### LESSON SKILL MATRIX

<table>
<thead>
<tr>
<th>Skill</th>
<th>Exam Objective</th>
<th>Objective Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Crosstab Queries</td>
<td>Create crosstab queries</td>
<td>3.1.2</td>
</tr>
<tr>
<td>Creating a Subquery</td>
<td>Add new fields</td>
<td>3.2.2</td>
</tr>
<tr>
<td></td>
<td>Remove fields</td>
<td>3.2.3</td>
</tr>
<tr>
<td></td>
<td>Add conditional logic</td>
<td>3.3.2</td>
</tr>
<tr>
<td></td>
<td>Use comparison operators</td>
<td>3.3.4</td>
</tr>
<tr>
<td>Saving a Filter as a Query</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating Action Queries</td>
<td>Create action queries</td>
<td>3.1.4</td>
</tr>
<tr>
<td>Understanding Advanced Query Modifi-</td>
<td>Create multi-table queries</td>
<td>3.1.5</td>
</tr>
<tr>
<td>cation</td>
<td>Add calculated fields</td>
<td>3.3.1</td>
</tr>
<tr>
<td></td>
<td>Use basic operators</td>
<td>3.3.5</td>
</tr>
<tr>
<td></td>
<td>Format fields within queries</td>
<td>3.2.6</td>
</tr>
<tr>
<td></td>
<td>Group and summarize data</td>
<td>3.3.3</td>
</tr>
</tbody>
</table>

### KEY TERMS

- action query
- aggregate function
- append query
- calculated field
- cross join
- crosstab query
- delete query
- inner join
- join
- left outer join
- make table query
- outer join
- right outer join
- SELECT statement
- subquery
- unequal join
- update query
World Wide Importers is a car dealership that specializes in imported luxury cars. The company has recently opened a used car division that sells vehicles acquired in trade and expands the buyer’s purchasing options. As the office manager for the new division, you have started using Access to track inventory and sales. In this lesson, you learn how to create an action query, a crosstab query, and a subquery and how to save filters as a query. You also learn how to create joins, include calculated fields in a query, and create aggregated queries.

**CREATING CROSSTAB QUERIES**

Queries are powerful tools that can be used to retrieve exactly the data you need from your database, showing only the relevant records. Depending on the information you want to display, these advanced queries can help refine the results of your search or perform the actions you want. A crosstab query calculates a sum, average, count, or other type of total on records and then groups the results by two types of information: one down the left side of the datasheet and the other across the top. When you summarize data using a crosstab query, you select values from specified fields or expressions as column headings so you can view data in a more compact format than with a select query.

In Lesson 7, you learned how to create and modify several types of queries.

Creating Crosstab Queries

A crosstab query is a special type of query that displays its results in a grid similar to an Excel worksheet. Crosstab queries summarize your values and then group them by two sets of facts—a set of row headers down the side and a set of column headers across the top. A crosstab query typically includes data from more than one table and always includes three types of data: the data used as row headings, the data used as column headings, and the values that you want to sum or otherwise compute. A crosstab query does not always populate all the fields in the result set because the tables that you use in the query do not always contain values for every possible data point. In this exercise, you create a crosstab query.

The easiest way to create a crosstab query is to use the Crosstab Query Wizard. To run a crosstab query, double-click it in the Navigation Pane or click it and then press Enter. When you run a crosstab query, the results are displayed in Datasheet view.

**STEP BY STEP**

Create Crosstab Queries

GET READY. Before you begin these steps, be sure to LAUNCH Microsoft Access.

1. OPEN the Importers database from the data files for this lesson.
2. SAVE the database as ImportersXXX (where XXX is your initials).
3. On the CREATE tab, in the Queries group, click the Query Wizard button to display the New Query dialog box, shown in Figure 12-1.
4. Click **Crosstab Query Wizard** and click **OK** to display the Crosstab Query Wizard, shown in Figure 12-2.

5. Click **Table: Used Cars Sold** and then click **Next >** to display the next screen, shown in Figure 12-3.
6. In the Available Fields box, double-click Sold By to move it to the Selected Fields box and then click Next. The next screen appears, as shown in Figure 12-4.

7. Click Date Sold and then click Next to display the next screen, as shown in Figure 12-5.
8. Click **Month** and then click **Next >** to display the next screen, as shown in Figure 12-6.

9. In the Fields box, click **Sales Price**, and in the Functions box, click **Sum**. Click **Next >** to display the final screen, as shown in Figure 12-7.
10. Click **Finish** to display the results of the crosstab query, as shown in Figure 12-8.

A crosstab query does not always populate all the fields in the result set because the tables used do not always contain values for every possible data point.
11. Click the Close button to close the Used Cars Sold_Crosstab query.

PAUSE. LEAVE the database open to use in the next exercise.

**CREATING A SUBQUERY**

You can use a subquery to limit the amount of data returned by a query. A subquery is a SELECT statement that is inside another select or action query. A SELECT statement is a Structured Query Language (SQL) command that instructs the Microsoft Access database engine to return information from the database as a set of records. This type of statement invokes conditional logic using expressions that evaluate to true or false; if true, Access returns a result that meets the condition that you have specified; if false, Access can return a different result depending on what you've specified. For example, you can use a conditional expression to create a new field that displays an adjusted salary with a 3% bonus for all employees who had annual sales revenue of $50,000 or more and display a message like “Not Bonus Eligible” for everyone else. The data that appears after the expression is evaluated can be communicated on reports to provide useful information to employees who make decisions in your organization.

At a minimum, the syntax for a SELECT statement is:

SELECT fields FROM table

An asterisk (*) can be used to select all the fields in a table. The following example selects all the fields in the Inventory table:

SELECT * FROM Inventory

Clauses such as WHERE and ORDER BY can be used in a SELECT statement to restrict and organize your returned data. Table 12-1 shows some SELECT statements and the results that are returned.

