Contents

Introduction vi

Exam MO-201 Excel 2019 Expert x

Prerequisites xi

1 Manage workbook options and settings 1

Objective 1.1: Manage workbooks 2

Copy macros between workbooks 2
Enable macros in a workbook 4
Reference data in another workbook 6
Manage workbook versions 9

Objective 1.2: Prepare workbooks for collaboration 15

Restrict editing 15
Protect worksheets and cell ranges 17
Protect workbook structure 19
Configure formula calculation options 20
Manage comments 25

Objective 1.3: Use and configure language options 31

Configure authoring and display languages 31
Use language-specific features 33

2 Manage and format data 37

Objective 2.1: Fill cells based on existing data 38

Fill cells by using Flash Fill 38
Fill cells by using advanced Fill Series options 41
Objective 2.2: Format and validate data ........................................ 45
  Create custom number formats 45
  Configure data validation 50
  Group and ungroup data 53
  Calculate data by inserting subtotals and totals 56
  Remove duplicate records 58

Objective 2.3: Apply advanced conditional formatting and filtering .......................................................... 62
  Create custom conditional formatting rules 62
  Create conditional formatting rules that use formulas 66
  Manage conditional formatting rules 67

3 Create advanced formulas and macros 71

Objective 3.1: Perform logical operations in formulas ...................... 72
  Insert functions into a formula 72
  Perform logical operations by using the IF, AND, OR, and NOT functions 72
  Perform logical operations by using nested functions 77
  Perform multiple logical tests with the IFS function 78
  Perform logical operations with criteria 79
  Perform logical operations with multiple ranges and criteria 82
  Apply multiple exact-match criteria to a value 87

Objective 3.2: Look up data by using functions .............................. 91

Objective 3.3: Apply advanced date and time functions ................... 98
  Reference the date and time by using the NOW and TODAY functions 98
  Serialize numbers by using date and time functions 100

Objective 3.4: Perform data analysis ............................................. 106
  Consolidate data 106
  Perform what-if analysis by using Goal Seek and Scenario Manager 109
Introduction

The Microsoft Office Specialist (MOS) certification program has been designed to validate your knowledge of and ability to use programs in the Microsoft Office 365 and Office 2019 suite of programs. This book has been designed to guide you in studying the types of tasks you are likely to be required to demonstrate in Exam MO-201, Microsoft Excel Expert.

Exam Strategy  For information about the tasks you are likely to be required to demonstrate in the core Excel exam, Exam MO-200, Microsoft Excel 2019, see MOS Study Guide for Microsoft Excel Exam MO-200 by Joan Lambert (Microsoft Press, 2020).

Who this book is for


MOS exams for individual programs are practical rather than theoretical. You must demonstrate that you can complete certain tasks or projects rather than simply answer questions about program features. The successful MOS certification candidate will have at least six months of experience using all aspects of the program on a regular basis; for example, protecting a worksheet, applying conditional formatting rules, filling a series, and building a PivotTable.

As a certification candidate, you probably have a lot of experience with the program you want to become certified in. Many of the procedures described in this book will be familiar to you; others might not be. Read through each study section and ensure that you are familiar with the procedures, concepts, and tools discussed. In some cases, images depict the tools you will use to perform procedures related to the skill set. Study the images and ensure that you are familiar with the options available for each tool.
How this book is organized

The exam coverage is divided into chapters representing broad skill sets that correlate to the functional groups covered by the exam. Each chapter is divided into sections addressing groups of related skills that correlate to the exam objectives. Each section includes review information, generic procedures, and practice tasks you can complete on your own while studying. You can use the provided practice files to work through the practice tasks, and the result files to check your work. You can practice the generic procedures in this book by using the practice files supplied or by using your own files.

Throughout this book, you will find Exam Strategy tips that present information about the scope of study that is necessary to ensure that you achieve mastery of a skill set and are successful in your certification effort.

IMPORTANT The Excel 2019 program is not available from this website. You should purchase and install that program before using this book.

You will save the completed versions of practice files that you modify while working through the practice tasks in this book. If you later want to repeat the practice tasks, you can download the original practice files again.

The following table lists the practice files provided for this book.
<table>
<thead>
<tr>
<th>Folder and objective group</th>
<th>Practice files</th>
<th>Result files</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSExcelExpert2019\Objective1 Manage workbook options and settings</td>
<td>Excel_1-1a.xlsx Excel_1-1b.xlsm Excel_1-2a.xlsx Excel_1-2b.xlsx Excel_1-2c.xlsx Excel_1-2d.xlsx Excel_1-3a.xlsx</td>
<td>Excel_1-1a_results.xlsx Excel_1-2a_results.xlsx Excel_1-2b_results.xlsx Excel_1-2c_results.xlsx</td>
</tr>
<tr>
<td>MOSExcelExpert2019\Objective2 Manage and format data</td>
<td>Excel_2-1.xlsx Excel_2-2.xlsx Excel_2-3a.xlsx Excel_2-3b.xlsx Excel_2-3c.xlsx Excel_2-3d.xlsx Excel_2-3e.xlsx Excel_2-3f.xlsx Excel_2-3g.xlsx Excel_2-3h.xlsx Excel_2-3i.xlsx</td>
<td>Excel_2-2_results.xlsx Excel_2-3a_results.xlsx Excel_2-3b_results.xlsx Excel_2-3c_results.xlsx</td>
</tr>
<tr>
<td>MOSExcelExpert2019\Objective3 Create advanced formulas and macros</td>
<td>Excel_3-1.xlsx Excel_3-2.xlsx Excel_3-3.xlsx Excel_3-4.xlsx Excel_3-4a.xlsx Excel_3-4b.xlsx Excel_3-4c.xlsx Excel_3-4d.xlsx Excel_3-4e.xlsx Excel_3-4f.xlsx Excel_3-4g.xlsx Excel_3-4h.xlsx Excel_3-4i.xlsx Excel_3-5.xlsx Excel_3-6.xlsm</td>
<td>Excel_3-1_results.xlsx Excel_3-2_results.xlsx Excel_3-3_results.xlsx Excel_3-4_results.xlsx Excel_3-5_results.xlsx Excel_3-6_results.xlsm</td>
</tr>
<tr>
<td>MOSExcelExpert2019\Objective4 Manage advanced charts and tables</td>
<td>Excel_4-1.xlsx Excel_4-2.xlsx Excel_4-3.xlsx</td>
<td>Excel_4-1_results.xlsx Excel_4-2_results.xlsx Excel_4-3_results.xlsx</td>
</tr>
</tbody>
</table>
Ebook edition

If you’re reading the ebook edition of this book, you can do the following:

- Search the full text
- Print
- Copy and paste

Errata, updates, and book support

If you discover an error that is not already listed, please submit it to us at the same page.

Stay in touch

Let’s keep the conversation going! We’re on Twitter at

twitter.com/MicrosoftPress
Microsoft Office Specialist

Exam MO-201

Excel 2019 Expert

This book covers the skills you need to have for certification as a Microsoft Office Specialist Expert in Excel 2019. Specifically, you need to be able to complete tasks that demonstrate the following skill sets:

1. Manage workbook options and settings
2. Manage and format data
3. Create advanced formulas and macros
4. Create advanced charts and tables

With these skills, you can manage, format, populate, and enhance the types of workbooks most commonly used in a business environment.
Prerequisites

We assume that you have been working with Excel 2019 for at least six months and that you know how to carry out fundamental tasks that are not specifically mentioned in the objectives for this Microsoft Office Specialist exam. This level of proficiency includes familiarity with features and tasks such as the following:

- Creating workbooks
- Adding worksheets to existing workbooks
- Copying and moving worksheets
- Inserting and deleting cells, columns, and rows
- Customizing the Quick Access Toolbar and the ribbon
- Freezing panes and splitting the window
- Setting a print area and adding headers and footers
- Changing fonts and cell styles
- Wrapping text within cells
- Creating and editing tables
- Using relative, mixed, and absolute cell references
- Using functions
- Creating and editing charts, and adding data series
- Inserting text boxes, SmartArt, and other images
- Applying styles and effects to objects
- Positioning objects

For information about the prerequisite tasks, see *MOS Study Guide for Microsoft Excel Exam MO-200* by Joan Lambert (Microsoft Press, 2020).
Objective group 2

Manage and format data

The skills tested in this section of the Microsoft Office Specialist Expert exam for Microsoft Excel 2019 relate to filling cells with data, creating custom number formats, performing data validation, grouping data, inserting subtotals, removing duplicate records, and applying and managing conditional formatting rules. Specifically, the following objectives are associated with this set of skills:

2.1 Fill cells based on existing data
2.2 Format and validate data
2.3 Apply advanced conditional formatting and filtering

Many worksheets are drab, lifeless conglomerations of numbers, formulas, and text. If you’ll be sharing your worksheets with other people, your numbers will have much more impact and will be more easily understood if they’re pleasingly formatted and presented in a way that aids comprehension.

