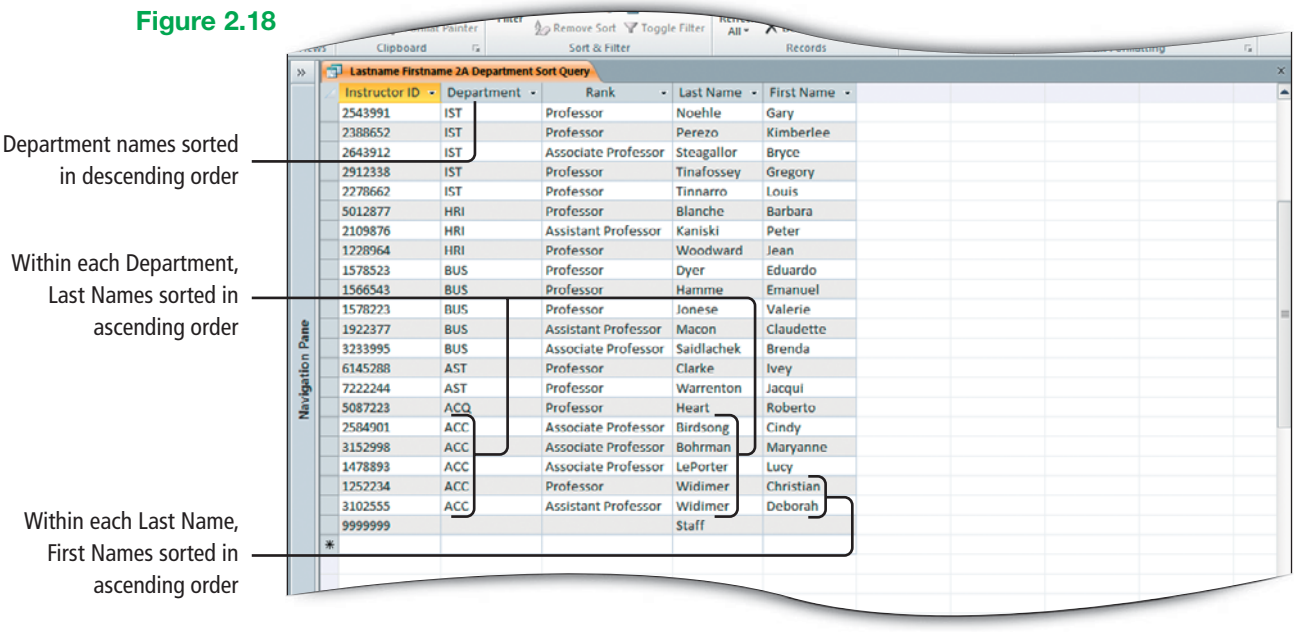
7 Switch to **Design** view. In the **Sort** row, click in the **Department** field, click the **Sort arrow**, and then click **Descending**. **Run** the query; if necessary, scroll down to display the last records, and then compare your screen with Figure 2.18.

In Design view, fields with a Sort designation are sorted from left to right. That is, the sorted field on the left becomes the outermost sort field, and the sorted field on the right becomes the innermost sort field.

Thus, the records are sorted first in descending alphabetical order by the Department field—the leftmost sort field. Then, within each same department name field, the Last Names are sorted in ascending alphabetical order. And, finally, within each same last name field, the First Names are sorted in ascending alphabetical order.

If you run a query and the sorted results are not what you intended, be sure that the fields are displayed from left to right according to the groupings that you desire.

Figure 2.18



8 Display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** ✕ the query. In the message box, click **Yes** to save the changes to the query design.

More Knowledge | Sorting

If you add a sort order to the *design* of a query, it remains as a permanent part of the query design. If you use the sort buttons in the Datasheet view, they will override the sort order of the query design, and can be saved as part of the query. A sort order designated in Datasheet view does not display in the Sort row of the query design grid.

Objective 7 | Specify Criteria in a Query

Queries locate information in a database based on ***criteria*** that you specify as part of the query. Criteria are conditions that identify the specific records for which you are looking.

Criteria enable you to ask a more specific question; therefore, you will get a more specific result. For example, if you want to find out how many instructors are in the IST department, limit the results to that specific department, and then only the records that match the specified department will display.

Activity 2.11 | Specifying Text Criteria in a Query

In this activity, you will assist Dean Judkins by creating a query to answer the question *How many instructors are in the IST Department?*

1 Be sure that all objects are closed and that the **Navigation Pane** is closed. Click the **Create tab**, and then in the **Queries group**, click the **Query Design** button.

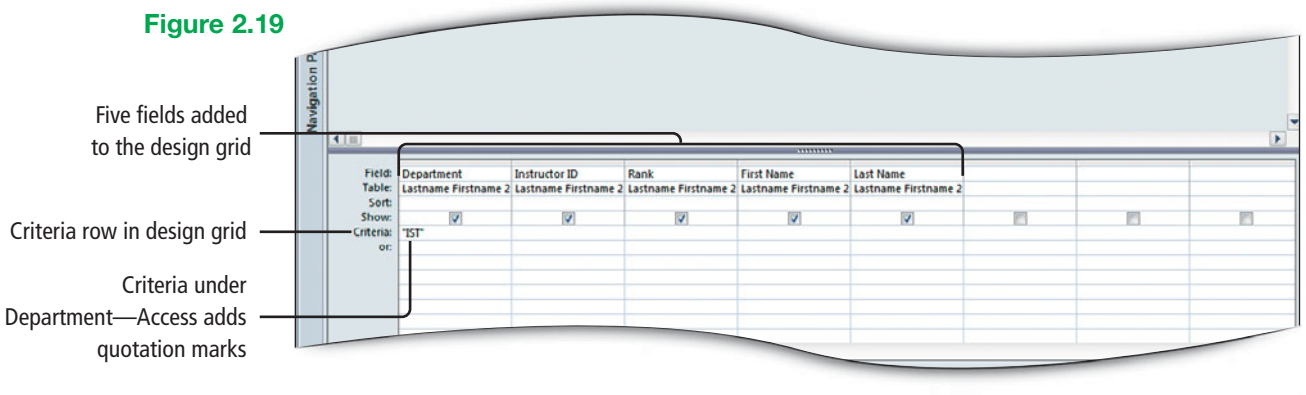
2 In the **Show Table** dialog box, **Add** the **2A Instructors** table to the table area, and then **Close** the **Show Table** dialog box.

3 Expand the field list to display all of the fields and the table name. Add the following fields to the design grid in the order given: **Department**, **Instructor ID**, **Rank**, **First Name**, and **Last Name**.

4 In the **Criteria** row of the design grid, click in the **Department** field, type **IST** and then press Enter. Compare your screen with Figure 2.19.

Access places quotation marks around the criteria to indicate that this is a ***text string***—a sequence of characters. Use the Criteria row to specify the criteria that will limit the results of the query to your exact specifications. The criteria is not case sensitive; so you can type *ist* instead of IST.

Figure 2.19



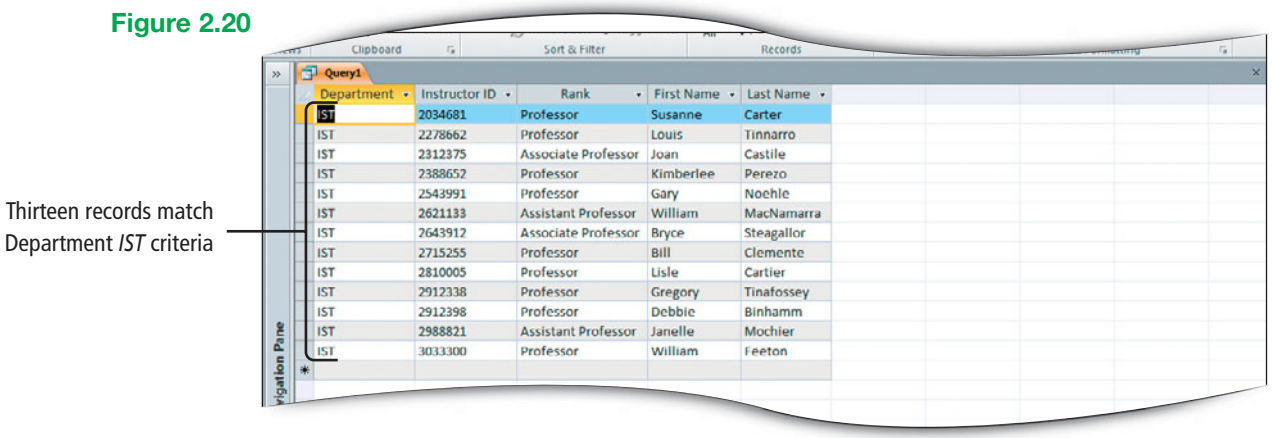
Note | Pressing Enter After Adding Criteria

If you press Enter or click in another column or row in the query design grid after you have added your criteria, you can see how Access alters the criteria so it can interpret what you have typed. Sometimes, there is no change, such as when you add criteria to a number or currency field. Other times, Access may capitalize a letter or add quotation marks or other symbols to clarify the criteria. Whether or not you press Enter after adding criteria has no effect on the query results. It is used here to help you see how the program behaves.

5 **Run** the query, and then compare your screen with Figure 2.20.

Thirteen records display that meet the specified criteria—records that have *IST* in the Department field.

Figure 2.20



Alert! | Do Your Query Results Differ?

If you mistype the criteria, or enter it under the wrong field, or make some other error, the result will display no records. This indicates that there are no records in the table that match the criteria as you entered it. If this occurs, return to Design view and re-examine the query design. Verify that the criteria is typed in the Criteria row, under the correct field, and without typing errors. Then run the query again.

6 **Save** the query as **Lastname Firstname 2A IST Query** and then display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

7 **Close** the query, **Open** the **Navigation Pane**, and then notice that the **2A IST Query** object displays under the **2A Instructors** table—its data source.

Recall that queries in the Navigation Pane display an icon of two overlapping tables.

Activity 2.12 | Specifying Criteria Using a Field Not Displayed in the Query Results

So far, all of the fields that you included in the query design have also been included in the query results. It is not required to have every field in the query display in the results. In this activity, you will create a query to answer the question, *Which instructors have a rank of Professor?*

1 **Close** the **Navigation Pane**. Click the **Create tab**, and then in the **Queries group**, click the **Query Design** button.

2 From the **Show Table** dialog box, **Add** the **2A Instructors** table to the table area, and then **Close** the dialog box. Expand the field list.

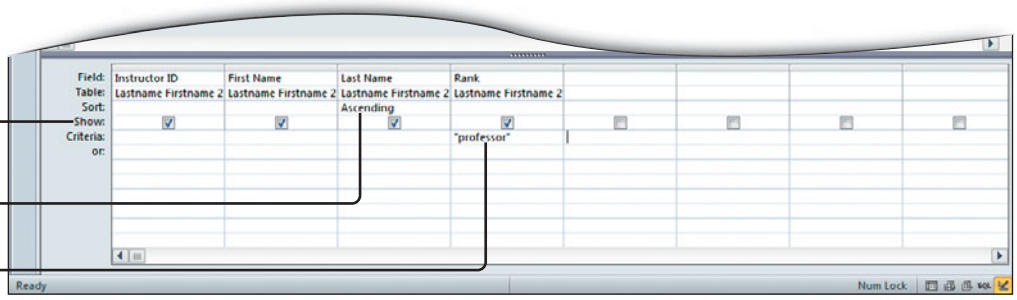
3 Add the following fields, in the order given, to the design grid: **Instructor ID**, **First Name**, **Last Name**, and **Rank**.

4 In the **Sort** row, in the **Last Name** field, click the **Sort arrow**; click **Ascending**.

5 In the **Criteria** row, click in the **Rank** field, type **professor** and then press Enter. Compare your screen with Figure 2.21.

Recall that criteria is not case sensitive. As you start typing *professor*, a list of functions display, from which you can select if including a function in your criteria. When you press Enter, the insertion point moves to the next criteria box and quotation marks are added around the text string that you entered.

Figure 2.21



Show row; check boxes selected for every field

Last Name field sorted in Ascending order

Criteria for Rank field

6 In the design grid, in the **Show** row, notice that the check box is selected for every field. **Run** the query to view the query results.

Nineteen records meet the criteria. In the Rank column each record displays *Professor*, and the records are sorted in ascending alphabetical order by the Last Name field.

7 Switch to **Design** view. In the design grid, under **Rank**, in the **Show** row, click to clear the check box.

Because it is repetitive and not particularly useful to have *Professor* display for each record in the query results, clear this check box so that the field does not display. However, you should run the query before clearing the Show check box to be sure that the correct records display.

8 **Run** the query, and then notice that the *Rank* field does not display.

The query results display the same 19 records, but the *Rank* field does not display. Although the Rank field is still included in the query criteria for the purpose of identifying specific records, it is not necessary to display the field in the results. When appropriate, clear the Show check box to avoid cluttering the query results with data that is not useful.

9 **Save** the query as **Lastname Firstname 2A Professor Rank Query** and then display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the query.

Activity 2.13 | Using *Is Null* Criteria to Find Empty Fields

Sometimes you must locate records where data is *missing*. You can locate such records by using ***Is Null***—empty—as the criteria in a field. Additionally, you can display only the records where a value *has* been entered in a field by using ***Is Not Null*** as the criteria, which will exclude records where the specified field is empty. In this activity, you will design a query to find out *Which scheduled courses have no credits listed?*

1 Click the **Create tab**. In the **Queries group**, click the **Query Design** button. Add the **2A Schedule** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list.

2 Add the following fields to the design grid in the order given: **Subject**, **Catalog#**, **Section**, **Course Name**, and **Credits**.

3 In the **Criteria** row, click in the **Credits** field, type **is null** and then press Enter.

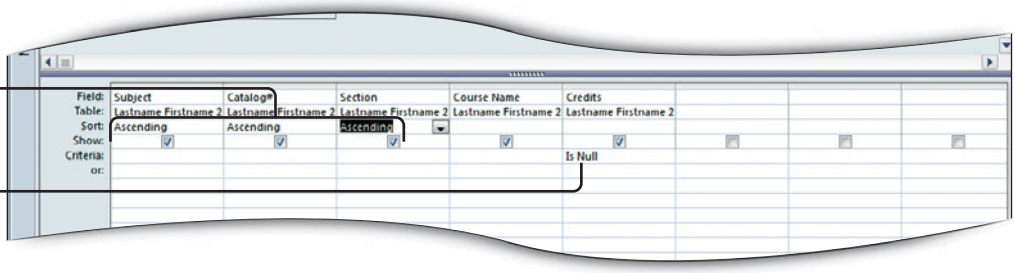
Access capitalizes *is null*. The criteria *Is Null* examines the field and looks for records that do *not* have any values entered in the Credits field.

4 In the **Sort** row, click in the **Subject** field, click the **Sort arrow**, and then click **Ascending**. **Sort** the **Catalog#** field in **Ascending** order, and then **Sort** the **Section** field in **Ascending** order. Compare your screen with Figure 2.22.

Figure 2.22

Three fields sorted in ascending alphabetical order

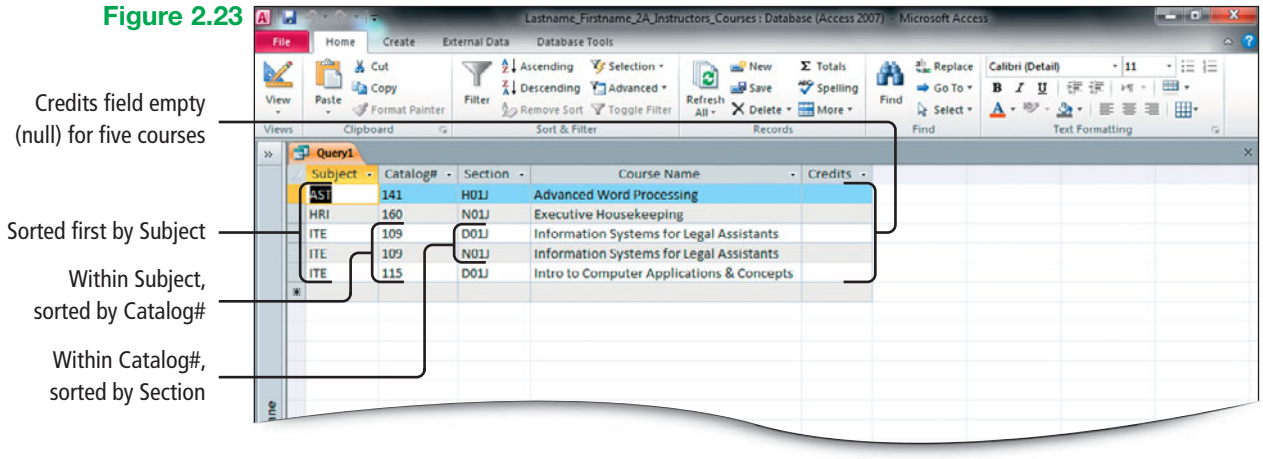
Is Null criteria in Credits field



5 **Run** the query, and then compare your screen with Figure 2.23.

Five scheduled courses do not have credits listed—the Credits field is empty. The records are sorted in ascending order first by the Subject field, then by the Catalog # field, and then by the Section. Using the information displayed in the query results, a course scheduler can more easily locate the records in the table to enter the credits.

Figure 2.23



6 **Save** the query as **Lastname Firstname 2A No Credits Query** and then display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

7 **Close** the query. **Open** the **Navigation Pane**, and then notice that the **2A No Credits Query** object displays under the **2A Schedule** table object, which is the query's data source.

8 From **Backstage** view, click **Close Database**, and then click **Exit** to close the Access program. As directed by your instructor, submit your database and the eight paper or electronic printouts—relationship report, sorted table, and six queries—that are the results of this project.

End **You have completed Project 2A**

Project 2B Athletic Scholarships Database

myitlab Project 2B Training

Project Activities

In Activities 2.14 through 2.26, you will assist Randy Shavrain, Athletic Director for Capital Cities Community College, in developing and querying his Athletic Scholarships database. Your results will look similar to Figure 2.24.

Project Files

For Project 2B, you will need the following files:

a02B_Athletes_Scholarships
a02B_Athletes (Excel file)

You will save your database as:

Lastname_Firstname_2B_Athletic_Scholarships

Project Results

Figure 2.24
Project 2B Athletic Scholarships

Objective 8 | Specify Numeric Criteria in a Query

Criteria can be set for fields containing numeric data. When you design your table, set the appropriate data type for fields that will contain numbers, currency, or dates so that mathematical calculations can be performed.

Activity 2.14 | Opening an Existing Database and Importing a Spreadsheet

In this activity, you will open, rename, and save an existing database, and then import an Excel spreadsheet that Mr. Shavrain wants to bring into Access as a new table.

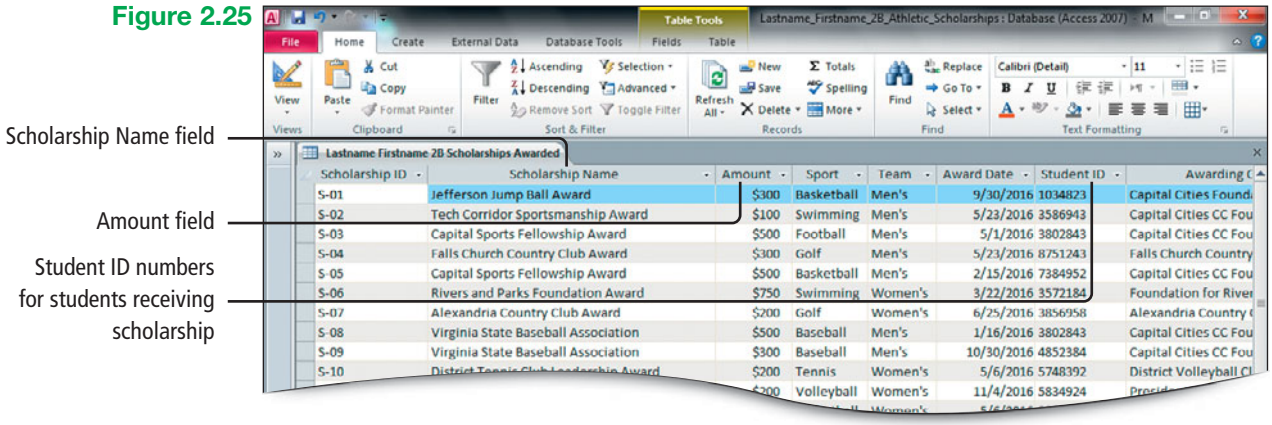
1. **Start** Access. In **Backstage** view, click **Open**. From your student files, open **a02B_Athletes_Scholarships**.