<table>
<thead>
<tr>
<th>SELECT statement</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT [FirstName], [LastName] FROM [Employees] WHERE [LastName] = “Cooper”;</td>
<td>Displays the values in the FirstName and LastName fields for employees whose last name is Cooper.</td>
</tr>
<tr>
<td>SELECT [ProductID], [ProductName] FROM [Products] WHERE [CategoryID] = Forms.[New Products].[CategoryID];</td>
<td>Displays the values for ProductID and ProductName in the Products table for records in which the CategoryID value matches the CategoryID value specified in an open New Products form. Note: “Forms.[New Products].[Category ID]” references the New Products form and the CategoryID field on the form in case a New Products table and field also exist. The ”.” character is used to substantiate these form and field relationships.</td>
</tr>
<tr>
<td>SELECT Avg([ExtendedPrice]) AS [Average Extended Price] FROM [Order Details Extended] WHERE [ExtendedPrice] &gt; 1000;</td>
<td>Displays in a field named Average Extended Price the average extended price of orders for which the value in ExtendedPrice field is more than 1,000.</td>
</tr>
<tr>
<td>SELECT [CategoryID], Count([ProductID]) AS [CountOfProductID] FROM [Products] GROUP BY [CategoryID] HAVING Count([ProductID]) &gt; 10;</td>
<td>Displays in a field named CountOfProductID the total number of products for categories with more than 10 products.</td>
</tr>
</tbody>
</table>
Lesson 12

A SELECT statement can be entered in a field or criteria cell in Design view. If you need more space in which to enter the SELECT statement in a field or criteria cell, press Shift+F2 and enter the statement in the Zoom box. You can see the entire SQL statement by switching to SQL view.

In a subquery, you use a SELECT statement to provide a set of one or more specific values to evaluate in the WHERE or HAVING clause expression. A subquery has three parts:

- **Comparison**: An expression and a comparison operator that compares the expression with the results of the subquery
- **Expression**: An expression for which the result set of the subquery is searched
- **Sqlstatement**: A SELECT statement, following the same format and rules as any other SELECT statement. It must be enclosed in parentheses.

Creating a Subquery

In this exercise, you create a subquery that returns only the records from the Inventory table whose asking price is equal to or greater than the average asking price. You also add, remove, and reposition fields while working within the query design grid.

Subqueries are created in Design view. Using the Show Table dialog box, you first need to select the table that contains the desired information. This will add the table window, which contains the table's field list, to Design view. You can then easily add fields from the table window to the design grid at the bottom of the screen by either double-clicking the field name or by clicking and dragging the field to the query design grid. You can remove fields from the query design grid by moving the mouse pointer above the field name you want to remove until the pointer changes to a bold down arrow, clicking, and then pressing Delete on the keyboard, or by clicking the Delete Columns button in the Query Setup group on the DESIGN tab. You can rearrange fields on the grid by moving the mouse pointer above the field you want to move until the pointer changes to a bold down arrow, then clicking and dragging the field to any position on the grid using the vertical placeholder bar that appears as a guide.

**STEP BY STEP**

**Create a Subquery**

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click **Query Design**. The query design grid opens, and the Show Table dialog box appears, as shown in Figure 12-9.
2. On the Tables tab, click **Inventory**, click **Add**, and then click **Close**. The table field list appears as a window in the upper section of the query design grid, as shown in Figure 12-10.
3. In the list of table fields, double-click Year, Make, Model, Trim, and AskingPrice to add those fields to the design grid, as shown in Figure 12-11.

Another Way
To quickly add all the fields in a table, double-click the asterisk (*) at the top of the list of table fields.

4. Move the insertion point above the Trim field on the design grid until it turns into a bold down arrow. Click to select and highlight the Trim column, as shown in Figure 12-12.
5. [Press the Delete key] on the keyboard and the Trim column is deleted. The AskingPrice field moves to the left to replace the Trim column.

6. Move the insertion point above the Model field on the design grid until it turns into a bold down arrow. Click to select and highlight the Model column.

7. On the DESIGN tab, in the Query Setup group, click the Delete Columns button. The Model column is deleted.

8. In the table field list in the Inventory table window, double-click the Model field to add it back to the query design grid as the last column.

9. Move the insertion point above the Model field on the design grid until it turns into a bold down arrow. Click to select and highlight the Model column. Click and hold the mouse button down and drag the Model field to the left until the black vertical placeholder bar is positioned between the Make and AskingPrice fields, then release the mouse button. Your screen should resemble Figure 12-13.
10. Place the insertion point in the Criteria row of the AskingPrice field and [press Shift+F2] to display the Zoom dialog box.

Take Note The Zoom dialog box is used to create a larger workspace to help you focus on expressions and limit typing errors.

11. Key the following expression in the Zoom dialog box, as shown in Figure 12-14, to have the query eventually return all car makes that have an asking price greater or equal to the average asking price for all car makes:

\[ \geq (\text{SELECT Avg(AskingPrice) FROM Inventory WHERE Make} = \text{Inventory.Make}) \]
Take Note  Since you are already working inside the Inventory table, the reference in the expression to the Inventory table preceding the Make field can be omitted and the expression can also be constructed as follows: 

\[ \geq (\text{SELECT Avg(AskingPrice) FROM Inventory WHERE Make = Make}) \]

12. Click OK to insert the expression in the Criteria row of the AskingPrice field.

13. On the DESIGN tab, in the Results group, click the lower half of the View button and click SQL View to see the entire expression, as shown in Figure 12-15.

Figure 12-15

SQL View

Subquery WHERE clause in SQL View

14. On the DESIGN tab, in the Results group, click Run. The query results are displayed, as shown in Figure 12-16.
15. Click the **FILE** tab and click **Save**.
16. In the Save As dialog box, key **Subquery** as the query name and click **OK**.
17. Click the **Close** button to close the Subquery query.

**PAUSE**. **LEAVE** the database open to use in the next exercise.
SAVING A FILTER AS A QUERY

A filter can be saved as a query so it can be run again anytime you want. If you often work with certain filters, you might want to save these filters so you are not wasting time defining them each time. You cannot save more than one filter for each table, query, or form; but you can save a filter as a query and then apply the query as a filter when and where you want.

Saving a Filter as a Query

In this exercise, you create a simple select query, filter it, and then save it.

