Microsoft ®

Working with Queries in Access 2000

The Richard Stockton College of New Jersey

The Richard Stockton College of New Jersey
Computer and Telecommunication Services
PO Box 195
Pomona, NJ  08240
(609) 652-1776
Stockton College Web Site:  www.stockton.edu
Computer Services Web Site: compserv.stockton.edu
CustomGuide.com granted to Computer and Telecommunication Services a license agreement to print an unlimited number of copies of the CustomGuide Courseware materials within Stockton College of New Jersey for training staff, faculty and students.

End users who receive this handout may not reproduce or distribute these materials without permission. Please refer to the copyright notice below for more information.

Written by Jonathan High, CustomGuide, Minneapolis. Thanks to my wife, Sue for her enduring support, patience, and love.

© 2000 by CustomGuide, Inc. 4941 Columbus Avenue, Minneapolis, MN 55417

This material is copyrighted and all rights are reserved by CustomGuide, Inc. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of CustomGuide, Inc.

We make a sincere effort to ensure the accuracy of the material described herein; however, CustomGuide makes no warranty, express or implied, with respect to the quality, correctness, reliability, accuracy, or freedom from error of this document or the products it describes. Data used in examples and sample data files are intended to be fictional. Any resemblance to real persons or companies is entirely coincidental.

The names of software products referred to in this manual are claimed as trademarks of their respective companies. CustomGuide is a registered trademark of CustomGuide, Inc.
# Table of Contents

**Chapter One: Working with Queries**

- Lesson 1-1: A Quick Review ................................................................. 5
- Lesson 1-2: Understanding the Different Types of Queries ................... 6
- Lesson 1-3: Creating a Multiple Table Query ........................................ 8
- Lesson 1-4: Creating a Calculated Field ............................................... 10
- Lesson 1-5: Working with Expressions and the Expression Builder .......... 12
- Lesson 1-6: Using an IIF Function ...................................................... 14
- Lesson 1-7: Summarizing Groups of Records ....................................... 16
- Lesson 1-8: Display Top or Bottom Values .......................................... 18
- Lesson 1-9: Parameter Queries ............................................................ 20
- Lesson 1-10: Finding Duplicate Records ........................................... 22
- Lesson 1-11: Finding Unmatched Records .......................................... 24
- Lesson 1-12: Crosstab Queries ............................................................ 26
- Lesson 1-13: Delete Queries ............................................................... 28
- Lesson 1-14: Append Queries .............................................................. 30
- Lesson 1-15: Make-Table Queries ....................................................... 32
- Lesson 1-16: Update Queries ............................................................... 34
- Chapter One Review ............................................................................. 36
Chapter Objectives:

- Understand the various types of queries and their purposes
- Create queries based on more than one table
- Create queries that calculate and summarize information
- Use the Expression Builder to create expressions
- Create parameter queries which prompt the user for information
- Create queries that find duplicate and unmatched records
- Create Action queries that delete, update, append, and export information

Chapter Task: Create a variety of queries that analyze and manipulate database information

Queries are the stars of Microsoft Access. Queries make sense out of all thousands of jumbled records and display exactly what you need to know. Queries discover things like what the average price of tea in China is or which customers bought the most parakeet food from your company. Queries can even make widespread changes to the records in your database without wearing out your mouse and keyboard! For example, a delete query can automatically delete a whole bunch of records that meet your criteria.

In this chapter you will learn how to harness the power of queries. First you will learn about all the different types of queries: simple select queries, parameter queries that prompt you for more information, crosstab queries that summarize records in an easy understand format, action queries that actually modify the records in your database.

All this power comes with a price tag: many people find that queries are one of the more difficult database objects, and learning how to fully utilize queries isn’t something you can learn in an afternoon. By the time you finish this chapter, however, you will have a good start towards understanding and mastering queries.

Prerequisites

- How to use menus, toolbars, dialog boxes, and shortcut keystrokes.
- How to open and modify database objects.
- A basic understanding of queries: how to specify criteria and sort information.
Lesson 1-1: A Quick Review

Before we start tackling the functions and types of queries, let’s take a step back and review. This lesson is more of a “cheat sheet” than an exercise. Hopefully it will help you remember what you already know about queries.

Table 1-1: Common Criteria Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>&quot;MN&quot;</td>
<td>Finds records equal to MN</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>&lt;&gt;&quot;MN&quot;</td>
<td>Finds records not equal to MN</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;10</td>
<td>Finds records less than 10</td>
</tr>
<tr>
<td>&lt;=</td>
<td>&lt;=10</td>
<td>Finds records less than or equal to 10</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;10</td>
<td>Finds records greater than 10</td>
</tr>
<tr>
<td>=&gt;</td>
<td>=&gt;10</td>
<td>Finds records greater than or equal to 10</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>BETWEEN 1/1/99 AND 12/31/99</td>
<td>Finds records between 1/1/99 AND 12/31/99</td>
</tr>
<tr>
<td>LIKE</td>
<td>LIKE &quot;S&quot;</td>
<td>Finds text beginning with the letter S</td>
</tr>
<tr>
<td>NOT</td>
<td>NOT &quot;MN&quot;</td>
<td>Finds records not equal to MN</td>
</tr>
<tr>
<td>IS NULL</td>
<td>IS NULL</td>
<td>Finds records whose fields are empty</td>
</tr>
<tr>
<td>IS NOT NULL</td>
<td>IS NOT NULL</td>
<td>Finds records whose fields contain values.</td>
</tr>
</tbody>
</table>
Lesson 1-2: Understanding the Different Types of Queries

Up until now, when you thought of a query you were actually probably thinking of a Select query—a particular type of query. Select queries are by far the most common and useful type of query in Access, however there are other types of queries that are also important. We will cover most of these query types in this chapter, so here's a quick “overview” of the different types of queries you’ll find in Microsoft Access.

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Query</td>
<td>The most basic and common type of query, select queries find and display the data you want from one or more tables or queries.</td>
</tr>
<tr>
<td>Parameter Query</td>
<td>Prompts the user for specific information every time the query is run.</td>
</tr>
<tr>
<td>Crosstab Query</td>
<td>Summarizes data in a table format that makes it easy to read and compare information.</td>
</tr>
<tr>
<td>Action Queries</td>
<td></td>
</tr>
<tr>
<td>Make-Table Query</td>
<td>Creates a new table from all or part of the data in one or more tables. Useful for backing up and exporting information.</td>
</tr>
<tr>
<td>Append Query</td>
<td>Appends or adds selected records from one table to another table. Useful for importing information into a table.</td>
</tr>
<tr>
<td>Delete Query</td>
<td>Deletes selected records from one or more tables.</td>
</tr>
<tr>
<td>Update Query</td>
<td>Updates selected information in a table. For example, you could raise the prices on all trips to Europe by 15%</td>
</tr>
<tr>
<td>Union Query</td>
<td>Union queries combine fields from two or more tables or queries into one field. Union Queries are written directly in SQL.</td>
</tr>
</tbody>
</table>
Lesson 1-3: Creating a Multiple Table Query

In Access you will often need to look at and analyze information that comes from not one but several different tables. Since Access is a relational database it’s easy to establish a relationship between two or more tables and to look at the information that goes together.

Just like it sounds, a multiple-table query blends together information from two or more related tables. Working with a multiple-table query usually isn’t much different than a single-table query. You tell Access which tables you want to use in your query and specify the fields and criteria you want to see. The main difference between a multiple-table query and a single-table query is that with multiple-table queries, Access creates a link between related tables. In query design view, this link (called a join) appears as a line that connects two more tables.