2. From **Backstage** view, click **Save Database As**. In the **Save As** dialog box, navigate to your **Access Chapter 2** folder, and then in the **File name** box, type **Lastname_Firstname_2B_Athletic_Scholarships** and then press Enter.

3. On the **Message Bar**, click the **Enable Content** button. In the **Navigation Pane**, **Rename 2B Scholarships Awarded** to **Lastname Firstname 2B Scholarships Awarded**, and then double-click to open the table. **Close** « the **Navigation Pane**, and then examine the data in the table. Compare your screen with Figure 2.25.

   In this table, Mr. Shavrain tracks the names and amounts of scholarships awarded to student athletes. Students are identified only by their Student ID numbers, and the primary key is the Scholarship ID field.

Figure 2.25



4. **Close** × the table. On the Ribbon, click the **External Data tab**, and then in the **Import & Link group**, click the **Excel** button. In the **Get External Data – Excel Spreadsheet** dialog box, to the right of the **File name** box, click **Browse**.

5. In the **File Open** dialog box, navigate to your student data files, and then double-click **a02B_Athletes**. Be sure that the **Import the source data into a new table in the current database** option button is selected, and then click **OK**.

   The Import Spreadsheet Wizard opens and displays the spreadsheet data.

6. Click **Next**. In the upper left portion of the **Import Spreadsheet Wizard** dialog box, select the **First Row Contains Column Headings** check box. Click **Next**, and then click **Next** again.

7 In the upper portion of the Wizard, click the **Choose my own primary key** option button, and then be sure that **Student ID** displays.

In the new table, Student ID will be the primary key. No two students have the same Student ID.

8 Click **Next**. In the **Import to Table** box, type **Lastname Firstname 2B Athletes** and then click **Finish**. In the **Get External Data – Excel Spreadsheet** Wizard, click **Close**, and then **Open** » the **Navigation Pane**. Widen the **Navigation Pane** so that the table names display fully.

9 **Open** the new **2B Athletes** table, and then on the **Home tab**, switch to **Design View**.

10 For the **Student ID** field, click in the **Data Type** box, click the **arrow**, and then click **Text**. For the **ZIP/Postal Code** field, change the **Data Type** to **Text**, and then set the **Field Size** to **5** Click in the **State/Region** field, set the **Field Size** to **2** and then switch back to **Datasheet View**, saving the changes.

Recall that numeric data that will not be used in any calculations, such as the Student ID, should have a Data Type of *Text*.

11 In the message box, click **Yes**—no data will be lost. **Close** « the **Navigation Pane**. Take a moment to review the imported data. Using the **Select All** button, apply **Best Fit** to all of the fields. Click in any field to cancel the selection, **Save** the table, and then **Close** × the table.

Activity 2.15 | Creating Table Relationships

In this activity, you will create a one-to-many relationship between the 2B Athletes table and the 2B Scholarships Awarded table by using the common field—*Student ID*.

1 Click the **Database Tools tab**, and then in the **Relationships group**, click the **Relationships** button.

2 In the **Show Table** dialog box, **Add** the **2B Athletes** table, and then **Add** the **2B Scholarships Awarded** table to the table area. **Close** the **Show Table** dialog box.

3 Move and resize the two field lists to display all of the fields and the entire table name, and then position the field lists so that there is approximately one inch of space between the two field lists.

Resizing and repositioning the field lists is not required, but doing so makes it easier for you to view the field lists and the join line when creating relationships.

4 In the **2B Athletes** field list, point to the **Student ID** field. Hold down the left mouse button, drag into the **2B Scholarships Awarded** field list on top of the **Student ID** field, and then release the mouse button to display the **Edit Relationships** dialog box.

5 Point to the title bar of the **Edit Relationships** dialog box, and then drag it below the two field lists. In the **Edit Relationships** dialog box, be sure that **Student ID** is displayed as the common field for both tables.

The two tables relate in a *one-to-many* relationship—*one* athlete can have *many* scholarships. The common field between the two tables is the Student ID field. In the 2B Athletes table, Student ID is the primary key. In the 2B Scholarships Awarded table, Student ID is the foreign key.

6 In the **Edit Relationships** dialog box, select the **Enforce Referential Integrity** check box. Click **Create**, and then compare your screen with Figure 2.26.

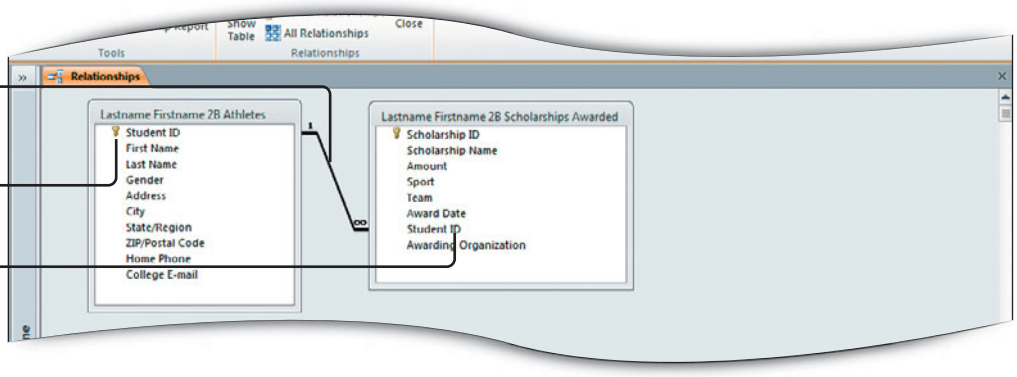
The one-to-many relationship is established. The *1* and ∞ indicate that referential integrity is enforced, which ensures that a scholarship cannot be awarded to a student whose Student ID is not in the 2B Athletes table. Similarly, you cannot delete a student athlete from the 2B Athletes table if there is a scholarship listed for that student in the 2B Scholarships Awarded table.

Figure 2.26

Join line—symbols indicate referential integrity enforced

Primary key in the *one* table

Foreign key in the *many* table



7 On the **Design tab**, in the **Tools group**, click the **Relationship Report** button. Create a paper or electronic printout if instructed to do so.

8 **Save** the report as **Lastname Firstname 2B Relationships** and then click **OK**. **Close** the report, and then **Close** the **Relationships** window.

9 **Open** the **Navigation Pane**, open the **2B Athletes** table, and then **Close** the **Navigation Pane**. On the left side of the table, in the first record, click the **plus sign** (+) to display the subdatasheet for the record.

In the first record, for *Joel Barthmaier*, one related record exists in the 2B Scholarships Awarded table. The related record displays because you created a relationship between the two tables using Student ID as the common field.

10 **Close** the **2B Athletes** table.

Activity 2.16 | Specifying Numeric Criteria in a Query

Mr. Shavrain wants to know *Which scholarships are in the amount of $300, and for which sports?* In this activity, you will specify criteria in the query so that only the records of scholarships in the amount of $300 display.

1 Click the **Create tab**. In the **Queries group**, click the **Query Design** button.

2 In the **Show Table** dialog box, **Add** the **2B Scholarships Awarded** table to the table area, and then **Close** the **Show Table** dialog box. Expand the field list to display all of the fields and the entire table name.

3 Add the following fields to the design grid in the order given: **Scholarship Name**, **Sport**, and **Amount**.

4 In the **Sort** row, click in the **Sport** field. Click the **Sort arrow**, and then click **Ascending**.

5 In the **Criteria** row, click in the **Amount** field, type **300** and then press Enter. Compare your screen with Figure 2.27.

When entering currency values as criteria, do not type the dollar sign. Include a decimal point only if you are looking for a specific amount that includes cents—for example 300.49. Access does not insert quotation marks around the criteria because the field's data type is Number.

Figure 2.27

Numeric criteria—no quotation marks

Sort in ascending order by *Sport*

6 On the **Design tab**, in the **Results group**, click the **Run** button to view the results.

Five scholarships were awarded in the exact amount of $300. In the navigation area, *1 of 5* displays to indicate the number of records that match the criteria.

7 On the **Home tab**, in the **Views group**, click the **View** button to switch to **Design** view.

Activity 2.17 | Using Comparison Operators

Comparison operators are symbols that evaluate each field value to determine if it is the same (=), greater than (>), less than (<), or in between a range of values as specified by the criteria.

If no comparison operator is specified, equal (=) is assumed. For example, in the previous activity, you created a query to display only records where the *Amount* is 300. The comparison operator of = was assumed, and Access displayed only records that had values equal to 300.

1 Be sure your query is displayed in **Design** view. In the **Criteria** row, click in the **Amount** field, delete the existing criteria, type **>300** and then press Enter.

2 On the **Design tab**, in the **Results group**, click the **Run** button.

Fourteen records have an Amount that is greater than $300. The results show the records for which the Amount is *greater than* $300, but do not display amounts that are *equal to* $300.

3 Switch to **Design** view. In the **Criteria** row, under **Amount**, delete the existing criteria. Type **<300** and then press Enter. **Run** the query.

Eleven records display and each has an Amount less than $300. The results show the records for which the Amount is *less than* $300, but does not include amounts that are *equal to* $300.

4 Switch to **Design** view. In the **Criteria** row, click in the **Amount** field, delete the existing criteria, type **>=300** and then press Enter.

5 **Run** the query, and then compare your screen with Figure 2.28.

Nineteen records display, including the records for scholarships in the exact amount of $300. The records include scholarships *greater than* or *equal to* $300. In this manner, comparison operators can be combined. This query answers the question, *Which scholarships have been awarded in the amount of $300 or more, and for which sports, with the Sport names in alphabetical order?*

Figure 2.28

Nineteen records with a scholarship amount of $300 or more

Query1

Scholarship Name	Sport	Amount
Virginia State Baseball Association	Baseball	$500
Virginia State Baseball Association	Baseball	$300
Roundball Academic Achievement Award	Basketball	$500
Capital Sports Fellowship Award	Basketball	$500
Hoops National Winner Award	Basketball	$400
Hoops Fellowship Award	Basketball	$500
Hoops Sports Award	Basketball	$500
DC Sports Award	Basketball	$500
Virginia Sportswomen Foundation Award	Basketball	$300
Jefferson Jump Ball Award	Basketball	$300
Capital Sports Fellowship Award	Football	$500
DC Science Achievement Award	Football	$750
The Touchdown Alumni Association Award	Football	$600
Falls Church Country Club Award	Golf	$300
Dolphin Club Award	Swimming	$300
Rivers and Parks Foundation Award	Swimming	$750
Capital Cities Country Club Foundation Award	Swimming	$400
Craig Fresch Foundation Award	Tennis	$400
Joseph Ingram Memorial Award	Tennis	$400

6 **Save** the query as **Lastname Firstname 2B $300 or More Query** and then display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

7 **Close** the query. **Open** the **Navigation Pane**, and notice that the new query displays under the table from which it retrieved the records—*2B Scholarships Awarded*.

Activity 2.18 | Using the Between … And Comparison Operator

The ***Between … And operator*** is a comparison operator that looks for values within a range. It is useful when you need to locate records that are within a range of dates; for example, scholarships awarded between August 1 and September 30. In this activity, you will create a new query from an existing query, and then add criteria to look for values within a range of dates. The query will answer the question *Which scholarships were awarded between August 1 and September 30?*

1 On the **Navigation Pane**, click the **2B $300 or More Query** object to select it. Display **Backstage** view and click **Save Object As**. In the **Save As** dialog box, type **Lastname Firstname 2B Awards Aug-Sep Query** and then click **OK**.

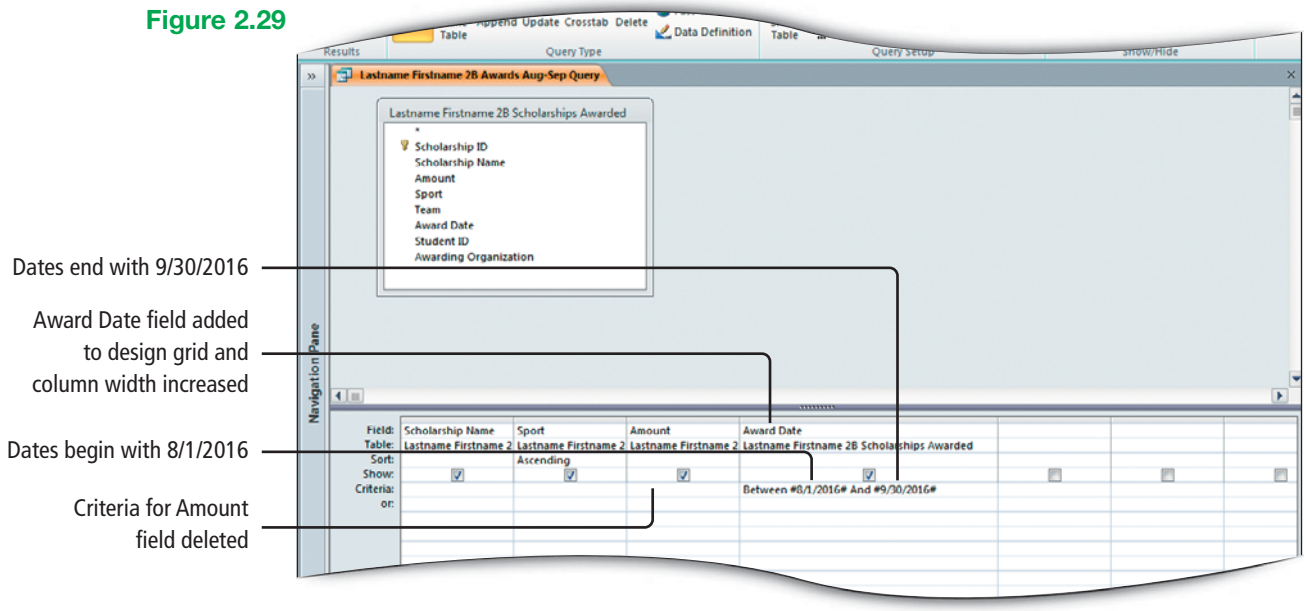
2 Click the **Home tab**. Open the **2B Awards Aug-Sep Query** object, **Close** the **Navigation Pane**, and then switch to **Design** view. From the **2B Scholarships Awarded** field list, add the **Award Date** as the fourth field in the design grid.

3 In the **Criteria** row, click in the **Amount** field, and then delete the existing criteria so that the query is not restricted by amount. In the **Criteria** row, click in the **Award Date** field, type **between 8/1/16 and 9/30/16** and then press Enter.

4 In the selection bar of the design grid, point to the right edge of the **Award Date** column to display the pointer, and then double-click. Compare your screen with Figure 2.29.

The width of the Award Date column is increased to fit the longest entry, enabling you to see all of the criteria. Access places pound signs (#) around dates and capitalizes *between* and *and*. This criteria instructs Access to look for values in the Award Date field that begin with 8/1/16 and end with 9/30/16. Both the beginning and ending dates will be included in the query results.

Figure 2.29



5 **Run** the query, and notice that three scholarships were awarded between 08/1/16 and 9/30/16.

6 Display the query in **Print Preview**, create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

7 **Close** the query. In the message box, click **Yes** to save the changes to the query design. **Open** the **Navigation Pane**, and notice that the new query displays under the table that is its data source—*2B Scholarships Awarded*.

Objective 9 | Use Compound Criteria

You can specify more than one condition—criteria—in a query; this is called ***compound criteria***. Compound criteria use AND and OR ***logical operators***. Logical operators enable you to enter criteria for the same field or different fields.

Activity 2.19 | Using AND Criteria in a Query

Compound criteria use an ***AND condition*** to display records in the query results that meet all parts of the specified criteria. In this activity, you will help Mr. Shavrain answer the question *Which scholarships over $500 were awarded for Football?*

1 **Close** the **Navigation Pane.** On the Ribbon, click the **Create tab**. In the **Queries group**, click the **Query Design** button. **Add** the **2B Scholarships Awarded** table to the table area. **Close** the **Show Table** dialog box, and then expand the field list.

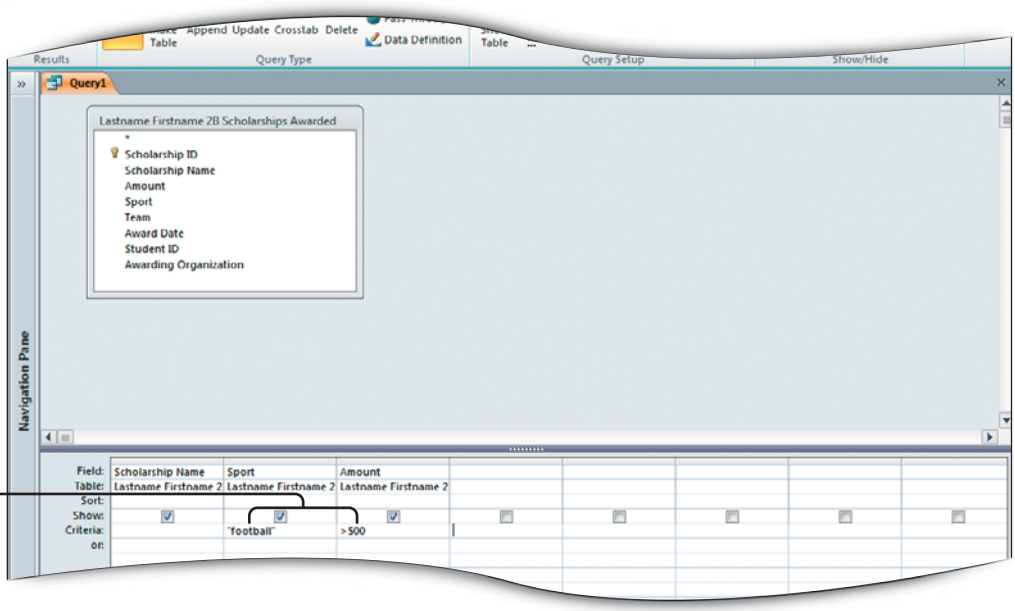
2 Add the following fields to the design grid in the order given: **Scholarship Name**, **Sport**, and **Amount**.

3 In the **Criteria** row, click in the **Sport** field, type **football** and then press Enter.

4 In the **Criteria** row, in the **Amount** field, type **>500** and then press Enter. Compare your screen with Figure 2.30.

You create the AND condition by placing the criteria for both fields on the same line in the Criteria row. The results will display only records that contain *Football* AND an amount greater than *$500*.

Figure 2.30



Criteria specified for Sport *AND* Amount

Access | Chapter 2

5 On the **Design tab**, in the **Results group**, click the **Run** button.

Two records display that match both conditions—Football in the Sport field *and* greater than $500 in the Amount field.

6 **Save** the query as **Lastname Firstname 2B Football and Over $500 Query** and then click **OK** or press Enter. **Close** the query.

7 **Open** the **Navigation Pane**, and then click one time to select the **2B Football and Over $500 Query** object. Display the query in **Print Preview**, create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

You can print any selected object from the Navigation Pane—the object does not have to be open to print.

8 **Close** the **Navigation Pane**.

