

WRITTEN BY
MICHAEL ALEXANDER
DICK KUSLEIKA

Access® 2016 BIBLE

THE COMPREHENSIVE TUTORIAL RESOURCE

DESIGN YOUR OWN
RELATIONAL DATABASES

BUILD POWERFUL
ANALYTICS AND REPORTING

PROGRAM AND PUBLISH
YOUR OWN APPLICATIONS

WILEY

Table of Contents

[Title Page](#)

[Introduction](#)

[Is This Book for You?](#)

[Conventions Used in This Book](#)

[How This Book Is Organized](#)

[How to Use This Book](#)

[What's on the Website](#)

[Part I: Access Building Blocks](#)

[Chapter 1: An Introduction to Database Development](#)

[The Database Terminology of Access](#)

[Relational Databases](#)

[Access Database Objects](#)

[A Five-Step Design Method](#)

[Chapter 2: Getting Started with Access](#)

[The Access Welcome Screen](#)

[How to Create a Blank Desktop Database](#)

[The Access 2016 Interface](#)

[Part II: Understanding Access Tables](#)

[Chapter 3: Creating Access Tables](#)

[Table Types](#)

[Creating a New Table](#)

[Creating tblCustomers](#)

[Changing a Table Design](#)

[Understanding tblCustomers Field Properties](#)

[Setting the Primary Key](#)

[Indexing Access Tables](#)

[Printing a Table Design](#)

[Saving the Completed Table](#)

[Manipulating Tables](#)

[Adding Records to a Database Table](#)

[Understanding Attachment Fields](#)

[Chapter 4: Understanding Table Relationships](#)

[Building Bulletproof Databases](#)

[Data Normalization and Denormalization](#)

[Table Relationships](#)

[Integrity Rules](#)

[Keys](#)

[Chapter 5: Working with Access Tables](#)

[Understanding Datasheets](#)

[Looking at the Datasheet Window](#)

[Opening a Datasheet](#)

[Entering New Data](#)

[Navigating Records in a Datasheet](#)

[Changing Values in a Datasheet](#)

[Using the Undo Feature](#)

[Copying and Pasting Values](#)

[Replacing Values](#)

[Adding New Records](#)

[Deleting Records](#)

[Displaying Records](#)

[Sorting and Filtering Records in a Datasheet](#)

[Aggregating Data](#)

[Printing Records](#)

[Previewing Records](#)

[Chapter 6: Importing and Exporting Data](#)

[How Access Works with External Data](#)

[Options for Importing and Exporting](#)

[Importing External Data](#)

[Exporting to External Formats](#)

[Chapter 7: Linking to External Data](#)

[Linking External Data](#)

[Working with Linked Tables](#)

[Splitting a Database](#)

[Part III: Working with Access Queries](#)

[Chapter 8: Selecting Data with Queries](#)

[Introducing Queries](#)

[Creating a Query](#)

[Working with Query Fields](#)

[Adding Criteria to Your Queries](#)

[Printing a Query's Recordset](#)

[Saving a Query](#)

[Creating Multi-Table Queries](#)

[Working with the Table Pane](#)

[Creating and Working with Query Joins](#)

[Chapter 9: Using Operators and Expressions in Access](#)

[Introducing Operators](#)

[Using Operators and Expressions in Queries](#)

[Entering Single-Value Field Criteria](#)

[Using Multiple Criteria in a Query](#)

[Entering Criteria in Multiple Fields](#)

[Chapter 10: Going Beyond Select Queries](#)

[Aggregate Queries](#)

[Action Queries](#)

[Crosstab Queries](#)

[Optimizing Query Performance](#)

[Part IV: Analyzing Data in Access](#)

[Chapter 11: Transforming Data in Access](#)

[Finding and Removing Duplicate Records](#)

[Common Transformation Tasks](#)

[Chapter 12: Working with Calculations and Dates](#)

[Using Calculations in Your Analyses](#)

[Using Dates in Your Analyses](#)

[Chapter 13: Performing Conditional Analyses](#)

[Using Parameter Queries](#)

[Using Conditional Functions](#)

[Chapter 14: The Fundamentals of Using SQL](#)

[Understanding Basic SQL](#)

[Getting Fancy with Advanced SQL Statements](#)