When you create a multiple-table query Access will usually link or join the tables automatically. Sometimes, however you will have to manually join two tables in the Query design window. You manually join two tables by dragging from one field list to the other matching field in the field list for the other table. If the tables don’t have any fields in common, you must add another table to act as a bridge between them. This lesson will give you some experience creating a multiple-table query.

1. Open the Lesson 7 database.
   You should already know how to create a query by now…
2. Click the Queries icon in the Objects bar then double-click the ✉️ Create query in design view icon.

   The Query design window and Show Tables dialog box both appear. You have to select the tables and/or queries you want to use.

3. Click the tblEmployees table and click Add.

   A field list for the tblEmployees table appears in the top half of the Query design window. You also want to add the tblCustomerTours table and tblTours table to the query.

4. Add the tblCustomerTours table and tblTours table to the query.

   When you have finished adding the tables and/or queries to your query you can close the Show Tables dialog box.

5. Click Close.

   If the two tables are related, Access will automatically connect their common fields with a join line. For example, Access automatically joins the tblTours table and tblCustomerTours tables because they are already related. If the tables aren’t related you will have to manually join the tables by dragging from one field list to the other matching field in the field list from the other table.

6. Click on EmployeeID in the tblEmployees field list and drag and drop it onto Employee in the tblCustomerTours field list.

   NOTE: You can remove a join from a query by clicking the join line (careful—there’s not much there to click!) and pressing <Delete>.

   Next you need to specify the fields you want to appear in the query results. You can add fields to the query design grid in two ways:
   - By double-clicking the field on the field list.
   - By clicking and dragging the field down to the design grid yourself.

   Because field lists don’t have much room to display their contents you may have to scroll up or down the list in order to find the field you want.

7. Double-click the LastName and FirstName fields in the tblEmployee field list.

   Access adds the LastName and FirstName fields from the tblEmployees table to the design grid. Next add the fields for the Tour and tblCustomerTours tables.

8. Double-click the TourName field in the tblTour field list and the Date and Cost fields in the tblCustomerTours field list.

   Next you need to specify any criteria for the query. For this exercise you want to see tours from the first quarter of the year.

9. Click the Date column’s Criteria row and type Between 1/1/00 and 3/31/00.

   You want to sort your query by date, so…

10. Click the Date field’s Sort Row and select Ascending.

    That’s it! You’ve just created a multiple-table query.

11. Save the Query as qryFirstQuarterTours.

    OK—let’s run our new query!

12. Click the Run button on the toolbar.

    Access displays the results of the query. The results show the names of the employees who sold tours and the names, dates, and costs of the tours.

13. Close the Query.
Lesson 1-4: Creating a Calculated Field

Normally when you create a database, you should only have to enter the information you need and not worry about data or values that Access calculate based on information already in the database. A calculated field performs some type of arithmetic on one or more fields in a database to come up with a completely new field. For example, if your database has an Order Total field and a Tax Rate field, Access can calculate these two fields to find out the Sales Tax for each order: \[ \text{Order Total} \times \text{Tax Rate} = \text{Sales Tax} \].

You must create an **expression** (or formula) to perform a calculation. To enter fields in an expression, type the field name in brackets \([\text{Order Total}]\). If a field name exists in more than one table you will need to enter the name of the table that contains the field in brackets \((\text{[Customer Tours]}!\text{Cost})\) followed by an exclamation mark (!). Then type the field name in brackets \([\text{[Order Total]}]\).

**Figure 1-6**
Creating a calculated field.

**Figure 1-7**
To enter fields in an expression type the field name in brackets \((\text{[Order Total]})\).

**Figure 1-8**
If a field name exists in more than one table you will need to enter the name of the table that contains the field in brackets \((\text{[Customer Tours]}!\text{Cost})\) followed by an exclamation mark (!). Then type the field name in brackets \((\text{[Order Total]})\).
The Richard Stockton College of New Jersey

Chapter One: Working with Queries

typing the table name ([Orders]), an exclamation mark (!), and then the field name ([Date]) or [Orders]! [Date]. Yes, calculated fields can be a little confusing at first...

This lesson will show you how to add a calculated field you a query.

1. **Click** the Queries tab, if necessary, click the qryEmployeeSales query, then **click Design**.
   
   Instead of creating a query from scratch, you can modify an existing query and save it with a different name.

2. **Click** the blank field cell of the fourth column and **type** Bonus: [Cost]*[Commission].
   
   The expression you entered will create a new calculated field to the query named Bonus that will display the results of the Cost field in the tblCustomerTours table multiplied by the Commission field in the tblEmployees table.

   Let’s see the results of our calculated field.

3. **Click** the Run button on the toolbar.
   
   Access displays the results of the query. The Bonus calculated field multiplies the Cost field by the Commission field in each record and displays the results.

   Save your changes in new query with a different name. Here’s how:

4. **Select** File → Save As/Export from the menu.
   
   The Save As dialog box appears.

5. **Type** qryEmployeeBonus in the New Name box and click OK.
   
   Access saves your changes in a new query named qryEmployeeBonus.

6. **Close** the query.

   You’re already familiar with some of the arithmetic operators used in expressions: They include math symbols such as the plus sign (+) to perform addition between values and the minus sign (-) to perform subtraction.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>^</td>
<td>Exponentiation</td>
</tr>
</tbody>
</table>

   You can also use the Expression Builder to help you create your calculated fields. **Click the Builder button** on the toolbar, double-click the field you want to use in the calculation, **click the button that corresponds to the calculation you want**, and then **click or type any other fields or values you want to use**.

   **Quick Reference**

   To Create an Expression or Calculation in a Query:
   1. Display the query in Design View.
   2. Click the Field Row of a blank column in the design grid.
   3. Enter the field name for the calculated field followed by a : (colon).
   4. Enter the expression you want Access to calculate, using the proper syntax. Or...

   **Click the Build button** on the toolbar, double-click the field you want to use in the calculation, **click the button that corresponds to the calculation you want**, and then **click or type any other fields or values you want to use** when you’re finished. **Click OK**.

   5. **Click** the View or Run button on the toolbar to see the results.
Lesson 1-5: Working with Expressions and the Expression Builder

You can add calculations to queries, forms, and reports by typing an expression or formula that tells Access exactly what to calculate. An expression can be any combination of values, identifiers (such as the value in a field), and operators that result in a value. Here's an example of an expression that calculates profit from two fields called Income and Expenses:

`Profit: [Income] - [Expenses]`

You can also use constants in an expression, such as:

`Commission: [Sales] * .15`

Number fields aren’t the only types of fields that you can use in expressions—you can also perform calculations with dates, times, and text data.

The problem with creating expressions is you have to enter a formula so that Access understands it (and believe me, Access isn’t very bright.) For example, when you create an expression, some types of information must be enclosed between special characters so that Access knows what type of information it is—table names and field names must be enclosed in [brackets], text strings in "quotation marks" and so on. Table 1-4: How Types of Data Should Look in an Expression has more information about how to use various elements in an expression.

If you know what you want an expression to do, but not how to write it, you can try using the Expression builder. The Expression builder lets you pick the fields, mathematical symbols, and functions you can use to create an expression.

This lesson will give you some more experience writing expressions—both on your own and using the Expression builder.

1. **Open the qrySales query in Design view.**

   First you need to insert a blank column at the beginning of the design grid. Here’s how:
2. Position the mouse over the top of the **Normal Price** field (↑ changes to ↓) and click to select the field. Press <Insert> to add a new column. A column is inserted at the beginning of the design grid. In the next step, you will learn how to create one of the most common database functions: how to combine the tblEmployee table’s FirstName and LastName fields to display the full name.

