

Queries

In this chapter you will learn:

- How to use a query to display selected records
- How to use character data in criteria in a query
- How to use wildcards in criteria
- How to edit a query
- How to use comparison operators and sorting
- How to use compound criteria and limit output
- How to perform calculations
- How to use computed fields
- How to compute statistics
- How to use an action query to update records
- How to use a parameter query
- How to use an action query to delete records
- How to print a query

Case: What SUP, Inc.

Kyle decides to take Meagan and Nathan to the next level of database use by explaining the query process.

“What good is all this information if you can’t use it?” Kyle asks. “Access provides you tables as a holder of information, but queries are used to ask questions of the data and to retrieve a subset of the total information available.”

“Do we have to learn a special language in order to ask questions of the database?” asks Meagan.

“No, but you do need to know some characteristics of the data so that you can use Access’s Query By Example (QBE) method of inquiry,” he responds.

Kyle decides that he’ll begin by explaining how to go about querying the database for specific information, such as what necklaces Coast has in inventory, and then continue with more complicated queries, such as which necklaces Coast has in inventory that are above one price but below another.

Querying Selected Records

Kyle explains that the querying you accomplished in your introduction to Access involved listing specific data from various tables. In these queries you

didn't establish any criteria to select specific records; you just specified what specific information you wanted from all records. The next level of inquiry requires that you establish specific criteria for certain fields to narrow down your search for particular records. These are referred to as *select* queries.

"We'll begin with character data, wildcards, and numbers to specify what records we're looking for," Kyle explains.

Using Character Data in a Select Query

Queries can be generated either by using the Query Wizard (as was done in our introduction to Access) or by using the Design view, which is more sophisticated and flexible. Kyle suggests you use the Design view to create a simple query of your inventory file, which has been updated to include a total of 20 records. If a customer were looking for a specific item that she saw elsewhere, then she might ask if you have a particular watch in stock. You could look that item up in your database by generating a query asking for a specific match to a specific name.

"Why don't we ask the database if it has a specific Seiko watch with a product number of SMA113?" asks Meagan.

"Good idea," Nathan responds. "Let's create such a query that displays the product number, name, and supplier."

To create a query using character data:

- 1 Start Access.
- 2 Click the **File** tab and then click **Open**.
- 3 Navigate the Open window to the location of this text's student files. (The location would be the CD provided with the text or from your computer lab's server.)
- 4 Double-click **ch11-01**, which should be located in a Ch 11 folder. (This is a modified version of the completed What SUP Inventory file from Chapter 10.) Click the **Enable Content** button.
- 5 Click the **File** tab and then click **Save Database As**.
- 6 Navigate the Save As window to the location where you want to save this file and type **Ch11-01_student_name** in the File Name: text box, replacing "student_name" with your name.
- 7 Click **Save**. Click the **Enable Content** button.
- 8 Click **Query Wizard** from the Queries group on the Create tab of the Ribbon.
- 9 Click **Simple Query Wizard** and then click **OK**. Click **Open** if a security notice appears.
- 10 Click **Table: Product Table** from the drop-down list of Tables/Queries.

- 11** Double-click **Product Number** and **Product Name** from the list of available fields.
- 12** Click **Table: Supplier Table** from the drop-down list of Tables/Queries.
- 13** Double-click **Supplier Name** from the list of available fields.
- 14** Click **Next** and then type **Product Query 1** as the title for the query and then click **Finish**.
- 15** Double-click the space between each field name to resize the columns so that all information is visible. Your screen should look like Figure 11.1.

Figure 11.1
Product Query 1

Product Number	Product Name	Supplier
137814	Shubu X	Boardworks
514102	Joy Ride	Boardworks
136305	Veneer Raven	Boardworks
415333	BW Hybrid	Boardworks
NA321	Nalu	Naish
NA841	Nalu Air	Naish
GS233	Glide GS	Naish
NA112	Makani	Naish
Z23111	Zoomer	Naish
CA392	Candice Appleby	Bark
BD741	Badfish	Bark
BD241	Badfish 2	Bark
234	Sleeping One	Ron House
254	Blaster	Ron House
*		

- 16** Click the **Home** tab of the Ribbon; click **View** and then click **Design View**.
- 17** Type **NA321** in the criteria section of the query in the ProductNumber column of the design grid, as shown in Figure 11.2.
- 18** Click **Run** from the Results group of the Design tab of the Ribbon.
- 19** Resize the column widths if necessary to view all of the information provided. The resulting Datasheet view of the query should look like the query solution in Figure 11.3.
- 20** Close the Query window and click **Yes** to save changes.

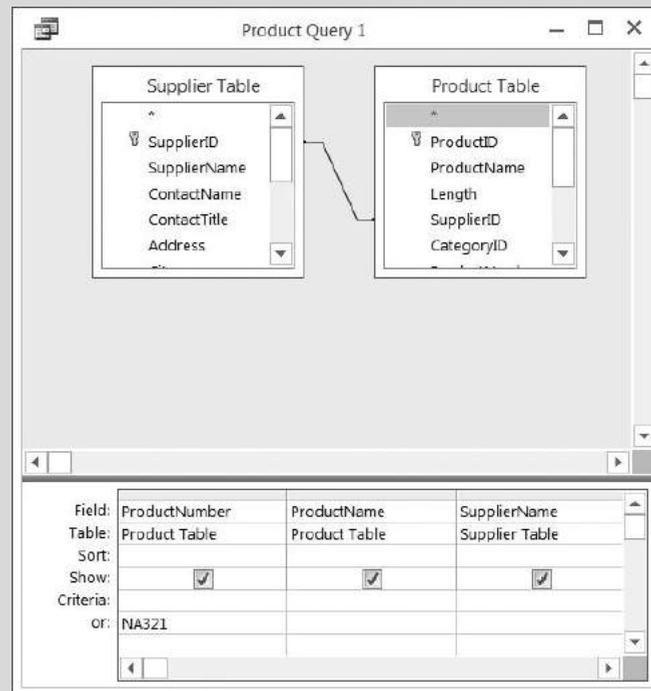


Figure 11.2

Product Query 1 (Design View)

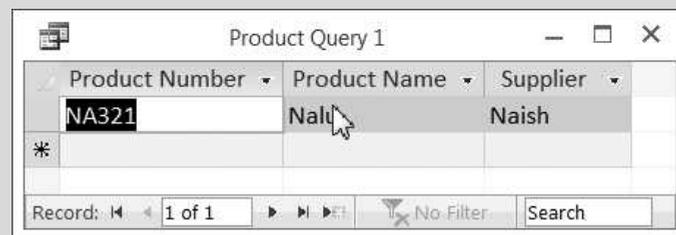


Figure 11.3

Results of Product Query 1

The character-based queries work well when you have a specific name or set of characters (like the product number) you are looking for. However, Access users are often in search of records that match some—but perhaps not all—of a specific criteria. Wildcards provide this capability.

Using Wildcards in a Select Query

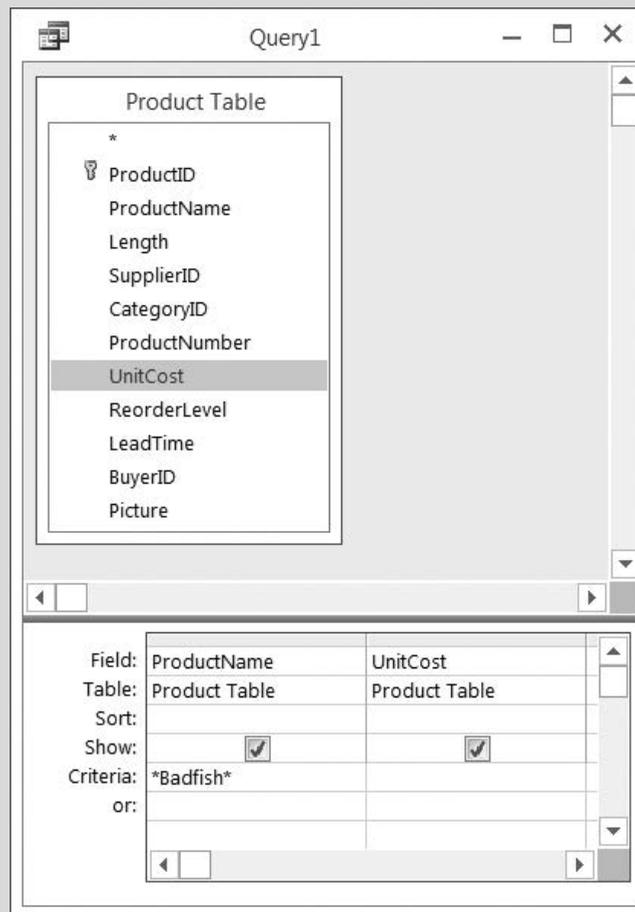
Queries using wildcards allow the user to find such things as field values, records, or filenames. The asterisk (*), which matches any number of characters, can be used as a wildcard in the first, last, or first *and* last position of the character string. For example, if Nathan wanted to list all the products he had in inventory that contained the text “Badfish” in the product name, he would type *Badfish* as the criteria under the ProductName column.

Kyle points out that there is a way to create a new query other than by using the Query Wizard. This method, which Kyle actually prefers, is called *query by design* and uses a different tool called the Query Design tool. He suggests you try this method out next to practice your use of wildcards.

To search for a product using wildcards using the Query Design tool:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2 Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window to add the Product table to the query, and then click the **Close** button.
- 3 Resize the Product Table window so that all fields are visible by clicking and holding the lower right-hand corner of the window and dragging down and to the right.
- 4 Double-click the **ProductName** and **UnitCost** fields from the Product table shown in the query.
- 5 Type ***Badfish*** in the criteria section of the query in the Product-Name column of the design grid, as shown in Figure 11.4.

Figure 11.4
Using Wildcards in a Query



- 6** Click **Run** from the Results group in the Design tab of the Ribbon. The resulting Datasheet view of the query should look like the query solution in Figure 11.5.

