

Problem Solving Using Access 2007

LEARNING OUTCOMES

1. Describe the process of using the Query Wizard using Access.
2. Describe the process of using the Design view for creating a query using Access.
3. Describe the process of adding a calculated field to a query using Access.
4. Describe the process of using aggregate functions to calculate totals in queries using Access.
5. Describe how to format results displayed in calculated fields using Access.

Introduction

A *query* is a tool for extracting, combining, and displaying data from one or more tables, according to criteria you specify. For example, in a book inventory database, you could create a query to view a list of all hardcover books with more than 500 pages that you purchased in the past five months. In a query, you can sort information, summarize data (display totals, averages, counts, and so on), display the results of calculations on data, and choose exactly which fields are shown. You can view the results of a query in a tabular format, or you can view the query's data through a form or on a report (which is covered in Plug-In T8, "Decision Making Using Access 2007"). In this plug-in, you will learn how to use the Query Wizard and Query-By-Example (QBE) tool to solve problems using Microsoft Access 2007.

Creating Simple Queries

Use the Query Wizard to create a select query. A *select query* displays data from a table or tables, based on the fields that you select, but it does not sort or filter the data. For example, if you owned a bicycle shop and wanted a list of customer names that rented bikes, you could use a simple query that would show fields from a CUSTOMER table.

To create a query using the Simple Query Wizard:

1. Open the file **T7_SlopesideBikes_Data.mdb** from the data file that accompanies this text.
2. You will see a **Security Warning**. Click on the **Options** button and select **Enable this content**. Click **OK**.

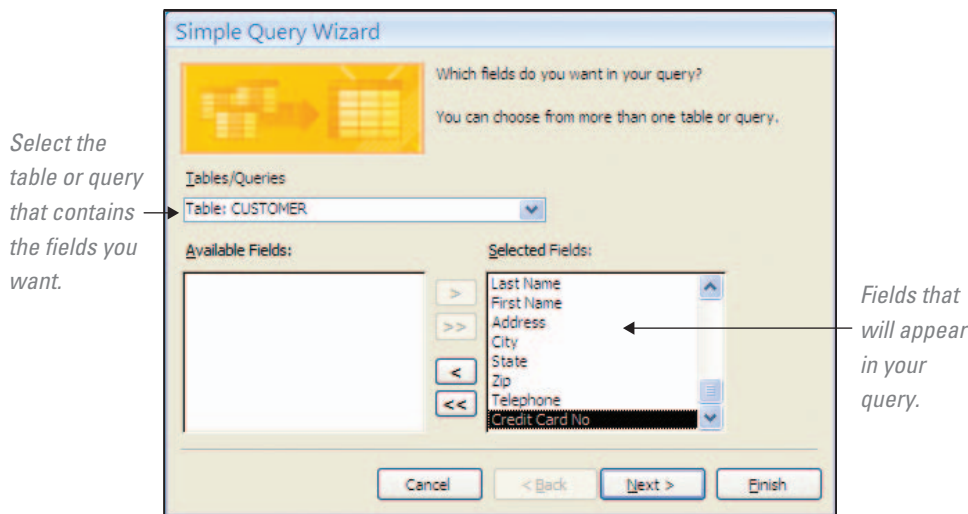


FIGURE T7.1
Create Query by Using Wizard

3. Click the **Create** tab and then click the **Query Wizard** button.
4. In the **New Query** dialog box, select **Simple Query Wizard** and then click **OK**.
5. Once the wizard opens, click the **Tables/Queries** drop-down arrow. Select **Table: CUSTOMER** (refer to Figure T7.1).
6. Add all the fields by clicking on the **right double arrow (>>)** button.
7. Click **Next**.
8. In the next window, make sure the radio button **Detail** is selected and click **Next**.
9. In the next window, type **Customer Query** as the Query title and select the radio button for **Open the query to view information**.
10. Click **Finish** to view the query in **Datasheet** view (refer to Figure T7.2).
11. Close the query (it will automatically be saved).

By modifying the query in Design view, you can specify that the query display only records that meet certain criteria or that the query display records in a specific order.

Note: Keep in mind that a query database object stores only the query definition—field names, data selection criteria, sorting orders, grouping information, and so on. It does not store the actual data that it displays; that data is stored only in the database tables. Consequently, every time you run a particular query, it shows the current state of the data stored in the database tables.