This chapter guides you in studying methods for using Flash Fill and advanced Fill Series options; creating custom number formats; configuring data validation; grouping and ungrouping data; inserting totals and subtotals; removing duplicate records from a table; and creating and managing conditional formatting rules.

To complete the practice tasks in this chapter, you need the practice files contained in the MOSExcelExpert2019\Objective2 practice file folder. For more information, see “Download the practice files” in this book’s introduction.
Objective group 2  Manage and format data

Objective 2.1: Fill cells based on existing data

Fill cells by using Flash Fill

If you have inherited workbooks from someone else or if you have imported data from external data sources, you have probably come across data that was either structured or formatted (or both) in such a way that it was either difficult to read or difficult to work with. It could be mainframe data that arrives as all-uppercase letters, dates that appear in non-date formats, phone numbers that don’t have dashes or parentheses, or fields that combine multiple pieces of data (such as first names and last names).

One way to tackle such data is to reenter it by hand in the structure or format you prefer or require. That works for a few records, but it gets tedious and time-consuming for dozens of records, and it becomes unworkable for hundreds or thousands of records.

Fortunately, creating your preferred data from existing data can often be done with only minimal work thanks to an Excel feature called Flash Fill. Given a column of original data, if you use the first cell in the next column to enter the corrected data (which could be data extracted from the original cell or the same data formatted in a different way), select the second cell, then run the Flash Fill command, Excel “recognizes” what you’re doing and automatically fills in the rest of the column with the corrected data.

Let’s look at a few examples:

- Column A contains a list of all-uppercase company names. In column B, I use cell B2 to type the title-case version of the text in cell A2 and then select cell B3. After I run the Flash Fill command, Excel recognizes the pattern and fills in the rest of the column with title-case versions of all the other cells in column A.
Objective 2.1: Fill cells based on existing data

You can use Flash Fill to convert all-uppercase text (far left) to title case (far right).

- Column A contains a list of full names, where each cell contains the first name, middle initial, and last name. In cells B2, C2, and D2, I typed the first name, middle initial, and last name, respectively, from cell A2. When I select cell B3 and run Flash Fill, Excel recognizes the pattern and fills in the rest of the first names; similarly, selecting C3 and running Flash Fill adds the middle initials, and selecting D3 and running Flash Fill adds the last names.

You can use Flash Fill to extract substrings (right) from longer strings (left).
Objective group 2  Manage and format data

- Column A contains a list of phone numbers without any parentheses or dashes. In cell B2, I typed the phone number from cell A2 and added the parentheses and dash. When I select cell B3 and run Flash Fill, Excel recognizes the pattern and fills the rest of the column with the formatted phone numbers.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phone 1</td>
</tr>
<tr>
<td>2</td>
<td>3175557605</td>
</tr>
<tr>
<td>3</td>
<td>3175559281</td>
</tr>
<tr>
<td>4</td>
<td>3175551452</td>
</tr>
<tr>
<td>5</td>
<td>3175555016</td>
</tr>
<tr>
<td>6</td>
<td>3175556433</td>
</tr>
<tr>
<td>7</td>
<td>3175559709</td>
</tr>
<tr>
<td>8</td>
<td>3175551890</td>
</tr>
<tr>
<td>9</td>
<td>3175559592</td>
</tr>
<tr>
<td>10</td>
<td>3175555877</td>
</tr>
<tr>
<td>11</td>
<td>3175554733</td>
</tr>
<tr>
<td>12</td>
<td>3175555123</td>
</tr>
<tr>
<td>13</td>
<td>3175551166</td>
</tr>
<tr>
<td>14</td>
<td>3175559826</td>
</tr>
<tr>
<td>15</td>
<td>3175555420</td>
</tr>
<tr>
<td>16</td>
<td>3175552765</td>
</tr>
<tr>
<td>17</td>
<td>3175557767</td>
</tr>
<tr>
<td>18</td>
<td>3175553038</td>
</tr>
<tr>
<td>19</td>
<td>3175555745</td>
</tr>
<tr>
<td>20</td>
<td>3175544553</td>
</tr>
<tr>
<td>21</td>
<td>3175550200</td>
</tr>
<tr>
<td>22</td>
<td>3175551919</td>
</tr>
<tr>
<td>23</td>
<td>3175559642</td>
</tr>
<tr>
<td>24</td>
<td>3175559113</td>
</tr>
</tbody>
</table>

You can use Flash Fill to add characters (right) to plain phone numbers (left).

To fill cells by using Flash Fill

→ Type the first value you want in the first cell of the new column, select the second cell in the new column, and then on the Data tab, in the Data Tools group, click Flash Fill. Alternatively, on the Home tab, in the Editing group, click Fill, then click Flash Fill.

Tip  You can also run the Flash Fill command by pressing Ctrl+E.

Or,

→ Type the first value you want in the first cell of the new column. Starting in the second cell of the new column, drag the Fill handle through all the cells in the new column that you want to fill, click Auto Fill Options, and then click Flash Fill.
Fill cells by using advanced Fill Series options

Worksheets often use text series (such as January, February, March; or Sunday, Monday, Tuesday) and numeric series (such as 1, 3, 5; or 2021, 2020, 2019). Instead of entering these series manually, you can create them automatically by using the Auto Fill feature. That is, you enter and select the first couple of values in the series, drag the fill handle over the range you want to fill, and then click Fill Series in the AutoFill Options list.

You can use the fill handle and the Fill Series option to extend an existing series.

Instead of using the fill handle to create a series, you can use the Series command to gain more control over the process. By using the Series command, you can specify a direction for the fill (rows or columns); a step value (the value by which each item in the series is changed to produce the next item); a stop value (the value at which Excel should terminate the series); whether you want the series to extend the trend of the initial values; the date units (such as day or month) for a date series; and the series type, which can be one of the following four values:

- **Linear** This option finds the next series value by adding the step value to (or subtracting the step value from) the preceding value in the series.
- **Growth** This option finds the next series value by multiplying the preceding value by the step value.
- **Date** This option creates a series of dates based on the option you select in the Date Unit group (Day, Weekday, Month, or Year).
- **AutoFill** This option works much like the fill handle. You can use it to extend a numeric pattern or a text series (for example, Qtr1, Qtr2, Qtr3).
To fill cells by using the Fill Series command

1. Enter the starting value in the first cell you want to use for the series. If you want to create a series out of a particular pattern (such as 2, 4, 6, and so on), fill in enough cells to define the pattern.

2. Select the entire range you want to fill.

3. On the Home tab, in the Editing group, click Fill, then click Series to open the Series dialog box.

4. Do either of the following to create the series, starting from the active cell:
   - Click Rows to create the series in rows.
   - Click Columns to create the series in columns.

5. In the Type group, click the type of series you want.

6. Do any of the following:
   - If you selected the Date type, click an option in the Date unit group.
   - If you selected the Linear or Growth type and want to extend a series trend, select the Trend check box.
   - If you selected a Linear, Growth, or Date series type, enter a number in the Step value box. This number is what Excel uses to generate the next value in the series.
   - To place a limit on the series, enter a number in the Stop value box.

7. Click OK to fill in the series and return to the worksheet.
Objective 2.1 practice tasks

The practice file for these tasks is located in the MOSExcelExpert2019\Objective2 practice file folder. The folder also contains a result file that you can use to check your work.

➤ Open the ExcelExpert_2-1 workbook, display the Flash Fill worksheet, and do the following:

- In cell B2, type the company name from cell A2 using title case, select cell B3, and then run the Flash Fill command from the Home tab.
- In cell D2, type the first name from the full name shown in cell C2, select cell D3, and then run the Flash Fill command from the Data tab.
- In cell E2, type the initial from the full name shown in cell C2, select cell E3, and then run the Flash Fill command from the Data tab.
- In cell F2, type the last name from the full name shown in cell C2, select cell F3, and then run the Flash Fill command from the Data tab.
- In cell H2, type the phone number shown in cell G2, but include parentheses around the area code, a space after the closing parenthesis, and a dash between the sixth and seventh digits. Drag the Fill handle from cell H2 down to cell H24, click Auto Fill Options, and then click Flash Fill.

➤ Display the Fill Series worksheet and do the following:

- In column A, below the Linear label, create a linear series that begins at 0, has a step value of 5, and has a stop value of 50.
- In column B, below the Growth label, create a growth series that begins at 1, has a step value of 2, and has a stop value of 250.
- Fill the range C2:C11 with a Date series that uses a day unit and a step value of 2.
Objective group 2  Manage and format data

- Fill the range D2:D11 with a Date series that uses a weekday unit and a step value of 1.
- Fill the range E2:E11 with a Date series that uses a month unit and a step value of 6.
- Save the workbook.
- Open the ExcelExpert_2-1_results workbook. Compare the two workbooks to check your work. Then close the open workbooks.
Objective 2.2: Format and validate data

Create custom number formats

One of the best ways to improve the readability of your worksheets is to display your data in a format that is logical, consistent, and straightforward. Formatting currency amounts with leading dollar signs, percentages with trailing percent signs, and large numbers with commas are a few of the ways you can improve your spreadsheet style. However, you can use Excel to go beyond these built-in formats to create custom number and date formats with which you can display your worksheet values exactly as you want them to be seen.