When you want to save a filter as a query, you first use the Query Wizard to create a new query using all the fields from a table or another query. You can then use the Filter by Form command to choose the data by which to filter the newly created query and then apply the filter. Next, on the HOME tab, in the Sort & Filter group, you can click the Advanced button and then click Advanced Filter/Sort. The new query design grid will appear. It will automatically include all the fields you’ve previously chosen to filter. On the HOME tab, in the Sort & Filter group, click the Advanced button and click Save As Query. Key a name for the query and click OK.

To apply the query as a filter, click the Advanced button and click Load from Query to display the Applicable Filter dialog box, as shown in Figure 12-17.

![Applicable Filter dialog box](image)

Only queries that are based on the same table or query as the form or datasheet will appear in the dialog box. Select the filter, click OK, and then apply the filter.

STEP BY STEP  
Save a Filter as a Query

USE the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click the Query Wizard button.
2. In the New Query dialog box, click Simple Query Wizard and click OK.
3. In the Tables/Queries drop-down list, click Table: Used Cars Sold.
4. Click the >> button to move all the fields from the Available Fields box to the Selected Fields box and then click Next >.
5. Click Next > again and then click Finish to display a simple select query.
6. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Filter by Form.
7. In Filter by Form view, click the Year field down arrow and click 2010, as shown in Figure 12-18.
8. On the HOME tab, in the Sort & Filter group, click the **Toggle Filter** button to apply the filter. The results are displayed, as shown in Figure 12-19.

**Figure 12-19**
Filter by Form results

Results filtered to show only 2010 cars
9. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Advanced Filter/Sort to display the new query design grid, as shown in Figure 12-20.

![New query design grid](image)

**Figure 12-20**

New query design grid

10. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Save As Query. The Save As Query dialog box appears, as shown in Figure 12-21.

![Save As Query dialog box](image)

**Figure 12-21**

Save As Query dialog box

11. Key Filter Query in the Query Name box and click OK.

12. Click the Close button to close the Used Cars Sold QueryFilter1 query.

13. On the HOME tab, in the Sort & Filter group, click the Toggle Filter button to remove the filter.

14. Click the Close button to close the Used Cars Sold Query and save the changes when prompted. If another dialog box appears informing you that another user may have changed the data, click Yes.

PAUSE. LEAVE the database open to use in the next exercise.
CREATING ACTION QUERIES

An action query changes the data in its datasource or creates a new table. There are four types of action queries—append, delete, update, and make table—and except for the make table query, action queries make changes to the data in the tables on which they are based.

As their name suggests, action queries make changes to the data in the tables they are based on (except for make table queries, which create new tables). There are four types of action queries:

- **Append query**: Adds the records in the query's result set to the end of an existing table
- **Delete query**: Removes rows matching the criteria that you specify from one or more tables
- **Update query**: Changes a set of records according to criteria that you specify
- **Make table query**: Creates a new table and then creates records in it by copying records from an existing table

Changes made by action queries cannot be easily undone, so if you later decide you didn't want to make those changes, usually you will have to restore the data from a backup copy. For this reason, you should always make sure you have a current backup of the underlying data before running an action query.

To minimize the risk involved in running an action query, you can first preview the data that will be acted upon by viewing the action query in Datasheet view before running it. When you are ready to run an action query, double-click it in the Navigation Pane or click it and then press Enter. Or, on the DESIGN tab, in the Results group, click Run.

Creating an Append Query

An append query adds a set of records from one or more source tables (or queries) to one or more destination tables. Typically, the source and destination tables reside in the same database, but they don’t have to. For example, suppose you acquire some new customers and a database that contains a table of information about those customers. To avoid entering that new data manually, you can append it to the appropriate table in your database. In this exercise, you practice creating an append query.

You can also use append queries to append fields that are based on criteria. For example, you might want to append only the names and addresses of customers who have outstanding orders. Or you can use append queries to append records when some of the fields in one table don’t exist in the other table. For example, suppose that your Customers table has 10 fields, and the fields in the Clients table in another database match 8 of your 10 fields. You can use an append query to add only the data in the matching fields and ignore the others.

You cannot use append queries to change the data in individual fields in existing records. To do that type of task, you use an update query. You can only use append queries to add rows of data.

**STEP BY STEP**

Create an Append Query

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click the **Query Design** button.
2. In the Show Table dialog box, double-click **Inventory** to add it to the upper section of the query design grid.
3. Click **Close** to close the Show Table dialog box.
4. In the list of table fields, double-click **Year**, **Make**, **Model**, and **Asking Price** to add those fields to the design grid. Your screen should look similar to Figure 12-22.
5. On the DESIGN tab, in the Results group, click Run. Verify that the query returned the records that you want to append, as shown in Figure 12-23.
If you need to add or remove fields from the query, switch back to Design view and double-click to add fields or select the fields that you don’t want and press Delete to remove them from the query.

6. Right-click the document tab (titled Query1) for the open query and click Design View on the shortcut menu.

7. On the DESIGN tab, in the Query Type group, click Append. The Append dialog box appears, as shown in Figure 12-24.

8. In the Table Name box, click the down arrow and click Used Cars Sold. This is the table you want to append to. The Current Database option button should be selected.

9. Click OK. Access automatically adds the names of the destination fields that match the source field names to the Append To row in the design grid. Because the Asking Price field doesn’t have a match, Access leaves that field blank.

10. Click the blank field in the Append To row under the Asking Price cell and select Sales Price as the destination field, as shown in Figure 12-25.
11. Right-click the document tab for the query, and then click Datasheet View to preview your changes.

12. Right-click the document tab for the query, and then click Design View.

13. On the DESIGN tab, in the Results group, click Run. An alert message appears, as shown in Figure 12-26.

14. Click Yes.

15. Open the Used Cars Sold table and scroll down to see that the records from the Inventory table have been appended to the end, as shown in Figure 12-27.