REORDERING COLUMNS IN THE SELECT QUERY DATASHEET

If you use the Query Wizard, the query datasheet displays fields in the order you added them. You can reorder columns by clicking and dragging.

To reorder columns (refer to Figure T7.3):

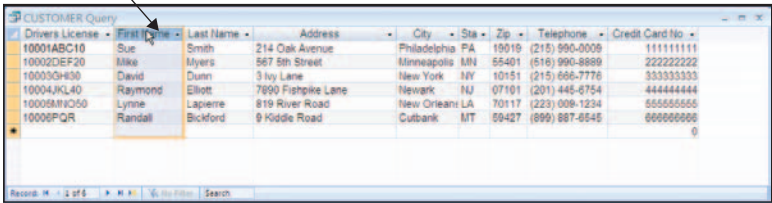
1. Open the **CUSTOMER Query** by double-clicking on it in the **Navigation Pane**.
2. Click the **First Name** field selector and then click it again and drag it to the **left** one column (i.e., **First Name** should now be to the left of **Last Name**).

	Last Name	First Name	Address	City	State	Zip	Telephone	Credit Card No.
10001ABCD	Smith	Sue	214 Oak Avenue	Philadelphia	PA	19019	(215) 990-0008	111111111
10002CDEF20	Myers	Mike	567 5th Street	Minneapolis	MN	55401	(616) 990-9889	222222222
10003GHI30	Dunn	David	3 Ivy Lane	New York	NY	10151	(215) 666-7776	333333333
10004JKL40	Elliott	Raymond	7890 Fishpike Lane	Newark	NJ	07101	(201) 445-6754	444444444
10005MN50	Lapierre	Lynne	819 River Road	New Orleans	LA	70117	(223) 009-1234	555555555
10006PQR	Bickford	Randall	9 Kiddle Road	Cutbank	MT	59427	(899) 887-8545	666666666

FIGURE T7.2
Query in Datasheet View

FIGURE T7.3
Reorder Column in
Datasheet View Query

To move a column, click and
drag a field selector.



- 3. Notice the dark, thick column border line that appears to the left of the selected column. As you move your mouse across the screen, the column border line will move with it.
- 4. Close and save the query.

Note: You can select adjacent columns by clicking a field selector and dragging the mouse across other field selectors, then clicking one of them again to drag the whole selection.

CREATING A SELECT QUERY IN DESIGN VIEW

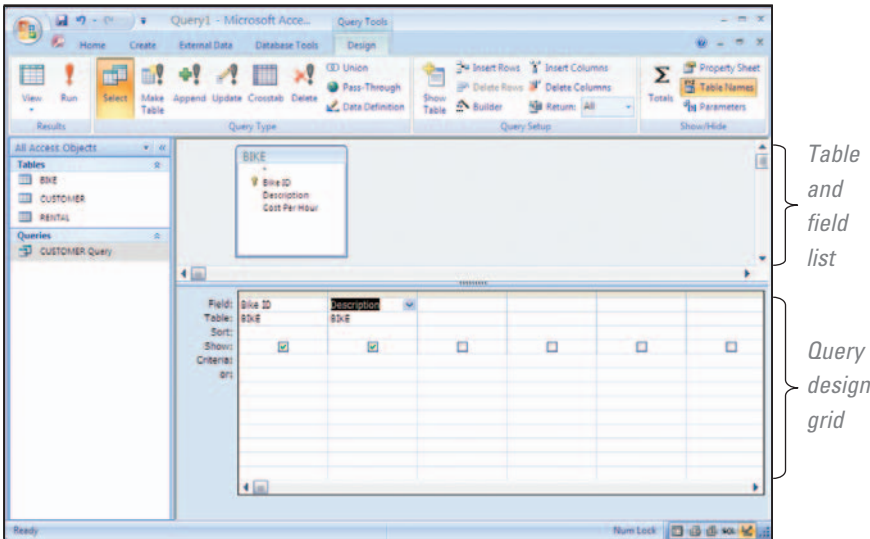
You do not have to use the Query Wizard to create select queries; you can create a new select query in Design view (which can be referred to as a *Query-By-Example*, or *QBE*, tool).