[Using SQL-Specific Queries](#)

[Chapter 15: Subqueries and Domain Aggregate Functions](#)

[Enhancing Your Analyses with Subqueries](#)

[Domain Aggregate Functions](#)

[Chapter 16: Running Descriptive Statistics in Access](#)

[Basic Descriptive Statistics](#)

[Advanced Descriptive Statistics](#)

[Part V: Working with Access Forms and Reports](#)

[Chapter 17: Creating Basic Access Forms](#)

[Formulating Forms](#)

[Working with Controls](#)

[Introducing Properties](#)

[Chapter 18: Working with Data on Access Forms](#)

[Using Form View](#)

[Changing Values in a Form](#)

[Printing a Form](#)

[Working with Form Properties](#)

[Adding a Form Header or Footer](#)

[Working with Section Properties](#)

[Changing the Layout](#)

[Converting a Form to a Report](#)

[Chapter 19: Working with Form Controls](#)

[Creating a Calculated Control](#)

[Working with Subforms](#)

[Form Design Tips](#)

[Tackling Advanced Forms Techniques](#)

[Using the Tab Control](#)

[Using Dialog Boxes to Collect Information](#)

[Designing a Form from Scratch](#)

[Chapter 20: Presenting Data with Access Reports](#)

[Introducing Reports](#)

[Creating a Report, from Beginning to End](#)

[Banded Report Design Concepts](#)

[Creating a Report from Scratch](#)

[Improving the Report's Appearance](#)

[Chapter 21: Advanced Access Report Techniques](#)

[Grouping and Sorting Data](#)

[Formatting Data](#)

[Adding Data](#)

[Adding Even More Flexibility](#)

[Part VI: Access Programming Fundamentals](#)

[Chapter 22: Using Access Macros](#)

[Understanding Macro Security](#)

[Submacros](#)

[Conditions](#)

[Temporary Variables](#)

[Error Handling and Macro Debugging](#)

[Embedded Macros](#)

[Macros versus VBA Statements](#)

[Chapter 23: Using Access Data Macros](#)

[Introducing Data Macros](#)

[Understanding Table Events](#)

[Using the macro designer for Data Macros](#)

[Understanding the Action Catalog](#)

[Creating Your First Data Macro](#)

[Managing Macro Objects](#)

[Recognizing the Limitations of Data Macros](#)

[Chapter 24: Getting Started with Access VBA](#)

[Introducing Visual Basic for Applications](#)

[Understanding VBA Terminology](#)

[Starting with VBA Code Basics](#)

[Creating VBA Programs](#)

[Understanding VBA Branching Constructs](#)

[Working with Objects and Collections](#)

[Exploring the Visual Basic Editor](#)

Chapter 25: Mastering VBA Data Types and Procedures

Using Variables

Working with Data Types

Understanding Subs and Functions

Creating Functions

Simplifying Code with Named Arguments

Chapter 26: Understanding the Access Event Model

Programming Events

Identifying Common Events

Paying Attention to Event Sequence

Chapter 27: Debugging Your Access Applications

Organizing VBA Code

Testing Your Applications

Traditional Debugging Techniques

Using the Access Debugging Tools

Trapping Errors in Your Code

Part VII: Advanced Access Programming Techniques

Chapter 28: Accessing Data with VBA

Working with Data

Understanding ADO Objects

Understanding DAO Objects

Writing VBA Code to Update a Table

Chapter 29: Advanced Data Access with VBA

Adding an Unbound Combo Box to a Form to Find Data

Filtering a Form

Chapter 30: Customizing the Ribbon

The Ribbon Hierarchy

[Editing the Default Ribbon](#)

[Working with the Quick Access Toolbar](#)

[Developing Custom Ribbons](#)

[Creating a Custom Ribbon](#)

[The Basic Ribbon XML](#)

[Adding Ribbon Controls](#)

[Attaching Ribbons to Forms and Reports](#)

[Removing the Ribbon Completely](#)

[Chapter 31: Preparing Your Access Application for Distribution](#)

[Defining the Current Database Options](#)

[Developing the Application](#)

[Polishing Your Application](#)

[Bulletproofing an Application](#)

[Securing the Environment](#)