16. Click the Close button to close the Used Cars Sold table.

17. Click the FILE tab and click Save.

18. In the Save As dialog box, key Append Query as the query name and click OK.

19. Click the Close button on Append Query to close the query.

PAUSE. LEAVE Access open to use in the next exercise.
Creating a Make Table Query

A make table query is an action query that creates a new table and then creates records in it by copying records from an existing table. You use a make table query when you want to create a new table based on query criterion or criteria from an existing table. For example, you may work for a telecommunications provider and be provided with a table with thousands of cellphone service customers who live in various states. You may find it easier to work with a subset of customers from New York, New Jersey, and Connecticut who have had cellphone service for less than one year, since you need to contact and produce reports on only those individuals. A make table query can be advantageous in this example, especially if you want to ensure the data is stored in a separate table (i.e., you want to export the data as a table to a new database). If you run a make table query with no criterion or criteria, Access will make a duplicate of the table titled after the name you provide. This is helpful when you need to copy the data in a table or to archive data, especially before you run an append, update, or delete query in case it amends the table data in a way that you don’t want. In this exercise, you practice creating a make table query to create a backup of a table.

**STEP BY STEP**

**Create a Make Table Query**

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click the **Query Wizard** button.
2. In the New Query dialog box, click **Simple Query Wizard** and click **OK**.
3. In the Tables/Queries drop-down list, click **Table: Sales Team**.
4. Click the **.** button to move all the fields from the Available Fields to the Selected Fields box and then click **Next >**.
5. Click **Finish** to display a simple select query.
6. Right-click the **Sales Team Query** document tab and click **Design View** to display the query in Design view, as shown in Figure 12-28.

![Query in Design view](image)
7. On the DESIGN tab, in the Query Type group, click **Make Table**. The Make Table dialog box appears, as shown in Figure 12-29.

![Make Table dialog box]

8. In the Table Name box, key **Sales Team Backup**. If it isn’t already selected, click **Current Database**, and then click **OK**.

9. On the DESIGN tab, in the Results group, click **Run**. An alert message appears, as shown in Figure 12-30.

![Make table alert message]

10. Click **Yes**. A new table named Sales Team Backup appears in the Navigation Pane.

11. Double-click **Sales Team Backup: Table** in the Navigation Pane to open the new table, as shown in Figure 12-31.

![New table]

New table now available in the Navigation Pane
12. Click the **Close** button to close the Sales Team Backup table.
13. Click the **Close** button to close the Sales Team query. Save the changes when prompted.
14. **LEAVE** the database open.

**PAUSE. LEAVE** Access open to use in the next exercise.

---

**Workplace Ready**

**USING QUERIES TO CREATE TARGETED MAILING LABELS**

Keeping track of sales data and contacts is vital to the success of any business. Access provides the tools you need to not only keep these records available and secure, but also to generate sales and provide customer service. Whether you need to mail a single sales brochure or do a mass mailing of two thousand, you can use Access to create the labels using the records you maintain in your Access databases.

Imagine you are a partner in a start-up software firm named Proseware, Inc., that has developed specialized software for colleges and universities. You have created an Access database with tables that include information for customers as well as sales leads for professors to whom you are marketing the product.

As you saw in Lesson 11, you can use the Label Wizard to create mailing labels and sort them by zip code for bulk mailing. You can create an action query, like a make table query, that uses one or more criteria to select only certain records for labels and makes a new table to include only those records you will use for mailing. For example, you can use a make table query to create a new table that includes professors who have expertise in a certain area, or those who instruct at a certain college or university. You can then use this new table as the source for the Label Wizard. Using the Label Wizard and a make table query, you can produce labels for mailings to your targeted audience quickly and efficiently.

---

**Creating an Update Query**

An **update query** is an action query that changes a set of records according to specified criteria. Use an update query when you need to add, change, or delete the data in one or more existing records. You can think of update queries as a powerful form of the Find and Replace dialog box. In this exercise, you practice making an Update Query.

When making an update query, you enter a select criterion and an update criterion. Unlike the Find and Replace dialog box, update queries can accept multiple criteria. You can use them to update a large number of records in one pass and to change records in more than one table at one time. You can also update the data in one table with data from another—as long as the data types for the source and destination fields match or are compatible.

To create an update query, first create or open a select query. On the DESIGN tab, in the Query Type group, click Update. Access adds the Update To row in the query design grid. Locate the field that contains the data you want to change, and type your change criteria in the Update To row for that field.

You can use any valid expression in the Update To row. Table 12-2 shows some common valid expressions and explains how they change data.
STEP BY STEP  

Create an Update Query

USE the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click the Query Wizard button.
2. In the New Query dialog box, click Simple Query Wizard and click OK.
3. In the Tables/Queries drop-down list, click Table: Inventory.
4. Click the >> button to move all the fields from the Available Fields to the Selected Fields box.
5. Click Trim and then the < button to move it back to the Available Fields box. Click Color and then the < button to move it back to the Available Fields box. Click Next >.
6. Click Next > again and then click Finish to display a simple select query in Datasheet view, as shown in Figure 12-32.
7. Right-click the **Inventory Query** document tab and click **Design View** to display the query in Design view.

8. Key **2012** in the Criteria row of the Year field.

9. On the **DESIGN** tab, in the Query Type group, click **Update**. Access adds the Update To row in the query design grid, as shown in Figure 12-33.
10. In the Update To row of the AskingPrice field, key \([\text{AskingPrice}] + 500\). The design grid should look similar to Figure 12-34.

![Figure 12-34 Select and update criterion](image)

11. On the DESIGN tab, in the Results group, click Run. An alert message appears, as shown in Figure 12-35.

![Figure 12-35 Update alert message](image)

12. Click Yes.

13. Right-click the Inventory Query document tab and click Datasheet View to display the update query results, as shown in Figure 12-36.
14. Click the Close button to close the Inventory query. Save the changes when prompted.

15. Double-click Inventory: Table in the Navigation Pane to open it. Notice that the asking price for all 2012 cars has been increased by $500.

16. Click the Close button to close the Inventory table.

17. LEAVE the database open.

PAUSE. LEAVE Access open to use in the next exercise.

Creating a Delete Query

A delete query is an action query that removes rows matching the criteria that you specify from one or more tables. A delete query is used to delete entire records from a table, along with the key value that makes a record unique. Typically, delete queries are used only when you need to change or remove large amounts of data quickly. To remove a small number of records, open the table inDatasheet view, select the fields or rows that you want to delete, and press Delete.

To create a delete query, first create or open a select query and add criteria to return the records you want to delete. On the DESIGN tab, in the Query Type group, click Delete. Access changes the select query to a delete query, hides the Show row in the lower section of the design grid, and adds the Delete row. The word Where appears in the Delete row for all fields of the query.