Product Name	Unit Cost
Badfish	\$410.00
Badfish 2	\$685.00
*	

Figure 11.5

Results of Wildcard Query

- 7** Click **View** and then **Design View**. Note that the Criteria has changed to Like “*Badfish*”, which is the proper syntax to be used when searching in a query.
- 8** Click the **Save** icon on the Quick Access Toolbar.
- 9** Type **Product Query 2** in the Save As text box and then click **OK**.
- 10** Click the **Home** tab and then close the query.

“That’s all well and good,” Meagan says. “But what if I wanted to add more fields to my query solution?”

“Good question,” Kyle responds. “It’s easy to change your existing query, but you’ll notice that Access has saved your query in a slightly different form from how you created it.”

Editing a Select Query

Once you’ve saved a query, it can be re-run any number of times. For instance, Product Query 1—which looked up a particular product number from the database—could be run again for the same or a different product number. Product Query 2 could be run again using the same criteria but asking for different fields of information.

Kyle suggests you demonstrate the query editing features of Access by modifying Product Query 1 to look up a different product number (LGR29) and adding the buyer’s name to the result. He also suggests you modify Product Query 2 to look up all product names containing the word “earrings” and adding the category name to the result.

To edit existing queries:

- 1** Right-click **Product Query 1** from the Query listing and then select **Design View** from the shortcut menu.

Trouble? You may have to resize the query window, tables, and so forth in order to view all the information. To do this, place the cursor over the edges you wish to resize.

- 2 Click **Show Table** from the Query Setup group on the Design tab of the Ribbon.
- 3 Double-click **Buyers Table** and then close the Show Table window.
- 4 Move the table windows around so that all fields and tables are viewable.
- 5 Double-click the **BuyerName** field from the Buyers table shown in the query to add that field to your query.
- 6 Type **234** in the criteria section of the query in the Product Number column replacing the previous value as shown in Figure 11.6.

Trouble? Before typing the new criteria specified for this query, note that the previous product number has automatically been placed in quotes. You don't need to type the quotes in before making this query, but Access knows that quotes are needed for future instances of the query.

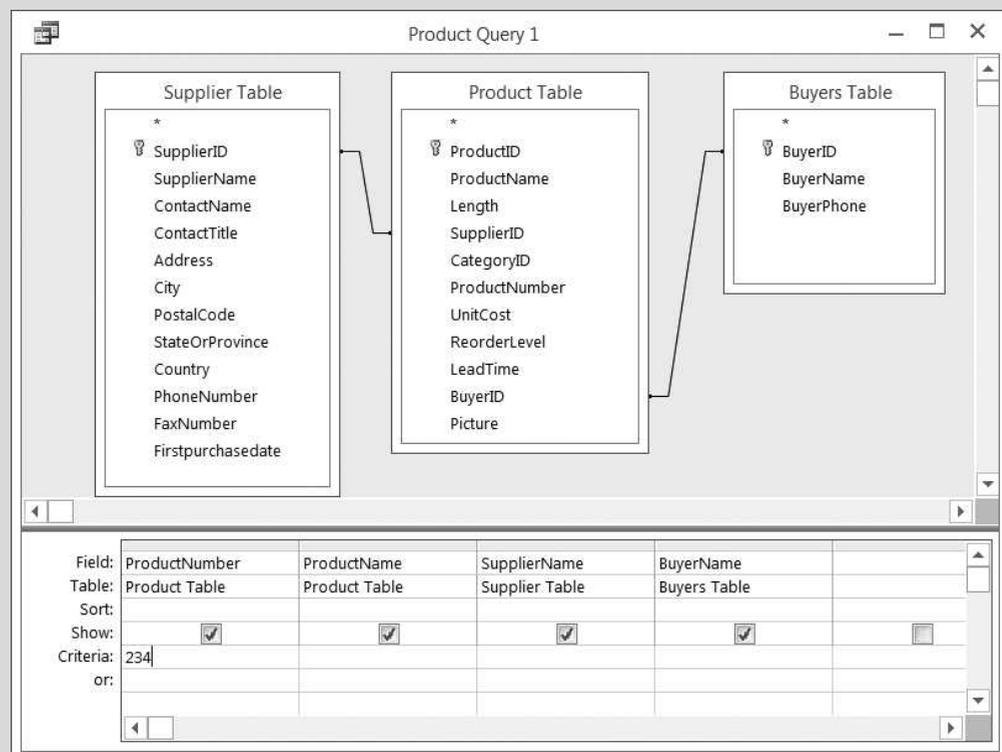


Figure 11.6
Editing Product Query 1

- 7** Click **Run** from the Results group in the Design tab of the Ribbon. The resulting Datasheet view of the query should look like the query solution in Figure 11.7.

Product Number	Product Name	Supplier	Buyer Name
234	Sleeping One	Ron House	Lopez
*			

Figure 11.7

Results of the Newly Modified Product Query 1

- 8** Click the **File** tab, then click **Save As**, then click **Save Object As**, and then click the **Save As** button.
- 9** Type **Product Query 3** as the new name and then click **OK**.
- 10** Click the **Home** tab and then close the Query window.
- 11** Right-click **Product Query 2** from the Query listing and then select **Design View** from the shortcut menu.
- 12** Click **Show Table** from the Query Setup group on the Design tab of the Ribbon.
- 13** Double-click **Category Table** and then close the Show Table window.
- 14** Move the table windows around so that all fields and tables are viewable.
- 15** Double-click the **CategoryName** field from the Category table shown in the query.
- 16** Select **Like “*Badfish*”** from the Criteria row of the ProductName column and press the [**Backspace**] key to delete this criteria. Then type **Race** in the Criteria row of the CategoryName column as shown in Figure 11.8.

Field:	ProductName	UnitCost	CategoryName
Table:	Product Table	Product Table	Category Table
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Race
or:			

Figure 11.8

Editing Product Query 2

- 17** Click **Run** from the Results group in the Design tab of the Ribbon. The resulting Datasheet view of the query should look like the query solution in Figure 11.9.

Figure 11.9
Results of the Newly Edited
Product Query 2

Product Name	Unit Cost	Category Name
Candice Appleby	\$1,150.00	Race
Glide GS	\$810.00	Race
Sleeping One	\$1,500.00	Race

- 18** Click the **File** tab, click **Save As**, then click **Save Object As**, and then click the **Save As** button.
- 19** Type **Product Query 4** as the new name and then click **OK**.
- 20** Click the **Home** tab and then close the Query window.

“How do we delete fields from a query?” Meagan asks.

“Let me show you by removing the UnitCost field from the Product Query 4 we just created,” Kyle answers.

To delete a field from a query:

- 1 Double-click **Product Query 4** to reopen the query you just worked on.
- 2 Click **View** and then click **Design View**.
- 3 Click the top of the UnitCost field column until it is selected (turns black).
- 4 Press the [**Delete**] key. Note that the UnitCost field column is removed.
- 5 Close Product Query 4 but do not save changes.

“I’m impressed!” exclaims Meagan. “This query stuff isn’t so difficult; in fact, it’s rather intuitive.”

Kyle continues with his explanation of queries by showing additional examples of the query process. So far he has used characters in his queries. Now it is time to demonstrate the use of numbers in queries and show how comparison operators can be used in a select query.

Using Comparison Operators and Sorting in a Select Query

Kyle asks if you would like to track items you’re running short of or items that you need to place on sale. Prior to creating a query to do this, you’ll need to add quantify as a field to the Product Table and then add values to that field for all products.

If so, he suggests you create a select query to find all inventory items for which quantity balances are running low (less than or equal to two items, for instance). To do so, Kyle explains that you simply modify your criteria to include comparison operators like the < symbol, which signifies “less than”. He cautions you, however, that the less-than symbol, just like the greater-than symbol (>), is not inclusive of the number specified. Thus, in order to generate a query that selects those records for which inventory quantities are less than or equal to a certain number, you must include both the < and the = symbols.

To add a quantity field and values and select records for which inventory quantities are less than or equal to 2:

- 1 Using what you learned in the previous chapter on tables, add a new field **Quantity** to the **Product Table** as a **Date Type Number**. Then enter quantity values as follows for each **ProductID**:

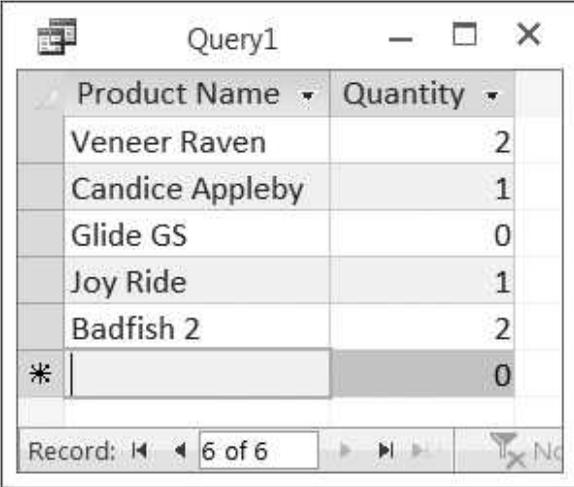
Product ID	Quantity
101	2
102	3
103	3
104	4
105	1
106	0
108	4
109	1
110	5
111	2
113	12
114	15
115	4
116	5

Then close the **Product Table**.

- 2 Click **Query Design** from the **Queries** group on the **Create** tab of the **Ribbon**.
- 3 Double-click **Product Table** from the list of tables in the **Tables** tab of the **Show Table** window to add the **Product** table to the query; then click the **Close** button.
- 4 Resize the **Product Table** window so that all fields are visible.
- 5 Double-click **Product Name** and **Quantity** fields for this query.
- 6 Type <=2 in the criteria row of the **Quantity** field of the design grid.
- 7 Click **Run**. Your results should look like Figure 11.10.
- 8 Click **File, Save As, Save Object As, and Save As** to save this new object as you’ve done before.
- 9 Type **Product Query 5** as the new name and then click **OK**.
- 10 Click the **Home** tab and then close the **Query** window.