[Part VIII: Access and Windows SharePoint Services](#)

[Chapter 32: Integrating Access with SharePoint](#)

[Introducing SharePoint](#)

[Understanding SharePoint Sites](#)

[Sharing Data between Access and SharePoint](#)

[Using SharePoint Templates](#)

[Chapter 33: Deploying Access Applications to SharePoint](#)

[Understanding Web Publishing with Access](#)

[Examining the Limitations of Access Web Applications](#)

[Publishing a Custom Access Application to SharePoint](#)

[End User License Agreement](#)

List of Illustrations

Chapter 2: Getting Started with Access

[Figure 2.1 The Access welcome screen provides a number of ways to start working with Access.](#)

[Figure 2.2 Enter the name of the new database in the File Name box.](#)

[Figure 2.3 Your new database is created.](#)

[Figure 2.4 The Access interface starts with the Ribbon at the top and the Navigation pane at the left.](#)

[Figure 2.5 Choosing an alternate display for the Navigation pane.](#)

[Figure 2.6 Contextual tabs contain commands that are specific to whichever object is active.](#)

[Figure 2.7 The Quick Access toolbar is located above the Ribbon.](#)

[Figure 2.8 Commands you can add to the Quick Access toolbar.](#)

[Figure 2.9 Adding more commands to the Quick Access toolbar.](#)

Chapter 3: Creating Access Tables

[Figure 3.1 The Create tab contains tools necessary for adding new objects to your Access database.](#)

[Figure 3.2 The new table in Datasheet view.](#)

[Figure 3.3 Field design tools are located on the Fields tab of the Ribbon.](#)

[Figure 3.4 A new table added in Design view.](#)

[Figure 3.5 The Design tab of the Ribbon.](#)

[Figure 3.6 The Property Sheet.](#)

[Figure 3.7 The Data Type drop-down list.](#)

[Figure 3.8 The Input Mask Wizard for creating input masks for Text field types.](#)

[Figure 3.9 A data-validation warning box appears when the user enters a value in the field that does not match the rule specified in the design of the table.](#)

[Figure 3.10 The Table Design View area on the Options screen contains a box for setting the AutoIndex on Import/Create options.](#)

[Figure 3.11 The property area for the Currency field named CreditLimit.](#)

[Figure 3.12 The Lookup tab for a Yes/No field.](#)

[Figure 3.13 Setting up a combo box as the display control for Credit Status.](#)

[Figure 3.14 Using a combo box as a lookup control to restrict user input on a field.](#)

[Figure 3.15 Creating a composite primary key.](#)

[Figure 3.16 frmIndexTest provides a quick and easy way to verify the importance of indexes.](#)

[Figure 3.17 Multiple-field \(composite\) indexes can enhance performance.](#)

[Figure 3.18 It's easy to set the properties of an index.](#)

[Figure 3.19 The Documenter dialog box.](#)

[Figure 3.20 Printing options in the Print Table Definition dialog box.](#)

[Figure 3.21 Pasting a table opens the Paste Table As dialog box.](#)

[Figure 3.22 Using Datasheet view to add records to a table.](#)

[Figure 3.23 Managing attachments in an Attachment field.](#)

Chapter 4: Understanding Table Relationships

[Figure 4.1 An Access table containing contact *and* orders data.](#)

[Figure 4.2 An unnormalized tblBookOrders table.](#)

[Figure 4.3 Only a slight improvement over the previous design.](#)

[Figure 4.4 First normal form at last!](#)

[Figure 4.5 Moving customer data to its own table.](#)

[Figure 4.6 The first step in making our table 2NF.](#)

[Figure 4.7 We have achieved second normal form.](#)

[Figure 4.8 An Excel worksheet used as a flat-file database.](#)

[Figure 4.9 The relationship between the Employees and Payroll tables is an example of a typical one-to-many relationship.](#)

[Figure 4.10 Most of the Collectible Mini Cars table relationships.](#)

[Figure 4.11 A one-to-one relationship.](#)

[Figure 4.12 The Northwind Traders database contains many examples of one-to-many relationships.](#)

[Figure 4.13 A database of students and the clubs they belong to is an example of a many-to-many relationship.](#)

[Figure 4.14 A typical database relationship.](#)