Activity 2.20 | Using OR Criteria in a Query

Use the ***OR condition*** to specify multiple criteria for a single field, or multiple criteria for different fields when you want to display the records that meet any of the conditions. In this activity, you will help Mr. Shavrain answer the question *Which scholarships over $400 were awarded in the sports of Baseball or Swimming, and what is the award date of each?*

1 Click the **Create tab**. In the **Queries group**, click the **Query Design** button.

2 **Add** the **2B Scholarships Awarded** table. **Close** the dialog box, expand the field list, and then add the following four fields to the design grid in the order given: **Scholarship Name**, **Sport**, **Amount**, and **Award Date**.

3 In the **Criteria** row, click in the **Sport** field, and then type **baseball**

4 In the design grid, on the **or** row, click in the **Sport** field, type **swimming** and then press Enter. **Run** the query.

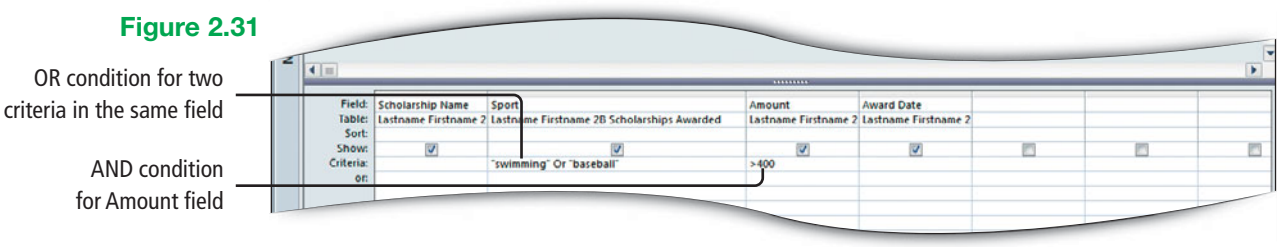
The query results display seven scholarship records where the Sport is either Baseball *or* Swimming. Use the OR condition to specify multiple criteria for a single field.

5 Switch to **Design** view. In the **or** row, under **Sport**, delete *swimming*. In the **Criteria** row, under **Sport**, delete *baseball*. Type **swimming or baseball** and then in the **Criteria** row, click in the **Amount** field. Type **>400** and then press Enter. Increase the width of the **Sport** column. Compare your screen with Figure 2.31.

This is an alternative way to use the OR compound operator in the Sport field. Because criteria is entered for two different fields, Access selects the records that are Baseball *or* Swimming *and* that have a scholarship awarded in an amount greater than $400.

If you enter swimming on the Criteria row and baseball on the or row, then you must enter >400 on both the Criteria row and the or row so that the correct records display when the query runs.

Figure 2.31



6 **Run** the query to display the two records that match the conditions.

7 **Close** the query. In the message box, click **Yes** to save changes to the query. In the **Save As** dialog box, type **Lastname Firstname 2B Swimming or Baseball Over $400 Query** and then click **OK**.

8 **Open** the **Navigation Pane**, increase the width of the **Navigation Pane** to display the full name of all objects, and then click one time to select the **2B Swimming or Baseball Over $400 Query** object. Display the query in **Print Preview**, create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the **Navigation Pane**.

Objective 10 | Create a Query Based on More Than One Table

In a relational database, you can retrieve information from more than one table. Recall that a table in a relational database contains all of the records about a single topic. Tables are joined by relating the primary key field in one table to a foreign key field in another table. This common field creates a relationship, so you can include data from more than one table in a query.

For example, the Athletes table contains all of the information about the student athletes—name, address, and so on. The Scholarships Awarded table includes the scholarship name, amount, and so on. When an athlete receives a scholarship, only the Student ID field is included with the scholarship to identify who received the scholarship. It is not necessary to include any other data about the athletes in the Scholarships Awarded table; doing so would result in redundant data.

Activity 2.21 | Creating a Query Based on More Than One Table

In this activity, you will create a query that selects records from two tables. This is possible because a relationship has been created between the two tables in the database. The query will answer the questions *What is the name, e-mail address, and phone number of athletes who have received swimming or tennis scholarships, and what is the name and amount of the scholarship?*

1 Click the **Create tab**. In the **Queries group**, click the **Query Design** button. **Add** the **2B Athletes** table and the **2B Scholarships Awarded** table to the table area, and then **Close** the **Show Table** dialog box. Expand the two field lists, and then drag the **2B Scholarships Awarded** field list to the right so that there is approximately one inch of space between the field lists.

The join line displays because you previously created a relationship between the tables. It indicates a *one-to-many* relationship—*one* athlete can have *many* scholarships.

2 From the **2B Athletes** field list, add the following fields to the design grid in the order given: **First Name**, **Last Name**, **College E-mail**, and **Home Phone**.

3 In the **Sort** row, click in the **Last Name** field. Click the **Sort arrow**, and then click **Ascending** to sort the records in alphabetical order by last name.

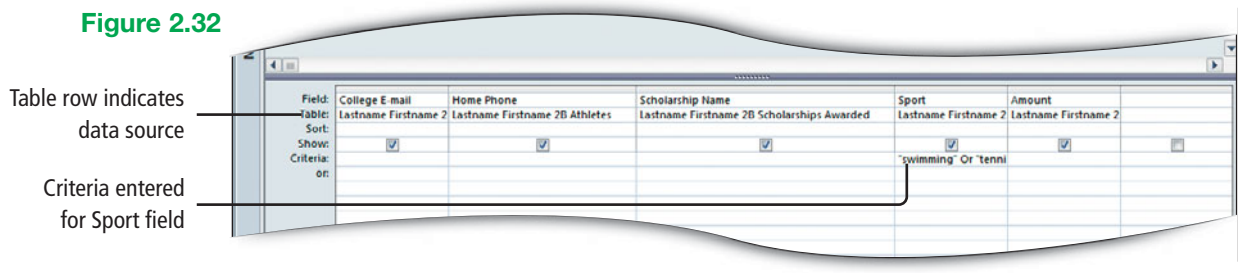
4 From the **2B Scholarships Awarded** field list, add the following fields to the design grid in the order given: **Scholarship Name**, **Sport**, and **Amount**.

5 In the **Criteria** row, click in the **Sport** field. Type **swimming or tennis** and then press Enter.

6 In the design grid, increase the width of the **Home Phone** and **Scholarship Name** columns to display the entire table name on the **Table** row. If necessary, scroll to the right to display the *Home Phone* and *Scholarship Name* fields in the design grid, and then compare your screen with Figure 2.32.

When extracting data from multiple tables, the information on the Table row is helpful, especially when different tables include the same field names, such as address, but different data, such as a student's address or a coach's address.

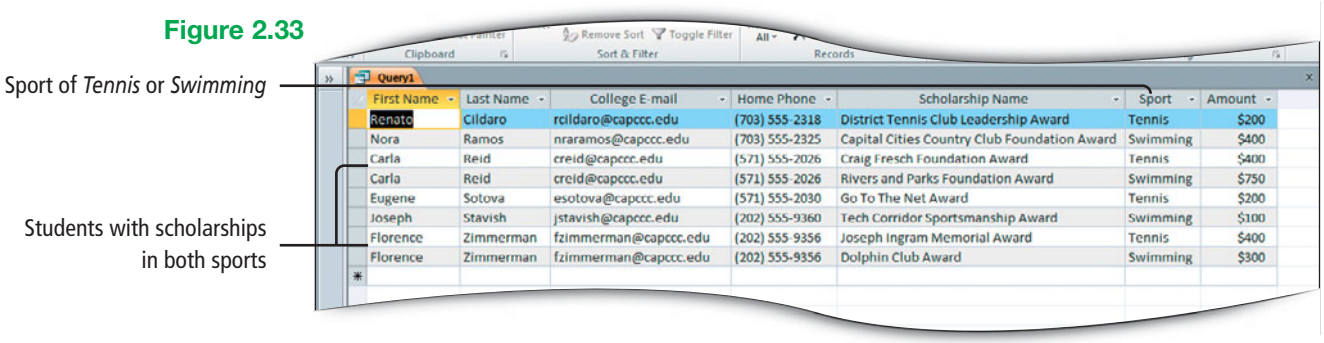
Figure 2.32



7 **Run** the query, and then compare your screen with Figure 2.33.

Information for eight student athletes displays. The First Name and Last Name fields are included in the query results even though the common field—*Student ID*—is *not* included in the query design. Because Student ID is included in both tables, and a one-to-many relationship was created between the tables, the Student ID field is used to select the records in both tables by using one query. Two students—*Carla Reid* and *Florence Zimmerman*—received scholarships in both Swimming and Tennis. Recall that *one* student athlete can receive *many* scholarships.

Figure 2.33

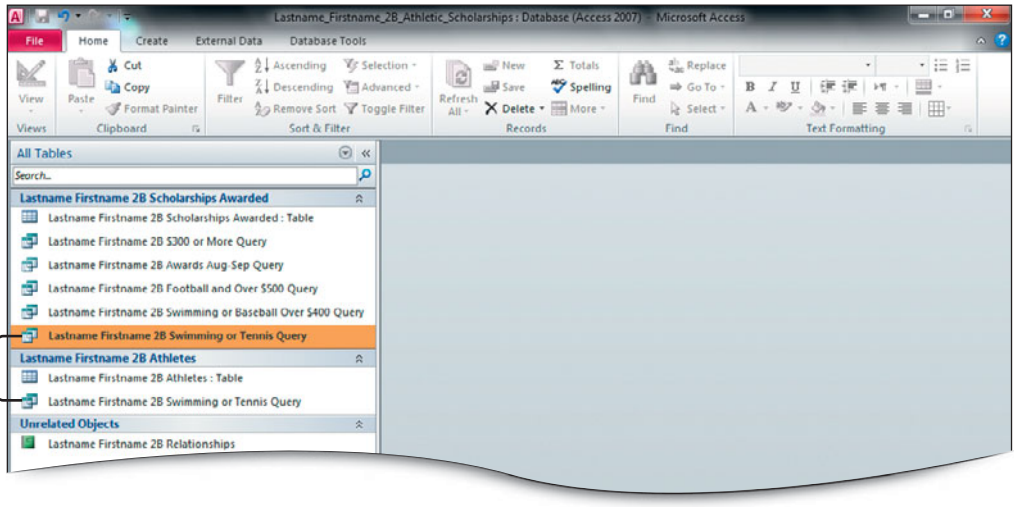


8 Save the query as **Lastname Firstname 2B Swimming or Tennis Query** and then display the query in **Print Preview**. Set the **Margins** to **Normal**, and then change the orientation to **Landscape**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

9 **Close** the query, **Open** the **Navigation Pane**, and then compare your screen with Figure 2.34.

Your new query—*2B Swimming or Tennis Query*—displays under both tables from which it retrieved records.

Figure 2.34



Objective 11 | Use Wildcards in a Query

Wildcard characters serve as a placeholder for one or more unknown characters in the criteria. When you are unsure of the particular character or set of characters to include in criteria, use wildcard characters in place of the characters.

Activity 2.22 | Using a Wildcard in a Query

Use the asterisk (*) to represent one or more characters. For example, if you use the * wildcard in the criteria Fo*, the results will display Foster, Forrester, Forrest, Fossil, or any word beginning with *Fo*. In this activity, you will use the asterisk (*) wildcard in the criteria row to answer the question *Which athletes received scholarships from local Rotary Clubs, country clubs, or foundations?*

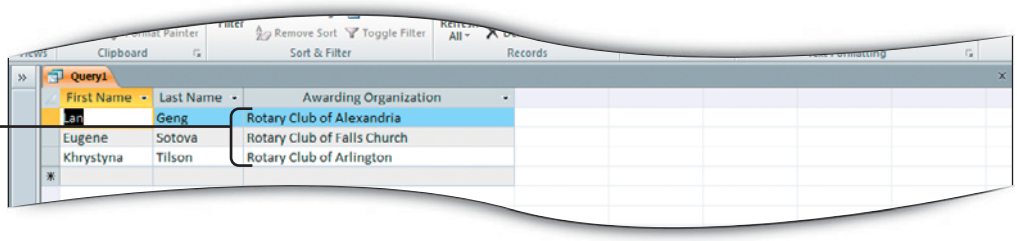
1. **Close** « the **Navigation Pane**. On the Ribbon, click the **Create tab**. In the **Queries group**, click the **Query Design** button.
2. **Add** both tables to the table area, **Close** the **Show Table** dialog box, and then expand the field lists.
3. Add the following fields to the design grid in the order given: from the **2B Athletes** table, **First Name** and **Last Name**; from the **2B Scholarships Awarded** table, **Awarding Organization**.
4. In the **Sort** row, click in the **Last Name** field. Click the **arrow**, and then click **Ascending**.
5. In the **Criteria** row, under **Awarding Organization**, type **rotary*** and then press Enter.

   The wildcard character * is a placeholder to match one or more characters. After pressing Enter, Access adds *Like* to the beginning of the criteria.
6. **Run** the query, and then compare your screen with Figure 2.35.

   Three athletes received scholarships from Rotary Clubs. The results are sorted alphabetically by the Last Name field.

Figure 2.35

Awarding Organization for all records begins with *Rotary*



7 Switch to **Design** view. On the **or** row, under **Awarding Organization**, type ***country club** and then press Enter.

The * can be used at the beginning, middle, or end of the criteria. The position of the * wildcard character determines the location of the unknown characters. Here you will search for records that end in *Country Club*.

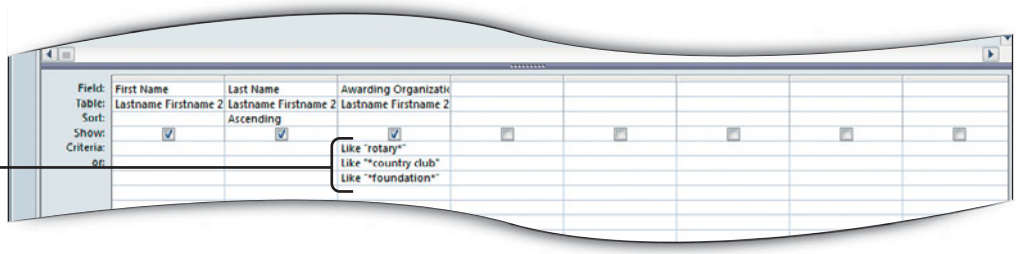
8 **Run** the query to display six records, and notice that three records begin with *Rotary*, and three records end with *Country Club*—sorted alphabetically by Last Name.

9 Switch to **Design** view. Under **Awarding Organization** and under **Like "*country club"**, type ***foundation*** and then press Enter. Compare your screen with Figure 2.36.

The query will also display records that have the word *Foundation* anywhere—beginning, middle, or end—in the field. Three *OR* criteria have been entered for the Awarding Organization field—the query results will display students who have received scholarships from an organization name that begins with Rotary, *or* that ends in County Club, *or* that has Foundation anywhere in the middle of the name.

Figure 2.36

Three variations of * wildcard placement



10 **Run** the query to display 28 records.

Twenty-eight scholarships were from a Country Club, *or* a Rotary Club, *or* a Foundation.

11 **Save** the query as **Lastname Firstname 2B Wildcard Query** and then display the results in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**.

12 **Close** the query, and then **Open** the **Navigation Pane**.

Because the 2B Wildcard Query object retrieved data from two tables, it displays below the 2B Scholarships Awarded table *and* the 2B Athletes table—the data sources.

More Knowledge | Search for a Single Unknown Character by Using the ? Wildcard

The question mark (?) is a wildcard that is used to search for unknown single characters. For each question mark included in criteria, any character can be inserted. For example, if you use *b?d* as a criteria, the query might locate bid, bud, bed, or any three-character word beginning with *b* and ending with *d*. If *b??d* is entered as the criteria, the results could include bind, bend, bard, or any four-character word beginning with *b* and ending with *d*.

Objective 12 | Use Calculated Fields in a Query

Queries can create calculated values that are stored in a ***calculated field***. A calculated field stores the value of a mathematical operation. For example, you can multiply two fields together, such as Total Credit Hours and Tuition Per Credit Hour to get a Total Tuition Due amount for each student without having to include a specific field for this amount in the table, which reduces the size of the database and provides more flexibility.

There are two steps to produce a calculated field in a query. First, name the field that will store the calculated values. Second, write the ***expression***—the formula—that performs the calculation. Each field name used in the calculation must be enclosed within its own pair of square brackets, and the new field name must be followed by a colon (:).

Activity 2.23 | Using Calculated Fields in a Query

For each scholarship received by student athletes, the Capital Cities Community College Alumni Association will donate an amount equal to 50 percent of each scholarship. In this activity, you will create a calculated field to determine the additional amount each scholarship is worth. The query will answer the question *What is the value of each scholarship if the Alumni Association makes a matching 50% donation?*

1. **Close** « the **Navigation Pane**, and then click the **Create tab**. In the **Queries group**, click the **Query Design** button.
2. **Add** the **2B Scholarships Awarded** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list. Add the following fields to the design grid in the order given: **Student ID**, **Scholarship Name**, and **Amount**.
3. In the **Sort** row, click in the **Student ID** field; sort **Ascending**.
4. In the **Field** row, right-click in the first empty column to display a shortcut menu, and then click **Zoom**.

   The Zoom dialog box gives you more working space so that you can see the entire calculation as you type it. The calculation can also be typed directly in the empty Field box in the column.
5. In the **Zoom** dialog box, type **Matching Donation:[Amount]*0.5** and then compare your screen with Figure 2.37.

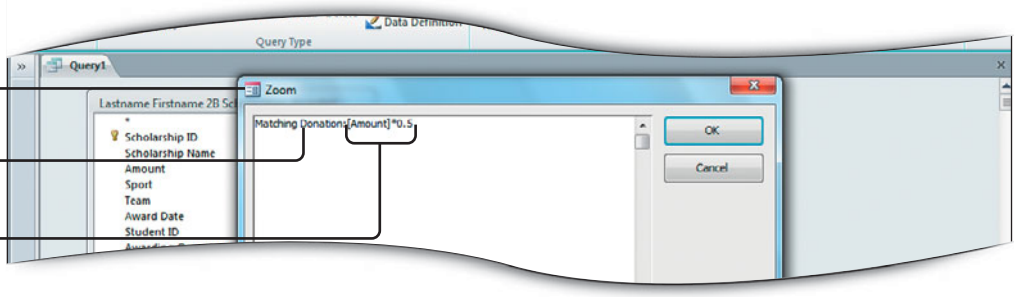
   The first element, *Matching Donation*, is the new field name where the calculated values will display. Following that is a colon (:), which separates the new field name from the expression. *Amount* is enclosed in square brackets because it is an existing field name in the 2B Scholarships Awarded table; it contains the numeric data on which the calculation will be performed. Following the right square bracket is an asterisk (*), which in math calculations signifies multiplication. Finally, the percentage (0.5 or 50%) displays.

Figure 2.37

Zoom dialog box

New field name for calculated value followed by a colon (:)

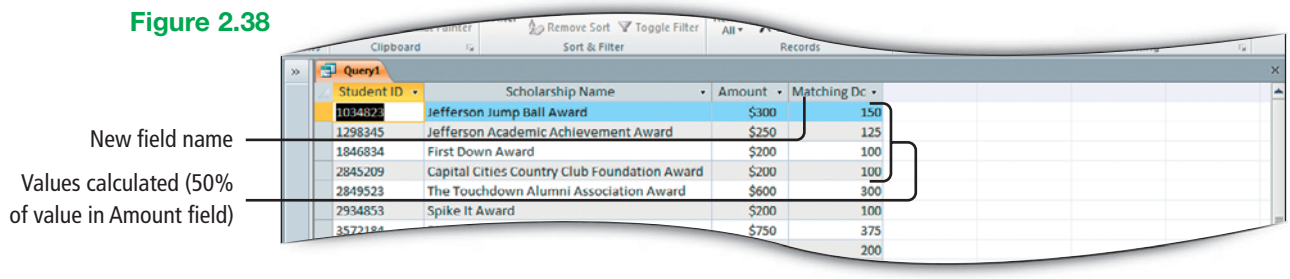
Expression—formula that calculates the value



6 In the **Zoom** dialog box, click **OK**, and then **Run** the query. Compare your screen with Figure 2.38.

The query results display three fields from the 2B Scholarships Awarded table plus a fourth field—*Matching Donation*—in which a calculated value displays. Each calculated value equals the value in the Amount field multiplied by 0.5.