Figure 11.10

Query Results for Products with a Quantity of 2 or Less



Product Name	Quantity
Veneer Raven	2
Candice Appleby	1
Glide GS	0
Joy Ride	1
Badfish 2	2
*	0

Record: 6 of 6

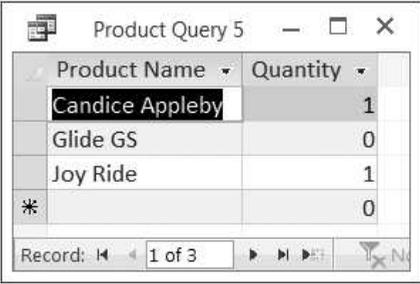
“Wouldn’t you get the same results if you set the criteria as <2?” Nathan asks. “No, you wouldn’t,” responds Kyle. “Let me demonstrate.”

To select records with criteria of <2:

- 1 Right-click **Product Query 5** and then click **Design View**.
- 2 Type <2 in the criteria row of the Quantity field of the design grid, replacing the <=2.
- 3 Click **Run**. Your results should look like Figure 11.11.

Figure 11.11

Query Results for Products with a Quantity of Less Than 2



Product Name	Quantity
Candice Appleby	1
Glide GS	0
Joy Ride	1
*	0

Record: 1 of 3

- 4 Close the query but do not save changes.

Kyle compares the two query results to demonstrate to Nathan the differences. He explains that the equal sign was important so that all records with a quantity of two or fewer were selected.

“What other comparison operators are available in Access?” asks Meagan.

Kyle responds that, in addition to the < and = symbols, Access allows you to select records using any combination of <, =, or >. For instance, you could ask for records in which the price was between \$500 and \$1,000 by using the < and > symbols together. Or you could not use the symbols and simply ask for the records in which the unit cost was between 500 and 1000.

“Could we also sort the results from lowest to highest cost?” asks Meagan.

“Yes,” Kyle answers. “Access queries allow for sorting the results in either ascending or descending order.

To select records with a cost between \$500 and \$1,000 and sort the results:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2 Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window to add the Product table to the query, and then click the **Close** button.
- 3 Resize the Product Table window so that all fields are visible.
- 4 Double-click **Product Name** and **UnitCost** fields for this query.
- 5 Type **>500 And <1000** in the criteria row of the UnitCost field of the design grid.
- 6 Click **Ascending** from the drop-down menu in the Sort cell of the UnitPrice field, as shown in Figure 11.12.

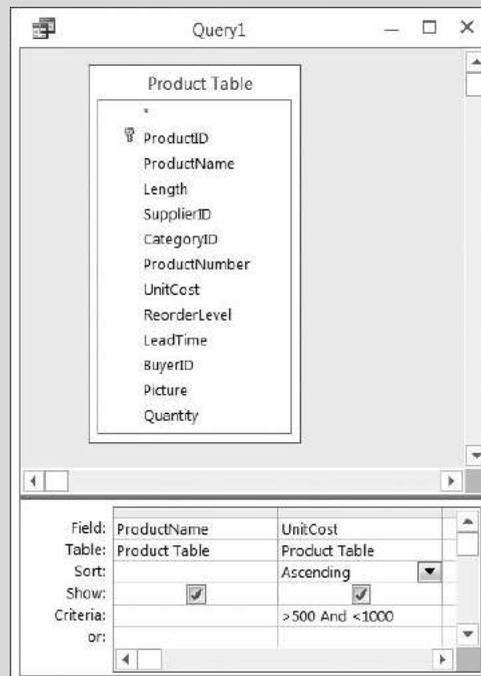


Figure 11.12

Design View for Comparison Query

- 7 Click **Run**. Your results should look like Figure 11.13.
- 8 Return to the Design View and type **Between 500 And 1000** in the criteria row of the UnitCost field.
- 9 Click **Run**. Your results will be slightly different since the concept of Between is inclusive of 500 and 1000, whereas >500 AND <1000 is exclusive of 500 and 1000.

Figure 11.13
Results of Unit Price
Query

Product Name	Unit Cost
Blaster	\$550.00
Nalu Air	\$625.00
Joy Ride	\$626.00
Badfish 2	\$685.00
Veneer Raven	\$700.00
Nalu	\$800.00
Glide G5	\$810.00

- 10** Click **File**, **Save As**, **Save Object As**, and **Save As** to save this new object as you've done before.
- 11** Type **Product Query 6** as the new name and then click **OK**.
- 12** Close the Query window.

Kyle explains that so far you've specified criteria for only one field. Access allows you to specify criteria for more than one field using specific or comparison operators.

Using Compound Criteria and Limiting Output in a Select Query

Compound criteria allow you to select records from a database using more than one field. These criteria can be applied to require that all criteria be met or they can be applied such that only one criterion need to be met before a record is selected. The field upon which the criteria are applied must be included in the query but need not be included in the output.

"Let's create a query that includes product name and unit cost for all products with a unit cost greater than \$500 and a buyer ID of 1 or 3 sorted by product name," suggests Kyle.

"Do we have to include the Buyers table in this query?" Meagan asks.

"No, since the buyer ID is a part of the Product table, it won't be necessary to include the Buyers table," answers Kyle.

To create a query that includes product name and unit price for all products with a unit cost greater than \$500 and a buyer ID of 1 or 3:

- 1** Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2** Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window to add the Product table to the query, and then click the **Close** button.
- 3** Resize the Product Table window so that all fields are visible.

- 4 Double-click **ProductName**, **UnitCost**, and **BuyerID** fields for this query.
- 5 Type **>500** in the criteria row of the UnitCost field of the design grid. Make sure the Show check box in the Unit Price field of the design grid is selected.
- 6 Type **602 Or 603** in the criteria row of the Buyer ID field of the design grid.
- 7 Be certain that the **Show** check box under all fields is checked.
- 8 Click **Ascending** from the drop-down menu in the Sort cell of the Product Name field. Your screen should look like Figure 11.14.

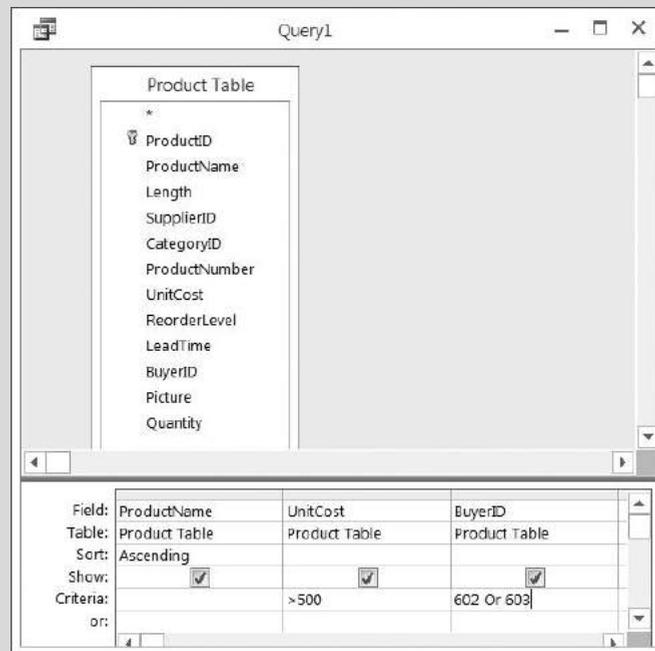


Figure 11.14

Query Design of Products with Unit Cost Greater than \$500 and Purchased by Buyer 602 or 603

- 9 Run the query. Your results should look like Figure 11.15.

Product Name	Unit Cost	BuyerID
Badfish 2	\$685.00	602
Candice Appleby	\$1,150.00	602
Glide GS	\$810.00	603
Joy Ride	\$625.00	603
Nalu	\$800.00	603
Veneer Raven	\$700.00	603
Zoomer	\$2,000.00	603

Figure 11.15

Query Results for Products with Unit Cost Greater than \$500 and Purchased by Buyer 602 or 603

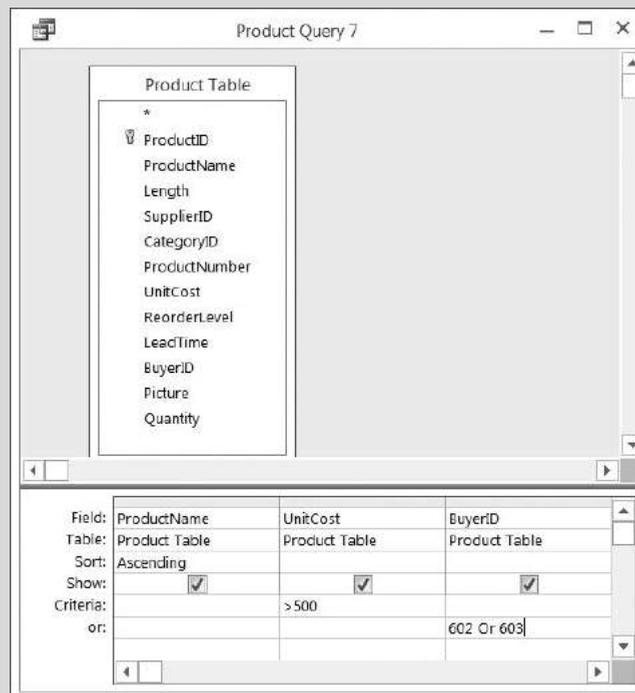
- 10 Click **File**, **Save As**, **Save Object As**, and **Save As** to save this new object as you've done before.
- 11 Type **Product Query 7** as the new name and then click **OK**.

As an alternative Kyle explains that the criteria specified on multiple fields can require that all records satisfy all criteria or that the criteria specified on multiple fields satisfy one or more criteria. For example, the previous query asked for all records in which the unit cost was greater than \$500 *and* for which the buyer was 602 or 603. Another query might ask for all records in which the unit cost was greater than \$500 *or* for which the buyer was 602 or 603. These variations, as you will see, yield different results.

To create a query that includes product name, unit price, and buyer ID for all products with a unit price greater than \$500 or a buyer ID of 602 or 603:

- 1** Using Product Query 7 that you just created, click **View** and then click **Design View**.
- 2** Click **602 Or 603** from the Criteria: row of the Buyer ID field and then press the [Delete] key.
- 3** Type **602 Or 603** in the or: row of the Buyer ID field, as shown in Figure 11.16.