When you click Run, Access prompts you to confirm the deletion. Click Yes to delete the data and then open the table to see that the records have been deleted.
STEP BY STEP  Create a Delete Query

USE the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click Query Wizard.
2. In the New Query dialog box, click Simple Query Wizard and click OK.
3. In the Tables/Queries drop-down list, click Table: Used Cars Sold.
4. Click the >> button to move all the fields from the Available Fields to the Selected Fields box and then click Next >.
5. Click Next > again.
6. Key Delete Query as the title and then click Finish to display the results of the simple select query.
7. Right-click the Delete Query document tab and click Design View to display the query in Design view.
8. Key <#3/31/2015# in the Criteria row of the Date Sold field, as shown in Figure 12-37.

Figure 12-37  Date Sold criteria
9. On the DESIGN tab, in the Results group, click Run to display the records to be deleted, as shown in Figure 12-38.

10. Right-click the Delete Query document tab and click Design View to display the query in Design view.

11. On the DESIGN tab, in the Query Type group, click Delete. Access hides the Show row in the lower section of the design grid and adds the Delete row, as shown in Figure 12-39.
12. On the DESIGN tab, in the Results group, click Run. An alert message appears, as shown in Figure 12-40.

13. Click Yes.

14. Double-click Used Cars Sold: Table in the Navigation Pane to open it. Notice that all the records for cars sold before March 31, 2015, have been deleted, as shown in Figure 12-41.
15. Click the Close button on Used Cars Sold to close the table.
16. Click the Close button on Delete Query to close the query. SAVE the changes when prompted.
17. LEAVE the database open.

UNDERSTANDING ADVANCED QUERY MODIFICATION
After a query has been created, you can modify it in various ways to suit your purposes—by creating a join, creating calculated fields, or using aggregated functions.

Creating a Join
Relational databases consist of tables that have logical relationships to each other. You use relationships to connect tables based on fields they have in common. A relationship between identical fields in different tables is represented by a join in Design view. When you include multiple tables in a query, you use joins to help you get the results you want. A join helps a query return only the records from each table you want to see based on how those tables are related to other tables in the query. When you add tables to a query, Access creates joins that are based on relationships that have been defined between the tables. You can manually create joins known as ad hoc relationships in queries, even if they do not represent relationships that have already been defined. In this exercise, you create a join between tables.
If the relationship is one-to-many, Access displays a “1” above the join line to show which table is on the “one” side and an infinity symbol (∞) to show which table is on the “many” side.

The four basic types of joins are inner joins, outer joins, cross joins, and unequal joins. An inner join includes rows in the query only when the joined field matches records in both tables. Inner joins are the most common type of join. Most of the time, you don’t need to do anything to use an inner join. Access automatically creates inner joins if you add two tables to a query and those tables each have a field with the same name and the same or compatible data type and one of the join fields is a primary key.

An outer join includes all of the rows from one table in the query results and only those rows from the other table that match the join field in the first table. You create outer joins by modifying inner joins. To create an outer join, double-click the line joining the tables to display the Join Properties dialog box. In the Join Properties dialog box, Option 1 represents an inner join. Option 2 is a left outer join, where the query includes all of the rows from the table on the left and only those records from the table on the right that match the join field in the left table. Option 3 is a right outer join, where the query includes all of the rows from the table on the right and only those rows from the table on the left that match the join field in the right table.

Because some of the rows on one side of an outer join will not have corresponding rows from the other table, some of the fields returned in the query results from the other table will be empty when the rows do not correspond.

To tell which table is the left table or the right table in a given join, double-click the join to view the Join Properties dialog box.

In a cross join, each row from one table is combined with each row from another table. Any time you run a query that has tables that are not explicitly joined, a cross join is produced. Cross joins are usually unintentional, but there are cases where they can be useful. A cross join can be used if you want to examine every possible combination of rows between two tables or queries.

If you want to combine the rows of two sources of data based on field values that are not equal, you use an unequal join. Typically, unequal joins are based on either the greater than (>), less than (<), greater than or equal to (≥), or less than or equal to (≤) comparison operators. Unequal joins are not supported in Design view. If you wish to use them, you must do so in SQL view.

If you create a join by mistake, for example, a join between two fields that have dissimilar data types, you can delete it. In the query design grid, click the join you want to remove and press Delete.

**STEP BY STEP**

**Create a Join**

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click **Query Design**.
2. In the Show Table dialog box, double-click **Sales Team** and **Used Cars Sold** to add them to the design grid.
3. Click **Close**.
4. In the Sales Team field list, double-click **E-mail Address**.
5. In the Used Cars Sold field list, double-click **Year**, **Make**, **Model**, and **Sales Price**. Your screen should look similar to Figure 12-42.
6. Double-click the join line between the tables, indicating which fields are joined. The Join Properties dialog box opens, as shown in Figure 12-43.

7. Click the option button for option 2: and then click OK to create a left outer join.

8. On the DESIGN tab, in the Results group, click Run.

9. The results of the query are displayed, as shown in Figure 12-44.
Advanced Queries

10. Save the query as Join Query and CLOSE.

11. LEAVE the database open.

PAUSE. LEAVE Access open to use in the next exercise.

Creating a Calculated Query Field

You can create a new field that displays the results of a calculation you define with an expression or that manipulates field values. A calculated field is a column in a query that results from an expression. For example, you can calculate a value; combine text values, such as first and last names; or format a portion of a date. You can also format the calculated field value by choosing an appropriate format in the Property Sheet pane. In this exercise, you use the Expression Builder to create a calculated query by subtracting two fields to determine a markup price in both U.S. dollars and Euros.