3. **Type Agent:[FirstName]&" "&[LastName] in the new blank Field row.**
   The ampersand (&) symbol is used to combine or **concatenate** two or more text fields. The " " adds a space between the [FirstName] and [LastName] fields.
   Next you need to enter an expression that calculates the cost of the tour by multiplying the Number of Tickets field by the Normal Price field. You will use the Expression Builder to help you write this expression.

4. **Click the first blank field row and click the **Build button** on the toolbar.**
   The Expression Builder appears, as shown in Figure 1-9. The Expression Builder contains an area where you can build the expression, buttons you can use to build the expression, and the fields and controls in the current query, report, or form.

5. **Double-click Normal Price, click the **Multiplication button**, and double-click Number of Tickets.**
   The completed expression appears in the Expression list box, as shown in Figure 1-9. You’re finished writing the expression so you can close the Expression Builder.

6. **Click OK.**
   The Expression Builder closes. Add a meaningful label to the new calculated field.

7. **In the new calculated field, replace the Expr1: label with Total:**
   The edited expression should read Total:[Normal Price]*[Number of Tickets]. Let’s see the results of the new query.

8. **Click the **Run button** on the toolbar.**
   The query combines the FirstName and LastName fields in the new Name field and displays the total sales price in the New Total Field.

9. **Click the **Save button** on the menu to save your changes and then close the query.**

Microsoft Access is very strict about how you write your expressions. If your expressions aren’t written in the correct syntax, they won’t work. Use the following table as a guideline for adding fields, text, and constants to your expressions.

<table>
<thead>
<tr>
<th>Type of Data</th>
<th>How It Should Look</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>&quot;Minneapolis&quot;</td>
</tr>
<tr>
<td>Date/Time</td>
<td>#20-Mar-99# (Access will add the # symbols)</td>
</tr>
<tr>
<td>Field Name</td>
<td>[Price]</td>
</tr>
<tr>
<td>Field Name in a Specific Table</td>
<td>[Products]![Price]</td>
</tr>
<tr>
<td>Concatenated (Combined) Text and Fields</td>
<td>[Last],&amp;&quot;,&quot;&amp;[First]</td>
</tr>
<tr>
<td>Calculated Field (Using Two Fields)</td>
<td>[SalePrice]-[Cost]</td>
</tr>
<tr>
<td>Calculated Field (Using a Field and a Constant)</td>
<td>[SalePrice]*0.1</td>
</tr>
</tbody>
</table>
Lesson 1-6: Using an IIF Function

**Figure 1-10**
The syntax for the IIF function.

**Figure 1-11**
Here the IIF function evaluates the value in the First Class field and returns 50 if the First Class field is True and 25 if the First Class field is false.

**Functions** are used to create more complicated calculations or expressions than operators can. For example, the SUM function adds several values together and the IPmt function calculates the loan payments based on an interest rate, the length of the loan, and the principal amount of the loan.

There are several hundred functions in Access, but all of them are used in a similar way. The name of the function followed by the arguments in parenthesis. An **argument** in Access is the value a function uses to perform its calculation—not the type you have over political views. For example, the argument in the formula \( \pi r^2 \) would be \( r \) or the radius, used to find the area of a circle.

This lesson introduces a very useful database function, the **IIF** function. The IIF function is a **conditional function** or **logical function** because it evaluates a condition and return one value if the condition is true and another value if the condition is false. For example, you could use the IIF function in an invoice to create a formula that would subtract a 5% discount from the invoice if the total were more than $500.00—otherwise the IIF Function wouldn’t subtract anything.

The IIF function contains 3 arguments, as shown in **Figure 1-10**. Since you can use the Expression Builder to help you create IIF function formulas you really don’t need to memorize the syntax of the function.

In this lesson you will use the IIF function to create a field that gives passengers a $50 rebate if they fly first class and a $25 rebate if they fly coach.

1. **Open the qryRebate query in Design view.**
   You need to create a calculated field that will determine how much of a rebate passengers should receive. Create this field in the first blank field row.

2. **Click the blank Field row in the second column and click the Build button on the toolbar.**
   The Expression Builder appears.
Chapter One: Working with Queries

1. Quick Reference

To Create an IIF (IF…THEN) Function:
1. Display the query in Design View.
2. Click the Field Row of a blank column in the design grid.
3. Enter the field name followed by a : (colon).
4. Type the expression using the syntax IIF(«expr», «truepart», «falsepart») or…
Click the Build button on the toolbar. Double-click the Functions folder in the bottom-left window. Double-click the IIF function in the bottom-left window.
5. Scroll down the bottom-middle window and find and select the Program Flow function category.
6. Double-click the IIF function in the right window.
Access adds IIF («expr», «truepart», «falsepart») to the expression box. Now that you know the proper syntax of the IFF function, you need to replace the argument names with the data values. You can double-click to select any argument name so that you can replace it with your own value.
7. Double-click the «expr» argument.
You need to specify the logical test—if the passenger flew first class or not.
8. Click the qryRebate folder in the right window and then double-click the First Class field in the middle window.
The Expression Builder adds the First Class field to the expression. Finish the rest of the logical test.
9. Type =True.
Your expression should read IIf ([First Class]=True, «truepart», «falsepart»). Now you have to replace the «truepart» and «falsepart» arguments with the values you want to use if the IIf statement is true or false.
10. Double-click the «truepart» argument and type 50, then double-click the «falsepart» argument and type 25.
11. Click OK.
The Expression Builder closes. We need to give the new calculated control a more meaningful name…
12. In the new calculated field, replace the Expr1: label with Rebate:
Let’s see the results of the new query.
13. Click the Run button on the toolbar.
The IIF expression in the new Rebate field evaluates the First Class field and returns a $50 rebate if the customer flew First Class and a $25 rebate if they didn’t.
Save your changes.
14. Save your changes and close the query.
Lesson 1-7: Summarizing Groups of Records

When you work with queries, you will often be less interested in the individual records and more interested in summarized information about groups of records. A query can calculate information about a group of records in one or more tables. For example, you could create a query that finds the total amount of tea your company sold to China in 1998 or how much all that tea cost. The **Total row** lets you group and summarize information in a query. The Total row normally is tucked away from view in the query design window—you can make the Total row appear by clicking the Total button on the toolbar or by selecting View → Totals from the menu. Once the Totals row is displayed you can tell Access how you want to summarize the fields.

1. **Open the qryTourSales query in Design view.**
   First you need to add the field that you want to group data by onto the design grid. You want to calculate the total sales and number of tickets sold for each tour package, so you will group the query by the TourName field.

2. **Double-click the TourName field in the tblTours table field list.**
   The TourName field appears as the first field in the design grid. Notice that Group By appears in the TourName Total row—the query will group the records by the TourName field. Next you need to add the fields you want to summarize.

3. **Double-click the Number of Tickets and Cost fields in the tblCustomerTours field list.**
   To summarize your query you must summon the Total row. To summon the Total row click on the Totals button on the toolbar or select View → Totals from the menu.

4. **Click the Totals button on the toolbar.**
   The Totals row appears in the design grid. Next you need to select the fields you want to summarize and the calculation you want to perform on them.
5. Click the **Number of Tickets** column’s **Total row** and click theMBED arrow that appears.  
A list of calculations appears. You simply select the calculation you want to perform on the field. **Table 1-5: Total Options** describes the available calculations.

6. **Select Sum** from the list.  
This will total the values in the Number of Tickets field.

7. Click the **Cost** column’s **Total row**, click theMBED arrow and select **Sum** from the list.  
You can specify criteria to limit the records you want to be calculated—simply enter the criteria in the Criteria row of any grouped or calculated fields. If the field you want to use for the criteria isn’t one of the grouped or calculated field you must use the Where option in the field’s Total row. The Where option limits the records used in the calculation without being included in the query results.