Figure 11.16
Query Design of
Products with Unit Cost
Greater than \$500
or Purchased by Buyer
602 or 603



- 4** Click **Run**. Your results should look like Figure 11.17. Note that in this query, the first product listed has a unit cost less than 500 but is listed since it had a BuyerID that matched the criteria in the query.
- 5** Click **File, Save As, Save Object As**, and **Save As** to save this new object as you've done before.

Product Name	Unit Cost	BuyerID
Badfish	\$410.00	602
Badfish 2	\$685.00	602
Blaster	\$550.00	601
BW Hybrid	\$110.00	603
Candice Appleby	\$1,150.00	602
Glide GS	\$810.00	603
Joy Ride	\$626.00	603
Nalu	\$800.00	603
Nalu Air	\$625.00	601
Shubu X	\$500.00	602
Sleeping One	\$1,500.00	601
Veneer Raven	\$700.00	603
Zoomer	\$2,000.00	603
*		0

Figure 11.17

Query Results for Products with Unit Cost Greater than \$500 or Purchased by Buyer 602 or 603

6 Type **Product Query 8** as the new name and then click **OK**.

7 Close this query.

Performing Calculations

Kyle remarks that there are many types of calculations you can perform in a query. For example, you can calculate the sum or average of the values in one field, multiply the values in two fields, or calculate the date three months from the current date. When you display the results of a calculation in a field, the results aren't actually stored in the underlying table. Instead, Microsoft Access reruns the calculation each time you run the query so that the results are always based on the most current information in the database.

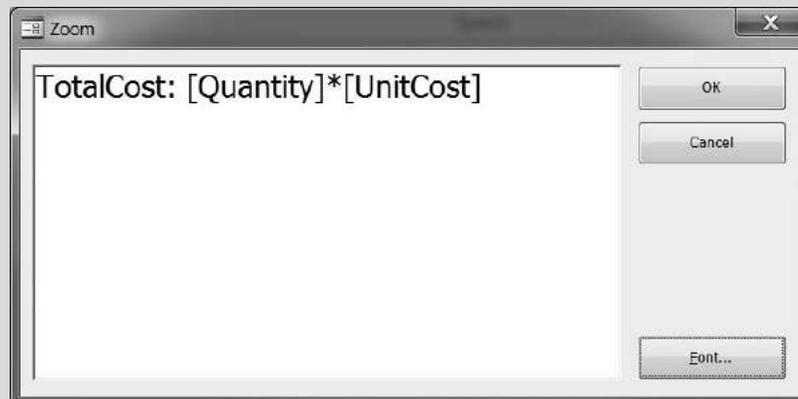
Displaying the Results of Calculations in a Field

To demonstrate the use of computations and the creation of a computed field, Kyle suggests that you and Nathan create a query that calculates the total cost of each item in inventory. He explains that to begin you'll need to create a query that includes the field's product name, quantity, and unit cost. Then you'll need to create a computed field to calculate the product of the quantity and the unit cost fields. In order to create a computed field, you'll need to type the name of the newly created field in a blank field row followed by the two fields used to compute total cost. Each field used must be surrounded by brackets, as with [Quantity]. After you've created the computed field, you'll often need to format the field to a number or currency depending on how you want it displayed.

To create an inventory cost query:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2 Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window to add the Product table to the query, and then click the **Close** button.
- 3 Resize the Product Table window so that all fields are visible.
- 4 Double-click **ProductName**, **Quantity**, and **UnitCost** fields for this query.
- 5 In the fourth column on the Field row of the design grid, type **TotalCost: [Quantity]*[UnitCost]**. (In this case TotalCost is the name of the new field and [Quantity] and [UnitCost] are the two fields being used to calculate the value for the newly created TotalCost field.
- 6 Right-click the newly created **TotalCost** field and then click **Zoom** to enlarge the formula.
- 7 Click **Font** and then click **16** in the Size column; then click **OK** to make the Zoom window text increase in size. Your Zoom window should look like Figure 11.18.

Figure 11.18
Creating a Calculated Field



- 8 Click **OK** to close the Zoom window after you verify its contents.
- 9 Right-click the **TotalCost** field again and then click **Properties**, which takes you to the Property Sheet window.
- 10 Click in the text box next to Format and then select **Currency** from the drop-down list of formats; then type **Total Cost** as the Caption, and then close the Property Sheet window.
- 11 Click **Run** to run the query to yield the results shown in Figure 11.19.
- 12 Click **File**, **Save As**, **Save Object As**, and **Save As** to save this new object as you've done before.
- 13 Type **Product Query 9** as the name of the new query and then click **OK**.
- 14 Close the query.

The screenshot shows a window titled 'Query1' displaying a table with the following data:

Product Name	Quantity	Unit Cost	Total Cost
Veneer Raven	2	\$700.00	\$1,400.00
Nalu	3	\$800.00	\$2,400.00
Shubu X	3	\$500.00	\$1,500.00
Nalu Air	4	\$625.00	\$2,500.00
Candice Appleby	1	\$1,150.00	\$1,150.00
Glide GS	0	\$810.00	\$0.00
Sleeping One	4	\$1,500.00	\$6,000.00
Joy Ride	1	\$626.00	\$626.00
Badfish	5	\$410.00	\$2,050.00
Badfish 2	2	\$685.00	\$1,370.00
Makani	12	\$135.00	\$1,620.00
BW Hybrid	15	\$110.00	\$1,650.00
Blaster	4	\$550.00	\$2,200.00
Zoomer	5	\$2,000.00	\$10,000.00
*	0		

At the bottom of the window, the status bar shows 'Record: 1 of 14', 'No Filter', and a search box.

Figure 11.19

Query with a Calculated Field

Once again, the resulting value established in the Cost field is there only temporarily and cannot be used in a later report or form unless this query is run as well. In addition to creating a computed field, Access will allow you to perform statistics and summations from within a query.

Computing Statistics

Access will allow you to calculate various statistical measures of your database, including (for example) sum, average, maximum, and minimum.

“What if I needed to know my total inventory cost?” asks Nathan.

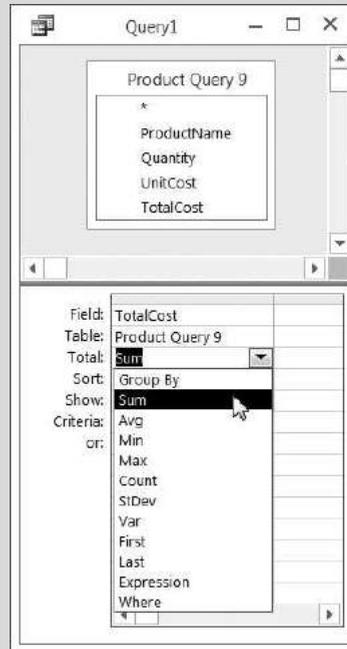
“No problem,” Kyle responds. “Access has a sum function that will work on the cost variable we just created. All we need to do is turn on the sum function and then create a sum query using the cost field we defined. To do so, we must select the query we just created instead of an existing table.”

To create a sum query:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2 Double-click **Product Query 9** from the list of queries in the Queries tab of the Show Table window then click the **Close** button.
- 3 Resize the Product Query 9 window so that all fields are visible.
- 4 Double-click **TotalCost**.
- 5 Click **Totals** from the Show/Hide group of the Design tab on the Ribbon.

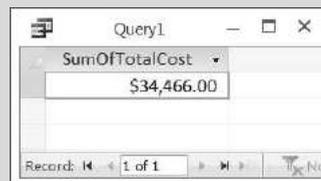
- 6** Click **Sum** from the drop-down list in the Total row, as shown in Figure 11.20.

Figure 11.20
Creating a Sum Query



- 7** Click **Run** to generate a summation of all inventory cost, as shown in Figure 11.21.

Figure 11.21
Sum Of Inventory Cost Query



- 8** Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 9** Type **Product Query 10** as the name of the new query and then click **OK**.
- 10** Close the query.

“Is there any way to find the total cost of inventory by category?” asks Nathan.

“Yes,” Kyle answers. “Access uses the concept of grouping to provide statistical information on a database.”

Kyle goes on to explain that, since the field he summed previously was a calculated field, it is imperative that the category field be a part of that query before a sum query is run. He suggests that you edit the inventory cost query to include the category field before modifying the inventory cost summary query you just created.

To create a sum query by category:

- 1 Right-click **Product Query 9** and then click **Design view** from the shortcut menu.
- 2 Click **Show Table** from the Query Setup group on the Design tab of the Ribbon.
- 3 Double-click **Category Table** and then close the Show Table window.
- 4 Double-click the **Category Name** field to add it to the query.
- 5 Close and save the revised query.
- 6 Right-click **Product Query 10** and then click **Design view** from the shortcut menu.
- 7 Click and drag the **Category Name** field to the left of the Cost field, as shown in Figure 11.22.

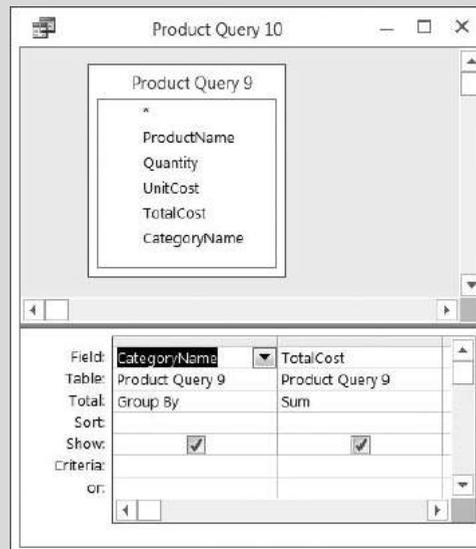


Figure 11.22

Modifying a Query to Group Totals by Category

- 8 Click **Run** to generate a summation of all inventory cost by category, as shown in Figure 11.23.