To create a select query in Design view (refer to Figure T7.4):

- 1. Click the **Create** tab and then click the **Query Design** button in the **Other** group.
- 2. Click the name of the **BIKE** table. Click the **Add** button.
- 3. Click the **Close** button to continue.
- 4. Double-click the **Bike ID** field in the field list at the top of the query window. You can also click the name and drag it to the query design grid below.
- 5. Double-click the **Description** field in the field list.
- 6. Run the query by clicking the **Run** button in the **Results** group (see Figure T7.5).
- 7. Close and save the query as **Bike List**.

Note: If you want to include all the fields from a table in your query, click and drag the asterisk (*) to the field row in the query design grid. Notice that rather than listing each field from the table separately, there is only one field called table. The * character represents a wild card. Rather than look for specific field names, the query will look for all the fields in that table. Therefore, if you later add or delete fields, you will not need to change the query design.

FIGURE T7.4
Query in Design View



Run button

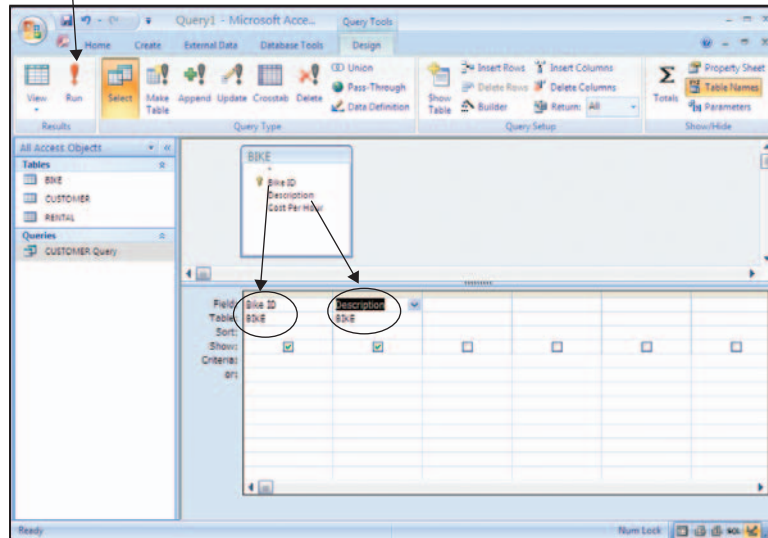


FIGURE T7.5

Run Query in Design View

You can add tables to the field list by clicking the toolbar button or selecting **Show Table** from the **Query Setup** group of the **Design** tab menu.

ADDING SELECTION CRITERIA TO A SELECT QUERY

Although a select query displays only the fields you select, by default, it will show all of the records. By modifying the select query in Design view, you can refine the query so that it shows only records that meet specific criteria. You can also hide fields so they do not display in Datasheet view.

To specify criteria for a select query:

1. In the **Navigation Pane**, click the **Bike List** query, right-click and click on **Design View**.
2. In Design view, double-click the **Cost Per Hour** field under **BIKE** in the field list.
3. Click in the **Criteria** cell under the **Cost Per Hour** field and type in **<15** (refer to Figure T7.6).

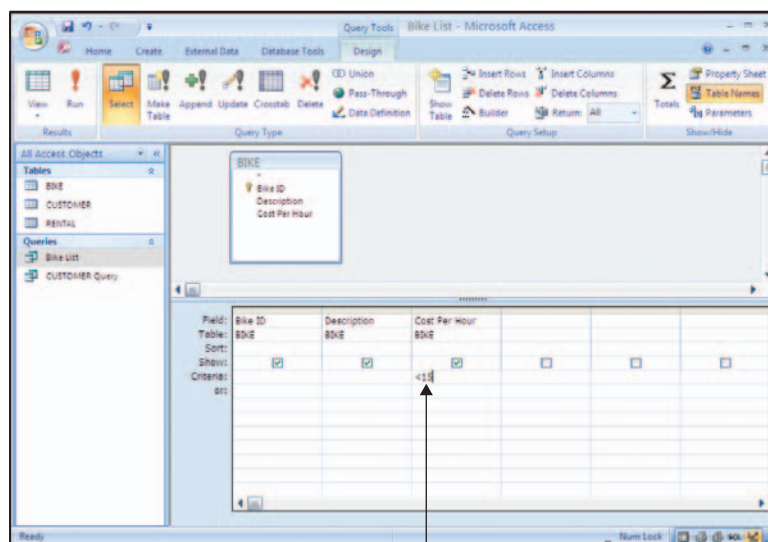


FIGURE T7.6

Criteria for a Select Query

Type the criteria value.