You want to calculate only those records from the second quarter of the year.

8. **Double-click the Date field in the tblCustomerTours field list.**  
Here’s how to add criteria to the Date field.

9. Click the **Date** column’s **Total row**, click theMBED arrow and select **Where** from the list.  
The Where option is used only to limit records—its results cannot be displayed in the results of the query. Access knows this and automatically turns off the Show checkbox.

10. Click the **Date** column’s **Criteria row** and type **Between 4/1/00 and 6/30/00**.  
You’re ready to see the results of the new query.

11. **Click the Run button on the toolbar.**  
Access displays the results of the query, which calculates the total sales and number of tickets sold for each tour package.

12. **Save the query as qryTourTotals and then close the query.**

---

**Table 1-5: Total Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group By</td>
<td>Groups the values in the field so that you can perform calculations on the groups.</td>
</tr>
<tr>
<td>Sum</td>
<td>Calculates the total (sum) of values in a field.</td>
</tr>
<tr>
<td>Avg</td>
<td>Calculates the average of values in a field.</td>
</tr>
<tr>
<td>Min</td>
<td>Finds the lowest value in a field.</td>
</tr>
<tr>
<td>Max</td>
<td>Finds the highest value in a field.</td>
</tr>
<tr>
<td>Count</td>
<td>Counts the number of entries in a field, not including blank (Null) records.</td>
</tr>
<tr>
<td>StDev</td>
<td>Calculates the standard deviation of values in a field.</td>
</tr>
<tr>
<td>Var</td>
<td>Calculates the variance of values in a field.</td>
</tr>
<tr>
<td>First</td>
<td>Finds the values from the first record in a field.</td>
</tr>
<tr>
<td>Last</td>
<td>Finds the values from the last record in a field.</td>
</tr>
<tr>
<td>Expression</td>
<td>Tells Access that you want to create your own expression to calculate a field.</td>
</tr>
<tr>
<td>Where</td>
<td>Specifies criteria for a field to limit the records included in a calculation.</td>
</tr>
</tbody>
</table>
Lesson 1-8: Display Top or Bottom Values

If all you care about is the highest or lowest values produced by a query you can use the Top Values list in the Query Design toolbar to display only these records. For example, you could use the top Values list to display the ten largest or smallest orders in the Invoices table.

This lesson explains how you can use the Top Values list to display the top or bottom values in a query.

1. **Open the qryTourSales query in Design view.**
   Move to the next step and add the fields you want to see in your query.

2. **Double-click the TourName field in the tblTours field list, and the Cost and Smoker fields in the tblCustomerTours field list.**
   You want to create a query that retrieves the five most expensive non-smoking tours. First you need to add the non-smoking criteria to the query.

3. **Click the Smoker column’s Criteria row and type False.**
   Next you have to sort the field that you want to display the top or bottom values. The Sort row works a little differently when you’re using top or bottom values:
   - **Ascending:** Displays bottom values
   - **Descending:** Displays top values
   You want the query to display the top values in the Cost field, so…
4. **Click the Cost field’s Sort Row and select Descending.**
   Next you have to use the Top Values list to specify the number of top values you want to display in your query results.

5. **Click the Top Values list arrow on the toolbar and select 5.**
   This will display the five most expensive tickets. You’re ready to run the query.

6. **Click the Run button on the toolbar.**
   Access displays the results of the query, as shown in Figure 1-14. (There are more than five records because some of the records have the same values.)

7. **Click the View button on the toolbar to display the query in Design view.**
   You can also use the Top Values feature to display the top or bottom values from a calculation. Let’s modify the query so that it calculates the total sales of non-smoking tour sales.

8. **Click the Totals button on the toolbar.**
   The Totals row appears. You need to specify which field you want to use to group the query, which field(s) you want to be calculated, and which fields are used to limit the number of record displayed in the query.

   You want to group records using the TourName field, and since its Total row already displays Group By you can leave it as it is. You will need to tweak both the Total row for both the Cost and Smoker fields, however.

9. **Click the Cost column’s Total row, click the arrow and select Sum from the list.**
   This will total the Cost field. The Smoker field is used as criteria to limit the records displayed to only nonsmokers—it shouldn’t be included in the query results. Move on to the next step to tell Access this.

10. **Click the Smokers column’s Total row, click the arrow and select Where from the list.**
   One more change before we run the query—this time instead of displaying the top values for the Cost field we want to display its bottom values.

11. **Click the Cost field’s Sort Row and select Ascending.**
    This will summarize the five tours with the lowest sales (for non-smokers). Let’s see the results…

12. **Click the Run button on the toolbar.**
    Access displays the results of the query, as shown in Figure 1-15.

13. **Save the query as qryBottomTours and then close the Query.**
    This table explains what each of the options in the Top Values list does.

<table>
<thead>
<tr>
<th><strong>Table 1-6: The Top Values List</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Do This…</strong></td>
</tr>
<tr>
<td>Click 5, 25, or 100 from the Top Values List</td>
</tr>
<tr>
<td>Type a number, such as 15 in the Top Values box</td>
</tr>
<tr>
<td>Click 5% or 25% from the Top Values List</td>
</tr>
<tr>
<td>Type a percentage, such as 20%, in the Top Values box</td>
</tr>
<tr>
<td>Click All from the Top Values list</td>
</tr>
</tbody>
</table>
Lesson 1-9: Parameter Queries

Getting tired of modifying a query every time you want to use a new criterion? A parameter query is your answer. A parameter query prompts the user for the query’s criteria. For example, you could create a Regional Sales query that would ask for the name of the state that you want filter by.

Creating a parameter query is easy. All you have to do is click the Criteria row for the field that you want to use a parameter and type a message, enclosed in [brackets] that you want Access to display when you run the query.

1. **Open the qryTourSales query in Design view.**
   As always, you need to add the field that you want to group data by onto the design grid. This time we want to create a query that summarizes total employee sales. Here are the fields that we will use in the query:

   2. **In the tblEmployees field list double-click the FirstName, LastName, and State fields and in the tblCustomerTours field list double-click the Cost and Date fields.**

   Next you need to tell the query that you want to group and summarize the query.

   3. **Click the Totals button on the toolbar.**

   The Totals row appears. You need to specify which field(s) you want to use to group the query, which field(s) you want to be calculated, and which fields are used to limit the number of record displayed in the query.
You want to group records using the FirstName, LastName, and State fields and find the total of the Cost field.

4. **Click the Cost column’s Total row, click the arrow and select Sum from the list.**
   
   This will total the Cost field. We will use the Date field as criteria to limit the records to those that fall between two dates. Instead of entering a criteria expression with two fixed date values, such as Between 1/1/00 and 3/31/00, we will create two parameters that will prompt the user to enter the two date values each time they run the query.

5. **Click the Date column’s Criteria row and type Between [Enter start date] and [Enter end date].**
   
   You’ve just created two parameters—the [Enter start date] parameter and the [Enter end date] parameter.
   
   Since the Date field is only being used as a criteria field you need to select the Where option from its Total row.

6. **Click the Date column’s Total row, click the arrow and select Where from the list.**
   
   The Show box automatically turns off, indicating that the Date field will not appear in the query results.
   
   Let’s add one more parameter—one that prompts the user to enter the state where the tour was sold.

7. **Click the State column’s Criteria row and type [Enter the state].**
   
   This will prompt the user to enter the name of the state.
   
   Since the State field is only being used as a criteria field you need to select the Where option from its Total row.

8. **Click the State column’s Total row, click the arrow and select Where from the list.**
   
   Let’s test our parameter query.