Figure 2.38



Alert! | Does Your Screen Differ?

If your calculations in a query do not work, switch to Design view and carefully check the expression you typed. Spelling or syntax errors prevent calculated fields from working properly.

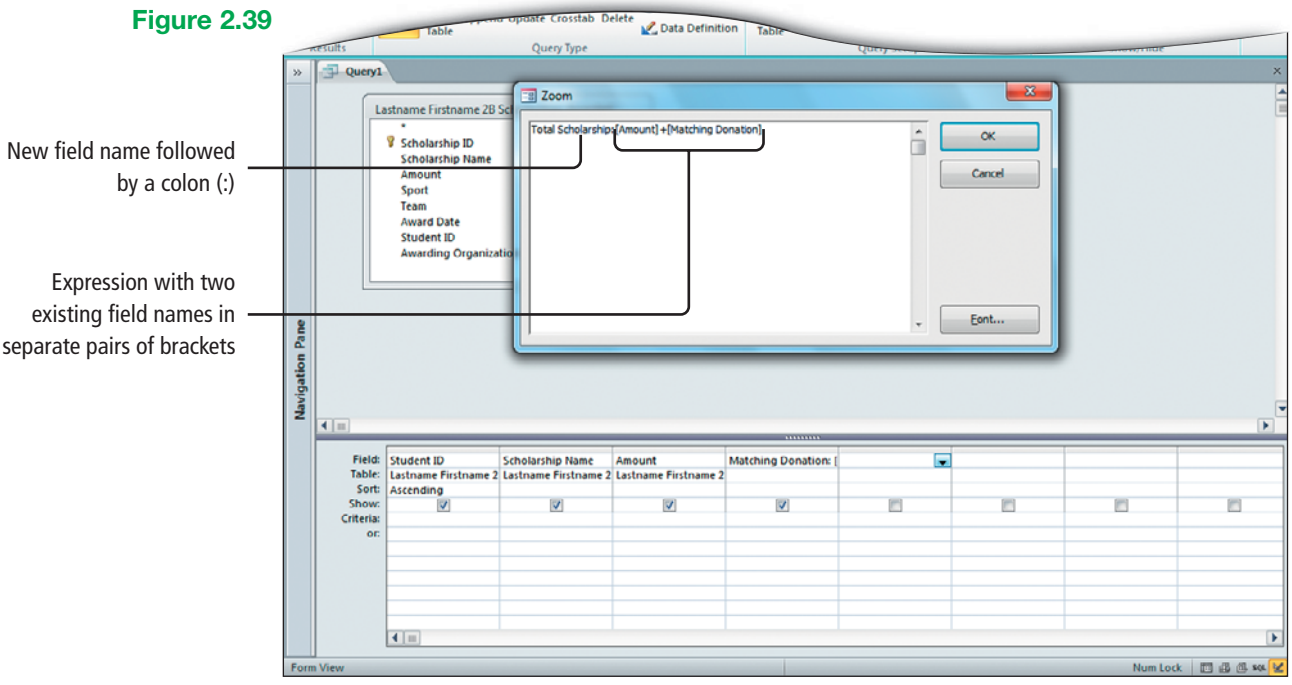
7 Notice the formatting of the **Matching Donation** field—there are no dollar signs, commas, or decimal places; you will adjust this formatting later.

When using a number, such as 0.5, in an expression, the values in the calculated field may not be formatted the same as in the existing field.

8 Switch to **Design** view. In the **Field** row, in the first empty column, right-click, and then click **Zoom**. In the **Zoom** dialog box, type **Total Scholarship:[Amount]+[Matching Donation]** and then compare your screen with Figure 2.39.

Each existing field name—*Amount* and *Matching Donation*—must be enclosed in separate pairs of brackets.

Figure 2.39



9 In the **Zoom** dialog box, click **OK**, and then **Run** the query to view the results.

Total Scholarship is calculated by adding together the Amount field and the Matching Donation field. The Total Scholarship column includes dollar signs, commas, and decimal points, which carried over from the Currency format in the Amount field.

10 Switch to **Design** view. In the **Field** row, click in the **Matching Donation** field box.

Another Way

Right-click the Matching Donation field name, and then click Properties.

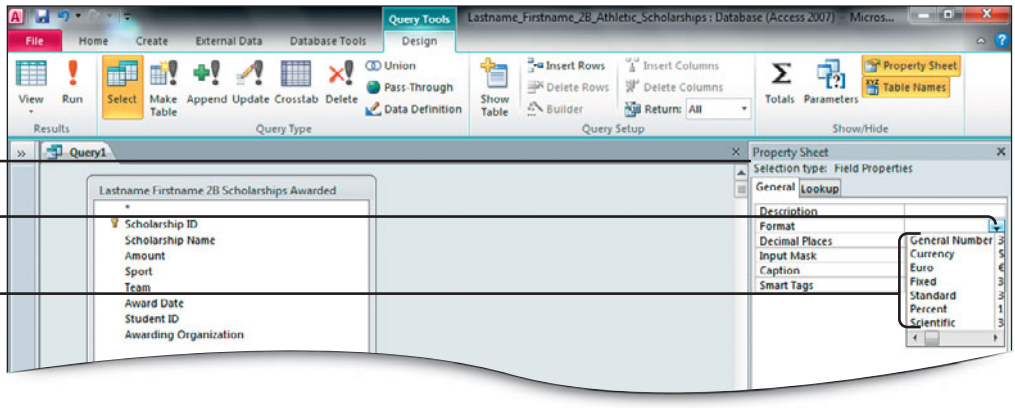
11 On the **Design tab**, in the **Show/Hide group**, click the **Property Sheet** button.

The ***Property Sheet*** displays on the right side of your screen. A Property Sheet is a list of characteristics—properties—for fields in which you can make precise changes to each property associated with the field. The left column displays the Property name, for example, Description. To the right of the Property name is the Property setting box.

12 In the **Property Sheet**, on the **General tab**, click in the **Format** property setting box, and then click the **arrow** that displays. Compare your screen with Figure 2.40.

A list of formats for the Matching Donation field displays.

Figure 2.40



13 In the list, click **Currency**. Click the next property, **Decimal Places**, click the **arrow**, and then click **0**.

14 In the design grid, in the **Field** row, click in the **Total Scholarship** field. On the **Property Sheet**, set the **Format** to **Currency** and the **Decimal Places** to **0**.

15 **Close** the **Property Sheet**, and then **Run** the query. Select all of the columns and apply **Best Fit**.

The Matching Donation and Total Scholarship fields are formatted as Currency with 0 decimal places.

16 **Save** the query as **Lastname Firstname 2B Matching Donations Query** and then display the query results in **Print Preview**. Change the **Orientation** to **Landscape**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the query.

Objective 13 | Calculate Statistics and Group Data in a Query

In Access queries, you can perform statistical calculations on a group of records. Calculations that are performed on a group of records are called ***aggregate functions***.

Activity 2.24 | Using the MIN, MAX, AVG, and SUM Functions in a Query

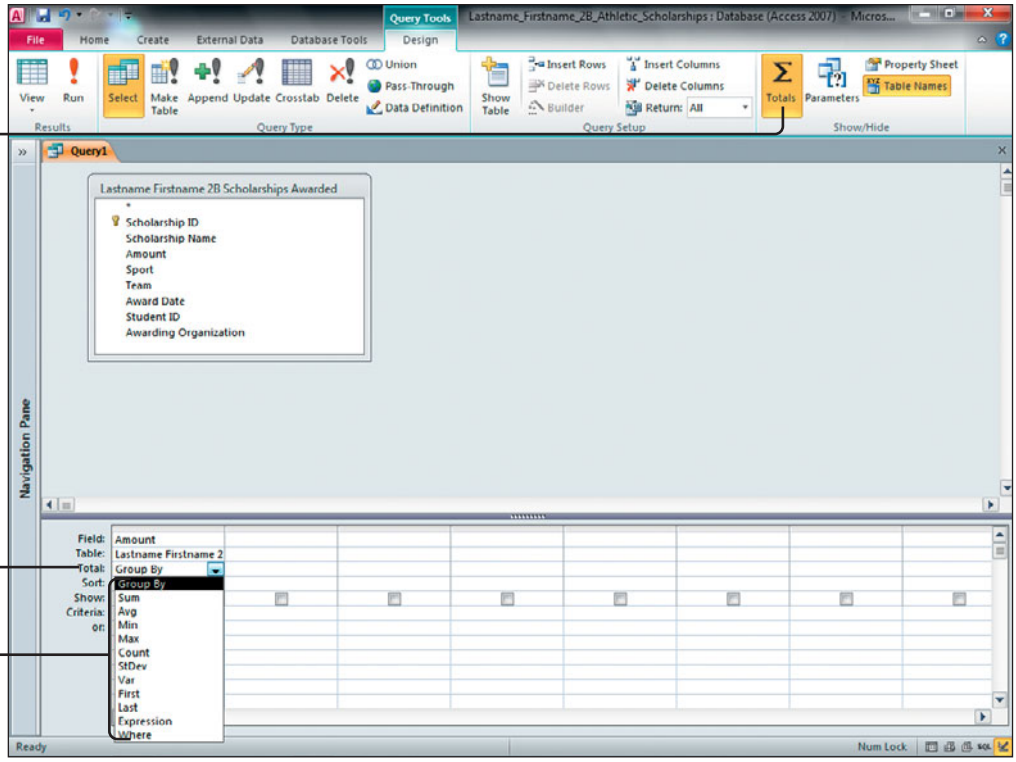
In this activity, you will use the minimum, maximum, average, and sum functions in a query to examine the amounts of scholarships awarded. The last query will answer the question *What is the total dollar amount of all scholarships awarded?*

1. On the **Create tab**, in the **Queries group**, click the **Query Design** button.
2. **Add** the **2B Scholarships Awarded** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list. Add the **Amount** field to the design grid.

   Include only the field you want to summarize in the query, so that the aggregate function (minimum, maximum, average, sum, and so forth) is applied to that single field.
3. On the **Design tab**, in the **Show/Hide group**, click the **Totals** button to add a **Total** row as the third row in the design grid. Notice that in the design grid, on the **Total** row, under **Amount**, *Group By* displays.

   Use the Total row to select the aggregate function that you want to use for the field.
4. In the **Total** row, under **Amount**, click in the **Group By** box, and then click the **arrow** to display the list of aggregate functions. Compare your screen with Figure 2.41, and take a moment to review the table in Figure 2.42.

Figure 2.41



Aggregate Functions

Function Name	What It Does
Sum	Totals the values in a field.
Avg	Averages the values in a field.
Min	Locates the smallest value in a field.
Max	Locates the largest value in a field.
Count	Counts the number of records in a field.
StDev	Calculates the Standard Deviation for the values in a field.
Var	Calculates the Variance for the values in a field.
First	Displays the First value in a field.
Last	Displays the Last value in a field.
Expression	Creates a calculated field that includes an aggregate function.
Where	Limits records to those that match a condition specified in the Criteria row.

Figure 2.42

5 From the list of functions, click **Min**, and then **Run** the query. Double-click the right edge of the column heading to apply Best Fit to the column.

Access calculates the minimum (smallest) scholarship award—*$100.00*. The field name *MinOfAmount* displays for the calculation. This query answers the question, *What is the minimum (smallest) scholarship amount awarded?*

6 Switch to **Design** view. In the **Amount** field, in the **Total** row, select the **Max** function, and then **Run** the query.

The maximum (largest) scholarship amount is *$750.00*.

7 Switch to **Design** view, select the **Avg** function, and then **Run** the query.

The average scholarship amount awarded is *$358.33*.

8 Switch to **Design** view. Select the **Sum** function, and then **Run** the query.

Access sums the Amount field for all records and displays a result of *$10,750.00*. The field name, SumOfAmount, displays. This query answers the question, *What is the total dollar amount of all the scholarships awarded?*

Activity 2.25 | Grouping Data in a Query

Aggregate functions can also be used to calculate totals by groups of data. For example, to group (summarize) the amount of scholarships awarded to each student, you include the Student ID field, in addition to the Amount field, and then group all of the records for each student together to calculate a total awarded to each student. Similarly, you can calculate how much is awarded for each sport.

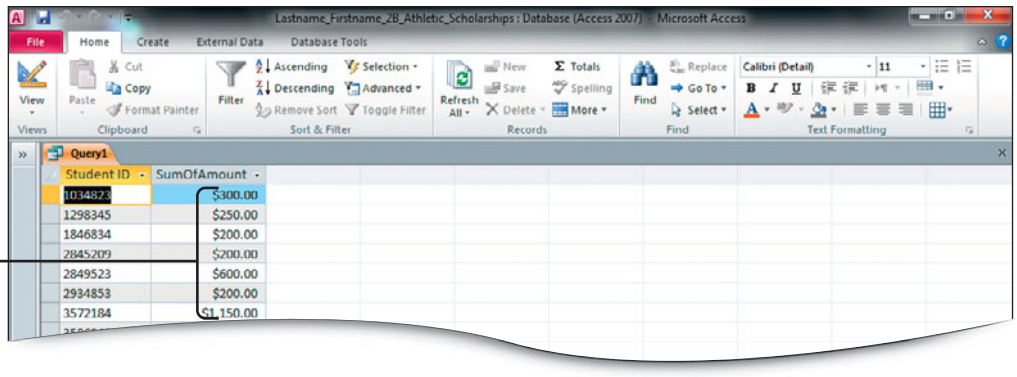
1 Switch to **Design** view. Drag the **Student ID** field to the first column of the design grid—**Amount** becomes the second column. On the **Total** row, under **Student ID**, notice that *Group By* displays.

This query groups—summarizes—the records by StudentID and calculates a total Amount for each student.

2 **Run** the query, and then compare your screen with Figure 2.43.

The query calculates the total amount of all scholarships for each student.

Figure 2.43



3 Switch to **Design** view. In the design grid, delete the **Student ID** field, and then drag the **Sport** field to the first column—**Amount** becomes the second column.

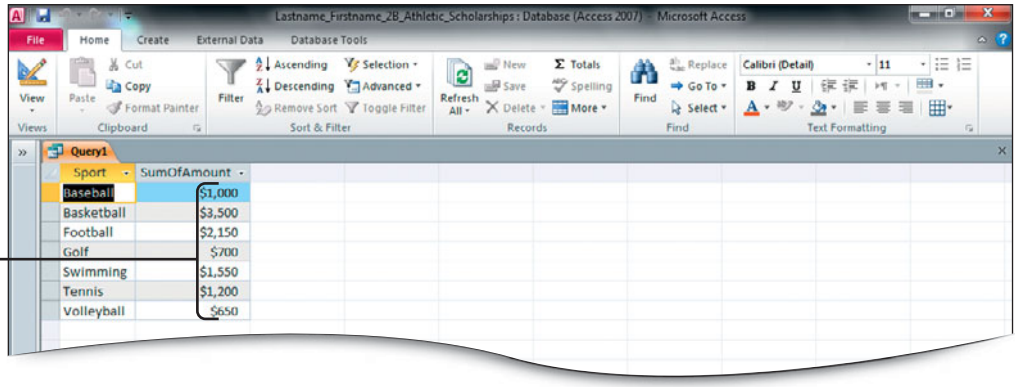
4 In the design grid, click in the **Amount** field, and then on the **Design tab**, in the **Show/Hide group**, click the **Property Sheet** button.

5 In the **Property Sheet**, set the **Format** to **Currency**, set the **Decimal Places** to **0**, and then **Close** ☒ the **Property Sheet**.

6 **Run** the query, and then compare your screen with Figure 2.44.

Access summarizes the data by sport. Basketball scholarships are the largest total Amount—*$3,500*.

Figure 2.44



7 **Save** the query as **Lastname Firstname 2B Total by Sport Query** and then display the query results in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the query.

Objective 14 | Create a Crosstab Query

A ***crosstab query*** uses an aggregate function for data that can be grouped by two types of information and displays data in a compact, spreadsheet-like format. A crosstab query always has at least one row heading, one column heading, and one summary field. Use a crosstab query to summarize a large amount of data in a small space that is easy to read.

Activity 2.26 | Creating a Crosstab Query Using One Table

In this activity, you will create a crosstab query that displays the total amount of scholarships awarded for each sport and for each team—women's or men's.

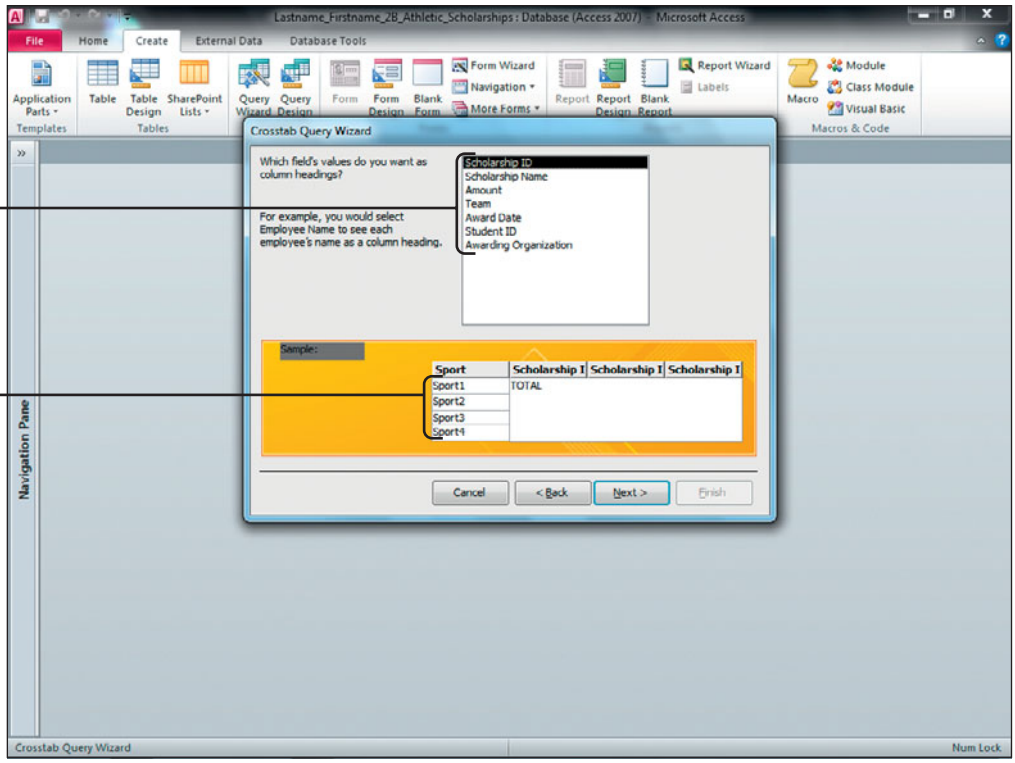
1 On the **Create tab**, in the **Queries group**, click the **Query Wizard** button. In the **New Query** dialog box, click **Crosstab Query Wizard**, and then click **OK**.

2 In the **Crosstab Query Wizard**, click **Table: 2B Scholarships Awarded** and then click **Next**.

3 To select the row headings, under **Available Fields**, double-click **Sport** to sort the scholarship amounts by the different sports. Click **Next**, and then compare your screen with Figure 2.45.