4. Run the query.
5. Close and save the query as **Bikes Under \$15**.

If you need to hide a query field in **Datasheet** view:

1. In **Design** view, click in the **Show** check box, in the query design grid, to remove the checkmark for the field you want to hide.
2. Run the query.

Note: Criteria expressions are not case-sensitive.

SORTING DATA IN A SELECT QUERY

You can control how records in a query appear in Datasheet view by using the *sort feature*.

To modify the select query to sort records:

1. Open the **Bikes Under \$15** query that you created in the previous section.
2. In **Design** view, click **Sort** under the **Description** field.
3. Click the drop-down arrow and select the **Ascending** button (**Sort Smallest to Largest**).
4. Run the query.
5. Close and save the query as **Sorted Bikes Under \$15**.

If you want to sort the results of a select query while in Datasheet view:

1. Click in the field that you want to sort on.
2. Click the **Sort Ascending** or **Sort Descending** toolbar button in the Sort & Filter group on the ribbon.

Note: If two fields have the sort option selected, Access will sort by the farthest left field first.

Advanced Queries

Database fields generally display the data that are entered into them. However, a calculated field will automatically figure its value based on values in other fields. Queries often include calculated fields that display values based on other values returned by the query.

To enter formulas in calculated fields, follow these steps:

1. Click the **Create** tab and then click the **Query Design** button.
2. Click the names of the **BIKE**, **CUSTOMER**, and **RENTAL** tables in the **Navigation Pane** (hold down the **CTRL** key to select each one). In the Show Table dialog box that appears, click the **Add** button.
3. Click the **Close** button to continue.
4. Double-click the **Last Name** and **First Name** fields from the **CUSTOMER** table, double-click the **Date** field from the **RENTAL** table, and double-click the **Description** field from the **BIKE** table.
5. To add a **new calculated field**, click inside the field row of a blank column, and type **Rental Amount**:—this is the name of an anonymous field (it is actually a variable name to hold the value of the calculated field).
6. Now you will enter the expression to be calculated. Type in: **([Time In]-[Time Out])*24*[Cost Per Hour]** (refer to Figure T7.7). **Note:** When referring to a field name, enclose the name in brackets. You can use standard mathematical operator symbols like * (multiplication), / (division), + (addition), - (subtraction), and ^ (exponentiation).
7. Click on the **Show** check box under the **Rental Amount** column.
8. Run the query (click **Run** button in toolbar) to see the results of the calculated field.
9. Close and save the query as **Rental Amount**.

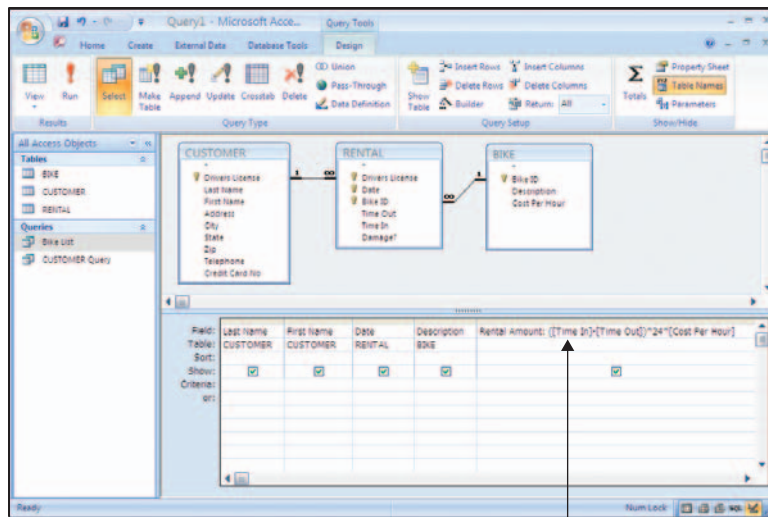


FIGURE T7.7

Expression to Be Calculated

Enter expression to be calculated.