9. **Click the Run button on the toolbar.**
   
   Access prompts you to enter the first parameters, as shown in Figure 1-17. You want to summarize records from Washington.

10. **Type WA and click OK.**
    
    Access prompts you for the next parameter—the start date.

11. **Type 1/1/00 and click OK.**
    
    Access prompts you for the last parameter—the end date.

12. **Type 6/30/00 and click OK.**
    
    Access displays the results of the parameter query, as shown in Figure 1-18.

13. **Save the query as qryParameter and then close the Query.**

Some advanced Access developers use custom-made forms to provide parameter queries with their information. By binding a parameter to the controls on a form (such as [frmCustomers]![Name]) users can fill out one dialog instead of having to fill out 5 or 6 pop-up dialog boxes.
Lesson 1-10: Finding Duplicate Records

The Find Duplicates Query Wizard helps you find records that have the same value in one or more fields. So when would need to use a Find Duplicates Query? Here are a few scenarios:

- To search for duplicate values in an Orders table to find out which customers have placed more than one order.
- To search for duplicate values in several fields to search for data-entry errors. For example, if you and another user each accidentally entered the same customers into a table you could search for duplicate values in the LastName and FirstName fields to find and delete the duplicated records.

Access provides a wizard to make creating a query that finds duplicate information a snap.

“Arrr-rrgh!” You shout as you realize that you and another co-worker have just entered the same customers into a database. Fortunately you can find the duplicated records by using the Find Duplicates Query Wizard.

1. From the Database window, click the Queries icon in the Objects bar and click the New button.

The New Query dialog box appears.
2. Select the **Find Duplicates Query Wizard** and click **OK**.
The first step of the Find Duplicates Query Wizard appears, as shown in [Figure 1-19](#).
You need to choose the table or query that you want to sift through for duplicate records.

3. Select the **tblCustomers** table and click **Next**.
Step two of the Find Duplicates Query Wizard appears, as shown in [Figure 1-20](#). Here you tell Access which field or fields might contain the duplicate information.
Since you are trying to find duplicate customers you decide to search the FirstName and LastName fields for duplicate values.

4. Double-click the **LastName** and **FirstName** fields.
The LastName and FirstName fields appear in the selected fields list.

5. Click **Next**.
The next step of the Duplicates Query Wizard appears. You can select any field (other than the ones you specified in Step 4) that you want to be displayed in the query.
You decide to display the Address and City fields so that you can verify that the records are indeed duplicates.

6. Double-click the **Address** and **City** fields. Click **Next** when you’re finished.
You’re finished! Well, almost. You have to give your query a name—or you can accept the default name Access gives the query.

7. Click **Finish**.
Access saves the query with the name “Find duplicates for Customers” and displays the results of the query, as shown in [Figure 1-21](#).

8. Close the query.

---

**Quick Reference**

To Use the Find Duplicate Wizard:
1. From the Database window, click the **Queries** icon in the Objects bar and click the **New button**.
2. Select the **Find Duplicates Query Wizard** and click **OK**.
3. Double-click the field(s) that may contain the duplicate values and click **Next**.
4. Double-click any additional fields that you want to appear in the query results and click **Next**.
5. Click **Finish**.
Lesson 1-11: Finding Unmatched Records

The Find Unmatched Records Query Wizard helps you find records in one table that do not have matching records in another table. Some scenarios when you might need to create such a query include:

- To find customers who have never placed an order.
- To find products that have never been purchased
- To find “orphan” records. If you haven’t enforced referential integrity in your related tables, deleting a record in one table could leave one or more orphan records in a related table. For example, if you delete a customer record from a Customer table you may leave several unmatched records for that customer in an Order table.

In this lesson you will use the Find Unmatched Records Query Wizard to create a query to find customers who have never booked a tour.

1. **From the Database window, click the Queries icon in the Objects bar and click the New button.**
   
The New Query dialog box appears.
2. **Select the Find Unmatched Records Query Wizard and click OK.**
   The first step of the Find Unmatched Records Query Wizard appears, as shown in Figure 1-22. You need to choose the table or query whose values you want to display in the query. Since you want to find customers without any tour packages you would select the tblCustomers table.

3. **Select the tblCustomers table and click Next.**
   Step two of the Find Unmatched Records Query Wizard appears. Here you have to tell Access which table contains the related records. Here’s where you would select the tblCustomerTours table.

4. **Select the tblCustomerTours table and click Next.**
   The third step of the Find Unmatched Records Query Wizard appears, as shown in Figure 1-23. Here you have to specify the related field to join the records in the first table to the records in the second table. Once you have selected the matching record in both tables, click the button to join the two tables.

5. **Verify that the CustomerID field is selected in both tables and click the button.**
   The Matching Field area displays the fields used to join the tables (CustomerID <= CustomerID).

6. **Click Next.**
   Almost there! Now you have to specify which fields you want to see in the query.

7. **Double-click the LastName, FirstName, City, and State fields.**
   The LastName, FirstName, City, and State fields should all appear in the Select Fields list.

8. **Click Next.**
   You have to give your query a name—or you can accept the default name access gives the query.

9. **Click Finish.**
   Access saves the query with the name “Customers Without Matching Customer Tours” and displays the results of the query, as shown in Figure 1-24.

10. **Close the query.**
Lesson 1-12: Crosstab Queries

There are many ways that queries can help you summarize and analyze all that information in your database. A **crosstab query** displays summarized information in a table format that makes it easy to analyze and compare data. Look at the information displayed in Figure 1-25—it's difficult to see the bottom line, isn't it? Now look what happens when the same information is placed in a crosstab query, as shown in Figure 1-26. Which do you think is easier to understand?

You can create a crosstab query in Design view or by using the Crosstab Query Wizard. The Crosstab Query Wizard is usually much easier, but it does have some limitations:

- If you need to use more than one table or query in the crosstab query, you will first need to create a separate query that has the tables you want to use.
- You can't specify any limiting criteria when using the Crosstab Query Wizard (of course you can always modify the crosstab query in Design view and add the criteria yourself).
In this lesson you will use the Crosstab Query Wizard to create a query that summarizes monthly ticket sales by tours.

1. **From the Database window, click the Queries icon in the Objects bar and click the New button.**
   The New Query dialog box appears.

2. **Select the Crosstab Query Wizard and click OK.**
   The first step of the Crosstab Query Wizard appears. You need to select the table or query that contains the values you want. For this exercise you will use the Tours by Query as the source for the crosstab query.

3. **Click the Queries option in the View section to display the queries in the database, select the qryToursByName query and click Next.**
   The second step of the Crosstab Query Wizard is which field you want to use as the row headings for the crosstab. You want to use the TourName field as row headings.

4. **Double-click the TourName field and click Next.**
   The next step is determining which field you want to use as your column headings. You want to use the Date field as the column heading.

5. **Double-click the Date field and click Next.**
   Because you selected a date field using the Crosstab Query Wizard asks by which interval you want to group the dates: By date, month, quarter, or year. For this exercise you want the date column to group dates by months.

6. **Double-click the Month option and click Next.**
   Probably the most important step in the Crosstab Query Wizard is determining which field you want to calculate where columns and rows intersect and the type of calculation you want to use to summarize the fields.

7. **Select the Number of Tickets field from the Field list and the Sum option from the Functions list, as shown in Figure 1-27.**
   This will calculate the total number of tickets sold for each tour, grouped by month.

8. **Click Next.**
   You have to give your crosstab query a name.

9. **Type qryTicketsByDate and click Finish.**
   Access saves the query with the name “qryTicketsByDate” and displays the results of the query, as shown in Figure 1-26. Let’s modify the crosstab query and add some limiting criteria.