Excel's built-in numeric formats give you a great deal of control over how your numbers are displayed, but they have their limitations. For example, there is no built-in format you can use to display a different currency symbol, such as the Euro symbol (€), or to display temperatures using, say, the degree symbol (°).

To overcome these limitations, you need to create your own custom numeric formats. You can do this either by editing an existing format or by entering your own format from scratch. The formatting syntax and symbols are explained in detail later in this section.

Every Excel numeric format, whether built-in or customized, has the following syntax:

\[ \text{positive format};\text{negative format};\text{zero format};\text{text format} \]

The four parts, separated by semicolons, determine how various numbers are presented. The first part defines how a positive number is displayed, the second part defines how a negative number is displayed, the third part defines how zero is displayed, and the fourth part defines how text is displayed. If you leave out one or more of these parts, numbers are controlled as shown in the following table.

<table>
<thead>
<tr>
<th>Number of parts used</th>
<th>Format syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Three</td>
<td>\text{positive format};\text{negative format};\text{zero format}</td>
</tr>
<tr>
<td>Two</td>
<td>\text{positive and zero format};\text{negative format}</td>
</tr>
<tr>
<td>One</td>
<td>\text{positive, negative, and zero format}</td>
</tr>
</tbody>
</table>

The following table lists the special symbols you use to define each of these parts.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Holds a place for a digit and displays the digit exactly as typed. Displays nothing if no number is entered. For example, if a cell’s custom format is ### and you enter 25 in the cell, Excel displays 25.</td>
</tr>
<tr>
<td>0</td>
<td>Holds a place for a digit and displays the digit exactly as typed. Displays zero if no number is entered. For example, if a cell’s custom format is 000 and you enter 25 in the cell, Excel displays 025.</td>
</tr>
<tr>
<td>?</td>
<td>Holds a place for a digit and displays the digit exactly as typed. Displays a space if no number is entered. For example, if a cell’s custom format is 0?? and you enter 25 in the cell, Excel displays 0 25.</td>
</tr>
<tr>
<td>. (period)</td>
<td>Sets the location of the decimal point. For example, if a cell’s custom format is #.0 and you enter 34.5 in the cell, Excel displays 34.50.</td>
</tr>
<tr>
<td>, (comma)</td>
<td>Sets the location of the thousands separator. Marks only the location of the first thousand. For example, if a cell’s custom format is #,### and you enter 12345 in the cell, Excel displays 12,345.</td>
</tr>
<tr>
<td>%</td>
<td>Multiplies the number by 100 (for display only) and adds the percent (%) character. For example, if a cell’s custom format is #% and you enter .75 in the cell, Excel displays 75%.</td>
</tr>
<tr>
<td>E+ e+ E- e-</td>
<td>Displays the number in scientific format. E- and e- place a minus sign in the exponent; E+ and e+ place a plus sign in the exponent. For example, if a cell’s custom format is 0.00E+00 and you enter 123456789 in the cell, Excel displays 1.23E+08. Similarly, if a cell’s custom format is 0.0E-00 and you enter 0.0000012 in the cell, Excel displays 1.2E-06.</td>
</tr>
<tr>
<td>/ (slash)</td>
<td>Sets the location of the fraction separator. For example, if a cell’s custom format is /0/ and you enter .75 in the cell, Excel displays 3/4.</td>
</tr>
<tr>
<td>$ () : - + &lt;space&gt;</td>
<td>Displays the character. For example, if a cell’s custom format is $$#0.00 and you enter 123.5 in the cell, Excel displays $123.50.</td>
</tr>
</tbody>
</table>
Objective 2.2: Format and validate data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Repeats whatever character immediately follows the asterisk until the cell is full. Doesn’t replace other symbols or numbers. For example, you can create a dot trailer in a cell by adding * to the format. So if the custom format is #*, and you enter 123 in the cell, Excel displays 123........... (where the dots continue until the cell is filled).</td>
</tr>
<tr>
<td>_(underscore)</td>
<td>Inserts a blank space the width of whatever character follows the underscore, which can often help you to align your numbers. For example, the custom format _(#.00) inserts a blank space the width of the opening parenthesis at the beginning of the displayed value.</td>
</tr>
<tr>
<td>\ (backslash)</td>
<td>Inserts the character that follows the backslash. See the next item for an example. In general, you need to use the backslash only for reserved characters (such as # or @) or for the following letters: B, D, E, G, H, M, N, S, and Y. (For all other letters, if you just enter a single character by itself, Excel will display that character.) For example, if a cell’s custom format is #.##\M and you enter 1.23 in the cell, Excel displays 1.23M.</td>
</tr>
<tr>
<td>“text”</td>
<td>Inserts the text that appears within the quotation marks. For example, if a cell’s custom format is “Part “#00-0000 and you enter 123456 in the cell, Excel displays Part #12-3456.</td>
</tr>
<tr>
<td>@</td>
<td>Displays the cell’s text. For example, if a cell’s custom format is @“ entry” and you enter credit in the cell, Excel displays credit entry.</td>
</tr>
<tr>
<td>[color]</td>
<td>Displays the cell contents in the specified color. For example, if the cell’s custom format is [green]0.00; [red]0.00, Excel displays positive cell values in green and negative cell values in red. The predefined color values you can use are black, white, red, green, blue, yellow, magenta, and cyan, and the color codes color1 through color565.</td>
</tr>
</tbody>
</table>

Although the built-in date and time formats of Excel are fine for most purposes, you might need to create your own custom formats. For example, you might want to display the day of the week (for example, “Friday”). Custom date and time formats generally are simpler to create than custom numeric formats. There are fewer formatting symbols, and you usually don’t need to specify different formats for different conditions. The following table lists the date and time formatting symbols.
Objective group 2  Manage and format data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Day number without a leading zero (1 to 31)</td>
</tr>
<tr>
<td>dd</td>
<td>Day number with a leading zero (01 to 31)</td>
</tr>
<tr>
<td>ddd</td>
<td>Three-letter day abbreviation (Mon, for example)</td>
</tr>
<tr>
<td>dddd</td>
<td>Full day name (Monday, for example)</td>
</tr>
<tr>
<td>m</td>
<td>Month number without a leading zero (1 to 12)</td>
</tr>
<tr>
<td>mm</td>
<td>Month number with a leading zero (01 to 12)</td>
</tr>
<tr>
<td>mmm</td>
<td>Three-letter month abbreviation (Aug, for example)</td>
</tr>
<tr>
<td>mmmm</td>
<td>Full month name (August, for example)</td>
</tr>
<tr>
<td>yy</td>
<td>Two-digit year (00 to 99)</td>
</tr>
<tr>
<td>yyyy</td>
<td>Full year (1900 to 2078)</td>
</tr>
<tr>
<td>h</td>
<td>Hour without a leading zero (0 to 24)</td>
</tr>
<tr>
<td>hh</td>
<td>Hour with a leading zero (00 to 24)</td>
</tr>
<tr>
<td>m</td>
<td>Minute without a leading zero (0 to 59)</td>
</tr>
<tr>
<td>mm</td>
<td>Minute with a leading zero (00 to 59)</td>
</tr>
<tr>
<td>s</td>
<td>Second without a leading zero (0 to 59)</td>
</tr>
<tr>
<td>ss</td>
<td>Second with a leading zero (00 to 59)</td>
</tr>
<tr>
<td>AM/PM, am/pm, A/P</td>
<td>Displays the time using a 12-hour clock</td>
</tr>
<tr>
<td>/ : . —</td>
<td>Symbols used to separate parts of dates or times</td>
</tr>
<tr>
<td>[color]</td>
<td>Displays the date or time in the color specified</td>
</tr>
</tbody>
</table>

The best way to become familiar with custom formats is to try your own experiments. Excel stores each format that you try. If you find that your list of custom formats is getting a bit unwieldy or that it’s cluttered with unused formats, you can delete those formats.

To open the Format Cells dialog box, do one of the following:

- On the Home tab, in the Cells group, click Format, then click Format Cells.
- Right-click the cell or range, then click Format Cells.
- Press Ctrl+1.
To create and apply a custom number format

1. Select the cell or range of cells you want the new format to apply to.
2. Open the **Format Cells** dialog box.
3. On the **Number** tab, in the **Category** list, click **Custom**.
4. To base the custom number format on an existing format, click the base format in the **Type** list.
5. Edit or enter the symbols that define the number format.