[Figure 4.15 Double-click a table name to add it to the Relationships window.](#)

[Figure 4.16 You enforce referential integrity in the Edit Relationships dialog box.](#)

[Figure 4.17 A one-to-many relationship between tblBookOrders5 and tblBookOrderDetails.](#)

[Figure 4.18 The Join Properties dialog box, used to set up the join properties between tblCustomers and tblSales. Notice that it specifies all records from the Contacts table.](#)

[Figure 4.19 A dialog box warning that referential integrity can't be enforced because of integrity violations.](#)

[Figure 4.20 A simple validation rule goes a long way toward preserving the database's integrity.](#)

Chapter 5: Working with Access Tables

[Figure 5.1 A typical Datasheet view. Each row represents a single record in the table; each column represents a single field \(like Description or ModelYear\) in the table.](#)

[Figure 5.2 The Navigation buttons of a datasheet.](#)

[Figure 5.3 The Datasheet Ribbon's Home tab.](#)

[Figure 5.4 An empty datasheet. Notice that the first record is blank and has an asterisk in the record selector.](#)

[Figure 5.5 Entering a new record into the Datasheet view of tblProducts.](#)

[Figure 5.6 The error message Access displays when attempting to save a record with a duplicate primary key value entered into the new record. Use an AutoNumber field as your primary key to avoid this error.](#)

Figure 5.7 The warning Access displays when entering data that doesn't match the field's data type. Access gives you a few choices to correct the problem.

Figure 5.8 The Zoom window. Notice that you can see all of the field's data event at a larger font size.

Figure 5.9 The Find and Replace dialog box. The fastest way to activate it is to simply press Ctrl+F.

Figure 5.10 The Find and Replace dialog box with the Replace tab showing. In this case, you want to replace Mini Vans with Minivans.

Figure 5.11 The Delete Record dialog box warns you that you're about to delete a specific number of records. The default response is Yes (okay to delete), so be careful when deleting records.

Figure 5.12 Select and drag a column to change the field order.

Figure 5.13 The Column Width dialog box.

Figure 5.14 Changing a row's height. Position the mouse as shown and drag to the desired height.

Figure 5.15 Changing the datasheet's font directly from the Ribbon. Choose font type style, size, and other font attributes for the entire datasheet.

Figure 5.16 Use the Datasheet Formatting dialog box to customize the look of the datasheet.

Figure 5.17 Different line styles and row colors for the datasheet.

Figure 5.18 Hide and unhide columns using the Unhide Columns dialog box.

Figure 5.19 The Sort & Filter group lets you change the record order and reduce the number of visible

ROWS.

Figure 5.20 Using Filter by Selection. In this case, you see all trucks that are not Mack models.

Figure 5.21 Filtering the Make field. Use the column filter menu to select criteria for a field.

Figure 5.22 Using Filter by Form lets you set multiple conditions for filtering at one time. Notice the Or tab at the bottom of the window.

Figure 5.23 The datasheet Totals row.

Figure 5.24 The Microsoft Office Print menu.

Chapter 6: Importing and Exporting Data

Figure 6.1 The External Data tab of the Ribbon hints at the variety of external data sources available to Access.

Figure 6.2 Many types of Access database objects can be imported from one Access database into another.

Figure 6.3 The Saved Import Steps feature lets you save frequently executed import processes for future use.

Figure 6.4 The Saved Imports feature lets you rerun previous saved import processes.

Figure 6.5 Access can import data from an Excel spreadsheet, but there are some restrictions.

Figure 6.6 Excel worksheet columns should contain consistent data.

Figure 6.7 The Import Spreadsheet Wizard.

Figure 6.8 Does the first row contain column headings?