Category Name	SumOfTotalCost
Flatwater	\$6,000.00
Inflatable	\$4,000.00
Paddles	\$3,270.00
Race	\$7,150.00
River	\$11,996.00
Surf	\$2,050.00

Figure 11.23

Results of a Query to Group Totals by Category

- 9 Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 10 Type **Product Query 11** as the name of the new query and then click **OK**.
- 11 Close the query.

“That’s great information,” says Nathan. “Is there a way to find out how many Flatwater, Inflatable, et cetera, we have on hand or how many different products we have in each category?”

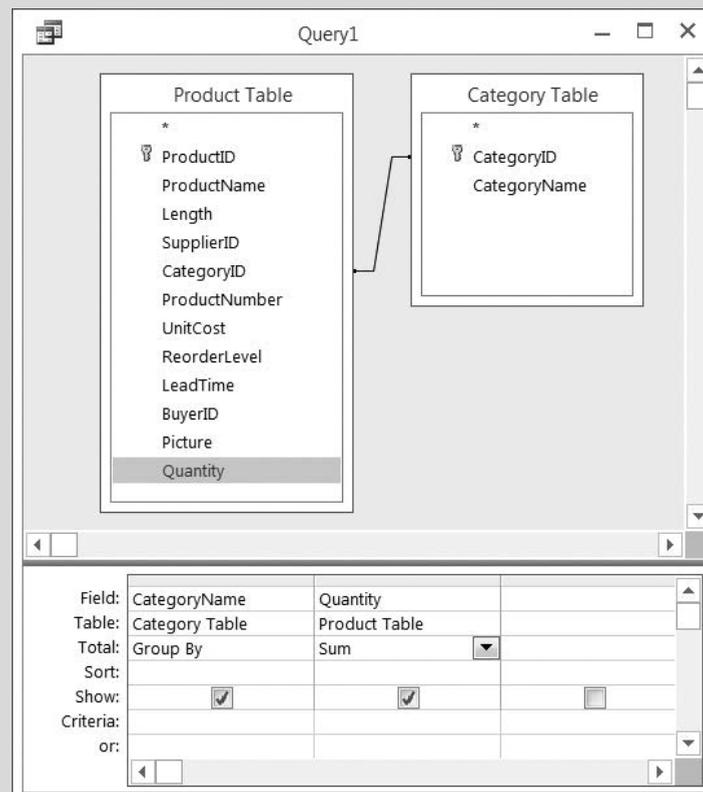
“Sure,” responds Kyle. “We can use the sum feature again to sum up the quantity of each category and the count feature to count the number of products we have by category.”

Kyle explains that the sum feature in this case can be used on the product and category tables because the information we want to display and sum is found in the tables and not in any particular query. Likewise, the count feature is used on the same two tables. He recommends that you and Meagan give it a try.

To create a sum query of quantities by category:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2 Double-click **Category Table** and **Product Table** from the list of tables in the Tables tab of the Show Table window then click the **Close** button.
- 3 Resize the window so that all fields are visible.
- 4 Double-click the **Category Name** and the **Quantity** fields to add them to the query.
- 5 Click **Totals** from the Show/Hide group of the Design tab on the Ribbon.
- 6 Click **Sum** from the drop-down list in the Total row of the Quantify field; then click **Group By** from the drop-down list in the Total row of the CategoryName field, as shown in Figure 11.24.

Figure 11.24
Design View of Query
to Sum Quantities
by Category



- Click **Run** to generate a summation of all quantities by category, as shown in Figure 11.25.

Category Name	SumOfQuantity
Flatwater	9
Inflatable	7
Paddles	27
Race	5
River	8
Surf	5

Figure 11.25

Results of a Query to Group Quantities by Category

- Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- Type **Product Query 13** as the name of the new query and then click **OK**.
- Close the query.

“That was fairly easy,” Meagan says. “Let’s let Nathan see if he can create the count query.”

To create a query to count the number of products by category:

- Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- Double-click **Category Table** and **Product Table** from the list of tables in the Tables tab of the Show Table window; then click the **Close** button.
- Resize the window so that all fields are visible.
- Double-click the **CategoryName** and the **ProductName** fields to add them to the query.
- Click **Totals** from the Show/Hide group of the Design tab on the Ribbon.
- Click **Count** from the drop-down list in the Total row of the Product-Name field and then click **Group By** from the drop-down list in the Total row of the CategoryName field, as shown in Figure 11.26.
- Click **Run** to generate a count of all products by category, as shown in Figure 11.27.
- Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- Type **Product Query 12** as the name of the new query and then click **OK**.
- Close the query.

Figure 11.26
Design View of
Query to Count
Products by Category

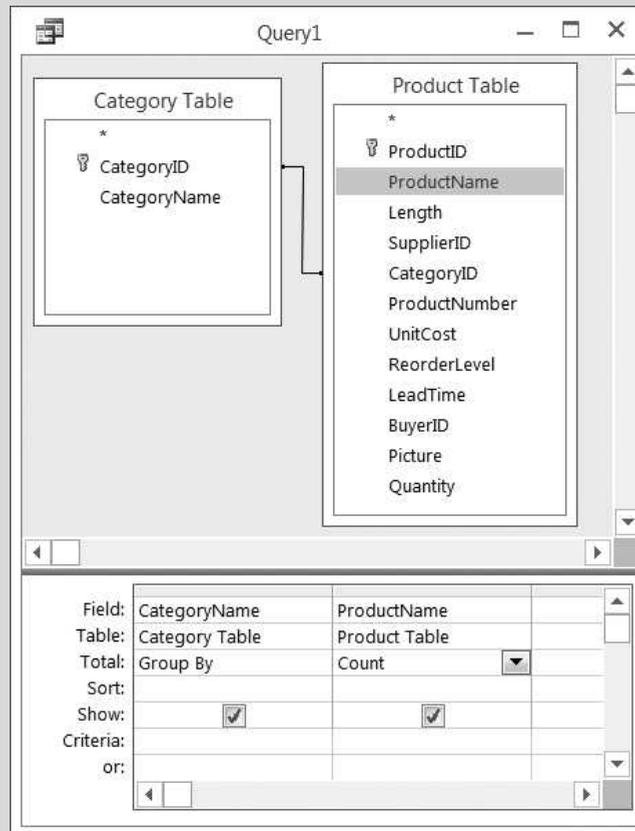


Figure 11.27
Results of a Query
to Count Products
by Category

Next Kyle will explain the process of using action queries to affect many records at one time.

Action Queries (Update, Parameter, and Delete)

An action query is a query that makes changes to many records in just one operation. There are three types of action queries that Kyle would like to explain: update, parameter, and delete. An *update query* makes global changes to a group of records in one or more tables. For example, you can raise costs or

sales prices by 10 percent for all products, or you can raise salaries by 5 percent for the people within a certain job category. With an update query, you can change data in existing tables. A *parameter query* is a query that, when run, displays its own dialog box prompting you for information—such as criteria for retrieving records or a value you want to insert in a field. A *delete query* removes a group of records from one or more tables. With delete queries, you always delete entire records, not just selected fields within records.

Update Queries

First, Nathan realizes that his database has a unit cost field for all products but it doesn't contain a unit price field. The unit cost field represents what he pays to his suppliers. A unit price would represent the unit price he charges each customer. Thus he needs to add a unit price field to the Product Table that is based on prices in effect at the beginning of the year. Later he decides to use an update query to increase the price of all products by 5 percent. To do so he'll need to use an update query that includes the Product Table and the newly created field UnitPrice. You should always back up your database before you run an update query, because you cannot undo the results of an update query. Making a backup ensures that you can always reverse your changes.

To add a unit price field and related unit prices to the Product Table:

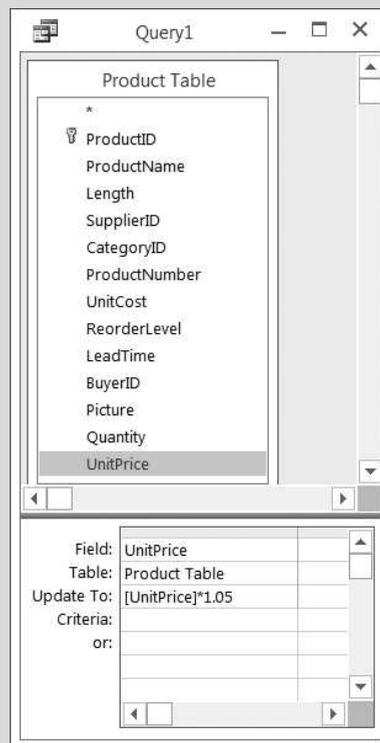
- 1 Right-click **Product Table** and select **Design View**.
- 2 Type **UnitPrice** as the name of a new field.
- 3 Select Data Type **Number** for the new field.
- 4 Select **Currency** as the Format and type **Unit Price** as the Caption.
- 5 Click **View** and then **Datasheet View** and then click **Yes** to save the table.
- 6 Add the following unit prices to each product:

ProductID	UnitPrice
101	1425
102	1625
103	1100
104	1250
105	2300
106	1700
108	3100
109	1200
110	850
111	1500
113	200
114	210
115	1000
116	3000

To back up a database and then create an update query to increase the price of all products by 5 percent:

- 1** Click the **File** tab.
- 2** Click **Save As**, then click **Save Database As**, then click **Backup Database**, and then click **Save As**.
- 3** Accept the file name provided, which should read Ch11-01_student_name followed by today's date.
- 4** Navigate the window to the location where you want to save your backup file, and then click **Save**.
- 5** Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 6** Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window; then click the **Close** button.
- 7** Resize the window so that all fields are visible.
- 8** Double-click **UnitPrice** to add it to the query.
- 9** Click **Update** from the Query Type group of the Design tab of the Ribbon.
- 10** Type **[UnitPrice]*1.05** in the Update To: row of the design grid, as shown in Figure 11.28.

Figure 11.28
Update Query



- 11** Click **Run** and then click **Yes** to update 14 records.