The sports are displayed as *row headings*; here you are prompted to select *column* headings.

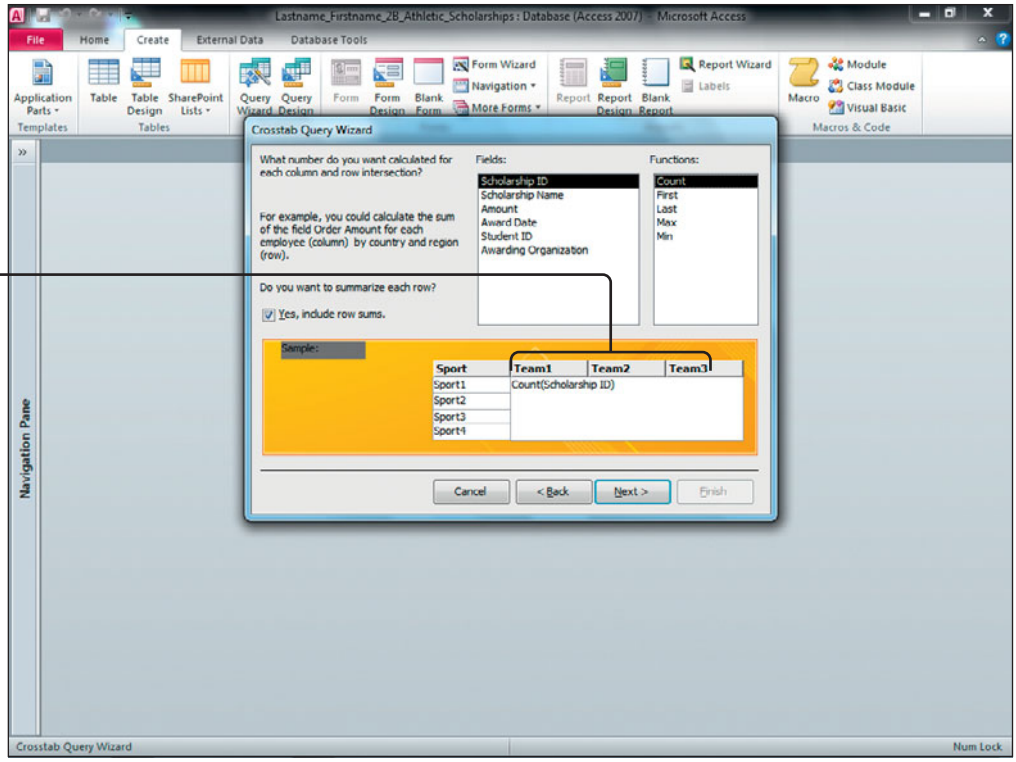
Figure 2.45



4 To select the column headings, in the field list, click **Team**. Click **Next**, and then compare your screen with Figure 2.46.

The teams will be listed as column headings; here you are prompted to select a field to summarize.

Figure 2.46



Teams display as column headings

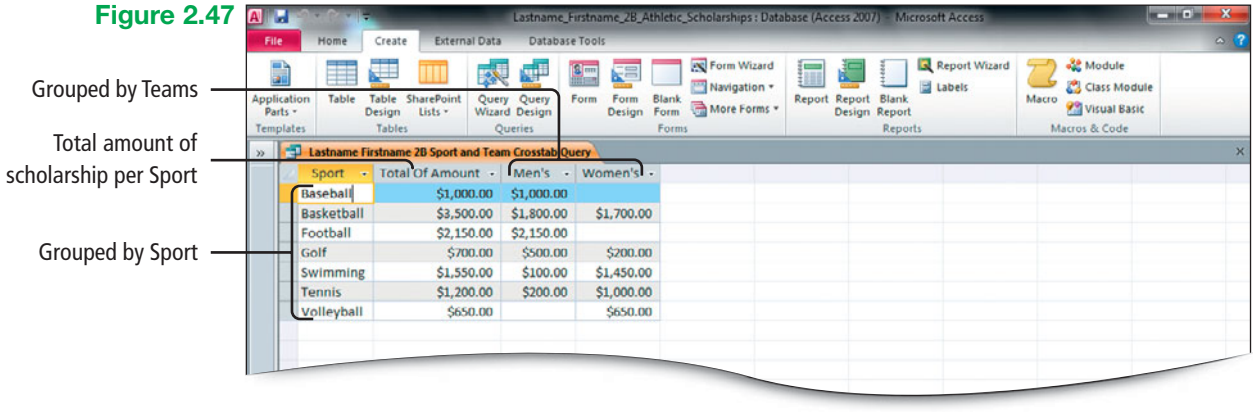
5 Under **Fields**, click **Amount**. Under **Functions**, click **Sum**, and then click **Next**.

The crosstab query will sum the Amount field for each sport and team.

6 In the **What do you want to name your query?** box, type **Lastname Firstname 2B Sport and Team Crosstab Query** and then click **Finish**. Apply **Best Fit** to the columns, click in any field to cancel the selection, and then compare your screen with Figure 2.47.

The crosstab query displays the total amount of scholarships awarded by sport and also by men's or women's teams. For example, for the sport of Golf, a total of $700 was awarded in scholarship money; $500 to men's teams and $200 to women's teams. A crosstab query is useful to display a summary of data based on two different fields—in this case, by sport and by teams.

Figure 2.47



7 Display the query results in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the query, and click **Yes** to save changes to the query layout.

8 **Open** the **Navigation Pane**. In **Backstage** view, click **Close Database**, and then click **Exit**. As directed by your instructor, submit your database and the ten paper or electronic printouts—relationship report and nine queries—that are the results of this project.

More Knowledge | Creating a Crosstab Query Using Two Related Tables

To create a crosstab query using fields from more than one table, you must first create a select query with the fields from both tables that will be included in the crosstab query.

End **You have completed Project 2B**

Content-Based Assessments

Summary

Sorting data in a table reorders the records based on one or more fields. Use queries to ask complex questions about the data in a database in a manner that Access can interpret. Save queries so they can be run as needed against current records. Use queries to limit the fields that display, add criteria to restrict the number of records in the query results, create calculated values, include data from more than one table, and to display data grouped by two types of information.

Key Terms

Aggregate functions 152
AND Condition 143
Ascending order.....123
Between ... And operator.....142
Calculated field.....149
Comparison operators.....141
Compound criteria 143
Criteria.....132
Crosstab query.....155
Data source.....126
Descending order.....123
Design grid 127
Expression 149
Field list.....119
Foreign key 121
Innermost sort field 125
Is Not Null 135
Is Null.....135
Join line.....121
Logical operators.....143
Message Bar.....117
One-to-many relationship 118
OR condition.....144
Outermost sort field.....125
Property Sheet 151
Referential integrity 121
Relationship 118
Run.....128
Select query 126
Sorting.....123
Subdatasheet 123
Subset 126
Table area 127
Text string 133
Trust Center 117
Wildcard character.....147

Matching

Match each term in the second column with its correct definition in the first column by writing the letter of the term on the blank line in front of the correct definition.

_____ 1. The area below the Ribbon that displays information such as security alerts.
_____ 2. An area where you can view the security and privacy settings for your Access installation.
_____ 3. An association that you establish between two tables based on common fields.
_____ 4. A relationship between two tables where one record in the first table corresponds to many records in the second table.
_____ 5. A list of field names in a table.
_____ 6. The field that is included in the related table so the field can be joined with the primary key in another table for the purpose of creating a relationship.
_____ 7. A set of rules that ensures that the data between related tables is valid.
_____ 8. The line joining two tables that visually indicates the common fields and the type of relationship.
_____ 9. A format for displaying related records in a datasheet when you click the plus sign (+) next to a record in a table on the one side of a relationship.
_____ 10. The process of arranging data in a specific order based on the value in a field.

A Ascending
B Descending
C Field list
D Foreign key
E Innermost
F Join line
G Message Bar
H One-to-many relationship
I Outermost
J Referential integrity
K Relationship
L Select query
M Sorting
N Subdatasheet
O Trust Center

_____ 11. A sorting order that arranges text in alphabetical order (A to Z) or numbers from lowest to highest.

_____ 12. A sorting order that arranges text in reverse alphabetical order (Z to A) or numbers from highest to lowest.

_____ 13. When sorting on multiple fields in Datasheet view, the field that is used for the first level of sorting.

_____ 14. When sorting on multiple fields in Datasheet view, the field that is used for the second level of sorting.

_____ 15. A database object that retrieves (selects) specific data from one or more tables and then displays the results in Datasheet view.

Multiple Choice

Circle the correct answer.

1. The lower area of the Query window that displays the design of the query is the:
   A. design grid **B.** property sheet **C.** table area

2. The process in which Access searches the records in the table, finds the records that match specified criteria, and then displays the records in a datasheet is:
   A. select **B.** run **C.** sort

3. Conditions in a query that identify the specific records for which you are looking are known as:
   A. aggregate functions **B.** criteria **C.** expressions

4. A criteria that searches for fields that are empty is:
   A. Is Empty **B.** Is Not Null **C.** Is Null

5. The symbols of =, >, and < are known as:
   A. aggregate functions **B.** comparison operators **C.** logical operators

6. A comparison operator that looks for values within a range is:
   A. And **B.** Between ... And **C.** Or

7. The logical operator that requires all conditions to be met is:
   A. AND **B.** Is Null **C.** OR

8. A wildcard character that serves as a placeholder for one or more unknown characters is the:
   A. * **B.** ? **C.** /

9. A field that stores the value of a mathematical operation is:
   A. an aggregate field **B.** a calculated field **C.** an expression

10. A query that uses an aggregate function for data that can be grouped by two types of information is:
    A. an aggregate query **B.** a calculated query **C.** a crosstab query

Content-Based Assessments

Apply 2A skills from these Objectives:

1. Open an Existing Database
2. Create Table Relationships
3. Sort Records in a Table
4. Create a Query in Design View
5. Create a New Query from an Existing Query
6. Sort Query Results
7. Specify Criteria in a Query

Skills Review | Project 2C Music Department

In the following Skills Review, you will assist Dr. William Jinkens, the Capital Cities Community College Music Director, in using his database to answer various questions about the instruments in the Music Department's inventory. Your results will look similar to Figure 2.48.

Project Files

For Project 2C, you will need the following file:

a02c_Music_Department

You will save your database as:

Lastname_Firstname_2C_Music_Department

Project Results

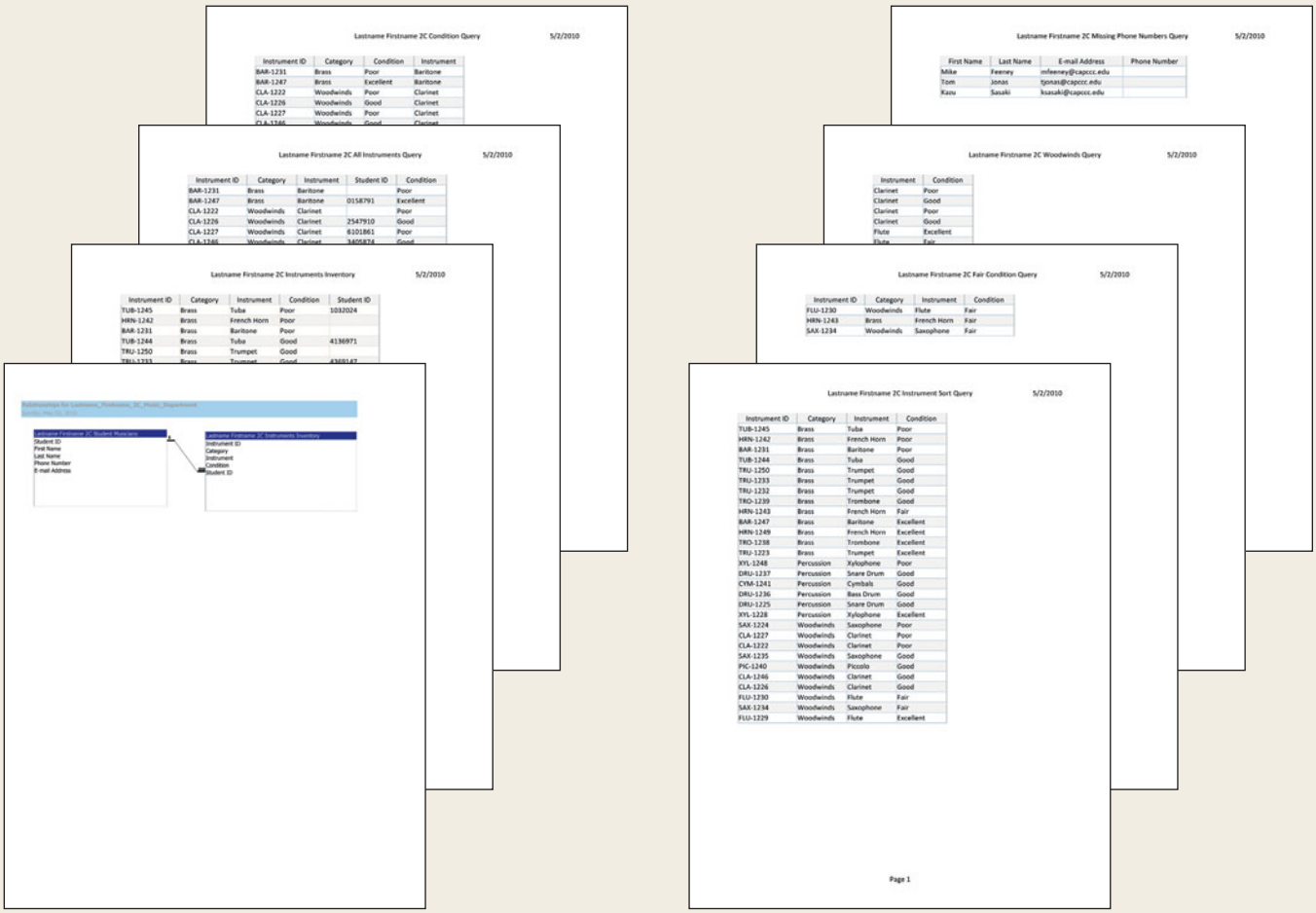


Figure 2.48

(Project 2C Music Department continues on the next page)

Skills Review | Project 2C Music Department (continued)

1 **Start** Access. In **Backstage** view, click **Open**. Navigate to the student files that accompany this textbook, and then open the **a02C_Music_Department** database.

a. Click the **File tab** to return to **Backstage** view, and then click **Save Database As**. In the **Save As** dialog box, navigate to your **Access Chapter 2** folder. In the **File name** box, select the file name, and then type **Lastname_Firstname_2C_Music_Department** and then press Enter. In the **Message Bar**, click **Enable Content**.

b. **Rename** the **2C Student Musicians** table to **Lastname Firstname 2C Student Musicians** and then **Rename** the **2C Instruments Inventory** to **Lastname Firstname 2C Instruments Inventory** Widen the **Navigation Pane** to display fully both table names.

2 **Open** both tables to examine the contents of each, **Close** the tables, and then **Close** the **Navigation Pane**.

a. Click the **Database Tools tab**, and in the **Relationships group** click the **Relationships** button. Drag the **Show Table** dialog box down into the lower right portion of your screen.

b. In the **Show Table** dialog box, click your **2C Student Musicians**, and then click **Add**. Double-click your **2C Instruments Inventory** to add the table to the **Relationships** window. In the **Show Table** dialog box, click **Close**.

c. Drag the **2C Instruments Inventory** field list—the field list on the right—to the right about 3 inches. In the **2C Student Musicians** field list—the field list on the left—position your mouse pointer over the lower right corner of the field list to display the  pointer, and then drag to the right to increase the width of the field list until the entire name of the table in the title bar displays and all of the field names display. Then use the  pointer to resize the **2C Instruments Inventory** field list so that all of the field names and the table name display.

d. In the **2C Student Musicians** field list, point to **Student ID**, hold down the left mouse button, and then drag down and to the right to the **2C Instruments Inventory** field list until your mouse pointer is on top of **Student ID**. Then release the mouse button. Drag the **Edit Relationships** dialog box to the lower portion of your screen. The relationship between the two tables is a one-to-many relationship; *one* student can play *many* instruments.

e. In the **Edit Relationships** dialog box, click to select the **Enforce Referential Integrity** check box, and then click the **Create** button. On the **Design tab**, in the **Tools group**, click the **Relationship Report** button. On the **Print Preview tab**, in the **Page Size group**, change the **Margins** to **Normal**, and then create a paper or electronic printout as directed.

f. **Save** the relationship report with the default name. **Close** the report, and then **Close** the **Relationships** window. From the **Navigation Pane**, open the **2C Instruments Inventory** table, and then **Close** the **Navigation Pane**.

3 In the **Condition** field, click any record. On the **Home tab**, in the **Sort & Filter group**, click the **Descending** button to sort the records from *Poor* to *Excellent*. In the field names row, click the **Category arrow**, and then click **Sort A to Z** to sort the records first by *Category* and then by *Condition*.

a. Display **Backstage** view, click **Print**, and then click **Print Preview**. Create a paper or electronic copy as directed. **Close Print Preview**, **Close** the table, and then click **No**; you do not need to save the sort changes.

4 Click the **Create tab**, and then in the **Queries group**, click the **Query Design** button. In the **Show Table** dialog box, double-click your **2C Instruments Inventory** table, and then **Close** the **Show Table** dialog box. Expand the field list.

a. Double-click **Instrument ID** to add the field to the design grid. Point to the **Category** field, hold down the left mouse button, and then drag the field down into the design grid until you are pointing to the **Field** row in the second column. Release the mouse button.

b. In design grid, in the **Field** row, click in the third column, and then click the **arrow** that displays. From the list, click **Instrument** to add the field to the design grid. Using the technique of your choice, add the **Student ID** field to the fourth column and the **Condition** field to the fifth column in the design grid.

c. On the **Design tab**, in the **Results group**, click the **Run** button. This query answers the question, *What is the Instrument ID, Category, Instrument, Student ID, and Condition of all of the instruments in the 2C Instruments Inventory table?*

d. **Save** the query as **Lastname Firstname 2C All Instruments Query** and then click **OK**. Display the query in **Print Preview**, and then create a paper or electronic printout as directed. **Close Print Preview**.

(Project 2C Music Department continues on the next page)

Skills Review | Project 2C Music Department (continued)

5 Display **Backstage** view, click **Save Object As**. In the **Save As** dialog box, type **Lastname Firstname 2C Condition Query** and then click **OK** to create a new query based on an existing query. Click the **Home tab**, and then switch to **Design** view.

a. In the design grid, point to the thin gray selection bar above the **Student ID** field name until the ↓ pointer displays. Click to select the **Student ID** column, and then press Del.

b. In the gray selection bar, select the **Instrument ID** column. Point to the **selection bar** to display the pointer, and then drag to the right until a dark vertical line displays on the right side of the **Condition** column. Release the mouse button to position the **Instrument ID** field in the fourth column.

c. **Run** the query. The query results display four fields. This query answers the question, *What is the Category, Instrument, Condition, and Instrument ID for every Instrument in the 2C Instruments Inventory table?*

d. Display the query in **Print Preview**, and then create a paper or electronic printout as directed. **Close Print Preview**, **Close** the query, and in the message box, click **Yes** to save the changes to the design—you moved two fields. **Open** the **Navigation Pane**.

6 **Open** your **2C All Instruments Query**. **Save** the query object as **Lastname Firstname 2C Instrument Sort Query** and then click the **Home tab**. **Close** the **Navigation Pane**. Switch to **Design** view.

a. In the design grid, delete the **Student ID** field. In the **Sort** row, click in the **Category** field. Click the **Sort arrow**, and then in the list, click **Ascending**. In the **Sort** row, click in the **Condition** field, click the **Sort arrow**, and then click **Descending**. **Run** the query. This query answers the question, *For every Instrument ID, within each Category (with Category sorted in ascending order), what Instruments are in the inventory and what is the instrument's Condition (with Condition sorted in descending order)?*

b. Display the query in **Print Preview**. Create a paper or electronic printout if instructed to do so, and then **Close Print Preview**. **Close** the query. In the message box, click **Yes** to save the changes to the query design.