[Figure 6.9 You can override any of the default settings Access has chosen.](#)

[Figure 6.10 Specify a primary key for the new table.](#)

[Figure 6.11 The first screen of the Import Text Wizard.](#)

[Figure 6.12 The second Import Text Wizard screen.](#)

[Figure 6.13 A typical fixed-width text file.](#)

[Figure 6.14 The Import Text Wizard screen for fixed-width text files.](#)

[Figure 6.15 The Import Specification dialog box for importing a fixed-width text file.](#)

[Figure 6.16 The Export XML dialog box.](#)

[Figure 6.17 Advanced XML export options.](#)

[Figure 6.18 Exporting XML schema information.](#)

[Figure 6.19 XML presentation options.](#)

[Figure 6.20 An exported XML file in plain text.](#)

[Figure 6.21 Access understands XML data.](#)

[Figure 6.22 The Export - HTML Document dialog box.](#)

[Figure 6.23 The same wizard imports objects and links to external Access tables.](#)

[Figure 6.24 Importing Access objects.](#)

[Figure 6.25 The initial Outlook import options.](#)

[Figure 6.26 Importing Outlook objects into Access.](#)

[Figure 6.27 Naming the table in an ODBC destination.](#)

[Figure 6.28 Selecting the ODBC driver.](#)

Figure 6.29 SQLite3 requires additional information, such as the database name, to make the connection.

Figure 6.30 The Employees table has successfully exported to a SQLite3 database.

Figure 6.31 The Microsoft Word Mail Merge Wizard allows you to export data to existing or new documents.

Figure 6.32 The Departments table from Chapter06_1 is the specified data source.

Figure 6.33 A completed Word Merge.

Figure 6.34 Select PDF or XPS as a file format.

Figure 6.35 A table published in the PDF file format.

Chapter 7: Linking to External Data

Figure 7.1 Linked tables in an Access database. Notice that each linked table has an icon indicating its status as a linked table.

Figure 7.2 A query using externally linked tables.

Figure 7.3 The Datasheet view of externally linked data.

Figure 7.4 Use the Get External Data dialog box to select the type of operation you want to perform on the external data sources.

Figure 7.5 Use the Link Tables dialog box to select the Access table(s) for linking.

Figure 7.6 The Navigation pane with tblSalesPayments added. Hovering over the linked table icon reveals its source.

Figure 7.7 The first screen of the Get External Data – Excel Spreadsheet dialog box.

[Figure 7.8 The main Link Spreadsheet Wizard screen.](#)

[Figure 7.9 The Link HTML Wizard screen showing the data in the HTML file.](#)

[Figure 7.10 The Link Specification screen that is used to name the column headings \(field names\) for the linked table.](#)

[Figure 7.11 A typical fixed-width text file.](#)

[Figure 7.12 CSV data is more compact than fixed-width text, but it's more difficult to read.](#)

[Figure 7.13 Opening linked tables in Design view comes with a warning.](#)

[Figure 7.14 The Linked Table Manager enables you to relocate external tables that have been moved.](#)

[Figure 7.15 A new row in the text file will sync with Access automatically.](#)

[Figure 7.16 A database kept on a file server can generate a large amount of traffic on the network.](#)

[Figure 7.17 A database kept on a file server can generate a large amount of traffic on the network.](#)

[Figure 7.18 The Database Splitter is a simple wizard.](#)

[Figure 7.19 Specify the permanent location of the back-end database in the Create Back-end Database dialog box.](#)

[Figure 7.20 The Database Splitter creates links for all tables in the database.](#)

Chapter 8: Selecting Data with Queries

[Figure 8.1 The Show Table dialog box and the query design window.](#)

Figure 8.2 The query design window with tblProducts added.

Figure 8.3 The Query Tools Design Ribbon.

Figure 8.4 To add fields from your table to the QBE grid, simply double-click or drag the field.

Figure 8.5 Selecting multiple fields to add to the QBE grid.

Figure 8.6 Adding the asterisk to the QBE grid selects all fields in the table.

Figure 8.7 Click the Run button to display the results of your query.

Figure 8.8 Right-click on the queries tab header and select Design View to return to the QBE grid.

Figure 8.9 Selecting a column in the QBE grid. The pointer changes to a downward-pointing arrow when you move over the selection row.

Figure 8.10 Moving the Category field to between ProductID and Description. Notice the QBE field icon below the arrow near the Description column.

Figure 8.11 Resizing columns in the QBE grid.

Figure 8.12 The Show check box is unchecked for the State field so that field will not show in the results.

Figure 8.13 An ascending sort has been specified for the LastName and FirstName fields.

Figure 8.14 The order of the fields in the QBE grid is critical when sorting on multiple fields.

Figure 8.15 Specifying Cars as the query's criteria.