   ![Format Cells dialog box]

   *Define custom number formats in the **Type** box.*

6. When you are done, click **OK** to return to the worksheet.

To delete custom number formats

1. Display the **Number** tab of the **Format Cells** dialog box.
2. In the **Category** list, click **Custom**.
3. In the **Type** list, click the format you want to remove.

   **Tip** You can delete only custom formats; you can't delete built-in formats.
4. Click **Delete** to remove the format from the list.

5. Click **OK** to close the **Format Cells** dialog box and return to the worksheet.

### Configure data validation

Formulas are only as good as the data they’re given. For basic data entry errors (for example, entering the wrong date or transposing a number’s digits), there’s not much you can do other than exhort yourself or the people who use your worksheets to enter data carefully. Fortunately, you have a bit more control when it comes to preventing the entry of improper data such as data that is the wrong type (for example, entering text in a cell that requires a number) or data that falls outside of an allowable range (for example, entering 200 in a cell that requires a number between 1 and 100).

You can prevent these kinds of improper entries, to a certain extent, by adding comments that describe what is allowable inside a particular cell. However, this requires other people to both read and act on the comment text. You can also use custom numeric formatting to “format” a cell with an error message if the wrong type of data is entered. This is useful, but it works only for certain kinds of input errors.

Exam Strategy

The Excel Expert exam emphasizes collaboration between users, so be sure to also study the information related to setting up workbooks for other people to view and edit in “Objective 1.2: Prepare workbooks for collaboration.”

The best solution for preventing data entry errors is to use the data-validation feature of Excel. With data validation, you create rules that specify exactly what kind of data can be entered and in what range that data can fall. You can also specify pop-up input messages that appear when a cell is selected and error messages that appear when data is entered improperly.

You configure data-validation rules on the Settings tab of the Data Validation dialog box. The following validation types are available:

- **Any Value** Allows any value in the range (that is, it removes any previously applied validation rule). If you’re removing an existing rule, be sure to also clear the input message, if any.

- **Whole Number** Allows only whole numbers (integers). You use the Data list to select a comparison operator (such as Between, Equal To, or Less Than) and then enter the specific criteria. For example, if you click the Between option, you must enter Minimum and Maximum values.
Objective 2.2: Format and validate data

- **Decimal**  Allows decimal numbers or whole numbers. You use the Data list to select a comparison operator and then enter the specific numeric criteria.

- **List**  Allows only values specified in a list. You specify the allowable values in the Source box on the Settings tab of the Data Validation dialog box, either by specifying a range on the same sheet or a range name on any sheet that contains the list of allowable values (preceding the range or range name with an equal sign) or by entering the allowable values directly into the Source box (separated by commas). You have the option of allowing the user to select from the allowable values by using a list.

- **Date**  Allows only dates. (If the user includes a time value, the entry is invalid.) You use the Data list to select a comparison operator and then enter the specific date criteria (such as a Start date and an End date).

- **Time**  Allows only times. (If the user includes a date value, the entry is invalid.) You use the Data list to select a comparison operator and then enter the specific time criteria (such as a Start time and an End time).

- **Text Length**  Allows only alphanumeric strings of a specified length. You use the Data list to select a comparison operator and then enter the specific length criteria (such as Minimum and Maximum lengths).

- **Custom**  You can use this option to enter a formula that specifies the validation criteria. You can either enter the formula directly in the Formula box on the Settings tab of the Data Validation dialog box (again preceding the formula with an equal sign) or enter a reference to a cell that contains the formula. For example, if you’re restricting cell A2 and you want to be sure the entered value is not the same as what’s in cell A1, you would enter the formula =A2<>A1.

On the Settings tab of the Data Validation dialog box, set up the criteria for your validation rule.
Objective group 2  Manage and format data

To configure data validation for a cell or range

1. Select the cell or range to which you want to apply the data-validation rule.
2. On the **Data** tab, in the **Data Tools** group, click **Data Validation** to open the **Data Validation** dialog box.
3. On the **Settings** tab, in the **Allow** list, click one of the validation types.
4. Enter the validation criteria you require.
5. To allow blank entries, either in the cell itself or in other cells specified as part of the validation settings, leave the **Ignore blank** check box selected. If you clear this check box, Excel treats blank entries as zero and applies the validation rule accordingly.
6. If the range had an existing validation rule that also applied to other cells, you can apply the new rule to those other cells by selecting the **Apply these changes to all other cells with the same settings** check box.
7. If you want a message to appear when the user selects the restricted cell or any cell within the restricted range, on the **Input Message** tab, do the following:
   - Verify that the **Show input message when cell is selected** check box is selected.
   - In the **Title** box, enter a title for the message.
   - In the **Input message** box, enter the message that you want Excel to display. For example, you could use the message to give the user information about the type and range of allowable values.

![Data Validation dialog box](image)

You can configure an input message to appear when a workbook user selects the cell.
8. If you want a dialog box to appear when the user enters invalid data, click the Error Alert tab, then do the following:

Select the Show error alert after invalid data is entered check box.

In the Style list, click the error style you want: Stop, Warning, or Information.

In the Title box, enter a title for the message.

In the Error message box, enter the message that you want Excel to display.

**IMPORTANT** Only the Stop style prevents users from entering invalid data.

---

You can configure an error alert to appear when a workbook user enters an invalid entry.

9. Click OK to apply the data-validation rule.

**Group and ungroup data**

You can control a worksheet range display by grouping the data based on the worksheet formulas and data. Grouping the data creates a worksheet outline, which you can use to “collapse” sections of the sheet to display only summary cells, or to “expand” hidden sections to show the underlying detail.

Not all worksheets can be grouped, so you need to make sure your worksheet is a candidate for outlining. First, the worksheet must contain formulas that reference cells or ranges directly adjacent to the formula cell. Worksheets with SUM functions that subtotal cells above or to the left are particularly good candidates for outlining.

Second, there must be a consistent pattern to the direction of the formula references. For example, a worksheet with formulas that always reference cells above or to the
left can be outlined. Excel will not outline a worksheet with, say, SUM functions that reference ranges above and below a formula cell.

If your worksheet meets these criteria, then you can use Excel’s Auto Outline command to automatically group the worksheet data. Otherwise, you can group data manually.

When Excel creates an outline, it divides your worksheet into a hierarchy of levels. These levels range from the worksheet detail (the lowest level) to the grand totals (the highest level). Excel outlines can handle up to eight levels of data. In the Budget worksheet shown below, for example, Excel created three levels for both the column and the row data:

- In the columns, the monthly figures are the details, so they’re the lowest level (level 3). The quarterly totals are the first summary data, so they’re the next level (level 2). Finally, the grand totals (not shown) are the highest level (level 1).
- In the rows, the individual sales and expense items are the details (level 3). The sales and expenses subtotals are the next level (level 2). The Gross Profit row is the highest level (level 1).

![Budget Worksheet](image)

When you create an outline, Excel adds outline tools to the worksheet.

To help you work with your outlines, Excel adds the following tools to your worksheet:

- **Level bars**—These bars indicate the data included in the current level. Click a bar to hide the rows or columns marked by a bar.
- **Collapse symbol**—Click this symbol to hide (or collapse) the rows or columns marked by the attached level bar.
Objective 2.2: Format and validate data

- **Expand symbol**—When you collapse a level, the collapse symbol changes to an expand symbol (+). Click this symbol to display (or expand) the hidden rows or columns.

- **Level symbols**—These symbols tell you which level each level bar is on. Click a level symbol to display all the detail data for that level.

  Tip: To toggle the outline symbols on and off, press Ctrl+8.

To group a worksheet using Auto Outline

➔ On the **Data** tab, in the **Outline** group, click **Group**, then click **Auto Outline**.

To group data manually

➔ Select the rows or columns you want to group, then on the **Data** tab, in the **Outline** group, click **Group**, then click **Group**.

Or,

1. Select a cell in each row or column you want to group.
2. On the **Data** tab, in the **Outline** group, click **Group**, then click **Group**. Excel asks whether you want to group rows or columns.
3. Select either **Rows** or **Columns**, as appropriate for your data, then click **OK**.

![Group dialog box](image)

*When you group data manually, Excel needs to know whether you’re grouping rows or columns.*

To remove an outline from a worksheet

➔ On the **Data** tab, in the **Outline** group, click **Ungroup**, then click **Clear Outline**.

To ungroup data

➔ Select the rows or columns you want to ungroup, then on the **Data** tab, in the **Outline** group, click **Ungroup**, then click **Ungroup**.
Calculate data by inserting subtotals and totals

Although you can use formulas and worksheet functions to summarize your data in various ways—including sums, averages, counts, maximums, and minimums—if you’re in a hurry, or if you just need a quick summary of your data, you can get Excel to do the work for you. The secret here is a feature called automatic subtotals, which are formulas that Excel adds to a worksheet automatically.

Excel sets up automatic subtotals based on data groupings in a selected field. For example, if you ask for subtotals based on the Customer field, Excel runs down the Customer column and creates a new subtotal each time the name changes. To get useful summaries, you should sort the range on the field containing the data groupings you’re interested in.