7 Click the **Create tab**, and then in the **Queries group**, click the **Query Design** button. **Add** your **2C Instruments Inventory** table to the table area, and then **Close** the **Show Table** dialog box. Expand the field list. Add the following fields to the design grid in the order given: **Instrument ID**, **Category**, **Instrument**, and **Condition**.

a. In the design grid, on the **Criteria** row, click in the **Condition** field, type **fair** and then press Enter. **Run** the query; three records display that meet the specified criteria—records that have *fair* in the Condition field.

b. **Save** the query as **Lastname Firstname 2C Fair Condition Query** and then create a paper or electronic printout as directed. **Close Print Preview**, and then **Close** the query.

c. **Create** a new query in **Query Design** view. **Add** the **2C Instruments Inventory** table to the table area, and then expand the field list. Add the following fields, in the order given, to the design grid: **Category**, **Instrument**, and **Condition**.

d. In the **Criteria** row, click in the **Category** field, type **woodwinds** and then press Enter. Under **Category**, in the **Show** row, click to clear the check box, and then **Run** the query. Ten instruments are categorized as woodwinds. Recall that if all results use the same criteria, such as *woodwinds*, it is not necessary to display the data in the query results.

e. **Save** the query as **Lastname Firstname 2C Woodwinds Query** and then create a paper or electronic printout as directed. **Close Print Preview**. **Close** the query.

f. **Create** a new query in **Query Design** view. **Add** the **2C Student Musicians** table to the table area, and then expand the field list. Add the following fields, in the order given, to the design grid: **First Name**, **Last Name**, **E-mail Address**, and **Phone Number**.

g. In the **Criteria** row, click in the **Phone Number** field, type **is null** and then press Enter. In the **Sort** row, click in the **Last Name** field, click the **Sort arrow**, and then click **Ascending**. **Run** the query. Three student musicians do not have a phone number stored in the 2C Student Musicians table.

h. **Save** the query as **Lastname Firstname 2C Missing Phone Numbers Query** and then create a paper or electronic printout as directed. **Close Print Preview**, and then **Close** the query. **Open** the **Navigation Pane**.

i. Display **Backstage** view, click **Close Database**, and then click **Exit** to close the Access program. As directed by your instructor, submit your database and the eight paper or electronic printouts—relationship report, sorted table, and six queries—that are the results of this project.

End **You have completed Project 2C**

Content-Based Assessments

Apply 2B skills from these Objectives:

- 8 Specify Numeric Criteria in a Query
- 9 Use Compound Criteria
- 10 Create a Query Based on More Than One Table
- 11 Use Wildcards in a Query
- 12 Use Calculated Fields in a Query
- 13 Calculate Statistics and Group Data in a Query
- 14 Create a Crosstab Query

Skills Review | Project 2D Concerts and Sponsors

In the following Skills Review, you will assist Dr. William Jinkens, the Capital Cities Community College Music Director, in answering questions about concerts, sponsors, box office receipts, dates, and concert locations. Your results will look similar to Figure 2.49.

Project Files

For Project 2D, you will need the following files:

a02D_Concerts_Sponsors
a02D_Sponsors (Excel file)

You will save your database as:

Lastname_Firstname_2D_Concerts_Sponsors

Project Results

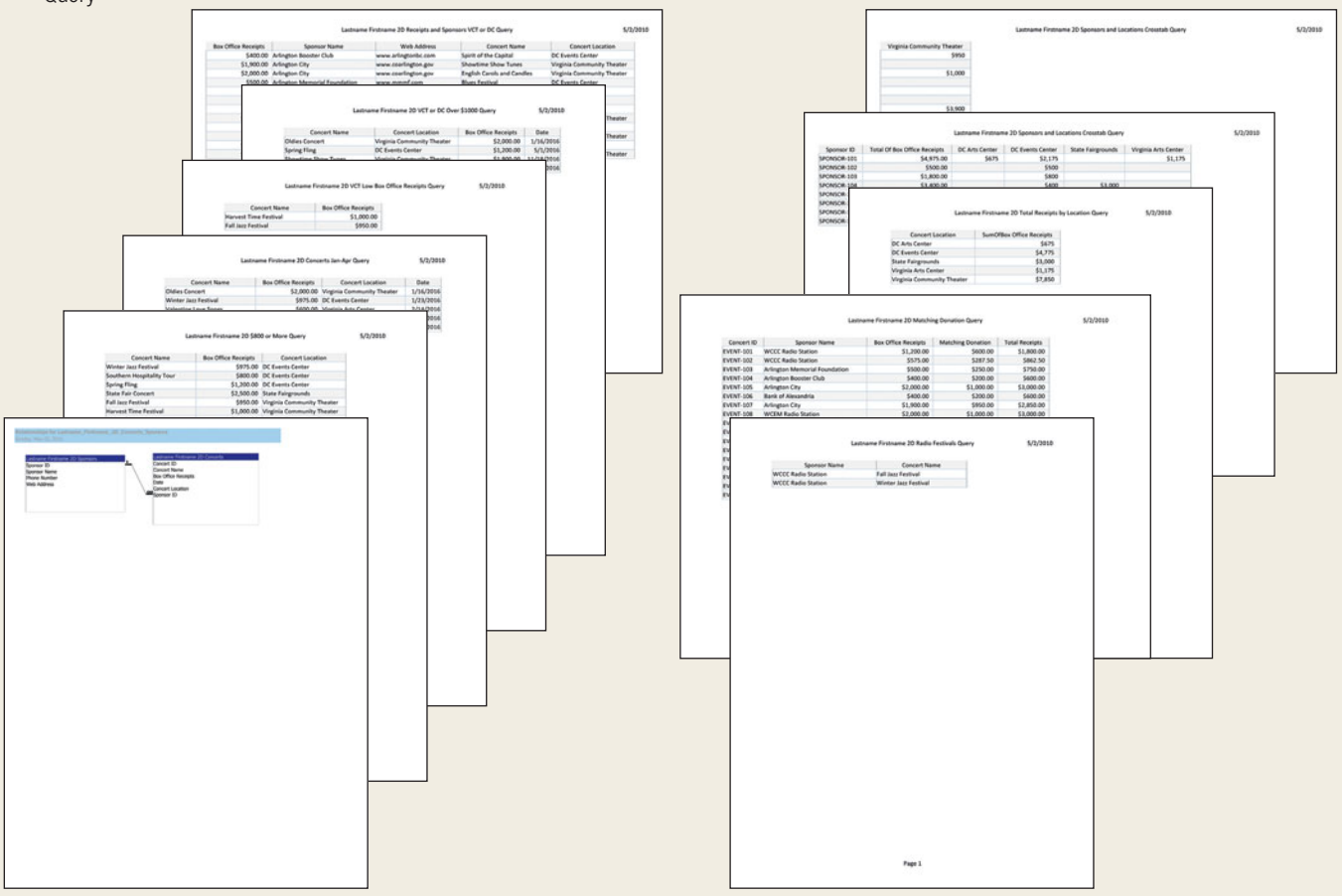


Figure 2.49

(Project 2D Concerts and Sponsors continues on the next page)

Skills Review | Project 2D Concerts and Sponsors (continued)

1 **Start** Access. In the **Backstage** view, click **Open**. Navigate to the student files that accompany this textbook, and then open the **a02D_Concerts_Sponsors** database.

a. Click the **File tab** to return to **Backstage** view, and then click **Save Database As**. In the **Save As** dialog box, navigate to your **Access Chapter 2** folder. In the **File name** box, select the file name, and then type **Lastname_Firstname_2D_Concerts_Sponsors** and then press Enter. In the **Message Bar**, click **Enable Content**. **Rename** the **2D Concerts** table to **Lastname Firstname 2D Concerts** and then widen the **Navigation Pane** to display the entire table name.

b. Click the **External Data tab**, and then in the **Import & Link group**, click the **Excel** button. In the **Get External Data – Excel Spreadsheet** dialog box, click **Browse**. Navigate to your student files, and then double-click the Excel file **a02D_Sponsors**. Be sure that the **Import the source data into a new table in the current database** option button is selected, and then click **OK**.

c. In the **Import Spreadsheet Wizard**, select the **First Row Contains Column Headings** check box, and then click **Next**. Click **Next** again. Click the **Choose my own primary key** option button, and then be sure that **Sponsor ID** displays. Click **Next**. In the **Import to Table** box, type **Lastname Firstname 2D Sponsors** and then click **Finish**. In the Wizard, click **Close**. The imported Excel spreadsheet becomes the second table in the database.

d. From the **Navigation Pane**, open your **2D Sponsors** table. Apply **Best Fit** to all columns, and then **Close** the table, saving changes to the design. Click the **Database Tools tab**, and in the **Relationships group**, click the **Relationships** button. **Add** the **2D Sponsors** table, and then **Add** the **2D Concerts** table to the table area. **Close** the **Show Table** dialog box. Expand and move the field lists as necessary.

e. In the **2D Sponsors** field list, point to the **Sponsor ID** field, hold down the left mouse button, drag into the **2D Concerts** field list, position the mouse pointer on top of the **Sponsor ID** field, and then release the mouse button. In the **Edit Relationships** dialog box, select the **Enforce Referential Integrity** check box, and then click the **Create** button. A one-to-many relationship is established; *one* sponsor organization can sponsor *many* concerts.

f. On the **Design tab**, in the **Tools group**, click the **Relationship Report** button. Create a paper or electronic printout as directed, and then **Close** the report. In the message box, click **Yes**; and then in the **Save As** dialog box, click **OK** to save the report with the default name. **Close** the **Relationships** window, and then **Close** the **Navigation Pane**.

2 Click the **Create tab**, and then in the **Queries group**, click the **Query Design** button. **Add** the **2D Concerts** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list.

a. Add the following fields to the design grid in the order given: **Concert Name**, **Box Office Receipts**, and **Concert Location**. Click in the **Sort** row under **Concert Location**, click the **Sort arrow**, and then click **Ascending**. In the **Criteria** row, under **Box Office Receipts**, type **>=800** press Enter, and then **Run** the query. Nine records meet the criteria. This query answers the question, *Which concerts had Box Office Receipts of $800 or more, and where was each concert held in alphabetical order by Concert Location?*

b. **Save** the query as **Lastname Firstname 2D $800 or More Query** and then create a paper or electronic printout as directed. **Close Print Preview**.

c. With the query still open, display **Backstage** view, click **Save Object As**, type **Lastname Firstname 2D Concerts Jan-Apr Query** and then click **OK**. Click the **Home tab**, and then switch to **Design** view. From the **2D Concerts** field list, add **Date** as the fourth field in the design grid.

d. In the **Criteria** row, under **Box Office Receipts**, delete the existing criteria so that the query is not restricted by receipts. Click in the **Sort** row under **Concert Location**, click the **Sort arrow**, and then click **(not sorted)**. Click in the **Sort** row under **Date**, click the **Sort arrow**, and then click **Ascending**.

e. Click in the **Criteria** row under **Date**, type **between 1/1/16 and 4/30/16** and then press Enter. **Run** the query. Five records meet the criteria. This query answers the question, *What is the Concert Name, Box Office Receipts, Concert Location, and Date, in chronological order between January 1, 2016, and April 30, 2016, of concerts held?* **Print** or submit electronically as directed. **Close Print Preview**, **Close** the query, and then click **Yes** to save the changes to the query design.

(Project 2D Concerts and Sponsors continues on the next page)

Skills Review | Project 2D Concerts and Sponsors (continued)

3 **Create** a query in **Query Design** view. **Add** the **2D Concerts** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list. Add the following fields to the design grid in the order given: **Concert Name**, **Concert Location**, and **Box Office Receipts**.

a. In the **Criteria** row, under **Concert Location**, type **Virginia Community Theater** and then press Enter. In the **Criteria** row, under **Box Office Receipts**, type **<=1000** and then press Enter. In the **Concert Location** field, clear the **Show** check box. **Run** the query; two records display. This query answers the question, *Which concerts that were held at the Virginia Community Theater had Box Office Receipts of $1,000 or less?*

b. **Save** the query as **Lastname Firstname 2D VCT Low Box Office Receipts Query** and then create a paper or electronic printout as directed. **Close** the query.

c. **Create** a query in **Query Design** view. **Add** the **2D Concerts** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list. Add the following fields to the design grid: **Concert Name**, **Concert Location**, **Box Office Receipts**, and **Date**.

d. In the **Criteria** row, under **Concert Location**, type **Virginia Community Theater or DC Events Center** and press Enter. In the **Criteria** row, under **Box Office Receipts**, type **>1000** and then press Enter. In the **Sort** row, under **Date**, click the **Sort arrow**, and then click **Ascending**.

e. **Run** the query. Four records display. This query answers the question, *Which concerts held at either the Virginia Community Theater or the DC Events Center had Box Office Receipts of more than $1,000 and on what dates, in chronological order, were the concerts held?*

f. **Save** the query as **Lastname Firstname 2D VCT or DC Over $1000 Query** and then create a paper or electronic printout as directed. **Close Print Preview**, and then **Close** the query.

4 **Create** a query in **Query Design** view, **Add** both tables to the table area, and then expand the field lists. Reposition the field lists so that **2D Sponsors** is on the left side. From the **2D Sponsors** field list, add the following fields to the design grid in the order given: **Sponsor Name** and **Web Address**. Click in the **Sort** row under **Sponsor Name**, click the **Sort arrow**, and then click **Ascending**.

a. From the **2D Concerts** field list, add the following fields to the design grid in the order give: **Concert Name**, **Concert Location**, and **Box Office Receipts**.

b. In the **Criteria** row, under **Concert Location**, type **Virginia Community Theater** and then click in the **or** row, under **Concert Location**. Type **DC Events Center** and then press Enter.

c. In the design grid, select the **Box Office Receipts** field, and then drag it to the first field position in the grid. **Run** the query; 12 records display. This query answers the question, *What were the Box Office Receipts, Sponsor Name, sponsor Web Address, Concert Name, and Concert Location of all concerts held at either the Virginia Community Theater or the DC Events Center, sorted alphabetically by Sponsor Name?*

d. **Save** the query as **Lastname Firstname 2D Receipts and Sponsors VCT or DC Query** and then display the query results in **Print Preview**. Change the orientation to **Landscape**, change the **Margins** to **Normal**, and then create a paper or electronic printout as directed. **Close** the query.

5 **Create** a query in **Query Design** view, **Add** both tables to the table area, **Close** the **Show Table** dialog box, and then expand the field lists. Reposition the field lists so that **2D Sponsors** is on the left side.

a. From the **2D Sponsors** field list, add the **Sponsor Name** field to the design grid. From the **2D Concerts** field list, add the **Concert Name** field to the design grid.

b. In the **Criteria** row, under **Sponsor Name**, type ***radio*** and then press Enter. In the **Criteria** row, under **Concert Name**, type ***festival** and then press Enter.

c. **Run** the query; two records have the word *Radio* somewhere in the Sponsor Name and the word *Festival* at the end of the Concert Name. This query answers the question, *Which radio stations are sponsoring Festival-type concerts?* **Save** the query as **Lastname Firstname 2D Radio Festivals Query** and then create a paper or electronic printout as directed. **Close** the query.

6 **Create** a query in **Query Design** view. **Add** both tables to the table area, **Close** the **Show Table** dialog box, and then expand the field lists. If necessary, reposition the

(Project 2D Concerts and Sponsors continues on the next page)

Skills Review | Project 2D Concerts and Sponsors (continued)

field lists so that *2D Sponsors* is on the left side. From the field lists, add the following fields to the design grid in the order given: **Concert ID**, **Sponsor Name**, and **Box Office Receipts**. Click in the **Sort** row under **Concert ID**, click the **Sort arrow**, and then click **Ascending**.

a. Sponsors have indicated that they will donate an additional amount to the Music Department based on 50 percent of the Box Office Receipts. On the **Field** row, right-click in the first empty column to display a shortcut menu, and then click **Zoom**. In the **Zoom** dialog box, type **Matching Donation:[Box Office Receipts]*0.5** and then click **OK**.

b. **Run** the query to view the new field—*Matching Donation*. Switch to **Design** view. In the **Field** row, in the first empty column, right-click, and then click **Zoom**. In the **Zoom** dialog box, type **Total Receipts: [Box Office Receipts]+[Matching Donation]** and then click **OK**. **Run** the query to view the results.

c. Switch to **Design** view. In the field row, click in the **Matching Donations** field. In the **Show/Hide group**, click the **Property Sheet** button, and then set the **Format** to **Currency** and the **Decimal Places** to **2**. **Close** the **Property Sheet**.

d. **Run** the query. This query answers the question *In ascending order by Concert ID, assuming each sponsor makes a matching 50 percent donation based on each concert's Box Office Receipts, what is the Sponsor Name, Box Office Receipts, Matching Donation, and Total Receipts for each concert?*

e. Select all of the columns, and then apply **Best Fit**. **Save** the query as **Lastname Firstname 2D Matching Donation Query** and then display the query results in **Print Preview**. Change the orientation to **Landscape**, and then create a paper or electronic printout as directed. **Close** the query.

7 **Create** a query in **Query Design** view. **Add** the **2D Concerts** table to the table area, **Close** the **Show Table** dialog box, and then expand the field list. Add the **Box Office Receipts** field to the design grid.

a. On the **Design tab**, in the **Show/Hide group**, click the **Totals** button, which adds a *Total* row as the third row in the design grid. On the **Total** row, under **Box Office Receipts**, click in the **Group By** box. Click the **arrow**, and then click the **Sum** function.

b. With the field still selected, display the **Property Sheet**, set the **Decimal Places** to **0**, and then **Close** the **Property Sheet**. **Run** the query. The total Box Office Receipts for all the concerts was *$17,475*. Apply **Best Fit** to the **SumOfBox Office Receipts** column.

c. Switch to **Design** view. In the design grid, add the **Concert Location** field as the first field of the design grid. **Run** the query. This query answers the question *For each Concert Location, what are the total Box Office Receipts?*

d. **Save** the query as **Lastname Firstname 2D Total Receipts by Location Query** and then create a paper or electronic printout as directed. **Close** the query.

8 **Create** a query using the **Query Wizard**. In the **New Query** dialog box, click **Crosstab Query Wizard**, and then click **OK**. Be sure that **2D Concerts** is selected, and then click **Next**.

a. Under **Available Fields**, double-click **Sponsor ID** so that you can display Box Office Receipts by Sponsor ID, and then click **Next**. In the field list, click **Concert Location** to add the locations as column headings, and then click **Next**.

b. Under **Fields**, click **Box Office Receipts**. Under **Functions**, click **Sum**, and then click **Next**.

c. Name the query **Lastname Firstname 2D Sponsors and Locations Crosstab Query** and then click **Finish**. Click the **Home tab**, switch to **Design** view, click in the **Box Office Receipts** column, display the **Property Sheet**, and then set the **Decimal Places** to **0**. **Close** the **Property Sheet**, **Run** the query, and apply **Best Fit** to all of the columns. This query answers the question *By Sponsor ID, what are the total Box Office Receipts for each Concert Location?*

d. Display the query results in **Print Preview**. Change the orientation to **Landscape**, change the **Margins** to **Normal**, and then create a paper or electronic printout as directed—two pages result. **Close** the query, saving changes to the design.

e. **Open** the **Navigation Pane**. Increase the width of the **Navigation Pane** to display fully all of the object names. Display **Backstage** view, click **Close Database**, and then click **Exit**. As directed by your instructor, submit your database and the ten paper or electronic printouts—relationship report and nine queries—that are the results of this project.