Trouble? Do not run this query again. Since the query as written takes a current value and increases it 5 percent, if you were to run it again, it would take the new value and increase it 5 percent again. To verify your query ran correctly and only one time, check to see if ProductID 101 has a new unit price of \$1,496.00.

- 12** Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 13** Type **Product Query 14** as the name of the new query and then click **OK**.
- 14** Close the query.
- 15** Create and run a new query that displays the Product Name, Unit Cost, and Unit Price for all Flatwater boards. Your query should look like Figure 11.29.

Product Name	Unit Cost	Unit Price
Veneer Raven	\$700.00	\$1,496.00
Nalu	\$800.00	\$1,706.00
Blaster	\$550.00	\$1,050.00

Figure 11.29

Listing of Flatwater Boards with Unit Costs and Unit Prices

- 16** Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 17** Type **Product Query 14A** as the name of the new query, click **OK**, and then close the query.

“Very slick and efficient!” says Meagan. “This software will be very helpful for us in maintaining our inventory. But do we always have to update all records, or can we limit an update to just a few records?”

“You can limit your update by setting criteria for the update, just as with a select query,” Kyle answers. “Let’s have you update all records that have a reorder level of 2 to reflect a new reorder level of 4.”

To create an update query to update reorder levels:

- 1** Create a back up file and then click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 2** Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window; then click the **Close** button.
- 3** Resize the window so that all fields are visible.

- 4 Double-click **Product Name** and **Reorder Level** to add the fields to the query.
- 5 Click **Run** to view a listing of all products and their reorder level, as shown in Figure 11.30.

Figure 11.30
Products and Reorder
Level

Product Name	Reorder Level
Veneer Raven	5
Nalu	2
Shubu X	2
Nalu Air	2
Candice Appleby	3
Glide GS	1
Sleeping One	2
Joy Ride	2
Badfish	2
Badfish 2	3
Makani	10
BW Hybrid	20
Blaster	2
Zoomer	3
*	

- 6 Click **File**, **Save As**, **Save Object As**, and **Save As** to save this new object as you've done before.
- 7 Type **Product Query 15** as the name of the new query and then click **OK**.
- 8 Close the query.
- 9 Click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 10 Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window, and then click the **Close** button.
- 11 Resize the window so that all fields are visible.
- 12 Double-click the **ReorderLevel** to add it to the query.
- 13 Click **Update** from the Query Type group of the Design tab of the Ribbon.

- 14** Type **4** in the Update To: row and then type **=2** in the Criteria row of the design grid as shown in Figure 11.31.



Figure 11.31

Update Query to Change ReorderLevel

- 15** Click **Run** and then click **Yes** to update 7 records.
- 16** Click **File, Save As, Save Object As**, and **Save As** to save this new object as you've done before.
- 17** Type **Product Query 16** as the name of the new query and then click **OK**.
- 18** Close the query.
- 19** Double-click **Product Query 15** to see the new reorder level for all affected products, as shown in Figure 11.32.
- 20** Compare Figure 11.32 with Figure 11.30 and note that all products that previously had a recorder level of 2 now have a reorder level of 4.
- 21** Close the query.

Figure 11.32
New Reorder Levels for
All Products

Product Name	Reorder Level
Veneer Raven	5
Nalu	4
Shubu X	4
Nalu Air	4
Candice Appleby	3
Glide GS	1
Sleeping One	4
Joy Ride	4
Badfish	4
Badfish 2	3
Makani	10
BW Hybrid	20
Blaster	4
Zoomer	3
*	

Parameter Queries

Another type of action query is the parameter query, in which the query prompts the user with a dialog box asking for some criteria information to select records.

“When would I want to use this type of query?” Meagan asks.

“These queries,” Kyle explains, “are used when you’re not sure what criteria the user wants to specify for record selection. They provide flexibility in the query process and will be handy in the future when forms and reports are generated for users who aren’t that familiar with Access.”

“Let’s create a parameter query to provide information on who is responsible for buying each product,” suggests Nathan. “Can this query just give us information on one buyer?”

“Yes,” responds Kyle. “I suggest you and Meagan create this query now.”

To create a parameter query for buyers:

- 1 Click **Query Design** from the Queries group on the Create tab of the Ribbon.

- 2 Double-click **Product Table** and **Buyers Table** from the list of tables in the Tables tab of the Show Table window, and then click the **Close** button.
- 3 Resize the window so that all fields are visible.
- 4 Double-click the **BuyerName**, **ProductName**, and **Quantity** fields to add them to the query.
- 5 Type **[Enter Buyer's Name]** in the Criteria: row of the Buyer Name field of the design grid, as shown in Figure 11.33.

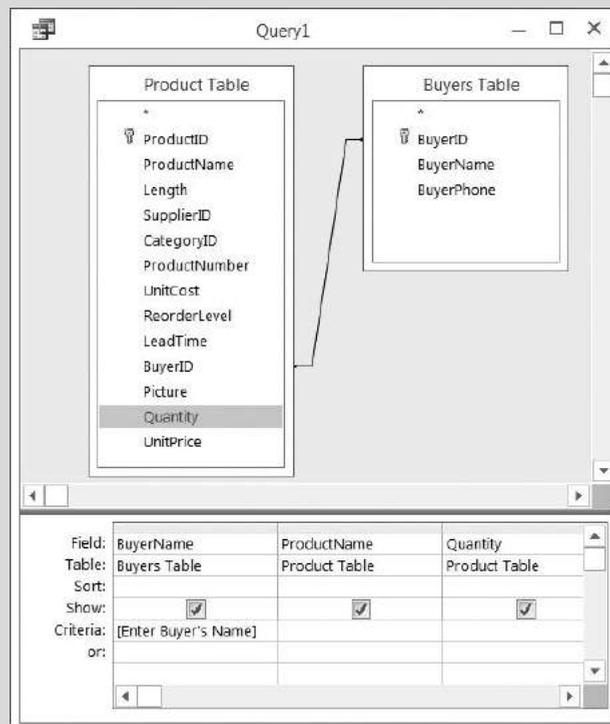


Figure 11.33

Creating a Parameter Query

- 6 Click **Run**.
- 7 Type **Lopez** in the Enter Parameter Value dialog box presented, as shown in Figure 11.34.



Figure 11.34

Enter Parameter Value Dialog Box

- 8 Click **OK** to produce the results, as shown in Figure 11.35.
- 9 Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 10 Type **Product Query 17** as the name of the new query and then click **OK**.

Figure 11.35
Products and Quantity
for Lopez

Buyer Name	Product Name	Quantity
Lopez	Nalu Air	4
Lopez	Sleeping One	4
Lopez	Makani	12
Lopez	Blaster	4

11 Close the query.

The last action query Kyle wants to demonstrate deletes specific records from your database.

Delete Queries

“Occasionally,” Kyle explains, “you might want to delete some records from your database based on specific criteria. For example, you may want to delete all records in a table that have a quantity balance of zero.”

“I’ll bet you set the criteria to = 0, right?” asks Meagan.

“Well, almost,” Kyle answers. “First you must create a specific delete query, include all the fields from the table, and specify the criteria that must be met for deletion of a record. But just as with the update queries you created previously, it always makes sense to back up your database before making massive changes.”

“Let’s give it a try,” says Nathan.

To back up a database and then create a delete query to delete specific records:

- 1** Double-click on **Product Table**.
- 2** Scroll down the table and note that ProductID 106 (Glide GS) has a quantity of 0.
- 3** Close the Product Table.
- 4** Create a backup of your database and then click **Query Design** from the Queries group on the Create tab of the Ribbon.
- 5** Double-click **Product Table** from the list of tables in the Tables tab of the Show Table window; then click the **Close** button.
- 6** Resize the window so that all fields are visible.

- 7 Double-click * (asterisk) to add all fields in the product table to the query.
- 8 Double-click **Quantity**.
- 9 Click **Delete** from the Query Type group of the Design tab of the Ribbon. Note that “From” appears on the Delete: line of the design grid under the field Product Table.* and that “Where” appears on the Delete: line under Quantity.
- 10 Type **0** in the Criteria: row of the design grid under the column Quantity, as shown in Figure 11.36.

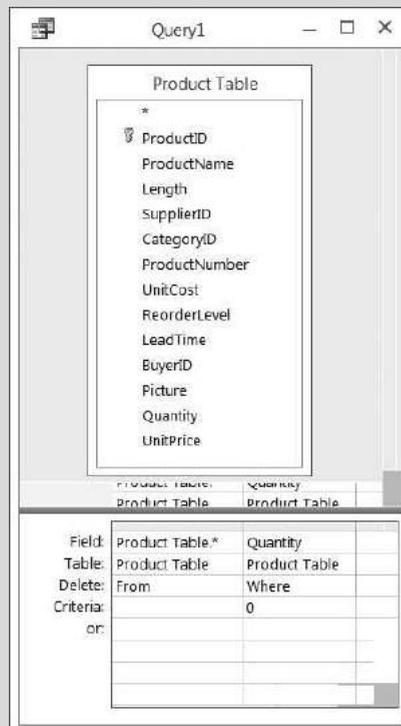


Figure 11.36

Creating a Delete Query

- 11 Click **Run** and then click **Yes** (in the dialog box presented) to delete one record from the Product table.
- 12 Click **File, Save As, Save Object As, and Save As** to save this new object as you've done before.
- 13 Type **Product Query 18** as the name of the new query and then click **OK**.
- 14 Close the query.
- 15 Double-click **Product Table** and note that there are no more products with a quantity of 0. (Product ID 106 has been removed.)
- 16 Close the Product table.

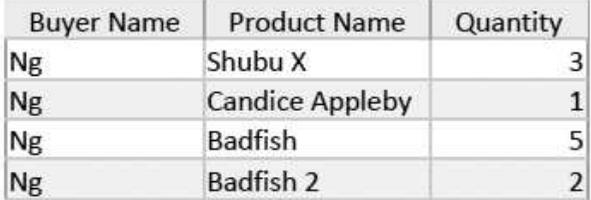
Print a Query

All that is left is to print a query. Kyle explains that printing a query is similar to printing a table (as you did in the previous chapter), but he'll review the process anyway.