Invoice data showing subtotals for each customer based on the values in the Total column.
Objective 2.2: Format and validate data

See Also  For more information on the outline tools that Excel adds to the worksheet when you insert subtotals, see “Group and ungroup data,” earlier in this topic.

Note that in the phrase automatic subtotals, the word subtotals is misleading because it implies that you can only summarize your data with totals. However, you can also count the values (all the values or just the numeric values), calculate the average of the values, determine the maximum or minimum value, and calculate the product of the values. For statistical analysis, you can also calculate the standard deviation and variance, both of a sample and of a population.

To insert subtotals and totals

1. Select a cell within the range you want to subtotal.
2. On the Data tab, in the Outline group, click Subtotal to open the Subtotal dialog box.
3. Use the At each change in list to select the column you want to use to group the subtotals.
4. In the Use function list, select Sum.
5. In the Add subtotal to list, select the check box for the column you want to summarize.
6. Click OK. Excel calculates the subtotals and adds them to the range and also adds a Grand Total row to the bottom of the range. Excel also adds outline symbols to the range.

Use the Subtotal dialog box to apply subtotals to a range.
To change the summary calculation

➜ In step 4 from the previous section, in the Use function list, select the function you want to use, such as Count, Average, Max, or Min.

Remove duplicate records

You can make your Excel data more accurate for analysis by removing duplicate records. Duplicate records throw off calculations by including the same data two or more times. To prevent this, you should delete duplicate records. Rather than looking for duplicates manually, you can use the Remove Duplicates command, which quickly finds and removes duplicates in even the largest ranges or tables.

Before you use the Remove Duplicates command, you must decide what defines a duplicate record in your data. You have two choices:

- Two records are duplicates if, for every column in the range or table, the records contain identical values.
- Two records are duplicates if, for only certain columns in the range or table, the records contain identical values.

To remove duplicate records

1. Click a cell inside the range or table.
2. On the Data tab, in the Data Tools group, click Remove Duplicates to open the Remove Duplicates dialog box.
3. If your range doesn’t have column headers, clear the My data has headers check box.
4. Select the check box beside each field that you want Excel to check for duplicate values.

Tip If your range or table has many fields, you may want Excel to use only one or two of those fields to look for duplicate records. Rather than clearing all the other check boxes manually, first click Unselect All in the Remove Duplicates dialog box to clear all the check boxes. You can then click to select just the check boxes you want Excel to use.
Objective 2.2: Format and validate data

Use the Remove Duplicates dialog box to specify which columns must contain identical data for the records to be considered duplicates.

**IMPORTANT** Excel does not give you a chance to confirm the deletion of the duplicate records, so be sure you want to do this before proceeding.

5. Click **OK**. Excel deletes any duplicate records that it finds and then displays a dialog box telling you the number of duplicate records that it deleted.

![Remove Duplicates dialog box]

Excel tells you how many duplicate records it deleted and how many unique records remain in the range or table.

6. Click **OK**.
Objective 2.2 practice tasks

The practice file for these tasks is located in the MOSExcelExpert2019\Objective2 practice file folder. The folder also contains a result file that you can use to check your work.

➤ Open the ExcelExpert_2-2 workbook, display the Custom Data Formatting worksheet, and do the following:

☐ Select cells A1:A4. Create and apply a custom number format that displays the thousands separator, always displays at least one number, displays a leading minus sign and red text if a negative number is entered, displays 0 if 0 is entered, and displays the message Enter a number if a non-numeric value is entered.

☐ Select cell B1. Create and apply a custom number format that displays the thousands separator and the decimal point, always displays at least one digit before and after the decimal point, and displays °C (the degree symbol and the letter C, for degrees Celsius) at the end.

☐ Select cells C1:C2. Create and apply a custom number format that displays a six-digit entry with a dash after the first two digits, the text Acct # before the digits, and the text Enter numbers only if the user includes any non-numeric characters in the entry.

☐ Select cell D1. Create and apply a custom date format that displays the two-digit month, day, and year, separated by periods.

☐ Select cell E1. Create and apply a custom time format that displays the two-digit hour and minute with nothing in between them, followed by a space and the text hours.

➤ Display the Data Validation worksheet and do the following:

☐ Select cell B2. Create and apply a data-validation rule that restricts data entry to values between 0 and 1 (that is, between 0% and 100%).

☐ Include an input message titled Interest Rate with the following text: Please enter a value between 0 and 1.
Objective 2.2: Format and validate data

- Then enter a stop-style error message titled \textit{Invalid Interest Rate} with the following text: \textit{The interest rate value you entered is invalid. Please enter a value between 0 and 1.}
- Select cell B3. Create and apply a data-validation rule that restricts data entry to positive values with a minimum of 1 and a maximum of 30.
- Include an input message titled \textit{Loan Period} with the following text: \textit{Please enter a value between 1 and 30 years.}
- Then enter a stop-style error message titled \textit{Invalid Loan Period} with the following text: \textit{The loan period value you entered is invalid. Please enter a value between 1 and 30 years.}
- Select cell B4. Create and apply a data-validation rule that restricts data entry to positive values.
- Include an input message titled \textit{Loan Principal} with the following text: \textit{Please enter a value greater than 0.}
- Enter a stop-style error message titled \textit{Invalid Loan Principal} with the following text: \textit{The loan principal value you entered is invalid. Please enter a value greater than 0.}

Display the \textbf{Grouping} worksheet and do the following:
- Run the Auto Outline command to group the worksheet data.
- Create a manual grouping of the Advertising, Rent, and Supplies expense items.

Display the \textbf{Subtotals} worksheet and create Sum subtotals for each customer based on the values in the Total field.

Display the \textbf{Remove Duplicates} worksheet and remove the six duplicate records in the range.

Save the workbook.

Open the \texttt{ExcelExpert_2-2_results} workbook. Compare the two workbooks to check your work. Then close the open workbooks.
Objective group 2  Manage and format data

Objective 2.3: Apply advanced conditional formatting and filtering

Create custom conditional formatting rules

Many Excel worksheets contain hundreds of data values. You can make sense of large sets of data by creating formulas, applying functions, and performing data analysis. However, there are times when you don't want to analyze a worksheet per se. Instead, all you want are answers to simple questions such as the following:

- Which cell values are less than 0?
- What are the top 10 values?
- Which cell values are above average, and which are below average?

Exam Strategy  Data analysis is a crucial component of the Excel Expert exam. This means that besides studying the conditional formatting techniques in this topic, be sure to also study the data analysis material in “Objective 3.4: Perform data analysis” and the PivotTable material in “Objective 4.2: Create and modify PivotTables.”

These simple questions aren’t easy to answer just by glancing at the worksheet, and the more numbers you’re dealing with, the harder it gets. To help you “eyeball” your worksheets and answer these and similar questions, Excel lets you apply conditional formatting to the cells. This is a special format that Excel applies only to cells that satisfy some condition, which Excel calls a rule. For example, you could apply formatting to show all the negative values in a red font, or you could apply formatting to highlight the top 10 values.

You can apply five types of conditional formatting rules:

- **Highlight cells rules**  A highlight cell rule applies a format to cells that meet specified criteria. You have seven choices: Greater Than, Less Than, Between, Equal To, Text That Contains, A Date Occurring, and Duplicate Values. In each case, you use a dialog box to specify the condition and the formatting that you want applied to cells that match the condition.
Objective 2.3: Apply advanced conditional formatting and filtering

For a Less Than conditional formatting rule, the specified formatting is applied to all cells that have a value that is less than the value specified in the dialog box.

- **Top/bottom rules** A top/bottom rule applies a format to cells that rank in the top or bottom (for numerical items, the highest or lowest) values in a range. You can select the top or bottom either as an absolute value (for example, the top 10 items) or as a percentage (for example, the bottom 25%). You can also format cells that are above or below the average. In each case, you use a dialog box to set up the specifics of the rule. For the Top 10 Items, Top 10%, Bottom 10 Items, and Bottom 10% rules, you use the dialog box to specify the condition and the formatting you want applied to cells that match the condition. For the Above Average and Below Average rules, you use the dialog box to specify the formatting only.

An example of a Top 10 conditional formatting rule.

- **Data bars** If you're interested in the relationship between similar values in a worksheet, you need some way to visualize the relative values in a range, and that's where data bars are useful. Data bars are colored, horizontal bars that appear “behind” the values in a range. (They’re like a bar chart.) Their key feature is that the length of the data bar that appears in each cell is related to the value in that cell: the larger the value, the longer the data bar. The cell with the highest value has the longest data bar, and the data bars that appear in the other cells have lengths that reflect their values. Excel configures its default data
bars with the longest data bar based on the highest value in the range and the shortest data bar based on the lowest value in the range. You can also set up data bars based on a specific range of values (for example, the values 0 and 100 for test scores) or as a percentage of the largest value.