End **You have completed Project 2D**

Content-Based Assessments

Apply 2A skills from these Objectives:

1. Open an Existing Database
2. Create Table Relationships
3. Sort Records in a Table
4. Create a Query in Design View
5. Create a New Query from an Existing Query
6. Sort Query Results
7. Specify Criteria in a Query

Mastering Access | Project 2E Grants and Organizations

In the following Mastering Access project, you will assist Susan Elkington, Director of Grants for the college, in using her database to answer questions about public and private grants awarded to the college departments. Your results will look similar to Figure 2.50.

Project Files

For Project 2E, you will need the following file:

a02E_Grants_Organizations

You will save your database as:

Lastname_Firstname_2E_Grants_Organizations

Project Results

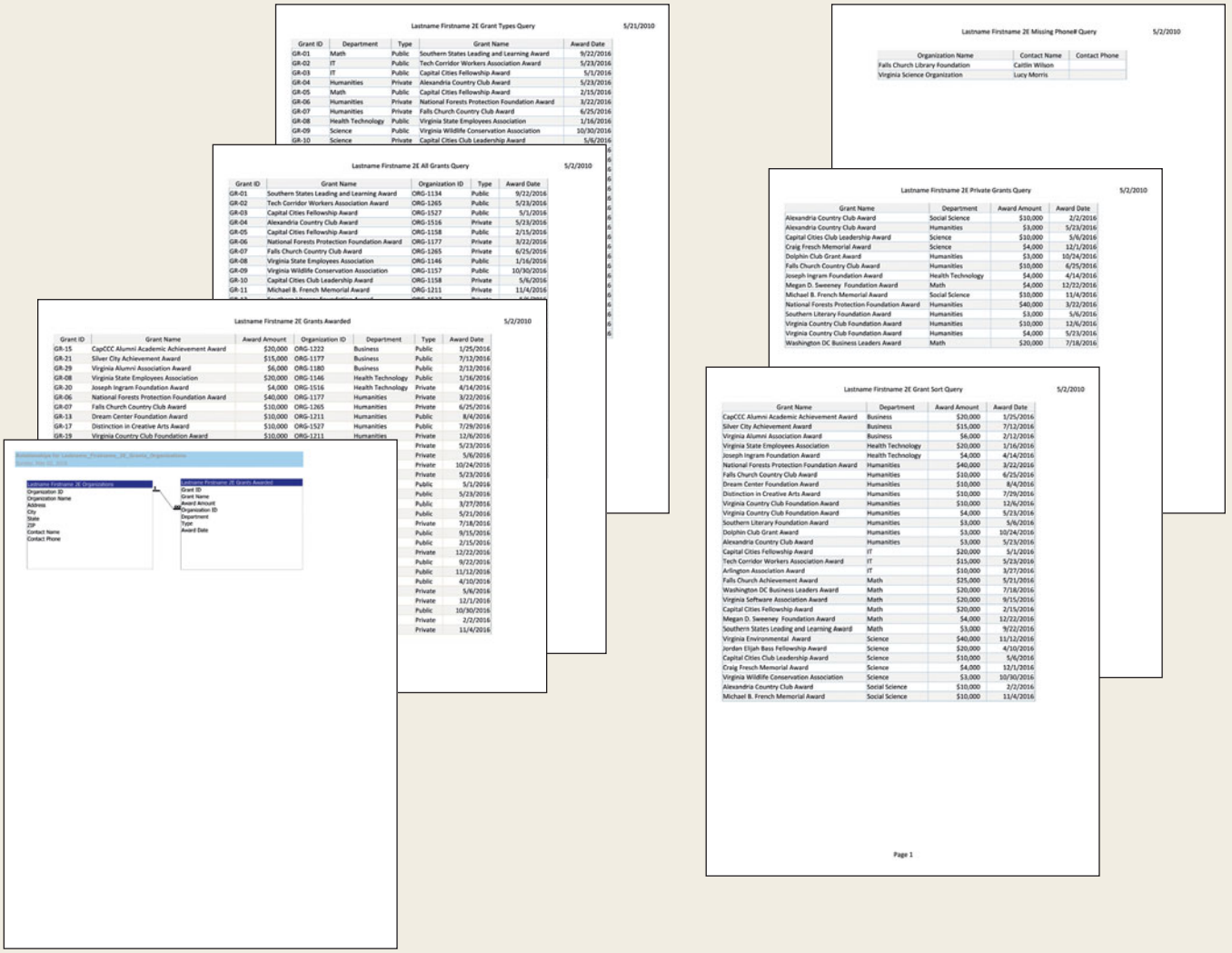


Figure 2.50

(Project 2E Grants and Organizations continues on the next page)

Content-Based Assessments

Mastering Access | Project 2E Grants and Organizations (continued)

1 **Start** Access. From your student files, open the **a02E_Grants_Organizations** database. Save the database in your **Access Chapter 2** folder as **Lastname_Firstname_2E_Grants_Organizations** and then enable the content. In the **Navigation Pane**, **Rename** the tables by adding **Lastname Firstname** to the beginning of each table name, and then widen the **Navigation Pane** to display fully both table names. **Open** both tables and examine their contents to become familiar with the data. **Close** both tables, and leave the **Navigation Pane** open.

2 Create a *one-to-many* relationship between the **2E Organizations** table and the **2E Grants Awarded** table based on the **Organization ID** field, and then **Enforce Referential Integrity**. *One* organization can award *many* grants. Create a **Relationship Report**, saving it with the default name. Create a paper or electronic printout as directed, and then **Close** all open objects, saving changes if prompted.

3 **Open** the **2E Grants Awarded** table, and then **Close** the **Navigation Pane**. **Sort** so that the records in the table are in alphabetical order by the **Department** and then in descending order by **Award Amount**. Create a paper or electronic printout as directed, being sure that the table prints on only one page by using **Landscape**, with **Normal** margins. **Close** the table, and do *not* save changes to the table.

4 **Create** a query in **Query Design** view, using the **2E Grants Awarded** table to answer the question, *What is the Grant ID, Grant Name, Award Amount, Type, and Award Date for all of the grants?* Display the fields in the order listed in the question. **Save** the query as **Lastname Firstname 2E All Grants Query** and then, with **Normal** margins, create a paper or electronic printout as directed. **Close Print Preview**, and leave the query open.

5 Use **2E All Grants Query** to create a new query. **Save** the **Object As Lastname Firstname 2E Grant Types Query** and then redesign the query to answer the question, *What is the Grant ID, Department, Type, Grant Name, and Award Amount for all grants?* Display the only the fields necessary to answer the question and in the order listed in the question. With **Normal** margins, create a paper or electronic printout as directed. **Close** the query, saving the design changes.

6 From the **Navigation Pane**, open the **2E All Grants Query**, and then **Close** the **Navigation Pane**. **Save** the **Object As Lastname Firstname 2E Grant Sort Query** and then switch to **Design** view. Redesign the query to answer the question, *What is the Grant Name, Department, Award Amount, and Award Date for grants sorted first in alphabetical order by Department and then in descending order by Amount?* Display only the fields necessary to answer the question and in the order listed in the question. With **Normal** margins, create a paper or electronic printout as directed. **Close** the query, saving changes to the query design.

7 **Open** the **Navigation Pane**, open **2E Grant Sort Query**, and then **Close** the **Navigation Pane**. **Save** the **Object As Lastname Firstname 2E Private Grants Query** and then switch to **Design** view. Redesign the query to answer the question, *What is the Grant Name, Department, Award Amount, and Award Date for grants that have a Type of Private, sorted in alphabetical order by Grant Name?* Do *not* display the **Type** field in the query results; display the fields in the order listed in the question. With **Normal** margins, create a paper or electronic printout as directed. **Close** the query, saving changes to the query design.

8 **Create** a query in **Query Design** view, using the **2E Organizations** table to answer the question, *What is the Organization Name and Contact Name where the Contact Phone number is missing from the table, sorted in alphabetical order by the Organization Name?* Two records meet the criteria. **Save** the query as **Lastname Firstname 2E Missing Phone# Query** and then create a paper or electronic printout as directed. **Close** the query.

9 **Open** the **Navigation Pane** and widen it so that all object names display fully. In **Backstage** view, click **Close Database**, and then click **Exit**. As directed by your instructor, submit your database and the seven paper or electronic printouts—relationship report, sorted table, and five queries—that are the results of this project.

End **You have completed Project 2E**

Content-Based Assessments

Apply 2B skills from these Objectives:

- 8 Specify Numeric Criteria in a Query
- 9 Use Compound Criteria
- 10 Create a Query Based on More Than One Table
- 11 Use Wildcards in a Query
- 12 Use Calculated Fields in a Query
- 13 Calculate Statistics and Group Data in a Query
- 14 Create a Crosstab Query

Mastering Access | Project 2F Events and Clients

In the following Mastering Access project, you will assist Hank Schwan, the Capital Cities Community College Facilities Manager, in using his database to answer questions about facilities that the college rents to community and private organizations at times when the facilities are not in use for college activities. Your results will look similar to Figure 2.51.

Project Files

For Project 2F, you will need the following files:

a02F_Events_Clients
a02F_Rental_Clients (Excel file)

You will save your database as:

Lastname_Firstname_2F_Events_Clients

Project Results

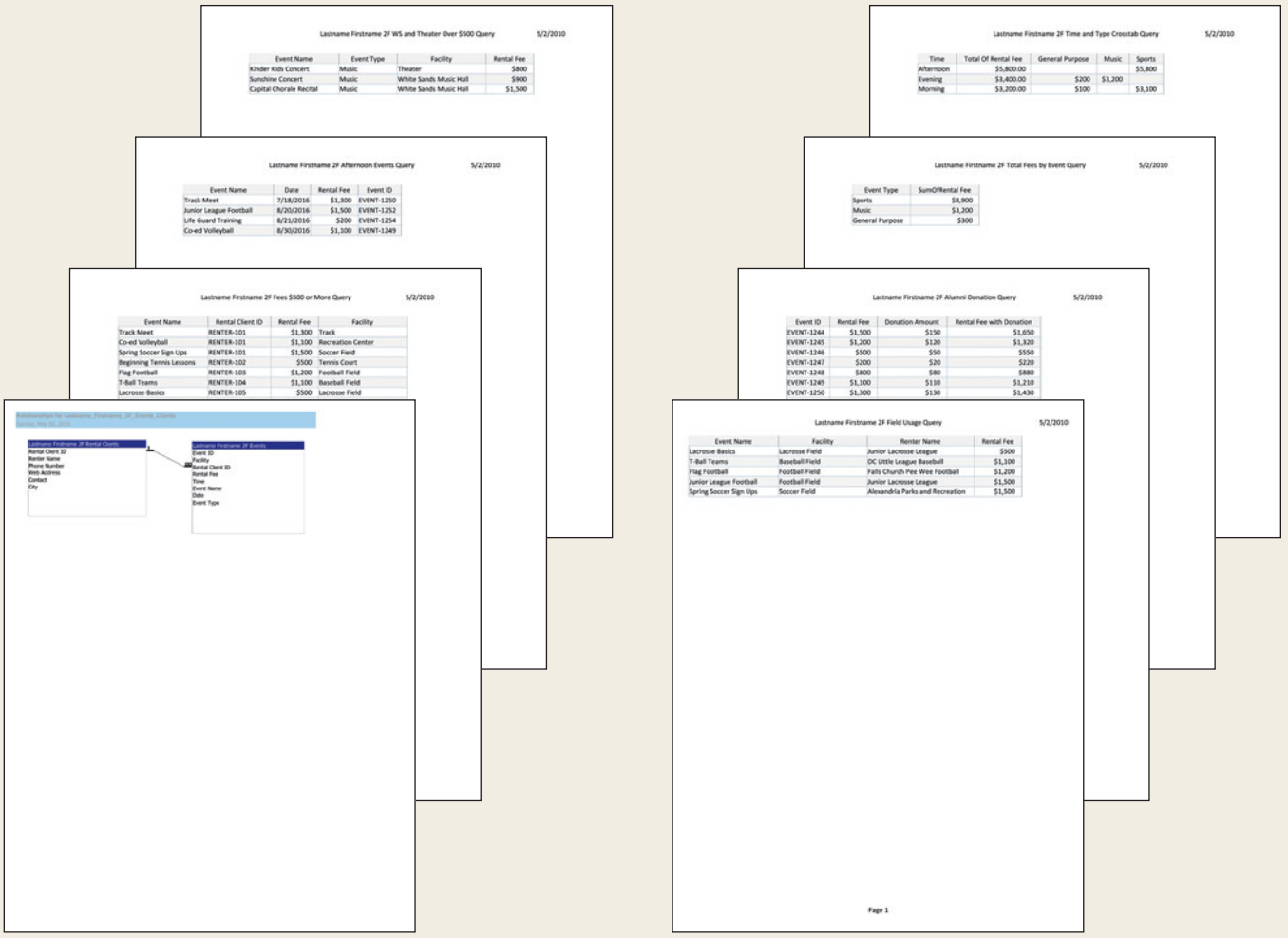


Figure 2.51

(Project 2F Events and Clients continues on the next page)

Content-Based Assessments

Mastering Access | Project 2F Events and Clients (continued)

1 **Start** Access. From your student files, open the **a02F_Events_Clients** database. Save the database in your **Access Chapter 2** folder as **Lastname_Firstname_2F_Events_Clients** and then enable the content. In the **Navigation Pane**, **Rename** the table by adding **Lastname Firstname** to the beginning of the table name.

2 **Import** the **a02F_Rental_Clients** Excel spreadsheet from the student data files that accompany this textbook into the current database as a new table. Designate the first row of the spreadsheet as column headings. Select the **Rental Client ID** field as the primary key. Name the table **Lastname Firstname 2F Rental Clients** and then widen the **Navigation Pane** to display fully the two table names. **Open** both tables and examine their contents to become familiar with the data. In the **2F Rental Clients** table, apply **Best Fit** to all of the columns. **Close** both tables, saving changes, and then **Close** the **Navigation Pane**.

3 Create a *one-to-many* relationship between the **2F Rental Clients** table and the **2F Events** table based on the **Rental Client ID** field, and then **Enforce Referential Integrity**. *One* rental client can have *many* events. Create a **Relationship Report**, saving it with the default name. Create a paper or electronic printout as directed, and then **Close** all open objects, saving changes if prompted.

4 **Create** a query in **Query Design** view using the **2F Events** table to answer the question, *What is the Event Name, Rental Client ID, and Rental Fee for events with fees greater than or equal to $500, in ascending order by Rental Client ID, and in which Facility was the event held?* Display the fields in the order listed in the question. Eleven records meet the criteria. **Save** the query as **Lastname Firstname 2F Fees $500 or More Query** Create a paper or electronic printout as directed. Leave the query open.

5 Using the **2F Fees $500 or More Query** object, create a new query, and save it as **Lastname Firstname 2F Afternoon Events Query** Redesign the query to answer the questions, *Which Events were held in the Afternoon between 7/1/16 and 8/31/16, in chronological order by date, what was the Rental Fee, and what was the Event ID?* (Hint: Open the 2F Events table to see how the Time field data is stored.). Do *not* display the **Time** field in the results, and do *not* restrict the results by **Rental Fee**. Four records meet the criteria. Create a paper or electronic printout as directed, **Close** the query, and save changes to the design.

6 **Create** a query in **Query Design** view using the **2F Events** table to answer the question, *Which Events and Event Types were held in either the White Sands Music Hall or the Theater that had Rental Fees greater than $500?* Display the fields in the order listed in the question. Three records meet the criteria. **Save** the query as **Lastname Firstname 2F WS and Theater Over $500 Query** and then create a paper or electronic printout as directed. **Close** the query.

7 **Create** a query in **Query Design** view using both tables to answer the question, *Which Events were held on one of the sports fields, for which Renter Name, and what was the Rental Fee in order of lowest fee to highest fee?* (Hint: Use a wildcard with the word *Field*.) Display the fields in the order listed in the question. Five records meet the criteria. **Save** the query as **Lastname Firstname 2F Field Usage Query** and then with **Normal** margins, create a paper or electronic printout as directed. **Close** the query.

8 The college Alumni Association will donate money to the Building Fund in an amount based on 10 percent of total facility rental fees. **Create** a query in **Query Design** view to answer the question, *In ascending order by Event ID, what will be the total of each Rental Fee if the Alumni Association donates an additional 10% of each fee?* (Hint: First compute the amount of the donation, name the new field **Donation Amount** and run the query to view the results. Then calculate the new rental fee and name the new field **Rental Fee with Donation**) **Run** the query.

Switch back to **Design** view, change the properties of the new fields to display in **Currency** format with **0** decimal places, and then **Run** the query again. For *EVENT-1244*, the *Donation Amount* is *$150* and the *Rental Fee with Donation* is *$1,650*. Apply **Best Fit** to the columns in the query results. **Save** the query as **Lastname Firstname 2F Alumni Donation Query** and then create a paper or electronic printout as directed. **Close** the query.

9 **Create** a query in **Query Design** view using the **2F Events** table and the **Sum** aggregate function to answer the question, *In descending order by Rental Fee, what are the total Rental Fees for each Event Type?* Change the properties of the appropriate field to display **Currency** format with **0** decimal places, and then **Run** the query. For a *Sports* Event Type, Rental Fees total *$8,900*. Apply **Best Fit** to the columns in the query results. **Save** the query as **Lastname Firstname 2F Total Fees by Event Query** and then create a paper or electronic printout as directed. **Close** the query.

(Project 2F Events and Clients continues on the next page)

Mastering Access | Project 2F Events and Clients (continued)

10 By using the **Query Wizard**, create a **Crosstab Query** based on the **2F Events** table. Select **Time** as the **row headings** and **Event Type** as the **column headings**. **Sum** the **Rental Fee** field. Name the query **Lastname Firstname 2F Time and Type Crosstab Query** Change the design to display **Currency** format with **0** decimal places in the appropriate column, and then apply **Best Fit** to all of the columns. This query answers the question *What are the total Rental Fees for each time of the day and for each Event Type?* Create a paper or electronic printout as directed. **Close** the query, saving changes to the design.

11 **Open** the **Navigation Pane** and widen it so that all object names display fully. In **Backstage** view, click **Close Database**, and then click **Exit**. As directed by your instructor, submit your database and the eight paper or electronic printouts—relationship report and seven queries—that are the results of this project.

End **You have completed Project 2F**

Content-Based Assessments

Apply 2A and 2B skills from these Objectives:

1. Open an Existing Database
2. Create Table Relationships
3. Sort Records in a Table
4. Create a Query in Design View
5. Create a New Query from an Existing Query
6. Sort Query Results
7. Specify Criteria in a Query
8. Specify Numeric Criteria in a Query
9. Use Compound Criteria
10. Create a Query Based on More Than One Table
11. Use Wildcards in a Query
12. Use Calculated Fields in a Query
13. Calculate Statistics and Group Data in a Query
14. Create a Crosstab Query

myitlab grader

Mastering Access | Project 2G Students and Scholarships

In the following Mastering Access project, you will assist Thao Nguyen, Director of Academic Scholarships, in using her database to answer questions about scholarships awarded to students. Your results will look similar to Figure 2.52.

Project Files

For Project 2G, you will need the following file:

a02G_Students_Scholarships

You will save your database as:

Lastname_Firstname_2G_Students_Scholarships

Project Results

Figure 2.52

(Project 2G Students and Scholarships continues on the next page)

Mastering Access | Project 2G Students and Scholarships (continued)

1 **Start** Access. From your student files, open the **a02G_Students_Scholarships** database. Save the database in your **Access Chapter 2** folder as **Lastname_Firstname_2G_Students_Scholarships** and then enable the content. **Rename** both tables by adding **Lastname Firstname** to the beginning of the table name, and then widen the **Navigation Pane** to display fully the object names.