Note: When using a calculated field to combine text fields, you usually want to hide the text fields that are being combined and show only the new calculated field.

USING THE EXPRESSION BUILDER

You do not always have to type your calculated field expressions. The *Expression Builder* lets you construct expressions with just a few clicks of the mouse. It even has a built-in checker, so you will know right away if you have made a mistake.

To create a calculated field with the Expression Builder:

1. Click the **Create** tab and then click the **Query Design** button.
2. Click the names of the **BIKE**, **CUSTOMER**, and **RENTAL** tables in the **Navigation Pane** (hold down the **CTRL** key to select each one). Click the **Add** button.
3. Click the **Close** button to continue.
4. Double-click the **Last Name** and **First Name** fields from the **CUSTOMER** table, double-click the **Date** field from the **RENTAL** table, and double-click the **Description** field from the **BIKE** table.
5. Position the insertion point in the field row of the next blank column in the query design grid. Click the **Build-er** toolbar button, found in the **Query Setup** group.
6. In the Expression Builder dialog box (see Figure T7.8), click the (button, and then double-click the **Tables** folder in the lower left pane.
7. Double-click the **Rental** subfolder. In the middle pane, double-click on **Time In**, click on the - button, double-click on **Time Out**, then double-click the) button.
8. Continue to build the expression in the Expression Builder dialog box.
9. Click **OK** to add the expression to your query.
10. Run the query to see the results of the calculated field.
11. Close and save the query as **Rental Amount 2**.

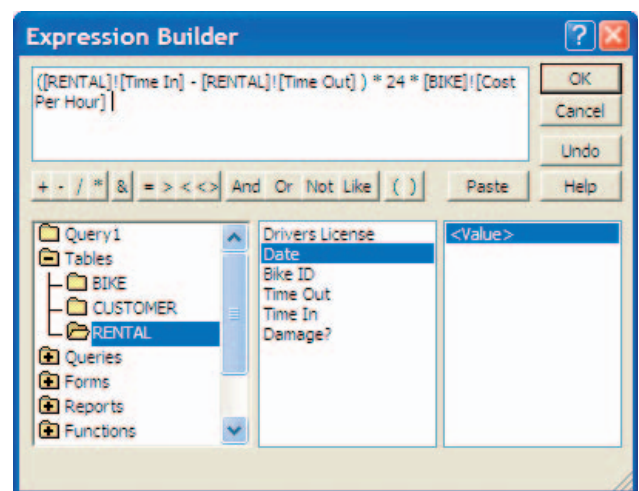


FIGURE T7.8

Query Expression Builder

When you create a new calculated field, by default, Access will set the Show option in the query design grid to off. Be sure to check the **Show** check box, or your new field will not show when you run the query.

When you create a calculated field with the Expression Builder, Access will automatically name the field **Expr1**. Rename the field by clicking in the field name box and typing the name you want before the colon that precedes the expression. Be sure not to delete the colon after the name; Access needs it to know that this is a calculated field.

USING AGGREGATE FUNCTIONS TO CALCULATE TOTALS IN QUERIES

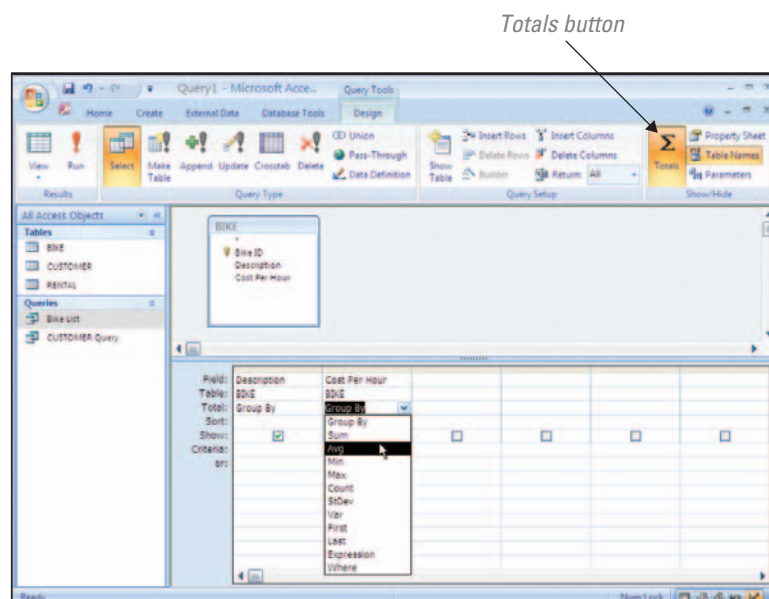
Access includes a group of powerful built-in commands known as aggregate (or total) functions. Using *aggregate functions*, you can easily calculate totals for groups of records returned by the query specifications. Aggregate functions can calculate the sum, minimum, maximum, average, count, variance, and standard deviation. In the Query Design window, you should include a field to group by as well as the field that contains the values needed for the calculation. The values of aggregate calculations are not stored in the table; instead, Access recalculates the totals each time the query is run.