Figure 8.16 Specifying text and date criteria in the same query.

[Figure 8.17 The query design window with four tables added. Notice that the join lines are already present.](#)

[Figure 8.18 The QBE grid with table names displayed. Notice that it shows all four table names.](#)

[Figure 8.19 An inner join operation will select only the records that have matching values in both tables. The arrows point to the records that will be included in the results.](#)

[Figure 8.20 A left outer join operation will select all records from the first table and only those records from the second table that have matching values in both tables. The arrows point to the records that will be included in the results.](#)

[Figure 8.21 A right outer join operation will select all records from the second table and only those records from the first table that have matching values in both tables. The arrows point to the records that will be included in the results.](#)

[Figure 8.22 An ad hoc join between tblProducts and tblCategories.](#)

[Figure 8.23 Selecting an outer join for the query.](#)

[Figure 8.24 A right outer join corrects the “missing products” problem in Figure 8.22.](#)

Chapter 9: Using Operators and Expressions in Access

[Figure 9.1 The QBE pane shows a simple criterion asking for all models where the category is Cars.](#)

[Figure 9.2 Using And and Or criteria in a query.](#)

[Figure 9.3 Creating complex criteria without using the And/Or operators.](#)

[Figure 9.4 The SQL view for the query in Figure 9.3. Notice that it contains a single OR and two AND operators \(in the WHERE clause\).](#)

[Figure 9.5 The query design window showing tblCustomers open.](#)

[Figure 9.6 Using the Like operator in a select query.](#)

[Figure 9.7 The results of using the Like operator with a select query in a Memo field. The query looks for the word *red* in the Features field.](#)

[Figure 9.8 Using the Not operator in criteria.](#)

[Figure 9.9 Criteria set for products with low inventory.](#)

[Figure 9.10 Using the Or operator. Notice the two Or operators under the State field—AZ Or CA Or NY.](#)

[Figure 9.11 Using the Or cell of the QBE pane. You can place criteria vertically in the QBE grid.](#)

[Figure 9.12 Using the In operator to find all records for buyer state being either AZ, CA, or NY.](#)

[Figure 9.13 Using an And operator to specify complex query criteria.](#)

[Figure 9.14 Using the Between...And operator. The results are the same as the query in Figure 9.13.](#)

[Figure 9.15 Use Is Null to select rows containing fields that contain no data.](#)

[Figure 9.16 The QBE pane with And/Or criteria between fields using the Criteria and Or rows.](#)

[Figure 9.17 Using the Or operator between fields.](#)

[Figure 9.18 A simple rearrangement in the QBE grid results in a very different query.](#)

[Figure 9.19 Using Ands and Ors in a select query.](#)

[Figure 9.20 Using multiple Ands and Ors across fields. This is a rather complex select query.](#)

Chapter 10: Going Beyond Select Queries

[Figure 10.1 Running this query will return all the records in your dataset, not the summary you need.](#)

[Figure 10.2 Activating Totals in your design grid adds a Total row to your query grid that defaults to Group By.](#)

[Figure 10.3 Change the aggregate function under the LineTotal field to Sum.](#)

[Figure 10.4 After running your query, you have a summary showing your total revenue by period.](#)

[Figure 10.5 This query returns only one line showing total records for the 201201 period.](#)

[Figure 10.6 This query results in a few more records, but if you add up the counts in each group, they'll total 503.](#)

[Figure 10.7 The Expression aggregate function allows you to perform the designated custom calculation on each Period group separately.](#)

[Figure 10.8 Running this query will cause an error message because you have no aggregation defined for Period.](#)

[Figure 10.9 Adding a Where clause remedies the error and allows you to run the query.](#)

[Figure 10.10 Create this query in Design view.](#)

[Figure 10.11 Enter the name of your new table.](#)

[Figure 10.12 Click Yes to run your query.](#)

Figure 10.13 This query will select all records with a Period earlier than 201206.

Figure 10.14 Click Yes to continue with your delete action.

Figure 10.15 The warning message tells you that you'll lose records during the append process.

Figure 10.16 This query selects all records contained in Periods 201201 thru 201205.

Figure 10.17 Enter the name of the table to which you want to append your query results.