You can use expressions that perform arithmetic operations in calculated fields using basic operators (+, −, *, /) to add, subtract, multiply, and divide the values in two or more fields. You can also perform arithmetic operations on dates or use expressions that manipulate text. Table 12-3 shows examples of expressions and basic operators that can be used in calculated fields. The results can then be formatted by selecting the calculated field and then selecting a format in the Format box of the Property Sheet pane. For example, a number can be formatted in several different ways, including as Currency or Scientific values. The Property Sheet button in the Show/Hide group on the DESIGN tab displays the Property Sheet pane.
**Table 12-3**

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrimeShip: [Ship] * 1.1</td>
<td>Creates a field called PrimeShip, and then displays shipping charges plus 10 percent in the field.</td>
</tr>
<tr>
<td>OrderAmount: [Quantity] * [Price]</td>
<td>Creates a field called OrderAmount, and then displays the product of the values in the Quantity and Price fields.</td>
</tr>
<tr>
<td>LeadTime: [RequiredDate] – [ShippedDate]</td>
<td>Creates a field called LeadTime, and then displays the difference (in number of days) between the values in the RequiredDate and ShippedDate fields.</td>
</tr>
<tr>
<td>TotalInventory: [UnitsInStock] + [UnitsOnOrder]</td>
<td>Creates a field called TotalInventory, and then displays the sum of the values in the UnitsInStock and UnitsOnOrder fields.</td>
</tr>
<tr>
<td>FullName: [FirstName] &amp; “ “ &amp; [LastName]</td>
<td>Creates a field called FullName that displays the values in the FirstName and LastName fields, separated by a space. (“&amp;” is called the concatenation operator and is used to put strings together.)</td>
</tr>
<tr>
<td>Address2: [City] &amp; “ “ &amp; [Region] &amp; “ “ &amp; [PostalCode]</td>
<td>Creates a field called Address2 that displays the values in the City, Region, and PostalCode fields, separated by spaces. (“&amp;” is called the concatenation operator and is used to put strings together.)</td>
</tr>
</tbody>
</table>

A well-designed database does not store simple calculated values in tables. For example, a table might store an employee’s hire date, but not how long she has worked for the company. If you know both today’s date and the employee’s date of hire, you can always calculate her employment length, so there is no need to store that in the table. Instead, you create a query that calculates and displays the pertinent value. The calculations are made every time you run the query, so if the underlying data changes, so do your calculated results.

To create a calculated field, first open or create a query and switch to Design view. In the Field row of the first blank column in the design grid, key the expression. You can use the Zoom box to access a larger screen area to help you enter the expression or, as you learned in Lesson 8, you can use the Expression Builder to easily select the elements of the expression (fields, operators, and built-in functions) from menus. To name the field, key a name followed by a colon before the expression. If you do not supply a name, Access will use a generic name for the field, for example, EXPR1. The string following the colon is the expression that supplies the values for each record. To see the SQL code, you can switch to SQL View.

**STEP BY STEP**

**Create a Calculated Query Field**

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click **Query Design**.
2. In the Show Table dialog box, double-click **Inventory** to add the table to the design grid.
3. Click **Close**.
4. In the Inventory field list, double-click **Year**, **Make**, **Model**, **AskingPrice**, and **TradeInValue**.
5. Click the **Field** cell in the first blank column (to the right of the TradeInValue field) and click the **Builder** button in the Query Setup group to open the Expression Builder dialog box.
6. In the blank area of the dialog box, key the following:
   
   **Markup**: [AskingPrice]

7. In the Expression Elements category, click **Operators**. The dialog box should resemble Figure 12-45.
8. In the Expression Values category, double-click the minus sign (–). The minus sign should appear in the expression and next to the AskingPrice field.

9. In the Expression Elements category, click the expand button next to ImportersXXX.accdb. Tables, Queries, Forms, and Reports should appear under ImportersXXX.accdb.

10. In the Expression Elements category, click the expand button next to Tables to expand it. The available table names appear. Click Inventory. The available fields from the Inventory table should appear in the Expression Categories box. Your screen should resemble Figure 12-46.
11. In the Expression Categories box, double-click TradeInValue; [Inventory]! [TradeInValue] should appear in the expression and next to the minus sign (−).

**Take Note**
The part of the expression that reads [Inventory]! [TradeInValue] specifies that the TradeInValue field originates from the Inventory table; however, even though Access automatically formats it this way, this expression format is not required since you’re already referencing the Inventory table in the Table row of the design grid.

12. Click OK.
13. Click the Field cell in the first blank column (to the right of the newly created calculated field) and click the Builder button in the Query Setup group to open the Expression Builder dialog box.
14. In the blank area of the dialog box, key the following:

   **Markup in Euros:** ([AskingPrice] – [TradeInValue]) * .7534

15. Click OK.
16. On the DESIGN tab, in the Show/Hide group, click the Property Sheet button. The Property Sheet pane is displayed, as shown in Figure 12-47.

**Figure 12-47**
Property Sheet Pane

17. Click the empty cell next to the Format box, then click the down arrow to display the Format menu, as shown in Figure 12-48. Select Euro from the options that appear to format the expression result as Euro.
18. On the DESIGN tab, in the Results group, click Run. The query with the new calculated Markup and Markup in Euros fields is displayed, as shown in Figure 12-49.
19. SAVE the query as Calculated Query and CLOSE.

20. LEAVE the database open.

PAUSE. LEAVE Access open to use in the next exercise.

Creating Aggregated Queries

An aggregate function performs a calculation on a set of values and then returns a single value. You can add, count, or calculate other aggregate values and display them in a special row, called the Total row, which appears below the asterisk (*) row in Datasheet view. You can use a different aggregate function for each column and you can also choose not to summarize a column. You can use aggregated functions to count the data returned by a query, calculate average values, and find the smallest, largest, earliest, and latest values using a feature called the Total row. You can work with the Total row in both query Design and query Datasheet views. In this exercise, you create an aggregated query using the Total row in both query Design and query Datasheet views.

You can also apply aggregated functions in Design view where you have the ability to use the Group By function in the Total row on the design grid. The Group By function can be used in combination with other fields and aggregated functions. For example, if you're managing a human resource database, you can group by employees' gender and display the average salary per group.
The following aggregated functions are available in both Datasheet view and Design view:

- **Count**: Counts the number of items in a column
- **Sum**: Sums a column of numbers
- **Average**: Averages a column of numbers
- **Maximum**: Finds the highest value in a column
- **Minimum**: Finds the lowest value in a column
- **Standard Deviation**: Measures how widely values are dispersed from an average value (a mean) in a column
- **Variance**: Measures the statistical variance of all values in a column

The following additional aggregated functions are available in Design view:

- **First**: Finds the first value in a field
- **Last**: Finds the last value in a field
- **Expression**: Groups data based on an expression you can specify
- **Where**: Groups data based on criteria you can specify

In Lesson 9, you use the Total row with tables to provide a summary of table data.