To use aggregate functions in queries:

1. Click the **Create** tab and then click the **Query Design** button.
2. Click the name of the **BIKE** table and click the **Add** button.
3. Click the **Close** button to continue.
4. Double-click the **Description** and **Cost Per Hour** fields.
5. Click the **Totals** button (the Sigma icon on the menu bar in the **Show/Hide** group) to add the **Total** row to the query design grid (refer to Figure T7.9).
6. Click in the **Total** row for the **Description** field. Click the drop-down arrow and select the **Group By** function.
7. Click in the **Total** row for the **Cost Per Hour** cell. Click the arrow and select the **Avg** function (refer to Figure T7.9).
8. Click the **Run** button on the toolbar.
9. Notice that the query does not return individual records. Instead, there is one row for each unique value in the **Group by** field. Each row shows the calculated total for that group of records (refer to Figure T7.10).
10. Close and save the query as **Bike Average Cost Per Hour**.

Note: Queries with aggregate calculations are often used to form the basis for strategic analysis or statistical reports.

FIGURE T7.9
Calculate Totals in a Query



Description	Cost Per Hour
Marin Downhill FRS	\$18.00
Specialized Rockhopper	\$12.00
Specialized Stumpjumper FSR	\$18.00
Specialized Stumpjumper Hard	\$20.00
Trek Fuel 70	\$12.00
Trek Fuel 80	\$15.00
Trek Fuel 90	\$16.00

FIGURE T7.10

Calculate Totals for a Group

FORMATTING RESULTS DISPLAYED IN A CALCULATED FIELD

You can specify the format for calculated field results by changing the *format property*.

To use the format property for calculated fields:

1. Open the **Rental Amount** query in **Design** view.
2. Right-click in the **Rental Amount** calculated field.
3. Click **Properties...** from the shortcut menu.
4. Click in the **Format** box.
5. Click the drop-down arrow.
6. Scroll down and click the **Currency** format.
7. Close the **Field Properties** dialog box by clicking the **close box** in the upper right corner.
8. Click the **Run** button.
9. Close and save the query.

Note: Be careful that you are viewing the field properties and not the query properties. If the field does not have the Show check box checked, you will open the Query Properties dialog box instead of the Field Properties dialog box.

CREATING AND RUNNING QUERIES TO MODIFY DATA

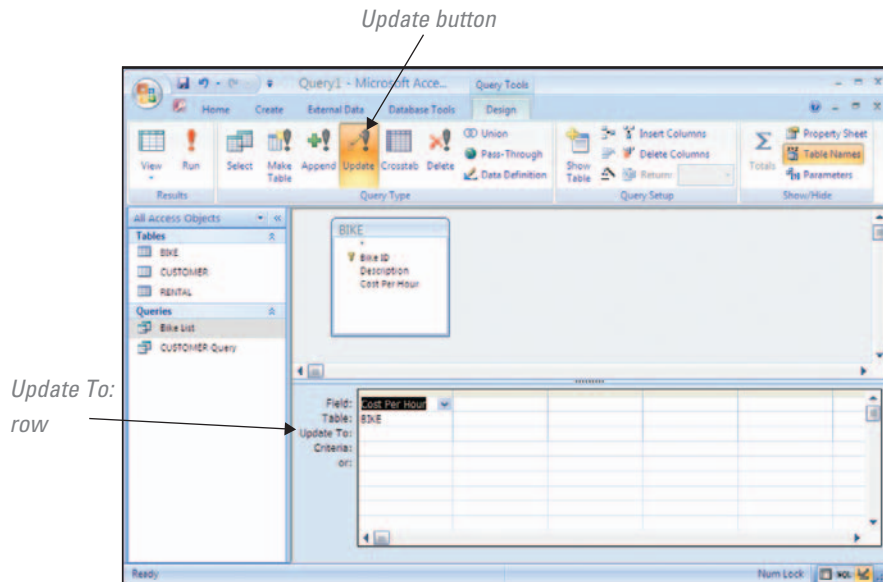
So far, this plug-in has described one major type of query: the select query. Four additional types of queries, which are known as *action queries*, can actually change your data:

1. A Make Table query.
2. An Append query.
3. A Delete query.
4. An Update query.

To use action queries:

1. Create a new query in **Design** view as you've done previously.
2. Select the **BIKE** table and click **Add** in the **Show Table** dialog box that appears.

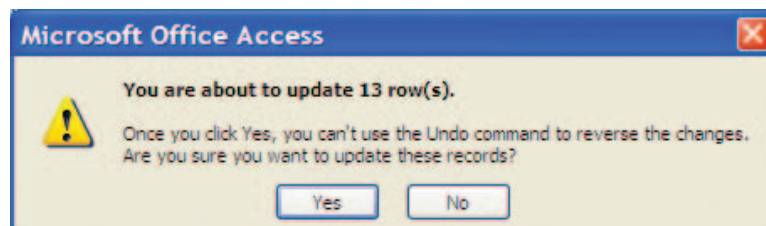
FIGURE T7.11
Query Type Option



3. Click **Close** to continue.
4. Double-click the **Cost Per Hour** field to add it to the query grid.
5. Convert the query to an action query by choosing the **Update** button in the **Query Type** group on the toolbar (refer to Figure T7.11).
6. Notice that you now have another line—**Update To:**—on the query design grid.
7. Click inside the **Update To:** row and type in **[Cost Per Hour] +.50** (this will add 50 cents to each cost per hour).
8. Run the query. You will first get a warning message, like that shown in Figure T7.12. Click **Yes**.
9. Close and save the query as **Update Query**.

Note: Make Table, Append, and Delete queries work the same way as the Update Query example above. Before running any of the queries, make a backup copy of your database. These queries can permanently modify your database, possibly removing a large amount of data.

FIGURE T7.12
Query Type Warning





A *query* is a tool for extracting, combining, and displaying data from one or more tables, according to criteria you specify. You can use the Query Wizard or Query-By-Example tool to select specific data from a table or tables. You can sort the rows of a query by a particular field, hide a particular field, enter a value to search on, summarize information, and even modify data using a query.



1. ProSwing Analysis

Carol Redden had been a part-time tennis instructor for several years. Frustrated with the tennis equipment available in the market, she started her own company. With the aid of her attorneys and investors, she founded ProSwing to design tennis equipment and apparel.

Carol developed a strong product line and was becoming successful in the United States, but she was having trouble with the firm's international marketing strategy. Carol thought that the most effective way to advertise was by getting players to use and wear her products. She wanted to be sure that the players using her line were the players followed by the fans (her target market). She knew that this would bring ProSwing attention from tennis players around the world. She decided to concentrate on the tennis players with the highest current winnings, since they were likely to be very popular and visible to the fans. She would approach the top 25 international men and women players about using ProSwing products. She would then additionally advertise during tennis matches in the four countries most represented by the top players in order to represent both the men and women top players.

Carol prepared a database after finding recent sport statistics on the Web and compiled data on the top players. Included in the database are the player's name, country of residence, most recent winnings, and gender. Carol needs your help to organize the data to plan her international strategy. She has provided you with the database file **T7_ProSwing_Data.mdb** in order to complete the following:

1. Create a query that will sort the player data in descending order by winnings. Save the query as **Player Winnings Query**.
2. Create a query that will sort the data in ascending order by Country Name and then in descending order by Winnings within each country. Change the order of the columns so that Country is the first column, followed by Winnings, and then Player. Save the query as **Country Winnings Query**.
3. Create a query that will total the winnings for each country. Use the Country Name for the criteria and sort the Winnings in descending order. Save the query as **Total Winnings for Country Query**.