Figure 10.18 In the Append To row, select the name of the field in your destination table where you want to append the information resulting from your query.

Figure 10.19 Click Yes to continue with your append action.

Figure 10.20 This query will select all customers that are in the 33605 zip code.

Figure 10.21 In this query, you are updating the zip code for all customers that have a code of 33605 to 33606.

Figure 10.22 Click Yes to continue with your update action.

Figure 10.23 The basic structure of a crosstab query.

Figure 10.24 Select Crosstab Query Wizard from the New Query dialog box.

Figure 10.25 Select the data source for your crosstab query.

Figure 10.26 Select the ProductID field and then click the Next button.

[Figure 10.27 Select the OrderDate field then click the Next button.](#)

[Figure 10.28 Select Quarter and then click Next.](#)

[Figure 10.29 Select the LineTotal and Sum, and then click the Next button.](#)

[Figure 10.30 Click Finish to see your query results.](#)

[Figure 10.31 A powerful analysis in just a few clicks.](#)

[Figure 10.32 Create an aggregate query as shown here.](#)

[Figure 10.33 Set each field's role in the Crosstab row.](#)

[Figure 10.34 You can define a criterion to filter your crosstab queries.](#)

[Figure 10.35 This crosstab query will display all regions as columns in alphabetical order.](#)

[Figure 10.36 The Column Headings attribute is set to have the columns read in this order: West, Canada, Midwest, North, Northeast, South, Southeast, and Southwest.](#)

Chapter 11: Transforming Data in Access

[Figure 11.1 Are there duplicate records in this table? It depends on how you define one.](#)

[Figure 11.2 If you get this error message when trying to set a primary key, you have duplicate records in your dataset.](#)

[Figure 11.3 Select the Find Duplicates Query Wizard and then click OK.](#)

[Figure 11.4 Select the dataset in which you want to find duplicates, and then click Next.](#)

Figure 11.5 Select the field(s) that make up a unique record in your dataset.

Figure 11.6 Select the field(s) you want to see in your query.

Figure 11.7 Name your query and click Finish.

Figure 11.8 Your Find Duplicates query.

Figure 11.9 Activate the Paste Table As dialog box to copy your table's structure into a new table called LeadList_NoDups.

Figure 11.10 Set as a primary key the field(s) that best defines a unique record.

Figure 11.11 Now you can append all records excluding the duplicates.

Figure 11.12 This query will update the null values in the DollarPotential field to a value of 0.

Figure 11.13 This query will update blank values in the Segment field to a value of "Other."

Figure 11.14 This query concatenates the row values of the Type field and the Code field.

Figure 11.15 The MyTest field now contains the concatenated values of the Type field and the Code field.

Figure 11.16 This query concatenates the row values of the Type field and the Code field and separates them with a colon.

Figure 11.17 The MyTest field now contains the concatenated values of the Type field and the Code field, separated by a colon.

Figure 11.18 The Address field is in all lowercase letters.

[Figure 11.19 This query will convert addresses to proper case.](#)

[Figure 11.20 The leading spaces are preventing an accurate aggregation.](#)

[Figure 11.21 Simply pass the field name through the Trim function in an update query to remove the leading and trailing spaces.](#)

[Figure 11.22 This query finds all instances of “blvd” and replaces them with “Boulevard.”](#)

[Figure 11.23 The phone number has been transformed into a standard format by adding the appropriate characters to key positions with the string.](#)

[Figure 11.24 This query will update the MyTest field with a properly formatted phone number.](#)

[Figure 11.25 You need to parse the values in the ContactName field into three separate fields.](#)

[Figure 11.26 This query will update the Contact_LastName and Contact_FirstName fields.](#)

[Figure 11.27 Check your progress so far.](#)

[Figure 11.28 This query parses out the first name and the middle initial from the Contact_FirstName field.](#)

[Figure 11.29 With two queries, you've successfully parsed the ContactName field into three separate fields.](#)

Chapter 12: Working with Calculations and Dates

[Figure 12.1 In this query, you're using a constant to calculate a 10 percent price increase.](#)

[Figure 12.2 In this query, you're using two fields in a Dollar Variance calculation.](#)

Figure 12.3 In this query, you're using the aggregation results for each market as expressions in your calculation.