A Green - Yellow - Red color scale shows the lowest values in the deepest shade of red and the highest values in the deepest shade of green.
Objective 2.3: Apply advanced conditional formatting and filtering

- **Icon sets** You use *icon* sets to visualize the relative values of cells in a range. Excel adds an icon to each cell in the range, and that icon tells you something about the cell’s value relative to the rest of the range. For example, the highest values might get an up arrow, the lowest values a down arrow, and the values in between a horizontal arrow. Icon sets take advantage of symbols that people have strong associations with.

  For example, a check mark means something is good or finished or acceptable, whereas an X means something is bad or unfinished or unacceptable. A green circle is positive, whereas a red circle is negative (similar to traffic lights).

<table>
<thead>
<tr>
<th>Sales Rep</th>
<th>2019 Sales</th>
<th>2020 Sales</th>
<th>% +/-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nancy Voorhees</td>
<td>$960,336</td>
<td>$960,492</td>
<td>-4%</td>
</tr>
<tr>
<td>Andrew Cerealis</td>
<td>$717,631</td>
<td>$777,583</td>
<td>15%</td>
</tr>
<tr>
<td>Jan Rotas</td>
<td>$802,751</td>
<td>$864,579</td>
<td>8%</td>
</tr>
<tr>
<td>Mariya Sereikno</td>
<td>$755,327</td>
<td>$827,213</td>
<td>4%</td>
</tr>
<tr>
<td>Steven Thorpe</td>
<td>$643,508</td>
<td>$711,213</td>
<td>10%</td>
</tr>
<tr>
<td>Michael Nepper</td>
<td>$1,051,321</td>
<td>$846,994</td>
<td>12%</td>
</tr>
<tr>
<td>Robert Zare</td>
<td>$722,450</td>
<td>$626,945</td>
<td>3%</td>
</tr>
<tr>
<td>Laura Giussani</td>
<td>$902,059</td>
<td>$874,472</td>
<td>12%</td>
</tr>
<tr>
<td>Anne Hellweg-Larson</td>
<td>$659,380</td>
<td>$722,932</td>
<td>26%</td>
</tr>
</tbody>
</table>

A 3 Arrows icon set indicates low, middle, and high values in a data set by using arrows of different colors and directions.

**To create a custom conditional formatting rule**

1. Select the range to which you want the custom conditional formatting applied.
2. On the **Conditional Formatting** menu, click **New Rule** to open the **New Formatting Rule** dialog box.
3. In the **Select a Rule Type** group, click the type of rule you want to apply.

The controls in the **Edit The Rule Description** area vary depending on the rule type you select.
4. Select the rule type's conditions and formatting.

5. Click OK to close the dialog box and return to the workbook.

Create conditional formatting rules that use formulas

Excel comes with another conditional formatting component that makes this feature even more powerful: you can apply conditional formatting based on the results of a formula. In particular, you can set up a logical formula as the conditional formatting criterion. If that formula returns TRUE, Excel applies the formatting to the cells; if the formula returns FALSE, instead, Excel doesn't apply the formatting.

When comparing worksheet values in a conditional formatting rule's logical formula, you generally set up the expression to compare the value you seek with a specific value from the range, and then use a mixed-reference format for that specific value so that Excel can compare all the values in the range to the target value. For example, suppose you have a list of percentage increases in the range D5:D13 and a “target” percentage in cell D2. You want to apply a format only on those entries where the percentage increase is greater than or equal to the target value. Here's the logical formula to use:

$$=D5 \geq D2$$

The mixed-reference format $D5$ tells Excel to keep the column (D) fixed while varying the row number (in this example, 5 through 13), and in each case compare the resulting cell value with the value in $D2$.

To create a custom conditional formatting rule based on a formula

1. Select the range to which you want the custom conditional formatting applied.

2. In the New Formatting Rule dialog box, in the Select a Rule Type list, click Use a formula to determine which cells to format.

3. In the Format values where this formula is true box, enter your logical formula.
Objective 2.3: Apply advanced conditional formatting and filtering

4. Click **Format** to open the **Format Cells** dialog box.

5. On the **Number**, **Font**, **Border**, and **Fill** tabs, specify the formatting you want Excel to apply when the formula evaluates to TRUE, then click **OK**.

6. In the **New Formatting Rule** dialog box, click **OK**.

Manage conditional formatting rules

Conditional formatting rules are extremely useful and powerful tools, so you might find that you use them frequently. As you use them, however, your need to manage those rules will increase. For example, you’ll often need to edit existing rules to update the conditions or change the formatting. Similarly, if you’ve applied two or more rules to the same range, you should know how to change the order that Excel uses to apply those rules. Finally, you also need to know how to delete existing rules that you no longer require.

To edit a conditional formatting rule

1. Click any cell in the range that has the conditional formatting applied.

2. On the **Conditional Formatting** menu, click **Manage Rules** to open the **Conditional Formatting Rules Manager** dialog box.

3. Click the rule you want to modify, then click **Edit Rule** to open the **Edit Formatting Rule** dialog box.
4. Make your changes to the rule type, rule conditions, or rule formatting, then click **OK**.

5. In the **Edit Formatting Rule** dialog box, click **OK**.

6. In the **Conditional Formatting Rules Manager** dialog box, click **OK**.

**To change the order in which conditional formatting rules are applied**

1. Click any cell in the range that has the conditional formatting applied.

2. Open the **Conditional Formatting Rules Manager** dialog box.

3. Click a rule, then click either the up arrow or the down arrow until the rule is in the position you prefer.

4. Repeat for the other rules you want to move, then click **OK**.

**To delete a conditional formatting rule**

1. Click any cell in the range that has the conditional formatting applied.

2. Open the **Conditional Formatting Rules Manager** dialog box.

3. Click the rule you want to remove, then click **Delete Rule**. Excel removes the rule.

4. Click **OK** to close the dialog box and return to the worksheet.

**Tip** If you want to delete multiple rules, a quicker method is to click any cell in the range that has the rules applied, click the Home tab, click Conditional Formatting in the Styles group, click Clear Rules, and then click Clear Rules From Selected Cells. If you want to delete every rule in the current worksheet, click Clear Rules From Entire Sheet.
Objective 2.3 practice tasks

The practice files for these tasks are located in the MOSExcelExpert2019\Objective2 practice file folder. The folder also contains result files that you can use to check your work.

➤ Open the ExcelExpert_2-3a workbook and do the following:

- On the Student Grades worksheet, for the cells in the Grade column, create and apply a custom conditional formatting rule that applies the 4 Traffic Lights Icon Set formatting based on the cell values. Display the black icon for values less than 50; the red icon for values from 50 to 59; the yellow icon for values from 60 to 79; and the green icon for values of 80 and greater.

- On the Projects worksheet, for the cells in the Percent column, create and apply a custom conditional formatting rule that applies the Data Bar format with a green gradient fill based on the cell values. Set the minimum value to 0 and the maximum value to 1.

- On the Product Inventory worksheet, for the Product Name column, create and apply a custom conditional formatting rule that applies an Orange fill color to cells that contain duplicate product names.

- Save the workbook.

- Open the ExcelExpert_2-3a_results workbook. Compare the two workbooks to check your work, and then close the open workbooks.

➤ Open the ExcelExpert_2-3b workbook and do the following:

- On the Customers worksheet, for the cell range A4:K94, create and apply a formula-based conditional formatting rule that evaluates cell B1 and applies a Yellow fill color to all the cells of each row that has the same country/region in column I. Test the conditional formatting rule by changing the country/region in cell B1.
Objective group 2  Manage and format data

- On the **Accounts Receivable Data** worksheet, for the cell range A4:G55, use the MOD function (refer to Excel Help if you are not familiar with this function) to create and apply a formula-based conditional formatting rule that applies a light gray fill color to every other row. Test the conditional formatting rule by deleting a row.

- On the **Products** worksheet, for the cell range A2:B78, create and apply a formula-based conditional formatting rule that applies a Red fill color and Bold font to the cells in the rows that contain the highest and lowest values in the Change In Units Sold column. Test the conditional formatting rule by entering new high and low values.

- Save the workbook.

- Open the **ExcelExpert_2-3b_results** workbook. Compare the two workbooks to check your work, and then close the open workbooks.

➤ Open the **ExcelExpert_2-3c** workbook, display the **Accounts Receivable Data** worksheet, and do the following for the cell range A4:G55:

  - Create and apply a conditional formatting rule that applies a Light Green fill color to the cells in each row that has an Invoice Amount value greater than $2,000.

  - Create and apply a conditional formatting rule that applies an Orange fill color to the cells in each row that has a Days Overdue value greater than or equal to 30.

  - Edit the first rule so that it applies the formatting to rows that have an Invoice Amount value that is greater than $1,500.

  - Change the order of the conditional formatting rules so that Excel applies the Invoice Amount rule before it applies the Days Overdue rule.

- Save the workbook.

