13. In the Navigation Pane, copy the PriceStatistics query, and then rename the copied query as **PriceStatisticsByCompany**.
14. Modify the PriceStatisticsByCompany query so that the records are grouped by the Company field in the Company table. The Company field should appear first in the query datasheet. Save and run the query, and then close it.
15. Compact and repair the Supplier database, and then close it.

Case Problem 1

If you have a SAM 2010 user profile, your instructor may have assigned an autogradable version of this assignment. If so, log into the SAM 2010 Web site at www.cengage.com/sam2010 to download the instructions and start files.

APPLY

Data File needed for this Case Problem: Pinehill.accdb (*cont. from Tutorial 2*)

Pine Hill Music School After reviewing the Pinehill database, Yuka Koyama wants to modify some records and then view specific information about the students, teachers, and contracts for her music school. She asks you to update and then query the Pinehill database to perform these tasks. Complete the following:

1. Open the **Pinehill** database located in the Access1\Case1 folder, and then click the Enable Content button next to the Security Warning, if necessary.
2. In the **Teacher** table, change the following information for the record with TeacherID 55-5310: Degree is **BM** and Hire Date is **3/12/2012**. Close the table.
3. In the **Student** table, find the record with the StudentID HAV7535, and then delete the related record in the subdatasheet for this student. Delete the record for StudentID HAV7535, and then close the Student table.
4. Create a query based on the Student table that includes the LastName, FirstName, and MobilePhone fields, in that order. Save the query as **StudentPhoneList**, and then run the query.
5. In the results of the StudentPhoneList query, change the mobile phone number for Andrea Barreau to **503-579-2277**. Close the query.
6. Use Design view to create a query based on the Teacher and Contract tables. Display the LastName field from the Teacher table, and the StudentID, EndDate, LessonType, LessonLength, and LessonCost fields, in that order, from the Contract table. Sort in ascending order first on the teacher's last name, and then in ascending order by the StudentID. Save the query as **LessonsByTeacher**, and then run it.
7. Display Backstage view, and then save the LessonsByTeacher query as **CurrentLessons**.
8. Modify the CurrentLessons query to display all contracts that end on or after 7/1/2013. Save your changes, and then run the query.
9. Display Backstage view, and then save the CurrentLessons query as **CurrentGuitarLessons**.
10. Modify the CurrentGuitarLessons query to display only those records for guitar lesson contracts that end on or after 7/1/2013. Do not include the LessonType field values in the query results. Run and save the query.
11. In the CurrentGuitarLessons query datasheet, calculate the total monthly amount for current guitar lessons.
12. Change the alternate row color in the CurrentGuitarLessons query datasheet to the Theme Color named Purple, Accent 4, Lighter 60% (third row, eight column), and then change the font size to 12. Resize all columns in the datasheet to fit the data, and then save and close the query.
13. Compact and repair the Pinehill database, and then close it.

Follow the steps and use the figures as guides to create queries for a fitness center.

CREATE

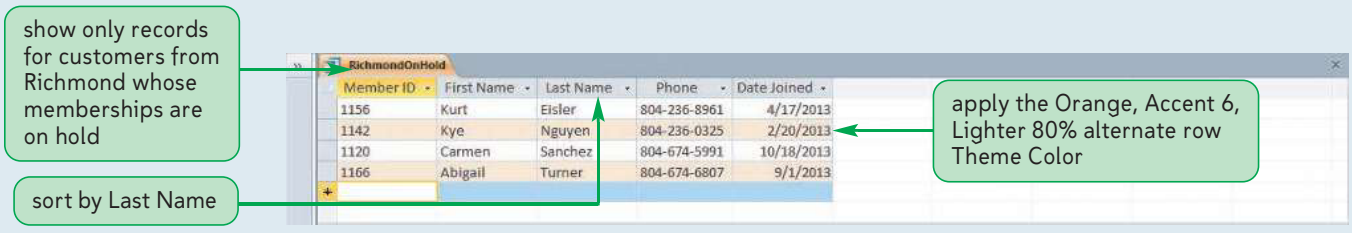
Case Problem 2

Data File needed for this Case Problem: Fitness.accdb (*cont. from Tutorial 2*)

Parkhurst Health & Fitness Center Martha Parkhurst needs to change a few records in the Fitness database and analyze the records for members enrolled in different programs at the fitness center. To help her perform these tasks, you'll update the Fitness database and create queries to answer her questions. Complete the following:

1. Open the **Fitness** database located in the Access1\Case2 folder, and then click the Enable Content button next to the Security Warning, if necessary.
2. In the **Member** table, find the record for MemberID 1158, and then change the Street value to **89 Mockingbird Lane** and the Phone to **804-751-1847**. Close the table.
3. In the **Program** table, find the record for ProgramID 205. In the subdatasheet, delete the related record from the Member table. Then delete the record for ProgramID 205 in the Program table. Close the table.
4. Use Design view to create a query that lists members who are required to have physical examinations. In the query results, display the First, Last, and DateJoined fields from the Member table, and the MonthlyFee field from the Program table. Sort the records in descending order by the date joined. Select records only for members required to take a physical. (*Hint:* The PhysicalRequired field is a Yes/No field that should not appear in the query results.) Save the query as **PhysicalsNeeded**, and then run the query.
5. Use the PhysicalsNeeded query datasheet to update the Member table by changing the Date Joined value for Ed Curran to **10/18/2013**.
6. Use the PhysicalsNeeded query datasheet to display the total Monthly Fee for the selected members. Save and close the query.
7. Use Design view to create a query that lists the MemberID, First, Last, DateJoined, ProgramType, and MonthlyFee fields for members who joined the fitness center between June 1 and June 30, 2013. Save the query as **JuneMembers**, run the query, and then close it.
8. Create and save the query to produce the results shown in Figure 3-42. Close the query when you are finished.

Figure 3-42 **RichmondOnHold query results**



EXPLORE

9. Create and save the query to produce the results shown in Figure 3-43. Close the query when you are finished.

Figure 3-43 SelectedCities query results

EXPLORE

10. Create and save the query to produce results that display statistics for the MonthlyFee field, as shown in Figure 3-44. Close the query when you are finished.

Figure 3-44 FeeStatistics query results

11. In the Navigation Pane, copy the FeeStatistics query and rename the copied query **FeeStatisticsByCity**.
12. Modify the FeeStatisticsByCity query to display the same statistics grouped by City, with City appearing as the first field. (*Hint:* Add the Member table to the query.) Run the query, and then save and close it.
13. Compact and repair the Fitness database, and then close it.

Explore some new skills to create queries for a recycling agency.

CHALLENGE

Case Problem 3

Data File needed for this Case Problem: Rossi.accdb (*cont. from Tutorial 2*)

Rossi Recycling Group Tom Rossi needs to modify some records in the Rossi database, and then he wants to find specific information about the donors, agencies, and donations to his not-for-profit agency. Tom asks you to help him update the database and create queries. Complete the following:

1. Open the **Rossi** database located in the Access1\Case3 folder, and then click the Enable Content button next to the Security Warning, if necessary.
2. In the **Donor** table, delete the record with DonorID 36065. (*Hint:* Delete the related record first.) Close the table.
3. Create a query based on the Agency table that includes the Agency, FirstName, LastName, and City fields, in that order. Save the query as **AgenciesByCity**, and then run it.
4. Modify the AgenciesByCity query design so that it sorts records in ascending order first by City and then by Agency. Save and run the query.
5. In the AgenciesByCity query datasheet, change the contact for the Community Development agency to **Beth Dayton**. Close the query.

6. Use Design view to create a query that displays the DonorID, FirstName, and LastName fields from the Donor table, and the Description and DonationValue fields from the Donation table for all donations over $50. Sort the query in ascending order by donation value. Save the query as **LargeDonations**, and then run the query.
7. Display Backstage view, and then save the LargeDonations query as **LargeCashDonations**.

EXPLORE

8. Modify the LargeCashDonations query to display only those records with donations valuing more than $50 in cash. Do not include the Description field values in the query results. Use the query datasheet to calculate the average cash donation. Save and close the query.
9. Use Design view to create a query that displays the AgencyID (from the Agency table), and the DonationID, DonationDate, and Description fields from the Donation table. Save the query as **SeniorDonations**, and then run the query.
10. Filter the results of the SeniorDonations query datasheet to display records for all donations to the SeniorCare Program (AgencyID K82).

EXPLORE

11. Format the datasheet of the SeniorDonations query so that it does not display gridlines, uses an alternate row Standard Color of Maroon 2, and displays a font size of 12. (*Hint*: Use the Gridlines button in the Text Formatting group on the Home tab to select the appropriate gridlines option.) Resize the columns to display the complete field names and values. Save your changes.
12. Display Backstage view, and then save the SeniorDonations query as **ComputerOrYouthDonations**.
13. Modify the ComputerOrYouthDonations query to display donations of "Computer equipment" or those to the After School Youth agency (AgencyID Y68). Sort the records in ascending order first by Description and then by AgencyID. Run, save, and then close the query.

EXPLORE

14. Use Design view to create a query (based on all three tables in the database) that displays the DonorID (Donor table), Agency, Description, and DonationValue fields for all donations that require a pickup. (*Hint:* The Pickup field is a Yes/No field that should not appear in the query results.) Save the query as **DonationsAfterPickupCharge**. Create a calculated field named **Net Donation** that displays the results of subtracting $8.75 from the DonationValue field values. Display the results in ascending order by donation value. Run the query, and then modify it to format the calculated field as Currency. Run the query again and resize the columns in the datasheet to their best fit. Save and close the query.