Many of the aggregated functions work only on data fields set to specific data types. For example, if you are in a column that only displays text values, some functions—such as Sum or Average—are not relevant, and are therefore not available.

**STEP BY STEP**

**Create an Aggregated Query**

**USE** the database that is open from the previous exercise.

1. On the CREATE tab, in the Queries group, click **Query Design**.
2. In the Show Table dialog box, double-click **Inventory** to add the table’s field list window to the design grid.
3. Click **Close**.
4. In the Inventory field list, double-click **Year, Make, Model, Mileage, AskingPrice**, and **TradeInValue** to add them to the design grid.
5. On the DESIGN tab, in the Results group, click **Run**.
6. On the HOME tab, in the Records group, click the **Total** button. Scroll down, if necessary, to see the Total row at the bottom of the result set.
7. In the Total cell of the **Year** field, click the **down arrow** to display the menu and click **Count**, as shown in Figure 12-50. Notice the cell displays 26, denoting 26 values (years) in the Year column.
8. Click the down arrow in the Total cell of the Mileage field and click Average.
9. Click the down arrow in the Total cell of the AskingPrice field and click Maximum.
10. Click the down arrow in the Total cell of the TradeInValue field and click Sum. Your Total row should appear similar to Figure 12-51.
11. Switch to Design view and remove the Model, Mileage, AskingPrice, and TradeInValue fields from the design grid. The Year and Make fields should be the only ones remaining on the grid.

12. On the DESIGN tab, in the Show/Hide group, click the Totals button. A new Totals row should appear below the Table row on the design grid.

13. Click the Group By cell below the Make field cell and click the down arrow to display the aggregate function menu.

14. Click the Count aggregate function to select it from the menu, as shown in Figure 12-52.
15. Switch to Datasheet view. Your screen should resemble Figure 12-53. Notice the records in the Make field are grouped by Year and counted with the results appearing in a new column named CountOfMake. Also notice the Year field is grouped and each year remains counted as applied from the aggregate function we created previously in Datasheet view.
Figure 12-53
Aggregate function results

16. SAVE the query as Aggregated Query and CLOSE it.
17. CLOSE the database.

PAUSE. CLOSE Access.

SKILL SUMMARY

<table>
<thead>
<tr>
<th>In This Lesson You Learned How To:</th>
<th>Exam Objective</th>
<th>Objective Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create crosstab queries</td>
<td>Create crosstab queries</td>
<td>3.1.2</td>
</tr>
<tr>
<td>Create a subquery</td>
<td>Add new fields</td>
<td>3.2.2</td>
</tr>
<tr>
<td></td>
<td>Remove fields</td>
<td>3.2.3</td>
</tr>
<tr>
<td></td>
<td>Add conditional logic</td>
<td>3.3.2</td>
</tr>
<tr>
<td></td>
<td>Use comparison operators</td>
<td>3.3.4</td>
</tr>
<tr>
<td>Save a filter as a query</td>
<td>Create action queries</td>
<td>3.1.4</td>
</tr>
<tr>
<td>Create action queries</td>
<td>Create multi-table queries</td>
<td>3.1.5</td>
</tr>
<tr>
<td>Understand advanced query modification</td>
<td>Add calculated fields</td>
<td>3.3.1</td>
</tr>
<tr>
<td></td>
<td>Use basic operators</td>
<td>3.3.5</td>
</tr>
<tr>
<td></td>
<td>Format fields within queries</td>
<td>3.2.6</td>
</tr>
<tr>
<td></td>
<td>Group and summarize data</td>
<td>3.3.3</td>
</tr>
</tbody>
</table>
Knowledge Assessment

Fill in the Blank

Complete the following sentences by writing the correct word or words in the blanks provided.

1. A(n) ________ is a SELECT statement that is inside another select or action query.
2. A(n) ________ removes rows matching the criteria you specify from one or more tables.
3. To minimize the risk of running an action query, you can first preview the data that will be acted upon by viewing the action query in ________ view before running it.
4. A(n) ________ includes all of the rows from one table in the query results and only those rows from the other table that match the join field in the first table.
5. A(n) ________ is a column in a query that results from an expression.
6. You can use the Group By function in the ________ row on the design grid in query Design view.
7. A(n) ________ performs a calculation on a set of values and then returns a single value.
8. A(n) ________ query always includes three types of data: the data used as row headings, the data used as column headings, and the values that you want to sum or otherwise compute.
9. To quickly add all the fields in a table to the design grid in Design view, double-click the ________ at the top of the list of table fields.
10. To be able to apply a filter when and where you want, save the filter as a(n) ________.

Multiple Choice

Select the best response for the following statements or questions.

1. What type of query displays its results in a grid similar to an Excel worksheet?
   a. crosstab
   b. append
   c. aggregated
   d. subquery
2. What can you use for a more intuitive interface in which to enter criteria or an expression in a field or criteria cell?
   a. Zoom box
   b. Field list pane
   c. control label
   d. Expression Builder
3. Which action query does not make changes to the data in the tables that it is based on?
   a. append
   b. make table
   c. update
   d. delete
4. Which type of query can be thought of as a powerful version of the Find and Replace dialog box?
   a. filter
   b. calculated field
   c. update
   d. crosstab
5. Which of the following is not a type of join?
   a. inner join
   b. exterior join
   c. cross join
   d. unequal join
6. Which of the following is not an aggregated function?
   a. Lowest
   b. Sum
c. Average
d. Count

7. Which of the following SELECT statements selects all the fields from the Inventory table?
   a. SELECT all fields FROM Inventory
   b. SELECT [ALL] from [INVENTORY]
   c. SELECT from INVENTORY (all fields)
   d. SELECT * FROM Inventory

8. For more space in which to enter the SELECT statement in a field or criteria cell, what do you press to display the Zoom box?
   a. Shift+F2
   b. Ctrl+2
   c. Shift+Enter
   d. Ctrl+Spacebar

9. To display the pane used to format a calculated value in query Design view, click the _____ in the Show/Hide group on the DESIGN tab of the Ribbon.
   a. Undo button
   b. Property Sheet button
   c. Datasheet View button
   d. Totals button

10. Which type of query adds the records in the query’s result set to the end of an existing table?
    a. append
    b. make table
    c. update
    d. delete

Competency Assessment

Project 12-1: Create a Calculated Query Field

In your job as a travel agent at Margie’s Travel, you are frequently asked the length of various trips. So that you don’t have to calculate it mentally, create a calculated field that will give you this information.