To print a query:

- 1 Double-click **Product Query 17** (as an example).
- 2 Type **Ng** as the Buyer's Name then click **OK**.
- 3 Click the **File** tab and then click **Print** located under the **File** tab and then click **Print**.
- 4 Click **OK** to print. The printed query should look like Figure 11.37.

Figure 11.37
Printed Product Query 17



Product Query 17

Buyer Name	Product Name	Quantity
Ng	Shubu X	3
Ng	Candice Appleby	1
Ng	Badfish	5
Ng	Badfish 2	2

End Note

In this chapter you have learned how to use select queries with characters, numbers, wildcards, comparison operators, and compound criteria. You've also performed calculations, computed statistics, created computed fields, and created parameter queries and action queries to update and delete records. In the next chapter you will learn how to create forms.

Chapter 11 Questions

- 1 Describe the difference between the Query Wizard and Design view methods of creating a query.
- 2 Describe the process for using criteria in a query using the Design view method.
- 3 How and why are wildcards used in a query?
- 4 Describe the process for adding to an existing query a field that exists in a table not currently included in the query.
- 5 What are the three key comparison operators used in a query?
- 6 What will compound criteria allow you to accomplish in Access?
- 7 Are the results of a calculation stored as a field in a table?
- 8 What is an action query? Give examples.
- 9 What is a parameter query?
- 10 When might you use a delete query?

Chapter 11 Assignments

- 1 Create select queries for What SUP using specific criteria (use the Ch11-02 student file).
 - a. Create and print a select query that lists the buyer name, buyer phone number, and product name of product number GS233. (*Hint:* The product number should not appear on your printed query if you answered the question correctly.) Save this query as Ch 11 Assignment 1a before you print it.
 - b. Create and print a select query that lists the category name, product name, product number, and lead times for all products with a lead time in excess of 10 days. Save this query as Ch 11 Assignment 1b before you print it.
- 2 Create select queries for What SUP. (Use the Ch11-02 student file, or use the file you used in answering the previous question.)
 - a. Create and print a select query that lists the product name, quantity, supplier contact name, and supplier phone number for any River SUPs. Save this query as Ch 11 Assignment 2a before you print it.

- b. Create and print a select query that lists the supplier name, city, and phone number for any supplier located in the state of New Jersey. Save this query as Ch 11 Assignment 2b before you print it.
- 3** Edit select queries for What SUP. (Use the Ch11-02 student file, or use the file you used in answering the previous question.)
 - a. Edit Query1 by adding the field PostalCode from the Supplier Table and deleting the field StateorProvince. Save this query as Ch 11 Assignment 3a before you print it.
 - b. Edit Query2 to include the field Category Name and to include only the category Race, sorted by product ID in ascending order. Save this query as Ch 11 Assignment 3b before you print it.
- 4** Create select queries for What SUP using comparison operators. (Use the Ch11-02 student file, or use the file you used in answering the previous question.)
 - a. Create and print a select query that lists the product name, unit cost, supplier name, and supplier contact name for any unit cost between \$300 and \$600, sorted in descending order by unit cost. Save this query as Ch 11 Assignment 4a before you print it.
 - b. Create and print a select query that lists the product name, unit cost, and category name for any unit cost less than or equal to \$700, sorted in ascending order by unit cost. Save this query as Ch 11 Assignment 4b before you print it.
- 5** Create additional select queries for What SUP using computed fields. (Use the Ch11-02 student file, or use the file you used in answering the previous question.)
 - a. Create and print a select query that lists the product name unit price, unit cost, and unit gross profit for all products. Sorted in descending order by unit gross profit price. (*Hint:* You'll need to create a computed field titled Unit Gross Profit whose formula is the product's unit price minus unit cost.) Save this query as Ch 11 Assignment 5a before you print it.
 - b. Create and print a select query that lists the product name, unit price, quantity, and sales value for all paddles. (*Hint:* You'll need to create another computed field, Sales Value, whose formula is the unit price times quantity; format the computed field to Currency using the Property Sheet.) Sort the query by unit price in ascending order. Save this query as Ch 11 Assignment 5b before you print it.
- 6** Create additional select queries for What SUP using statistics. (Use file you used in answering the previous question.)

- a. Create and print a select query that calculates the total sales value of all Flatwater SUPs in inventory. (*Hint:* You'll need to use your completed query from assignment 5b as the source for this query. If you did not complete 5b, do it now. Format this computed field to Currency with two decimal places with a caption Sales Value of all Flatwater SUPs.) Save this query as Ch 11 Assignment 6a before you print it.
 - b. Create and print a query that includes all products and lists category name, product name, unit price, unit cost, unit gross profit, quantity, and gross profit. To do this, create two computed fields: Unit Gross Profit (unit price less unit cost) and Gross Profit (unit gross profit times quantity). Both calculated fields should have a Currency format and a caption with spaces. Save this query as Ch 11 Assignment 6b before you print it.
 - c. Next create and print a query that sums the gross profit of each category, using the query just created in assignment 6b as the source. Format this totaled value as Currency with two decimal places and a caption of Gross Profit. Save this query as Ch 11 Assignment 6c before you print it.
- 7** Create action queries for What SUP. (Use the Ch11-02 student file, or use the file you used in answering the previous question.)
- a. Create and run an action query that increases the unit price for all products by 6 percent. Save this query as Ch 11 Assignment 7a. Create and print an additional query listing the product name and unit price after running the update query. Do not save this query.
 - b. Create a parameter query that lists products purchased from a particular supplier. The query should ask "Enter supplier's name:" and list the supplier name, product name, quantity, and unit cost. Save this query as Ch 11 Assignment 7b. Run and print the query entering Bark as the supplier.
 - c. Create and run a delete query that deletes all products where Quantity = 0. Save this query as Ch 11 Assignment 7c. Create and print a query listing the supplier ID and product name for all products still in the database.

Chapter 11 Case Problem 1:

KELLY'S BOUTIQUE

Note: You must have completed Case 1 in the previous chapter in order to continue working on this case.

In the last chapter you added and modified some tables for Kelly's Boutique. She would now like you to create, run, and print some select, parameter, and action queries. Make the following changes for Kelly, using the ch10-03_student_name file you created in Chapter 10. (*Note:* Open your ch10-03_student_name

file and then save it as `ch11-03_student_name` before making the indicated changes.)

Add a field `Quantity` (Data Type = Number) to the `Book` Table and then enter values as follows:

<u>Book Title</u>	<u>Quantity</u>
Angela's Ashes	18
Betsy - Tacy	4
Blueberries for Sal	5
Caddie Woodlawn	4
Deep End of the Ocean	2
Divine Secrets of the YaYa Sisterhood	3

<u>Book Title</u>	<u>Quantity</u>
Green Eggs and Ham	15
Harry Potter and the Chamber of Secrets	10
Harry Potter and the Prisoner of Azkaban	8
Harry Potter and the Sorcerer's Stone	6
Hop on Pop	8
The Cat in the Hat	4
The Notebook	3

Add a field `Markup` (Data Type = Number, Field Size = Decimal, Format = Percent, Scale = 2, Decimal Places = 0) to the `Dept` Table and then enter values as follows:

<u>Dept</u>	<u>Markup</u>
Adult	100%
Children	50%

- Create and print a select query that lists the author and book title for all books written by Seuss. Save this query as Ch 11 Kelly Case a before you print it.
- Create and print a select query that lists the book title and author for all books purchased and supervised by Barbara Manchester, sorted in ascending order by book title. Save this query as Ch 11 Kelly Case b before you print it.
- Create and print a select query that lists the book title and list price for all book titles that start with Harry. Save this query as Ch 11 Kelly Case c before you print it.
- Edit the query you just created. Add fields for author and for publisher and delete the list price field. Change the criteria from "starting with Harry" to "containing the word Secrets". (*Hint*: Place a wildcard character in front of and behind the word.) Save this query as Ch 11 Kelly Case d before you print it.
- Create and print a select query that lists the book title and the publisher contact person and phone number for all books with a list price

greater than \$17. Save this query as Ch 11 Kelly Case e before you print it.

- f. Create and print a select query that lists the ISBN, book title, and list price for all books with a list price greater than \$20 or less than \$10, sorted in ascending order by list price. Save this query as Ch 11 Kelly Case f before you print it.
- g. Create and print a select query that lists the book title, list price, quantity, and retail value (a computed field equal to list price multiplied by quantity), sorted in descending order by retail value. Be sure to format the field as Currency. Save this query as Ch 11 Kelly Case g before you print it.
- h. Create and print a select query that lists book title, list price, markup, unit cost, quantity, and cost. Unit cost is a computed field (list price divided by 1 plus markup). Cost is another computed field (unit cost times quantity). Be sure to format both computed fields as Currency and to sort the query alphabetically by book title. Save this query as Ch 11 Kelly Case h before you print it.
- i. Create and print a select query that sums the total cost of the book inventory. (*Hint:* Use the query just created in part h as your source for this new query and use the currency format.) Save this query as Ch 11 Kelly Case i before you print it.
- j. Modify the query created in part h so that it includes the Department field. Save the query as Ch 10 Kelly Case h 1, and then use that modified query as the source for a query that sums the cost of inventory by department. Save this latter query as Ch 11 Kelly Case j before you print it.
- k. Create and run an action query that increases all books' list price by 5 percent. Be sure to back up your file first! Save this query as Ch 10 Kelly Case k. Create another query that lists the book title and list price for all books, sorted alphabetically by book title. Save this query as Ch 11 Kelly Case k 1 and then print it.
- l. Create a parameter query that lists books from a particular publisher. The query should ask "Enter publisher's name:" and then list the publisher, book title, and quantity for that publisher. Save this query as Ch 11 Kelly Case l. Run and print the query after entering "Harper Collins" as the publisher.
- m. Create and run a delete query that deletes all products with a quantity of 0. Save this query as Ch 11 Kelly Case m.
- n. Create a query listing the book title and quantity for all the remaining books in inventory, sorted by quantity in descending order. Save this query as Ch 11 Kelly Case n and then print it.
- o. Make sure you keep a copy of this file for use in the next chapter.