EXPLORE

15. Use the **Donation** table to display the sum, average, and count of the DonationValue field for all donations. Then complete the following:
    a. Specify column names of **Total Donations**, **Average Donation**, and **Number of Donations**.
    b. Save the query as **DonationStatistics**, and then run it.
    c. Modify the field properties so that the values in the Total Donations and Average Donation columns display two decimal places and the Standard format. Run the query and resize the columns in the datasheet to their best fit. Save and close the query.
    d. In the Navigation Pane, create a copy of the DonationStatistics query named **DonationStatisticsByAgency**.
    e. Modify the DonationStatisticsByAgency query to display the sum, average, and count of the DonationValue field for all donations grouped by Agency, with Agency appearing as the first field. (*Hint*: Add the Agency table to the query.) Sort the records in descending order by Total Donations. Save, run, and then close the query.
16. Compact and repair the Rossi database, and then close it.

Explore some new skills to create queries for a luxury rental company.

CHALLENGE

Case Problem 4

Data File needed for this Case Problem: GEM.accdb (*cont. from Tutorial 2*)

GEM Ultimate Vacations Griffin and Emma MacElroy want to modify some records, and then analyze data about their clients and the luxury properties they rent. You'll help them update and query the GEM database. Complete the following:

1. Open the **GEM** database located in the Access1\Case4 folder, and then click the Enable Content button next to the Security Warning, if necessary.
2. In the **Guest** table, delete the record with a GuestID of 224, and then close the table.
3. Create a query based on the Property table that includes the PropertyName, Location, Country, NightlyRate, and PropertyType fields, in that order. Sort in ascending order based on the NightlyRate field values. Save the query as **PropertiesByRate**, and then run the query.

EXPLORE

4. In the results of the PropertiesByRate query, change the nightly rate for the Hartfield Country Manor property to $2,500, and then use the datasheet to display the number of properties (using the Property Type column) and the average nightly rate. Save and close the query.
5. Create a query that displays the GuestLast, City, State/Prov, ReservationID, StartDate, and EndDate fields. Save the query as **GuestTripDates**, and then run the query. Change the alternate row color in the query datasheet to the Standard Color Purple 2. In Datasheet view, use an AutoFilter to sort the query results from oldest to newest Start Date. Resize the datasheet columns to their best fit, and then save and close the query.
6. Create a query that displays the GuestLast, City, ReservationID, People, StartDate, and EndDate fields for all guests from Illinois (IL). Do not include the State/Prov field in the query results. Sort the query in ascending order by City. Save the query as **IllinoisGuests** and then run it. Resize datasheet columns to their best fit, as necessary, and then save and close the query.

EXPLORE

7. Create a query that displays the GuestLast, City, State/Prov, ReservationID, StartDate, and PropertyID fields for all guests who are not from Illinois or who are renting a property starting in the month of July 2013. (*Hint*: You must specifically type the quotation marks around the state abbreviation "IL" in the criteria.) Sort the query in descending order by StartDate. Save the query as **OutOfStateOrJuly**, and then run the query. Resize datasheet columns to their best fit, as necessary, and then save the query.
8. Display Backstage view, and then save the OutOfStateOrJuly query as **OutOfStateAndJuly**.
9. Modify the OutOfStateAndJuly query to select all guests who are not from Illinois and who are renting a property beginning in the month of July 2013. Sort the query in ascending order by StartDate. Run the query, and then save and close it.
10. Create a query that displays the ReservationID, StartDate, EndDate, PropertyID, PropertyName, People, and Rate fields for all reservations. Save the query as **RentalCost**. Add a field to the query named **Cost Per Person** that displays the results of dividing the Rate field values by the People field values. Display the results in descending order by Cost Per Person. Run the query. Modify the query by setting the Format property for the Cost Per Person field to Currency. Run the query, resize datasheet columns to their best fit, as necessary, and then save your changes.
11. Display Backstage view, and then save the RentalCost query as **TopRentalCost**.

EXPLORE

12. Modify the TopRentalCost query in Design view to display only the top five values for the Cost Per Person field. (*Hint:* Use the Return (Top Values) box in the Query Setup group on the Design tab.) Save, run, and then close the query.

13. Use the Reservation table to determine the minimum, average, and maximum rental rate values for all reservations. Then complete the following:
    a. Specify column names of **Lowest Rate**, **Average Rate**, and **Highest Rate**.
    b. Save the query as **RateStatistics**, and then run the query.
    c. In Design view, specify the Standard format and two decimal places for each column.
    d. Run the query, resize all the datasheet columns to their best fit, save your changes, and then close the query.
    e. Create a copy of the RateStatistics query named **RateStatisticsByCountry**.
    f. Revise the RateStatisticsByCountry query to display the rate statistics grouped by Country of the property, with Country appearing as the first field. Save your changes and then run and close the query.
14. Compact and repair the GEM database, and then close it.

ASSESS

SAM: Skills Assessment Manager

For current SAM information, including versions and content details, visit SAM Central (http://samcentral.course.com). If you have a SAM user profile, you may have access to hands-on instruction, practice, and assessment of the skills covered in this tutorial. Since various versions of SAM are supported throughout the life of this text, check with your instructor for the correct instructions and URL/Web site for accessing assignments.

ENDING DATA FILES