10. **Click the View button to display the crosstab query in Design view.**
    The crosstab query appears in Design view as shown in Figure 1-28. Notice the Crosstab row, which you use to determine if a field should be a column heading, row heading, or value.

11. **Double-click the First Class field in the qryToursByName list.**
    You only want to see tours for passengers without first class tickets.

12. **Click the First Class column’s Total row, click the arrow and select Where from the list. Click the Total row column’s Criteria row and type False.**

13. **Click the Run button on the toolbar.**
    Access displays the crosstab query, which only includes non-first class tickets.

14. **Close the query without saving your changes.**
Lesson 1-13: Delete Queries

Definitely the most dangerous of all queries is a delete query is an of action query that deletes a whole bunch of records at a time. Once you have deleted records using a delete query, you cannot undo the results—the records are gone forever! Creating a Delete query is no different than creating a Select query—with one very important difference: While a Select query displays the records that match your criteria a Delete query deletes those records. For this reason you should always preview the results of a Delete query in Datasheet view to see which records will be deleted.

If you want to delete records from multiple tables—for example a customer and all their orders—you need to do a few things first:

- Define relationships between the tables.
- Establish Referential Integrity for the join(s) between the tables and turn on the Cascade Delete Related Records option.

LeAnne Chang, one of North Shore Travel’s managers, decides to open her own competing travel agency across the street, and being the vindictive person she is, takes all her customers with her. In this lesson you will create a Delete query to delete all the tours sold by LeAnne Chang.

1. From the Database window, click the Queries icon in the Objects bar and click the New button.

   The New Query dialog box appears.

2. Select Design View and click OK.

   The Query design window and Show Tables dialog box both appear. You have to select the tables and/or queries you want to use in the Delete query.

3. Click the tblEmployees table and click Add.

   A field list for the tblEmployees table appears in the top half of the Query design window.
4. **Add the tblCustomerTours table and tblTours table to the query.**

   When you have finished adding the tables and/or queries you can close the Show Tables dialog box.

5. **Click Close.**

   If the tables are related, Access automatically connects their common fields with a join line. If the tables aren’t related, you will have to manually join them by dragging from one field list to the other matching field in the field list from the other table.

   You need to connect the tblEmployees table with the tblCustomerTours table.

6. **Click on EmployeeID field in the tblEmployees field list and drag and drop it onto the Employee field in the tblCustomerTours field list.**

   Next you need to tell Access that this is a Delete query.

7. **Click the Query Type button arrow on the toolbar and select Delete Query from the list.**

   Access converts the Select query to a Delete query and displays the Delete row in the query design grid. Now you have to tell Access what you want to delete.

8. **Drag the asterisk (*) from the top of the tblCustomerTours field list into the design grid.**

   Notice that From appears in the Delete cell for the asterisk field, indicating the records will be deleted from the tblCustomerTours table. Unless you want the Delete query to delete each and every record in the tblCustomerTours table you will need to add some limiting criteria. You decide to use the last name Chang as the limiting criteria.

9. **Drag the LastName field from the top of the Employees tblTours field list into the design grid.**

   This time Where appears in the Delete cell for the LastName field, indicating the LastName field will be used as the criteria to select which records will be deleted from the tblCustomerTours table.

10. **Click the LastName column’s Criteria row and type Chang.**

    Access will add the “quotation marks” around the text string Chang for you. That’s all there is to creating a Delete query. Before you run a delete query you should always preview the results in Datasheet View.

    **NOTE:** Before you run a Delete query, you should always preview the results in Datasheet view first. Click the View button on the toolbar to display the Delete query in Datasheet View.

11. **Click the View button on the toolbar to preview the results of the Delete query in Datasheet view.**

    The Delete query displays the records belonging to LeAnne Chang that will be deleted upon your command. Switch back to Design view.

12. **Click the View button on the toolbar to return to Design view.**

    OK, let’s run the Delete query and delete the selected records.

13. **Click the Run button on the toolbar.**

    Ever cautious Access asks if you really want the Delete query to delete the records.

14. **Click Yes.**

    Access silently deletes all 75 tour records for LeAnne Chang.

15. **Close the query without saving your changes and open the qryToursByName query.**

    Notice that all the tours for LeAnne Chang are gone.

16. **Close the qryToursByName query.**

---

**Quick Reference**

**To Create a Delete Query:**

1. From the Database window, click the Queries icon in the Objects bar and click the New button.

2. Click the Query Type button list on the toolbar and select Delete Query.

Or...

Select Query → Delete Query from the menu.

3. Click the View button to view the results of the delete query.

4. If you’re satisfied that the appropriate records will be deleted, click the Run button on the toolbar to delete the records.
Lesson 1-14: Append Queries

An append query takes a group of records from one or more tables or queries in your database and adds them to another existing table. Append queries are especially useful for importing information into a table. For example, you could use an Append query to import several dozen customer records from an Excel spreadsheet into an existing table. Of course you would have to know how to import the Excel spreadsheet first—and that’s another lesson in and of itself.

There are several rules that you must follow when using an Append query.

- The appended data must meet the data validation and referential integrity rules of the table it is being added to.
- The appended data must have its own unique primary key values. If the primary key field in the table to which the data is being added is an AutoNumber field, do not append that field—Access will generate new numbers for the new records.
- The type of data in the records you’re adding must match the type of data in the table to which you’re adding them.

In this lesson you will create an Append query to add a new group of tours to the tblCustomerTours table.

1. From the Database window, click the Queries icon in the Objects bar and click the New button.

The New Query dialog box appears.
2. Select Design View and click OK.
   The Query design window and Show Tables dialog box both appear. You have to select
   the tables and/or queries you want to use in the Append query.
3. Click the tblCaribbeanTours table, click Add and then click Close.
   Now you need to convert the Select query to anAppend query. Here’s how:
4. Select Query \(\rightarrow\) Append Query from the menu.
   The Append dialog box appears, as shown in Figure 1-31. You have to tell the Append
   query where you want the results of the query to be added.
5. Select tblCustomerTours from the Table Name list and click OK.
   The Append query will add the results of its query to the tblCustomerTours table.
   Notice an Append row appears in the design grid. Now you have to specify the fields
   you want to append.
6. Double-click the CustomerID, EmployeeID, and TourID fields in the
   tblCaribbeanTours field list.
   Since the field names are the same in both tables, Access automatically fills in the
   Append To row with the names of the fields you’re appending records to. If some of
   the fields you’re appending have a different field name you will have to specify to
   which field they should be added from the Append To row.
7. Double-click the No Tickets field in the tblCaribbeanTours field list.
   Since there isn’t a field named No Tickets in the tblCustomerTours table, Access
   doesn’t automatically fill in the Append To row. You will have to select the name of the
   field you want to append to.
8. Click the No Tickets field’s Append To and select Number of Tickets.
   This will append the fields from the No Tickets field in the tblCaribbeanTours table to
   the Number of Tickets field in the tblCustomerTours table. Move on to the next step
   and finish adding the remaining fields that you want to append.
9. Double-click the Date, FirstClass, Smoker, Ship Via, and Cost fields in
   the tblCaribbeanTours field list.
   NOTE: As with any Action query you should always preview the results in Datasheet
   view first. Click the View button on the toolbar to display the results of
   the Append query in Datasheet view.
10. Click the View button on the toolbar to preview the results of the Append
    query in Datasheet view.
    The Append query displays the records it will add or append to the tblCustomerTours
    table.
11. Click the View button on the toolbar to return to Design view and click
    the Run button on the toolbar.
    Access asks you to confirm the addition of the records to the tblCustomerTours table.
12. Click Yes.
    Access adds the records from the tblCaribbeanTours table to the tblCustomerTours
    table.
13. Close the query without saving your changes and open the
    qryToursBy_Name query.
    Notice the new tblCaribbeanTours records that have been added by the Append query.
14. Close the Tours by Name query.