- Open the **ExcelExpert_2-3c_results** workbook. Compare the two workbooks to check your work, and then close the open workbooks.
Objective group 3

Create advanced formulas and macros

The skills tested in this section of the Microsoft Office Specialist Expert exam for Microsoft Excel 2019 relate to the advanced use of formulas and data analysis. Specifically, the following objectives are associated with this set of skills:

3.1 Perform logical operations in formulas
3.2 Look up data by using functions
3.3 Apply advanced date and time functions
3.4 Perform data analysis
3.5 Troubleshoot formulas
3.6 Create and modify simple macros

Although Excel works well as a simple database, in most of your worksheets you’ll want to do more than just store data. You’ll also want to summarize your data, apply statistical methods, perform data analysis, automate tasks with macros, and perform other techniques that you can use to interrogate your worksheet data and draw conclusions that are relevant to your business requirements. This level of dynamism requires creating worksheet models that use formulas, functions, and macros.

This chapter guides you in studying methods for performing logical operations by using functions, nesting functions, performing statistical operations with functions, looking up data with functions, applying advanced date and time functions, analyzing and consolidating data, creating financial formulas, troubleshooting formula errors, and recording and editing simple macros.
To complete the practice tasks in this chapter, you need the practice files contained in the MOSExcelExpert2019\Objective3 practice file folder. For more information, see “Download the practice files” in this book’s introduction.

Objective 3.1: Perform logical operations in formulas

Insert functions into a formula
Formulas that combine operators with basic operands such as numeric and string values are the mainstay of any Excel spreadsheet. But to get the most benefit from the spreadsheet model, you need to expand your formula repertoire to include worksheet functions. Excel includes dozens of these functions, and they’re essential to making your worksheet easier to work with and more powerful.

To insert a function into a formula
1. Enter your formula up to the point where you want to insert the function.
2. On the Formulas tab, in the Function Library group, click the category that contains the function you want to use. Then on the category menu, click the function.
3. In the Function Arguments dialog box, enter the function arguments, then click OK.

Perform logical operations by using the IF, AND, OR, and NOT functions
In the computer world, we very loosely define something as intelligent if it can perform tests on its environment and act in accordance with the results of those tests. However, a computer is a binary machine, so “acting in accordance with the results of a test” means that it can do only one of two things. Even with this limited range of options, you can still bring a great deal of intelligence to your worksheets. Your
formulas will be able to test the values in cells and ranges and then return results based on those tests. This is all done with the logical functions in Excel, which are designed to create decision-making formulas.

The following table describes the most common logical functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF(logical_test, value_if_true, value_if_false)</td>
<td>Performs a logical test and returns a value based on the result</td>
</tr>
<tr>
<td>AND(logical1, logical2,...)</td>
<td>Returns TRUE if all the arguments are true</td>
</tr>
<tr>
<td>OR(logical1, logical2,...)</td>
<td>Returns TRUE if any argument is true</td>
</tr>
<tr>
<td>NOT(logical)</td>
<td>Reverses the logical value of the argument</td>
</tr>
</tbody>
</table>

You use the IF function to test some condition and then return a value based on the result of that test. This is the simplest version of the IF function:

\[
IF(\text{logical\_test}, \text{value\_if\_true})
\]

The \text{logical\_test} argument is a logical expression—that is, an expression that returns TRUE or FALSE (or their equivalent numeric values: 0 for FALSE and any other number for TRUE); the \text{value\_if\_true} argument is the value returned by the function if \text{logical\_test} evaluates to TRUE.

A \text{logical expression} compares two or more numbers, text strings, cell contents, or function results. If the expression is true, it’s given the logical value TRUE (which is equivalent to any nonzero value); if the expression is false, it’s given the logical value FALSE (which is equivalent to zero). The following table summarizes the comparison operators you can use in logical expressions.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to</td>
<td>=10=5</td>
<td>FALSE</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
<td>=10&gt;5</td>
<td>TRUE</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
<td>=10&lt;5</td>
<td>FALSE</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
<td>=&quot;a&quot;&gt;=&quot;b&quot;</td>
<td>FALSE</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
<td>=&quot;a&quot;&lt;=&quot;b&quot;</td>
<td>TRUE</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to</td>
<td>=&quot;a&quot;&lt;&gt;'b'</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
For example, suppose cell B2 contains a sales rep’s total sales and you want to give the rep a 10-percent bonus if those sales are greater than $100,000. Here’s a formula that does that:

=IF(B2 > 100000, B2 * 0.1)

The logical expression B1 > 100000 is used as the test. Assume that you add this formula to cell C2. If the logical expression proves to be true (that is, if the value in cell B2 is greater than 100,000), the function returns the value B2 * 0.1—that is, 10 percent of the number in B2—and that’s the value you see in cell C2.

When the IF function test returns FALSE (that is, the value in column B is less than or equal to 100,000), the function returns FALSE as its result. That’s not inherently bad, but the worksheet would look tidier (and, hence, be more useful) if the formula returned, for instance, the value 0 instead.

To do this, you need to use the full IF function syntax:

IF(logical_test, value_if_true, value_if_false)

The extra value_if_false argument is the value returned by the function if logical_test evaluates to FALSE. Here’s a modification of the sales rep bonus calculation that takes a FALSE result into account:

=IF(B2 > 100000, B2 * 0.1, 0)

It’s often necessary to perform an action if and only if two conditions are true. For example, you might want to pay a salesperson a bonus if and only if dollar sales exceed a certain amount and unit sales also exceed some minimum value. If the dollar sales, the unit sales, or both fall below the minimum, no bonus is paid. In Boolean logic, this is called an And condition because one expression and another must be true for a positive result.
In Excel, And conditions are handled by the AND logical function:

\[
\text{AND}(\text{logical1}, \text{logical2}, \ldots)
\]

Here, \textit{logical1} and \textit{logical2} arguments are the logical conditions to test. (The ellipsis means that you can enter as many conditions as you need.) The AND result is calculated as follows:

- If all the arguments return TRUE (or any nonzero number), AND returns TRUE.
- If one or more of the arguments return FALSE (or 0), AND returns FALSE.

Consider a formula that uses AND to check whether a sales rep's sales total is greater than $100,000 and units total is greater than 10,000. That formula could look like this:

\[
=\text{AND}(B2 > 100000, C2 > 10000)
\]

A formula that uses the AND function to test whether the values in columns B and C exceed a specified minimum.

In many worksheet models, you need to take an action if one thing or another is true. For example, you might want to pay a salesperson a bonus if she exceeds the dollar sales budget \textit{or} if she exceeds the unit sales budget. In Boolean logic, this is called an Or condition.

In Excel, Or conditions are handled by the OR function:

\[
\text{OR}(\text{logical1}, \text{logical2}, \ldots)
\]

Here, \textit{logical1} and \textit{logical2} arguments are the logical conditions to test. (The ellipsis means that you can enter as many conditions as you need.) The OR result is calculated as follows:

- If one or more of the arguments return TRUE (or any nonzero number), OR returns TRUE.
- If all of the arguments return FALSE (or 0), OR returns FALSE.
Consider the following formula, which uses OR to check whether a sales rep’s sales total is greater than $100,000 or the units total is greater than 10,000:

\[=\text{OR}(B2 > 100000, C2 > 10000)\]

You sometimes need to return the opposite of a logical expression. For example, you might want to set up a worksheet that looks for the sales reps who did not make their sales quota and then flag those reps for further training. If you have a worksheet that already returns a TRUE or FALSE value based on whether a sales rep meets his quota, flagging whether the sales rep needs more training is a matter of returning the opposite value: if a sales rep meets her quota (TRUE), she doesn’t need training (FALSE); if a sales rep doesn’t meet his quota (FALSE), he does need training (TRUE).

In Excel, you return the opposite of a logical expression by using the NOT function:

\[\text{NOT}(\text{logical})\]

Here, the \text{logical} argument is the logical condition to test. The NOT result is calculated as follows:

- If \text{logical} is TRUE (or any nonzero number), NOT returns FALSE.
- If \text{logical} is FALSE (or 0), NOT returns TRUE.

The following formula uses NOT in the Training Required? column (E) to return the opposite of the logical value in the Bonus column (D):

\[=\text{NOT}(D2)\]
To insert a logical function into a formula

- On the Formulas tab, in the Function Library group, click Logical, click one of the functions in the list (such as IF, AND, OR, or NOT), and then enter the arguments, if any, as described earlier in this topic.

Perform logical operations by using nested functions

You can create sophisticated logical tests by combining one or more logical functions within a single expression. In particular, you can create complex expressions by including one logical function within another. This is called nesting, and the inner function is called a nested function.

A good time to use nested IF functions arises when you need to calculate a tiered payment or charge. That is, if a certain value is X, you want one result; if the value is Y, you want a second result; and if the value is Z, you want a third result. For example, suppose you want to calculate tiered bonuses for a sales team as follows:

- If the salesperson did not meet the sales target, no bonus is given.
- If the salesperson exceeded the sales target by less than 10 percent, a bonus of $1,000 is awarded.
- If the salesperson exceeded the sales target by 10 percent or more, a bonus of $5,000 is awarded.