GET READY. LAUNCH Access if it is not already running.

1. OPEN the M Travel database from the data files for this lesson.
2. SAVE the database as M TravelXXX (where XXX is your initials).
3. On the CREATE tab, in the Queries group, click Query Design.
4. In the Show Table dialog box, double-click Events to add the table to the design grid.
5. Click Close.
6. In the Inventory field list, double-click Event, StartDate, and EndDate to add the fields to the design grid.
7. Click the Field cell in the first blank column and [press Shift+F2] to open the Zoom dialog box.
8. In the Zoom dialog box, key the following expression:
   TripLength: [EndDate] – [StartDate]
9. Click OK.
10. On the DESIGN tab, in the Results group, click Run. The query is displayed, with a new TripLength field calculating the number of days for each trip.
11. SAVE the query as Calculated Query and CLOSE.
12. CLOSE the database.

LEAVE Access running for the next project.
Project 12-2: Save a Filter as a Query

As sales manager for Fourth Coffee, you frequently run the same filters on the database. Now that you have learned to save a filter as a query, you can save yourself some time.

GET READY. LAUNCH Access if it is not already running.

1. OPEN Fourth Coffee from the data files for this lesson.
2. SAVE the database as Fourth Coffee XXX (where XXX is your initials).
3. On the CREATE tab, in the Queries group, click the Query Wizard button.
4. In the New Query dialog box, click Simple Query Wizard and click OK.
5. In the Tables/Queries drop-down list, click Table: Order Summary.
6. Click the >> button to move all the fields from the Available Fields to the Selected Fields box and then click Next >.
7. Click Next > again and then click Finish to display a simple select query.
8. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Filter by Form.
9. In the Filter by Form, click the down arrow in the Status field and click Completed.
10. On the HOME tab, in the Sort & Filter group, click the Toggle Filter button to apply the filter. The results are displayed.
11. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Advanced Filter/Sort to display the new query design grid.
12. On the HOME tab, in the Sort & Filter group, click the Advanced button and then click Save As Query. The Save As Query dialog box appears.
13. Key Filter Query in the Query Name box and click OK.
14. Click the Close button to close the Order Summary QueryFilter1 query.
15. On the HOME tab, in the Sort & Filter group, click the Toggle Filter button to remove the filter.
16. Click the Close button to close the Order Summary Query and save the changes when prompted. If presented with a message box stating the query has been changed, click Yes.
17. LEAVE the database open.

LEAVE Access open for the next project.

Proficiency Assessment

Project 12-3: Create a Subquery

You are interested in extracting specific information about monthly sales for all Fourth Coffee stores from the database. Create a subquery to determine which months and corresponding stores have sales that are above average.

USE the database that is open from the previous project.

1. On the CREATE tab, in the Queries group, click Query Design.
2. Use the Show Table dialog box to add the Monthly Sales by Store table to the upper section of the query design grid and then close the Show Table dialog box.
3. Add the Month, Store, and Sales fields to the design grid.
4. Place the insertion point in the Criteria row of the Sales field and display the Expression Builder.
5. Key the following expression in the Expression Builder, using the available categories and menus:
   > (SELECT AVG([Sales]) FROM [Monthly Sales by Store])
6. Click OK.
7. Use the Property Sheet pane to change the calculated field format to Euro.
8. On the DESIGN tab, in the Results group, click Run to display the query results.
9. SAVE the query as Subquery and CLOSE.
10. CLOSE the database.

LEAVE Access open for the next project.

Project 12-4: Create a Make Table Query
As the manager at Southridge Video, you want to archive the current table with information about used games. Use the make table action query to create a backup table.

GET READY. LAUNCH Access if it is not already running.
1. OPEN Games Southridge from the data files for this lesson.
2. SAVE the database as Games SouthridgeXXX (where XXX is your initials).
3. Create a simple select query named Games Query using all the fields in the Games: Table.
4. Display the query in Design view if it is not already.
5. On the DESIGN tab, in the Query Type group, click Make Table to display the Make Table dialog box.
6. In the Table Name box, key Games Backup. If it is not already selected, click Current Database, and then click OK.
7. On the DESIGN tab, in the Results group, click Run. An alert message appears.
9. CLOSE the Games Query and save the changes when prompted.
10. CLOSE the database.

LEAVE Access open for the next project.

Mastery Assessment

Project 12-5: Create a Crosstab Query
As a regional manager for Contoso Pharmaceuticals, you are in charge of overseeing the sales reps in your division. To determine the total samples given by each rep in the first two weeks of the quarter, you decide to create a crosstab query.

GET READY. LAUNCH Access if it is not already running.
1. OPEN Contoso Data from the data files for this lesson.
2. SAVE the database as Contoso DataXXX (where XXX is your initials).
3. Using the Samples Given: Table and the skills you have learned in this lesson, create and save a crosstab query named Samples Given_Crosstab, as shown in Figure 12-54.
LEAVE the database open for the next project.

**Project 12-6: Create an Update Query**

The name of one of the hospitals in your region has recently been changed. You need to create an update query to change the name in the database.

USE the database that is open from the previous project.

1. Create a select query named **Update Query** that includes all the fields in the **Doctors: Table**.
2. Switch to Design view.
3. Use criteria to select only the records that have Community Medical Center in the Hospital field.
4. Use the skills you have learned in this lesson to create and save an update query that will change the name of Community Medical Center to Community Regional Hospital.
5. Open the **Doctors: Table** to verify that the hospital name has been changed. Then, close the table and the query.
6. CLOSE the database.

CLOSE Access.