Quick Reference

To Create an Append Query:
1. In Query Design View
   create a select query, including any tables, fields, calculated fields, and criteria.
2. Click the Query button on the toolbar and select
   Append Query.
   Or...
   Select Query \(\rightarrow\) Append Query from the menu.
3. Select the table where
   you want to add the records to from the drop-down list.
4. Click one of the following options:
   Current Database: If the table is in the currently
   open database.
   Another Database: And type the name of the
   another database (including the path, if
   necessary).
5. Click OK.
6. Click the View button on the toolbar to view the results of the
   query or the Run button on the toolbar to append the records.
Lesson 1-15: Make-Table Queries

Like all queries, a make-table query asks a question of the information in one or more tables and then retrieves results. Instead of displaying the results however, a make-table query creates a new table with the results of the query. Make-table queries are useful for:

- Exporting a table to another database or application
- Creating a backup copy of a table
- Creating an archive table that stores old records
- Creating a table that includes information or fields from more than one table
In this lesson you will create a make-table query to create a table with information about all China tour records.

1. **Open the qryTourSales query in Design view.**
   First you need to add the fields that you want to include in your new table.

2. **Double-click the TourName field in the tblTours field list, the Number of Tickets, Date, and Cost fields in the tblCustomerTours field list, and the LastName and FirstName fields in the tblEmployees field list.**
   The make-table query will create a table with these fields. Next you need to specify any limiting criteria.

3. **Click the **TourName** column’s Criteria row and type China.**
   Access will add the “quotation marks” around the text string China for you. The make-table query will only include China tours when it creates the new table.
   Here’s how to change the query type to a Make-Table query:

4. **Click the **Query Type** button arrow and select **Make Table Query** from the list.**
   Access displays the Make Table dialog box, as shown in Figure 1-33. Here you need to tell Access the name of the new table.

5. **Type tblChinaTours in the Table Name box and click OK.**
   You’re ready to have the make-table query create the new table. Preview the results of the query first.
   **NOTE:** As with any Action query you should always preview the results in Datasheet view first. Click the View button on the toolbar to display the results of the make-table query in Datasheet view.

6. **Click the **View** button on the toolbar to preview the results of the make-table query in Datasheet view.**
   The Append query displays the records it will use to create the new tblChinaTours table.

7. **Click the **View** button on the toolbar to return to Design view and click the **Run button on the toolbar.**
   Access asks you to confirm the creation of the tblChinaTours table.

8. **Click Yes.**
   Access creates the new tblChinaTours table based on the results of the make-table query.

9. **Close the query without saving your changes and open the tblChina table.**
   Verify the results of the make-table query, then…

10. **Close the tblChinaTours table.**
Lesson 1-16: Update Queries

You can use an update query to change a pile of records at the same time. For example, you could create an update query to lower prices by 8% or to change the sales representative for all your clients in Oregon from Mr. Potter to George Bailey. Just like other action queries, you create an update query by first creating a select query and then converting the select query to an update query.

In this lesson you will create an update query to raise the prices of all trips to Europe by 10%.

1. From the Database window, click the Queries icon in the Objects bar and click the New button.
   The New Query dialog box appears.
2. Select Design View and click OK.
   The Query Design window and Show Tables dialog box both appear. You have to select the tables and/or queries you want to use in the update query.
3. Double-click the tblTours table and click Close.
   Now you need to convert the Select query to an Append query. Here’s how:
4. Click the Query Type button arrow and select Update Query from the list.
   Access converts the Select query to an Update query. Notice an Update To row appears in the design grid. Now you have to specify the fields you want to update.

5. Double-click the TourID, Normal Price, and First Class Price fields in the tblTours field list.
   The next step is a little bit tricky—you have to tell Access what fields to update and how to update them. You want to raise the price of both the Normal Price and First Class Price fields by 10%—you will have to write an expression (or formula) in the Update To rows of both fields to make this little bit of magic happen.

6. Click the Normal Price column’s Update To row and type 
   \([\text{Normal Price}] + ([\text{Normal Price}] * .1)\).
   So what does the confusing expression you entered actually mean? Let’s assume the Normal Price for a record is $100. The expression would then look something like this: 
   \([$100] + ([$100] * .1)\)
   Access first calculates anything it sees in parenthesis, so it multiples \(100 \times 0.1\) (or 10%) and comes up with $10. Access then adds this $10 to the Normal Price $10 + $100 and comes up with $110. Make a little more sense? Hope so—because you have to do the same thing to the First Class Price field.

7. Click the First Class Price column’s Update To row and type 
   \([\text{First Class Price}] + ([\text{First Class Price}] * .1)\).
   Next you need to specify any limiting criteria. You want to raise prices for only tours that are in Europe. Sadly, the only way to determine this is by looking at the values in the tblTours table. Oh, OK here are all the European TourIDs: 1, 2, 3, and 9. You will have to create an OR statement in your query to make sure you get them all.

8. Click the TourID field’s Criteria row and type 1.
   This will select the record whose TourID is 1 but what about the rest?

9. Press the <↓↓↓↓> (down arrow key) to select the TourID field’s or Criteria row and type 2.
   This will select the records whose TourID is 2. Go to the next step and finish adding the rest of the OR criteria.

10. Press the <↓↓↓↓> (down arrow key) to select the TourID field’s second or Criteria row, type 3, press the <↓↓↓↓> (down arrow key) to select the TourID field’s second or Criteria row and type 9.
    Compare your query to the one in Figure 1-36. Look the same? Let’s run the update query.

11. Click the Run button on the toolbar.
    Access asks about the updating the selected records, as shown in Figure 1-37.

12. Click Yes.
    Access updates the selected records by raising both their Normal Price and First Class Price fields by 10%.

13. Close the query without saving your changes and open the tblTours table
    Notice the prices of four European tours have been increased by ten percent.

14. Close the tblTours table.
Chapter One Review

Lesson Summary

A Quick Review

- To Create a Query in Design View: From the database window, click the Queries tab, click New, select Design View from the list and click OK. Select the table or query you want to use and click OK. Repeat as necessary for additional tables or queries you want to add to the query and click Close when you’re finished. Double-click the fields that you want to appear in the query or click and drag the fields onto the design grid. In the design grid, enter any desired search criteria for the field in the Criteria box and/or click the Sort box list arrow for the field and select a sort order. Click the Save button on the toolbar and give a name to the query.

Understanding the Different Types of Queries

- The Types of Queries Are:
  - Select Queries
  - Parameter Queries
  - Make-Table Queries
  - Append Queries
  - Select Queries
  - Delete Queries
  - Update Queries
  - Union Queries

Creating a Multiple Table Query

- To Create a Query in Design View: From the database window, click the Queries tab, click New, select Design View from the list and click OK. Select the table or query you want to use and click OK. Repeat as necessary for additional tables or queries that you want to add to the query and click Close when you’re finished. If Access doesn’t automatically join the tables click the related field in the first table and drag it to the related field in the second table. Repeat as necessary to connect all the tables. Double-click the fields that you want to appear in the query or click and drag the fields onto the design grid. In the design grid, enter any desired search criteria for the field in the Criteria box and/or click the Sort box list arrow for the field and select a sort order. Click the Save button on the toolbar and give a name to the query.