2. WasteNot Recycling

WasteNot Recycling is an organization that picks up recyclables from homeowners in Boulder, Colorado, as introduced in Plug-In T6. The CUSTOMER table holds static customer information such as name, address, and phone. The CUSTOMER RECORD table holds data about each recyclable pickup. Enough test data has been added to each table to test queries (use the **T7_WasteNotRecycling_Data.mdb** file associated with this text).

The owners of WasteNot Recycling have asked you to assist with creating several queries. Specifically they need to do the following:

1. Create a query using the CUSTOMER data that will select records for customers who had their first pickup in May 2004. Sort the records by customer's Last Name. Save the query as **May Pickup Query**.
2. Create a query on CUSTOMER RECORD to determine the total weights of paper and other products each customer has had picked up. Use the CUSTOMER Last Name and First Name in the query. Save the query as **Customer Weight Query**.
3. Create a query using the Name, Street, Address, and Weight fields from the CUSTOMER DETAIL table. Enter the criteria that will select customers with less than 10 pounds in either recyclable field. Save the query as **Low Volume Query**.

3. Scale Classic Cars

Johnny Krol runs a body shop that specializes in restoring classic cars. Johnny owns three classics and began collecting scale models when his wife put her foot down and said no to building more garage space for his cars.

Although Johnny frequently used the Internet and e-mail, he had never considered starting an e-business. The Scale Classics Web site began as a technology class project for Johnny's son J.J., who created a basic text and graphics informational site. Johnny liked the site, but wanted a complex site dedicated to the serious collector. He envisioned a storefront, auction house, and collector's forums and had been unable to find such a site in his online searches. Johnny hired a local consultant to build the site, found a processing house to manage orders and payments, and began shipping scale models from the body shop.

The storefront is largely for American classic cars, which come in 1/18, 1/24, 1/43, and 1/64 scale. Popular foreign cars are also available. Johnny has hired you to create some business analysis queries; specifically he wants you to:

1. Use the **T7_ClassicCars_Data.mdb** database.
2. Create a query to select any record with "coupe" in the model name. Include the Make and Model in the query. Save the query as **Coupe Query**.
3. Create a query to select the models that cost less than \$35. The query should display the Car ID, Make, Model, and Price fields and sort the result from the highest to the lowest price. Save the query as **Less Than \$35 Query**.
4. Create a query listing classic cars grouped by their make. The query should display only the Make, Model, and Price fields. Sort by the model and calculate the average price for each make. Save the query as **Cars By Model Query**.

4. BookFinder.com

BookFinder.com is an open marketplace for books online, a one-stop e-commerce search engine where you can search through more than 100 million new, used, rare, and out-of-print books for sale.

The site is produced by a team of high-tech librarians and programmers, working since 2004 to connect readers with the books they are looking for. They are part of their own audience; members of the BookFinder.com team are heavy readers and buy several dozen books every year using BookFinder.com. They also blog about their work, connecting readers and booksellers from around the world and supporting public access to a strong, diverse bookselling industry.

BookFinder.com was first developed in 2004 by then-19-year-old University of Denver undergraduate Charles Cook (as a personal Web site). Over the years, the site has grown to become one of the best online resources for book-related e-commerce, as evidenced

by the great feedback received from users and the press. Whether customers collect rare books or buy cheap paperbacks to read, BookFinder.com is an unbiased marketplace and search engine.

Charles needs help developing some custom queries for his sales and marketing team members. He has provided you with a sample data file, **T7_BookFinder_Data.mdb**. Specifically, what Charles needs is:

1. A query of all authors who have written a book that costs more than \$10 and is a novel. The query needs to have the author's First Name, Last Name, Title of Book, Price, and Category displayed. Save the query as **Authors Query**.
2. A query that calculates a 15 percent increase in the purchase price of each book. Format the calculated field with the Currency format. Sort the query on the new calculated field. Save the query as **Book Prices Query**.
3. A query that displays the minimum price of all the books. Save the query as **Min Book Price Query**.
4. A query that updates the price of each book by subtracting 50 cents. Save the query as **Less 50 Cents Query**.