Chapter 11 Case Problem 2:

WINE DEPOT

Note: You must have completed Case 2 in the previous chapter in order to continue working on this case.

In the last chapter you modified some tables for the Wine Depot. Now Barbara would like you to help her create some queries to extract some information from the database. Make the following changes for Barbara using the ch10-04_student_name file you created in Chapter 10. (*Note:* Open your ch10-04_student_name file and then save it as ch11-04_student_name before making the indicated changes.)

- a. Create a select query that lists the type, winery, and price for all wines on hand from winery #4. Save this query as Ch 11 Wine Depot Case a and then print it.
- b. Create a select query that lists the type, winery name, and price for all wines whose buyer name is Michael Jackson. Save this query as Ch 11 Wine Depot Case b and then print it.
- c. Create a select query that lists the SKU and type for wines types that contain the word “Blanc”. Save this query as Ch 11 Wine Depot Case c and then print it.
- d. Edit the query you just created in 2c above. Add fields for cost and vintage and remove the SKU field. Change the criteria of the query from “types that include the word Blanc” to “types that begin with the letter S”. Save this query as Ch 11 Wine Depot Case d and then print it.
- e. Create a select query that lists type, price, winery name, buyer name, and phone number for all wines with a price greater than \$35. Save this query as Ch 11 Wine Depot Case e and then print it.
- f. Create a select query that lists type, price, winery name, buyer name, and phone number for all wines with a price greater than \$35 but less than \$50. Save this query as Ch 11 Wine Depot Case f and then print it.
- g. Create a select query that lists SKU, type, winery name, price, quantity, and retail value (a computed field: quantity times price, formatted as currency) for all Chardonnay wines, sorted in descending order by retail value. Save this query as Ch 11 Wine Depot Case g and then print it.
- h. Create an update query that changes the price of all wines to be 150% of cost. (Remember to back up your file first.) Save this query as Ch 11 Wine Depot Case h.
- i. Create a parameter query that lists SKU, type, winery name, and price. The query should ask “What type of wine?” Sort the query in descending order by price. Save this query as Ch 11 Wine Depot Case i. Run the query for Pinot Noir wines and then print it.

- j. Create a select query that sums the total retail value of the Chardonnay wine inventory, formatted as currency. Be sure to list the wine type in your query. (*Hint:* Use the Ch 10 Wine Depot Case g query you created previously as the source for this new query.) Save this query as Ch 11 Wine Depot Case j and then print it.
- k. Create a select query that lists SKU, type, winery name, cost, quantity, and total cost (a computed field: quantity times cost, formatted as currency) for all wines, sorted in ascending order by SKU. Save this query as Ch 11 Wine Depot Case k and then print it.
- l. Create a select query that sums the cost of wine inventory by type (in currency format), sorted by type. (*Hint:* Use the Ch 11 Wine Depot Case k query as the source for your new query.) Save the new query as Ch 11 Wine Depot Case l and then print it.
- m. Make sure you keep a copy of this file for use in the next chapter.

Chapter 11 Case Problem 3: SNICK'S BOARD SHOP

Note: You must have completed Case 3 in the previous chapter in order to continue working on this case.

In the last chapter you modified some tables for the Snick's Board Shop. Now Caitlin would like you to help her create some queries to extract information from the database. Make the following changes for Caitlin, using the ch10-05_student_name file you created in Chapter 10. (*Note:* Open your ch10-05_student_name file and then save it as ch11-05_student_name before making the indicated changes.)

- a. Create a select query that lists the category name, manufacturer name, product name, and price for all products from Manufacturer ID 888. Save this query as Query A and then print it.
- b. Create a select query that lists the manufacturer name, product name, and price for all products with a Category ID of 1. Save this query as Query B and then print it.
- c. Create a select query that lists the product ID, category name, and style for styles that contain the word "Black". Save this query as Query C and then print it.
- d. Edit the query you just created in the previous step. Add the manufacturer name field and remove the category name field. Change the criteria of the query from styles that contain the word "Black" to styles that contain the word "Blue". Save this query as Query D and then print it.
- e. Create a select query that lists product name, price, and quantity for all products with a price greater than \$100. Save this query as Query E and then print it.

- f. Create a select query that lists product name, price, and quantity for all products with a price greater than \$100 but less than \$130. Save this query as Query F and then print it.
- g. Add a number field “Discount” to the Manufacturer table. Set the field size of this new field to Single and the Format to Percent. Set the Discount to 10% for manufacturers EMT and MOJ. All other manufacturers discount should be set to 0%. Create a select query that lists Product ID, Price, and Discounted Price (a new computed field: Price times 1 - Discount, formatted as currency) for all products sorted in ascending order by Discounted Price. Save this query as Query G and then print it.
- h. Create an update query that increases the price of all products by 10%. (Remember to back up your file first.) Save this query as Query H. Open Query G again and then print it with the new prices.
- i. Create a parameter query that lists the manufacturer name, product name, price, style, and quantity. The query should state “Enter Manufacturer ID”. Sort the query in ascending order by product name. Save this query as Query I. Run the query for Manufacturer ID SC and then print it.
- j. Create a select query that sums the total retail value of the complete boards in inventory, formatted as currency. (*Hint:* First create a query with the fields CategoryID, CategoryName, Price, Quantity, and a computed field titled Retail Value, which is the price times quantity.) Save this query as Query J - 1, and then create the summation query. Save this query as Query J - 2 and then print it.
- k. Create a select query that sums the retail value of all products by category (in currency format), sorted by type. (*Hint:* Use the Query J - 1 query, modify it by removing the criteria, then save it as Query K - 1, using Query K - 1 as the source for your new query.) Your new query should have two fields: CategoryName and SumOfRetailValue. Save the new query as Query K - 2 and then print it.
- l. Make sure you keep a copy of this file for use in the next chapter.

Chapter 11 Case Problem 4:

ROSEY’S ROSES

Note: You must have completed Case 4 in the previous chapter in order to continue working on this case. In the last chapter, you modified some tables for Rosey’s Roses. Now the company would like you to help it create some queries to extract information from the database. Make the following changes using the ch10-06_student_name file you created in Chapter 10. (*Note:* Open your ch10-06_student_name file and then save it as ch11-06_student_name before making the indicated changes.)

- a. Create a select query using the Grower ID field from the Grower table, the Description field from the Product table, and the Cost/Unit for all products from Grower ID 100. Do not show the Grower ID on the resulting query. Save this query as Query A and then print it.
- b. Create a select query that lists the Grower Name, product Description, and Quantity for all products where the Description field contains the word “Drift”. Save this query as Query B and then print it.
- c. Create a select query that lists the product Description, Grower Name, Contact, Phone Number, and Cost/Unit for all products with a Cost/Unit less than \$18. Save this query as Query C and then print it.
- d. Create a select query that lists the product Description, Grower Name, Contact, Phone Number, and Cost/Unit for all products with a Cost/Unit greater than \$18 but less than \$50. Save this query as Query D and then print it.
- e. Create and print a new table called Customer as follows:

Customer Table Structure:

Field Name	Data Type	Field Size/Format	Primary Key?
Customer Number	Text	2	Yes
Customer Name	Text	30	
Discount	Number	Single/Percent	

Customer Table Data:

Customer Number	Customer Name	Discount
10	Jan Muller	10%
11	Robert Frost	10%
12	Juliet Inch	0%

- f. Add a number field “Markup” to the Grower table. Set the field size of this new field to Single and the Format to Percent. Set the Markup to 100% for Grower ID 100 and 200 and then set the markup to 125% for Grower ID 300 and 400. Print the Grower table.

- g. Create and print a new table called Quote as follows:

Quote Table Structure:

Field Name	Data Type	Field Size	Primary Key?
Quote Number	Number		Yes
Customer Number	Text	2	
ID (from the Product table)	Number		
Quantity Ordered	Number		

Quote Table Data:

Quote Number	Customer Number	ID	Quantity Ordered
100	10	4	10
101	11	11	20
102	12	12	40

- h. Establish a relationship between the Quote table and the Product table (ID) and the Customer table and the Quote table (Customer Number) enforcing referential integrity. Print the Relationships report.
- i. Create a select query that contains the Quote Number, Customer Name, Product Table Description, Quantity Ordered, Sales Price (a new computed field: $\text{Cost/Unit} \times (1 + \text{Markup})$) formatted as currency, and Quote Amount (a new computed field: $\text{Quantity Ordered} \times \text{Sales Price}$) also formatted as currency. Save as Query E and then print.
- j. Modify the query you created in (i) above to include a new field called Cost (a computed field: $\text{Quantity Ordered} \times \text{Cost/Unit}$) and a new field called Gross Profit (a computed field: $\text{Quote Amount} - \text{Cost}$). Format both as currency. Save as Query F and then print.
- k. Create and run an update query that increases the Cost/Unit of all products by 7%. (Remember to back up your file first.) Save this query as Query G. Create a new select query that lists all products showing the ID, Description, and Cost/Unit fields. Save this query as Query H and then print it.
- l. Create a parameter query that lists the product Description and Grower Name. The query should state “For which type of rose?” (*Hint:* Include Type in your query but don’t show it when you run the query.) Save this query as Query I. Run the query for type Tree and then print it.
- m. Create a select query that sums the total cost of all roses currently in inventory, formatted as currency. (*Hint:* First create a query with

the fields product Description, Quantity, and Cost/Unit and a computed field titled Cost, which is the product of Quantity and Cost/Unit formatted as currency.) Save this query as Query J – 1, run it, and then print it. Then create the summation query, save it as Query J – 2, run it, and then print it.

- n. Create a select query that sums the cost of all products by category (in currency format), sorted by type. (*Hint:* Modify the Query J – 1 you previously created by adding the field Type. Save the new query as K – 1.) Your new query should use Query K – 1 as its source and have two fields: Type and Cost. Save the new query as Query K – 2, run it, and then print it.
- o. Make sure you keep a copy of this file for use in the next chapter.