Creating a Calculated Field

- To Create an Expression or Calculation in a Query: Display the query in Design View. Click the Field Row of a blank column in the design grid, enter the field name for the calculated field followed by a : (colon) followed by the expression you want Access to calculate, using the proper syntax (you can use the Expression Builder to help you create this expression).
Working with Expressions and the Expression Builder

- **To Create an Expression with the Expression Builder:** Display the query in **Design View**. Click the **Field Row** of a blank column in the design grid, click the **Build button** on the toolbar, double-click the field you want to use in the calculation, click the button that corresponds to the calculation you want, and then click or type any other fields or values you want to use. Click **OK** when you're finished.

Using an IIF Function

- **To Create an IIF (IF...THEN) Function:** Display the query in **Design View**. Click the **Field Row** of a blank column in the design grid, enter the field name followed by a : (colon) and type the expression using the syntax \texttt{IIF(«expr», «truepart», «falsepart»)} (you can also use the Expression Builder to help you create the IIF expression.)

Summarizing Groups of Records

- **To Calculate or Summarize a Group of Records:** Display the query in **Design View** and, if necessary, click the **Totals button** the toolbar. Move the field that you want to group data by onto the design grid and make sure **Group By** appears in that field’s Total Row. Move the field that you want to perform calculations on to the design grid. Choose the type of calculation that you want for each field from the drop-down list in the **Total row**.

Display Top or Bottom Values

- **To View Top or Bottom Values:** Display the query in **Design View**, click the first **Sort field** and select either Ascending (displays bottom values) or Descending (displays top values). Select an option from the **Top Values list** on the toolbar.

Parameter Queries

- **To Create a Parameter Query:** Display the query in **Design View**, click the **Criteria row** for the field you want to use for your parameter criteria and enter the text of the prompt surrounded by square brackets [ ].

Finding Duplicate Records

- **To Use the Find Duplicate Wizard:** From the Database window, click the **Queries icon** in the Objects bar and click the **New button**. Select the **Find Duplicates Query Wizard** and click **OK**, select the field(s) that may contain the duplicate values and click **Next**. Double-click any additional fields that you want to appear in the query results, click **Next** and then click **Finish**.

Finding Unmatched Records

- **To Use the Find Unmatched Wizard:** From the Database window, click the **Queries icon** in the Objects bar and click the **New button**. Select the **Find Duplicates Query Wizard** and click **OK**, select the table whose values you want to display and click **Next**. Select the table that contains the related records, click **Next**, specify the related fields that join the two tables then click the **button** to join the two tables and click **Next**. Double-click any additional fields that you want to appear in the query results, click **Next**, and then click **Finish**.
Crosstab Queries

To Create a Crosstab Query: From the Database window, click the Queries icon in the Objects bar and click the New button, then select the Crosstab Query Wizard and click OK. Select the table or query you want to use in the crosstab query and click OK. Select the field you want to use as the column headings, click Next, select the field you want to use as the row headings and click Next. Select the field you want to use as the row heading, click Next, type a name for the crosstab query and click Finish.

Delete Queries

To Create a Delete Query: In Query Design View create a select query, including any tables, fields, calculated fields, and criteria. Click the Query Type button list on the toolbar and select Delete Query or select Query → Delete Query from the menu. Click the View button to view the results of the delete query. If you're satisfied that the appropriate records will be deleted, click the Run button on the toolbar to delete the records.

Append Queries

To Create an Append Query: In Query Design View create a select query, including any tables, fields, calculated fields, and criteria. Click the Query Type button list on the toolbar and select Append Query or select Query → Append Query from the menu. Select the table where you want to add the records to from the drop-down list and select one of the following options: Current Database (If the table is in the currently open database) or Another Database: And type the name of the another database (including the path, if necessary). Click OK and click the View button on the toolbar to view the results of the query or the Run button on the toolbar to append the records.

Make-Table Queries

To Create a Make-Table Query: In Query Design View create a select query, including any tables, fields, calculated fields, and criteria. Click the Query Type button list on the toolbar and select Make-Table Query or select Query → Make Table Query from the menu. Type the name of the table you want to create, or click the drop-down list and select a table from the list if you want to replace the existing one. Click one of the following options: Current Database: If the table is in the currently open database or Another Database: And type the name of the another database (including the path, if necessary). Click OK. Click the View button on the toolbar to view the results of the query or the Run button on the toolbar to create the new table.

Update Queries

To Create an Update Query: In Query Design View create a select query, including any tables, fields, calculated fields, and criteria. Click the Query Type button list on the toolbar and select Append Query or select Query → Update Query from the menu. Enter an expression to update the selected field and enter any criteria, if needed, to select which records should be updated. Click the View button to view the results of the update query. If you're satisfied that the appropriate records will be updated, click the Run button on the toolbar to update the records.
Quiz

1. Which of the following criterion is NOT written using the proper syntax?
   A. “Harris”
   B. Between 1/1/2000 and 12/31/2000
   C. NO VALUE
   D. <500

2. Which of the following types of queries are action queries? (Select all that apply.)
   A. Parameter queries
   B. Append queries
   C. Update queries
   D. Crosstab queries

3. Which of the following expressions is not written in the correct syntax?
   A. [Order Total]*[Tax Rate]
   B. “Order Total”*0.1
   C. [tblCustomerTours]![Cost][tblEmployees]![Commission]
   D. 100+10

4. If you are having trouble remembering how to write expressions using the correct syntax, you can use the Expression Builder to help you create the expression (True or False?)

5. Rebate: IIF([Age]>65,”Senior”,“Adult”) This expression is an example of:
   A. Something I learned back in high school algebra and thought I would never see again.
   B. A financial expression.
   C. Something that belongs in a Microsoft Excel book.
   D. A conditional expression.

6. A query prompts a user for a date and then displays only records that contain the specified date. Which type of query is this?
   A. A parameter query.
   B. A crosstab query.
   C. An action query.
   D. An update query.

7. You must create a report if you want to calculate totals for a group of records, as queries can’t perform this task. (True or False?)

8. A query summarizes information in a grid, organized by regions and months. Which type of query is this?
   A. A parameter query.
   B. A crosstab query.
   C. An action query.
   D. An update query.
9. Your company finally agreed to buy you a nifty 3COM Palm palmtop. Now you want to extract your clients from the company's database and put them into a separate table that you can export to your Palm. Which type of query could help you accomplish this task?

A. A parameter query.
B. A crosstab query.
C. An update query.
D. A make-table query.

Homework

1. Open the Homework database.
2. Using the Customers and Insurance Claims tables, create a multi-table query that counts the total number of claims, grouped by the State field.
   \textbf{Hint:} Use the Count function on any field in the Insurance Claims table.
3. Sort the results of the query alphabetically by date.
4. Add criteria to the query that excludes records from the state of Wisconsin (WI).
5. Change the Select query to a Delete query and delete all records from the state of Wisconsin (WI).
6. Change the Delete query back into a Select query. Remove the current criteria and add parameter criteria that asks for the beginning and ending date.
7. Create a calculated field that combines the FirstName and LastName fields.
8. Save the query as “Homework Query”.

Quiz Answers

1. C.
2. B and C. Append queries and Update queries are both action queries.
3. B. Fields should be enclosed in [ ] brackets, not "" quotation marks.
4. True. The Expression Builder helps you create expressions in your queries (and in forms, reports, and macros too!)
5. D. The IIF function is a conditional expression that evaluates a field for a specified condition and returns one value if the condition is true and another value if the condition is false.
6. A. A parameter query prompts the user for information.
7. False. Queries can calculate groups of records—just click the Totals button on the toolbar and choose the type of calculation that you want for each field from the drop-down list in the Total row.
8. B. A query that summarizes information in a grid, organized by columns and rows sounds an awful lot like a crosstab query.
9. D. Since a Make-Table query creates a new table with the results of the query, this would be your best bet.