2 **Open** the two database tables to become familiar with the data. **Close** the tables, and then create a *one-to-many* relationship between the **2G Students** table and the **2G Scholarships Awarded** table based on the **Student ID** field, and then **Enforce Referential Integrity**; *one* student can have *many* scholarships. Create the **Relationship Report**, and create a paper or electronic printout as directed, saving it with the default name. **Close** all open objects.

3 Open the **2G Scholarships Awarded** table, and then **Sort** the appropriate fields in **Ascending** order so that the records are sorted by the **Major** field. Within each Major, the records should be sorted by **Scholarship Name**. Create a paper or electronic printout, being sure to print the results on one page. **Close** the table, and do *not* save changes to the table design. **Close** the **Navigation Pane**.

4 **Create** a query in **Query Design** view using the **2G Scholarships Awarded** table to answer the question, *In alphabetical order by Scholarship Name, what is the Amount and Major for scholarships greater than or equal to $500?* Display the fields in the order listed in the question. Ten records meet the criteria. **Save** the query as **Lastname Firstname 2G Scholarships $500 or More Query** and create a paper or electronic printout as directed. **Close Print Preview**, and leave the query open.

5 Using the **2G Scholarships $500 or More Query**, create a query. **Save** the **Object As Lastname Firstname 2G Scholarships 1st Qtr Query** and then redesign the query to answer the question *Which scholarships were awarded, in chronological order by Award Date, between 1/1/16 and 3/31/16, for what amount, and what was Student ID of the student?* Display the fields in the order listed in the question, display *only* the fields listed in the question, do not restrict the amount, and sort only by date. Eight records meet the criteria. Create a paper or electronic printout as directed. **Close** the query, saving changes.

6 **Create** a query in **Query Design** view using the **2G Scholarships Awarded** table to answer the question, *Which scholarships were awarded for either Nursing or CIS majors for amounts of more than $100, listed in descending order by amount?* Display the fields in the order listed in the question. Four records meet the criteria. (Hint: If five records display, switch to **Design view** and combine the majors on one criteria line using OR.) **Save** the query as **Lastname Firstname 2G Nursing or CIS More Than $100 Query** and then create a paper or electronic printout as directed. **Close** the query.

7 **Create** a query in **Query Design** view. Use the **2G Students** table and a wildcard to answer the question, *In alphabetical order by City and in alphabetical order by Last Name, what are the Student ID, City, First Name, and Last Name of students from cities that begin with the letter A?* Display the fields in the order listed in the question. Four records meet the criteria. **Save** the query as **Lastname Firstname 2G Cities Query** Create a paper or electronic printout as directed. **Close** the query.

8 **Create** a query in **Query Design** view using the **2G Students** table and all of the table's fields to answer the question *For which students is the ZIP Code missing?* Three students are missing ZIP Codes. **Save** the query as **Lastname Firstname 2G Missing ZIP Query** and then with **Normal** margins, create a paper or electronic printout as directed. **Close** the query. Using the information that displays in the query results, an enrollment clerk can use a reference to look up the ZIP codes for the students and then enter the ZIP codes in the student records in the underlying table.

9 For each scholarship, the Board of Trustees of the college will donate an amount equal to 50 percent of each scholarship. **Create** a query in **Query Design** view. Use both tables and calculated fields to answer the question, *In alphabetical order by scholarship name, and including the first and last name of the scholarship recipient, what will the value of each scholarship be if the Board of Trustees makes a matching 50 percent donation?* (Hint: First compute the amount of the donation, naming the new field **Donation** and then calculate the new scholarship value, naming the new field **New Value**).

Run the query, switch back to **Design** view, and as necessary, change the properties of all the numeric fields to display in **Currency** format with **0** decimal places, and then **Run** the query. For the *Alexandria Historical Society Scholarship*, the *Donation* is *$150* and the *New Value* is *$450*. Apply **Best Fit** to the columns in the query results. **Save** the query as **Lastname Firstname 2G Trustee Donation Query** and then create a paper or electronic printout as directed, being sure to print the results on one page. **Close** the query.

(Project 2G Students and Scholarships continues on the next page)

GO! with Microsoft® Access 2010 Comprehensive, by Shelley Gaskin and Nancy Graviett. Published by Prentice Hall.

Mastering Access | Project 2G Students and Scholarships (continued)

10 **Create** a new query in **Query Design** view. Use the **2G Scholarships Awarded** table and the **Sum** aggregate function to answer the question *For each major, in descending order by amount, what are the total scholarship amounts?* Display the fields in the order listed in the question. Use the **Property Sheet** to display the sums in the **Currency** format with **0** decimal places. *History* majors received *$1,850* in scholarships. Apply **Best Fit** to the columns in the query results. **Save** the query as **Lastname Firstname 2G Total Scholarships by Major Query** and then create a paper or electronic printout as directed. **Close** the query.

11 **Create** a **Crosstab Query** using the **2G Scholarships Awarded** table. Use the **Student ID** field as row headings and the **Major** field as column headings to answer the question *For each student or major, what is the total scholarship Amount awarded?* Name the query **Lastname Firstname 2G Student ID and Major Crosstab Query** In **Design** view, apply **0** decimal places to the appropriate fields. Apply **Best Fit** to the columns in the query results. **Save** the query, and then as directed, create a paper or electronic printout in **Landscape** orientation—the query results will print on two pages. **Close** the query.

12 **Open** the **Navigation Pane** and widen it to display all of the object names. In **Backstage** view, click **Close Database**, and then click **Exit**. As directed by your instructor, submit your database and the ten paper or electronic printouts—relationship report, sorted table, and eight queries—that are the results of this project.

End **You have completed Project 2G**

Content-Based Assessments

Apply a combination of the 2A and 2B skills.

GO! Fix It | Project 2H Social Sciences Division

Project Files

For Project 2H, you will need the following file:

a02H_Social_Sciences

You will save your database as:

Lastname_Firstname_2H_Social_Sciences

In this project, you will correct query design errors in a database used by the Dean of Social Sciences. From the student files that accompany this textbook, open the file a02H_Social_Sciences, and then save the database in your Access Chapter 2 folder as **Lastname_Firstname_2H_Social Sciences**

To complete the project you must find and correct errors in relationships, query design, and column widths. In addition to errors that you find, you should know:

- A relationship should be created between the 2H Social Sciences Faculty table and the 2H Anthropology Dept Course Schedule table. A relationship report should be created and named **Lastname Firstname 2H Relationship Report** One faculty member can teach many courses.
- You should add your last name and first name to each query name; do *not* rename the tables.
- Several queries do not accurately reflect the result implied in the query name. Open each query and examine and correct the design of any queries that do not accurately reflect the query name.
- Be sure that all of the object names in the Navigation Pane display fully.
- Create a paper or electronic printout of the relationship report and the four queries as directed by your instructor.

End **You have completed Project 2H**

Content-Based Assessments

Apply a combination of the 2A and 2B skills.

GO! Make It | Project 2I Faculty Awards

Project Files

For Project 2I, you will need the following file:

a02I_Faculty_Awards

You will save your database as:

Lastname_Firstname_2I_Faculty_Awards

Start Access, navigate to your student files, and then open the a02I_Faculty_Awards database file. Save the database in your Access Chapter 2 folder as **Lastname_Firstname_2I_Faculty Awards** Rename the two tables to include your name, create a relationship and relationship report. Then create two queries as shown in Figure 2.53. Create paper or electronic printouts as directed.

Project Results

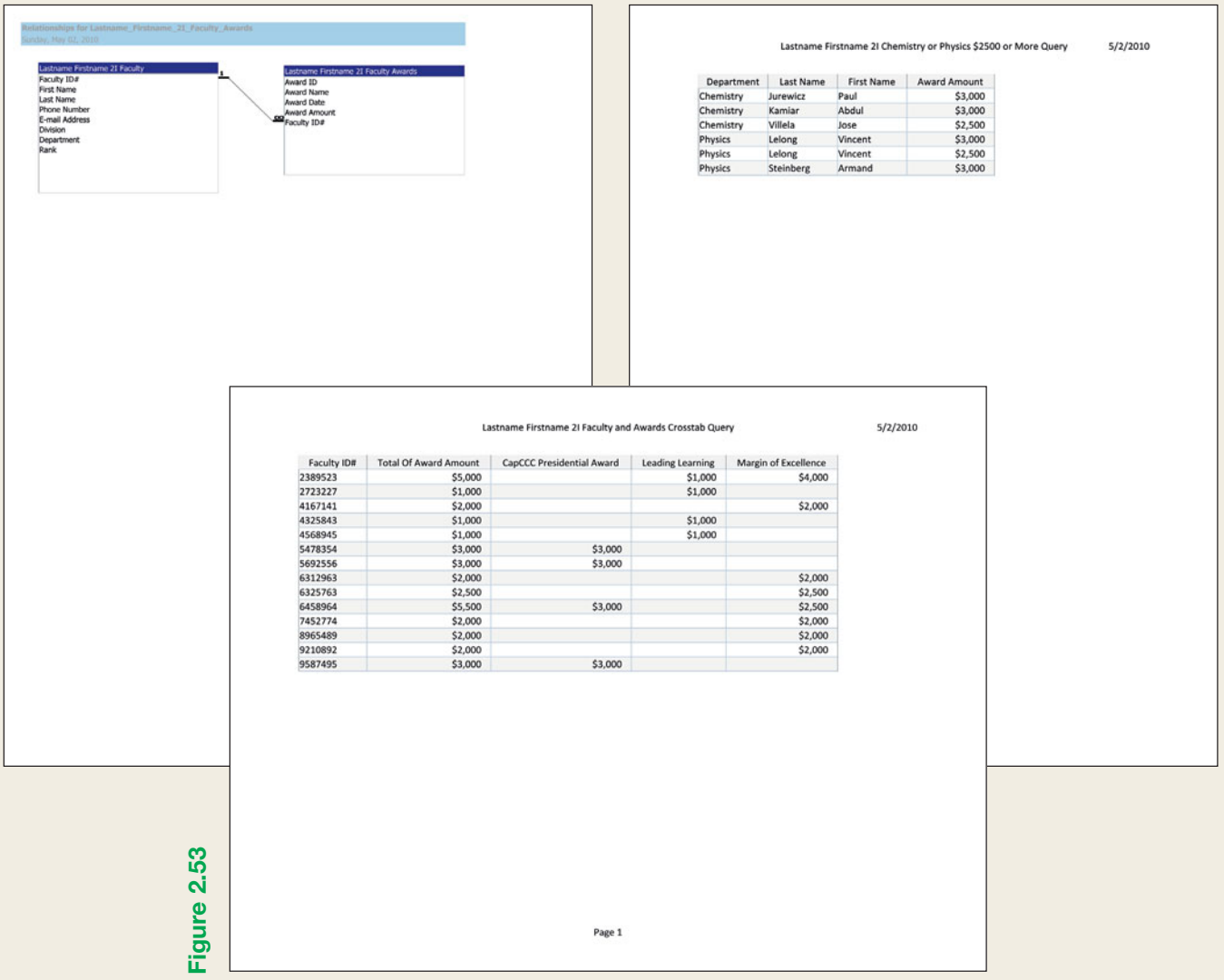


Figure 2.53

End You have completed Project 2I

Content-Based Assessments

Apply a combination of the 2A and 2B skills.

GO! Solve It | Project 2J Student Refunds

Project Files

For Project 2J, you will need the following file:

a02J_Student_Refunds

You will save your database as:

Lastname_Firstname_2J_Student_Refunds

Start Access, navigate to your student files and open the a02J_Student_Refunds database file. Save the database in your Access Chapter 2 folder as **Lastname_Firstname_2J_Student Refunds** Rename the tables by adding **Lastname Firstname** to the beginning of each table name. Create a relationship between the tables—one student can have many refunds—and then create a relationship report.

Create and save a query to answer the question, *What is the First Name and Last Name of the students who are eligible for a refund and who live in Alexandria or Falls Church sorted alphabetically by Last Name within City?* Create and save a query to answer the question, *What is the total Refund Amount for Full Time and Part Time Students?* Create and save a query to answer the question, *In ascending order by Refund Eligibility Date, what is the Last Name and First Name of students receiving a Refund of more than $50 between the dates of 8/1/16 and 12/31/16?* Apply Best Fit to all the query results, and then create paper or electronic printouts of the report and queries as directed.

Performance Element	Performance Level: Exemplary: You consistently applied the relevant skills	Proficient: You sometimes, but not always, applied the relevant skills	Developing: You rarely or never applied the relevant skills
Create relationship and relationship report	Relationship and relationship report created correctly.	Relationship and relationship report created with one error.	Relationship and relationship report created with two or more errors, or missing entirely.
Create City query	Query created with correct name, fields, sorting, and criteria.	Query created with three elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.
Create Refund query	Query created with correct name, fields, and criteria.	Query created with two elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.
Create Refund Eligibility query	Query created with correct name, fields, sorting, and criteria.	Query created with three elements correct and one incorrect.	Query created with two or more two elements incorrect, or missing entirely.

End **You have completed Project 2J**

Content-Based Assessments

Apply a combination of the 2A and 2B skills.

GO! Solve It | Project 2K Leave

Project Files

For Project 2K, you will need the following file:

a02K_Leave

You will save your database as:

Lastname_Firstname_2K_Leave

Start Access, navigate to your student files, and then open the a02K_Leave database file. Save the database in your Access Chapter 2 folder as **Lastname_Firstname_2K_Leave** Rename the tables by adding **Lastname Firstname** to the beginning of each table name. Create a relationship between the tables—one employee can have many leave transactions—and a relationship report.

Create and save a query to answer the question, *Which employees, identified alphabetically by Last Name, have used Personal Leave?* Create and save a query to answer the question, *Which employees, identified alphabetically by Last Name, have no Phone Number?* Create and save a query to answer the question, *How many Leave Transactions were for Vacation leave grouped by the Employee# field?* (Hint: In the Total row of your query, use the Count function.) Create and save a crosstab query to answer the question, *What is the total number of leave transactions for each Employee # (row) by Leave Classification (column)?* Apply Best Fit to all of the query results, and then create paper or electronic printouts of the report and queries as directed by your instructor.

		Performance Level		
		Exemplary: You consistently applied the relevant skills	**Proficient:** You sometimes, but not always, applied the relevant skills	**Developing:** You rarely or never applied the relevant skills
Performance Element	Create relationship and relationship report	Relationship and relationship report created correctly.	Relationship and relationship report created with one error.	Relationship and relationship report created with two or more errors, or missing entirely.
	Create Personal Leave query	Query created with correct name, fields, sorting, and criteria.	Query created with three elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.
	Create Phone Number query	Query created with correct name, fields, and criteria.	Query created with two elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.
	Create Vacation Leave query	Query created with correct name, fields, grouping, and aggregate function.	Query created with three elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.
	Create Crosstab query	Query created with correct name, row headings, column headings, and aggregate function.	Query created with three elements correct and one incorrect.	Query created with two or more elements incorrect, or missing entirely.

End **You have completed Project 2K**

Outcomes-Based Assessments

Rubric

The following outcomes-based assessments are *open-ended assessments*. That is, there is no specific correct result; your result will depend on your approach to the information provided. Make *Professional Quality* your goal. Use the following scoring rubric to guide you in *how* to approach the problem and then to evaluate *how well* your approach solves the problem.

The *criteria*—Software Mastery, Content, Format and Layout, and Process—represent the knowledge and skills you have gained that you can apply to solving the problem. The *levels of performance*—Professional Quality, Approaching Professional Quality, or Needs Quality Improvements—help you and your instructor evaluate your result.

	Your completed project is of Professional Quality if you:	Your completed project is Approaching Professional Quality if you:	Your completed project Needs Quality Improvements if you:
1-Software Mastery	Choose and apply the most appropriate skills, tools, and features and identify efficient methods to solve the problem.	Choose and apply some appropriate skills, tools, and features, but not in the most efficient manner.	Choose inappropriate skills, tools, or features, or are inefficient in solving the problem.
2-Content	Construct a solution that is clear and well organized, contains content that is accurate, appropriate to the audience and purpose, and is complete. Provide a solution that contains no errors in spelling, grammar, or style.	Construct a solution in which some components are unclear, poorly organized, inconsistent, or incomplete. Misjudge the needs of the audience. Have some errors in spelling, grammar, or style, but the errors do not detract from comprehension.	Construct a solution that is unclear, incomplete, or poorly organized; contains some inaccurate or inappropriate content; and contains many errors in spelling, grammar, or style. Do not solve the problem.
3-Format and Layout	Format and arrange all elements to communicate information and ideas, clarify function, illustrate relationships, and indicate relative importance.	Apply appropriate format and layout features to some elements, but not others. Overuse features, causing minor distraction.	Apply format and layout that does not communicate information or ideas clearly. Do not use format and layout features to clarify function, illustrate relationships, or indicate relative importance. Use available features excessively, causing distraction.
4-Process	Use an organized approach that integrates planning, development, self-assessment, revision, and reflection.	Demonstrate an organized approach in some areas, but not others; or, use an insufficient process of organization throughout.	Do not use an organized approach to solve the problem.

Outcomes-Based Assessments

Apply a combination of the 2A and 2B skills.

GO! Think | Project 2L Coaches

Project Files

For Project 2L, you will need the following file:

a02L_Coaches

You will save your database as:

Lastname_Firstname_2L_Coaches

Use the skills you have practiced in this chapter to assist Randy Shavrain, the Athletic Director, in answering questions about the coaches in your database **Lastname_Firstname_2L_Coaches** Create and save the relationship report with the default name, and save the queries you create with your name in the query title. Create paper or electronic printouts of the report and queries as directed by your instructor.

Mr. Shavrain needs to determine: 1) *In alphabetical order by Last Name, what is the Last Name and First Name of every coach involved with Dive activities?* 2) *In alphabetical order by Last Name, what is the Last Name and First Name of every coach involved with Basketball or Football activities?* 3) *In alphabetical order by Last Name, what is the Last Name and First Name of every coach with a Skill Specialty in Volleyball?*

End You have completed Project 2L

Apply a combination of the 2A and 2B skills.

GO! Think | Project 2M Club Donations

Project Files

For Project 2M, you will need the following file:

a02M_Club_Donations

You will save your database as:

Lastname_Firstname_2M_Club_Donations

Use the skills you have practiced in this chapter to assist Dr. Kirsten McCarty, Vice President of Student Services, in answering questions about donations collected by students to support student services in your database **Lastname_Firstname_2M_Club_Donations** Create and save the relationship report with the default name, and save the queries you create with your name in the query title. Create paper or electronic printouts of the report and queries as directed.

Dr. McCarty needs to determine: 1) *In ascending order by Last Name, what is the Last Name and First Name of donors who gave donations that are $25 or more?* 2) *What is the total of all donations grouped alphabetically by the Club Affiliation?* 3) *In alphabetical order by the student Last Name, and including the Donation ID#, what will the value of each donation be if the local sports store makes a matching 10 percent donation?* 4) *What are the total donations by Club Affiliation and Student ID?*

End You have completed Project 2M

Outcomes-Based Assessments

Apply a combination of the 2A and 2B skills.

You and GO! | Project 2N Personal Inventory

Project Files

For Project 2N, you will need the following file:

New blank Access database

You will save your database as:

Lastname_Firstname_2N_Personal_Inventory

Create a personal database containing a household inventory of your possessions. Name the new database **Lastname_Firstname_2N_Personal_Inventory** Create one or more tables with at least 10 records. Include fields such as item, room location, value, date of purchase. Your table should have items stored in several locations. Sort the table in descending order by the value of the item. Create a paper or electronic printout. Clear all sorts, and then close the table. Create at least three queries to answer specific questions about your inventory. Name the queries to reflect the question asked. Create paper or electronic printouts of your queries.

End **You have completed Project 2N**