Figure 12.4 This query uses the results of one calculation as an expression in another.

Figure 12.5 The results of this calculation will be difficult to read because they'll all be fractional numbers that have many digits trailing a decimal point. Forcing the results into round numbers will make for easier reading.

Figure 12.6 You can use your calculation as the argument in the Int function, allowing you to remove the fractional part from the resulting data.

Figure 12.7 Activate the Expression Builder by right-clicking inside the Field row of the query_grid and selecting Build.

Figure 12.8 The Expression Builder will display any database object you can use in your expression.

Figure 12.9 The Expression Builder displays all the functions available to you.

Figure 12.10 Access tells you which arguments are needed to make the function work.

Figure 12.11 Help files are available to explain each function in detail.

Figure 12.12 The function here will round the results of the calculation, $\frac{([Dim_Transactions]![Line_Total])}{13}$.

Figure 12.13 Your newly created expression will give you the average revenue per period for all transactions.

Figure 12.14 To demonstrate how null values can cause calculation errors, build this query in Design

view.

Figure 12.15 When any variable in your calculation is null, the resulting answer is a null value.

Figure 12.16 Pass the Forecast field through the Nz function to convert null values to zero.

Figure 12.17 The first record now shows a variance value.

Figure 12.18 You're adding 30 to each ship date, effectively creating a date that is equal to the ship date plus 30 days.

Figure 12.19 In this query, you're calculating the number of days between two dates.

Figure 12.20 This query returns the number of days between today's date and each order date.

Figure 12.21 No matter what day it is today, this query will return all orders older than 90 days.

Figure 12.22 You're calculating the difference between today's date and each manager's hire date.

Figure 12.23 This dataset shows the number of days, not the number of years.

Figure 12.24 Divide your original calculation by 365.25 to convert the answer to years.

Figure 12.25 Your query is now returning years, but you have to strip away the fractional portion of your answer.

Figure 12.26 Running this query will return the number of years each employee has been with the company.

Figure 12.27 The Year, Month, Day, and Weekday functions enable you to parse out a part of a date.

Figure 12.28 This query will give you the original order date, the date you should send a warning letter, and the date on which the order will be 30 days overdue.

Figure 12.29 You can group dates into quarters by using the Format function.

Figure 12.30 You can also use the Format function in a crosstab query.

Figure 12.31 You've successfully grouped your dates into quarters.

Chapter 13: Performing Conditional Analyses

Figure 13.1 This query has a hard-coded criterion for system period.

Figure 13.2 To create a parameter query, replace the hard-coded criteria with text enclosed in square brackets ([]).

Figure 13.3 The Parameter Value dialog box lets you specify the criteria each time you run your query.

Figure 13.4 You can employ more than one parameter in a query.

Figure 13.5 This query combines standard operators with parameters in order to limit the results.

Figure 13.6 If the parameter in this query is ignored, the query will return all records thanks to the wildcard (*).

Figure 13.7 You can use parameters in calculations, enabling you to change the calculations variables each time you run the query.

Figure 13.8 You can use parameters as arguments in functions instead of hard-coded values.

Figure 13.9 This query will perform a calculation on the Actual and the Forecast fields to calculate a percent to forecast.

Figure 13.10 The errors shown in the results are due to the fact that some revenues are being divided by zeros.

Figure 13.11 This IIf function enables you to test for forecasts with a value of zero and bypass them when performing your calculation.

Figure 13.12 The IIf function helped you avoid the division by zero errors.

Figure 13.13 This query will update the MyTest field to tag all customers that have a revenue potential at or above \$10,000 with the word "LARGE."

Figure 13.14 This query will update the MyTest field to tag all customers that have a revenue potential less than \$10,000 with the word "SMALL."

Figure 13.15 You can accomplish the same task in one query using the IIf function.

Figure 13.16 This query demonstrates how to create a crosstab analysis without using a crosstab query.

Figure 13.17 The resulting dataset gives you a clean crosstab-style view of your data.

Figure 13.18 Creating crosstab-style reports using the IIf function allows you to calculate more than one value.

Figure 13.19 This analysis would be impossible to create in a standard crosstab query, where multiple calculations are not allowed.

Figure 13.20 Using the Switch function is sometimes more practical than using nested IIf functions. This