Assuming that cell D2 contains the percentage that each salesperson’s actual sales were above or below her target sales, here’s a formula that handles these rules:

`=IF(D2 < 0, "", IF(D2 < 0.1, 1000, 5000))`
Objective group 3  Create advanced formulas and macros

A formula that uses nested IF functions to calculate a tiered bonus.

To nest logical functions

1. Insert the first logical function into your formula.
2. Enter the function arguments as described earlier in this topic, including the logical function you want to nest as one of the arguments.

Perform multiple logical tests with the IFS function

Nesting one IF function inside another is a handy way to perform a couple of logical tests, but the method quickly becomes unwieldy and difficult to decipher when the nesting goes three or more IF functions deep. If your data analysis requires more than two logical tests, you can make your worksheet model easier to read by turning to the IFS function:

$$IFS(logical\_test1, value\_if\_true1, [logical\_test2, value\_if\_true2,...])$$

The IFS function consists of a series of logical tests, each of which has an associated return value. IFS performs each logical test in turn, and when it comes across the first logical test to return TRUE, it returns that logical test’s associated value.

For example, here’s how you’d use IFS to calculate the tiered bonuses that I introduced in the previous section:

$$=IFS(D2 < 0, 0, D2 < 0.1, 1000, D2 >= 0.1, 5000)$$
Perform logical operations with criteria

You use the SUM, AVERAGE, and COUNT functions in Excel to return the total value, the mean value, and the number of values, respectively, within a specified range. These functions operate over the entire range, but it's often the case that you want the sum, the average, or the count of only those cells that meet some criterion. For example, you might want the total sales for just the customers from the United States. Even more sophisticated calculations are possible when you use multiple criteria. For example, you might want the total sales for those customers who are from the United States and who are based in Oregon.

These seem like they would require complex combinations of the IF, AND, and OR functions nested within the SUM, AVERAGE, or COUNT function. Fortunately, Excel includes three summary functions that you can use to specify multiple criteria without complex nesting: SUMIF, AVERAGEIF, and COUNTIF. In each case, the “IF” part of the function name implies that an IF function is built into each function, without being explicitly invoked.

The SUMIF function is similar to SUM, except that it sums only those cells in a range that meet a specified condition:

\[
\text{SUMIF}(\text{range, criteria[, sum_range]})
\]

Here, the range argument is the range of cells to use for the criteria; the criteria argument is the criteria, entered as text, that Excel applies to range to determine which cells to sum; sum_range is an optional argument that specifies the range from which the sum values are taken. Excel sums only those cells in sum_range that correspond to the cells in range and meet the criteria. If you omit sum_range, Excel uses range for the sum.

For the criteria argument, you generally use the following format:

"operator value"

Here, you replace operator with a comparison operator such as equal to (=) or greater than (>), and you replace value with a comparison value. Many Excel functions also support text-only criteria arguments that optionally use one or more wildcard characters: * to match any number of characters and ? to match any single character.
For example, the following formula sums only those values in the range B3:B21 that are greater than 100000:

\[ =\text{SUMIF}(B3:B21, \ "> 100000") \]

A formula that uses the \textit{SUMIF} function.

The \textit{COUNTIF} function counts the number of cells in a range that meet a single condition:

\[ \text{COUNTIF}(\text{range}, \ \text{criteria}) \]

Here, replace \textit{range} with the range of cells to use for the count; replace \textit{criteria} with the criteria, entered as text, that Excel applies to \textit{range} to determine which cells to count.

For example, you can use the following formula to count the number of cells in the range D3:D21 that are greater than or equal to 0.1:

\[ =\text{COUNTIF}(D3:D21, \ "\geq 0.1") \]
Objective 3.1: Perform logical operations in formulas

A formula that uses the COUNTIF function.

The AVERAGEIF function calculates the average of a range set for those items that meet a specified condition:

\[
\text{AVERAGEIF(range, criteria, average_range)}
\]

Here, the \textit{range} argument is the range of cells to use for the criteria; the \textit{criteria} argument is the criteria, entered as text, that Excel applies to \textit{range} to determine which cells to average; \textit{average_range} is an optional argument that specifies the range from which the average values are taken. Excel averages only those cells in \textit{average_range} that correspond to the cells in \textit{range} and meet the \textit{criteria}. If you omit \textit{average_range}, Excel uses \textit{range} for the average.

For example, the following formula averages only those values in the range D3:D21 that correspond to those values in the range C3:C21 that are greater than or equal to 10,000:

\[
=\text{AVERAGEIF(C2:C21, "> 10000", D3:D21)}
\]
To insert a SUMIF function into a formula

➤ On the Formulas tab, in the Function Library group, click Math & Trig, click SUMIF, and then enter the arguments, as described earlier in this topic.

To insert an AVERAGEIF or COUNTIF function into a formula

➤ On the Formulas tab, in the Function Library group, click More Functions, click Statistical, click AVERAGEIF or COUNTIF, and then enter the arguments, as described earlier in this topic.

Perform logical operations with multiple ranges and criteria

The SUMIF, COUNTIF, and AVERAGEIF functions that I describe in the previous section replace simple nested functions with a single condition. However, there are many situations in which your logical formula requires multiple criteria applied to multiple ranges. To handle these situations, Excel includes five summary functions that you can use to specify multiple criteria: SUMIFS, AVERAGEIFS, COUNTIFS, MAXIFS, and MINIFS. In each case, the “IFS” part of the function name implies that multiple IF functions are built into each function, without being explicitly invoked.
Objective 3.1: Perform logical operations in formulas

The SUMIFS function sums cells in a range that correspond to those cells in one or more ranges that meet one or more criteria:

\[ \text{SUMIFS(sum_range, range1, criteria1[, range2, criteria2, ...])} \]

The sum_range argument is the range from which the sum values are taken. Excel sums only those cells in sum_range that correspond to the cells that meet the criteria. The range1 argument is the first range of cells to use for the sum criteria, and the criteria1 argument is the first criterion, entered as text, that determines which cells to sum. Excel applies the criterion to range1. You can enter up to 127 range/criterion pairs.

Consider the following SUMIFS function example. In this model, SUMIFS is used to calculate the total bonuses (range E3:E21) just for those sales reps whose total units (range C3:C21) are greater than 10,000:

\[ =\text{SUMIFS(E3:E21, C3:C21, " > 10000")} \]

The AVERAGEIFS function averages cells in a range that correspond to those cells in one or more ranges that meet one or more criteria:

\[ \text{AVERAGEIFS(average_range, range1, criteria1[, range2, criteria2, ...])} \]

A formula that uses SUMIFS to apply a criterion to a sum.

The AVERAGEIFS function averages cells in a range that correspond to those cells in one or more ranges that meet one or more criteria:
Objective group 3  Create advanced formulas and macros

The *average_range* argument is the range from which the average values are taken. Excel averages only those cells in *average_range* that correspond to the cells that meet the criteria. The *range1* argument is the first range of cells to use for the average criteria, and the *criteria1* argument is the first criterion, entered as text, that determines which cells to average. Excel applies the criterion to *range1*. You can enter up to 127 range/criterion pairs.

In the following example, AVERAGEIFS is used to calculate the average sales (range B3:B21) just for those sales reps whose total units (C3:C21) are greater than 10,000 and whose return rate (D3:D21) is less than 10 percent:

\[=AVERAGEIFS(B3:B21, C3:C21, " > 10000", D3:D21, " < 0.1")\]

A formula that uses AVERAGEIFS to apply multiple criteria to an average.

The COUNTIFS function counts the number of cells in one or more ranges that meet one or more criteria:

\[\text{COUNTIFS(range1, criteria1[, range2, criteria2, ...])}\]

The *range1* argument is the first range of cells to use for the count, and the *criteria1* argument is the first criterion, entered as text, that determines which cells to count. Excel applies the criterion to *range1*. You can enter up to 127 range/criterion pairs.
In the following example, COUNTIFS is used to count the number of sales reps whose dollar sales (range B3:B21) are greater than $100,000, whose unit sales (C3:C21) are greater than 10,000, and whose return rate (D3:D21) is less than 10 percent:

\[=\text{COUNTIFS}(B3:B21, " > 100000", C3:C21, " > 10000", D3:D21, " < 0.1")\]

The MAXIFS function returns the maximum value from a range that corresponds to those cells in one or more ranges that meet one or more criteria:

\[\text{MAXIFS}(\text{max\_range}, \text{range1}, \text{criteria1},[, \text{range2}, \text{criteria2}, ...])\]

The \text{max\_range} argument is the range from which the maximum value is taken. Excel looks for the maximum only in those cells in \text{max\_range} that correspond to the cells that meet the criteria. The \text{range1} argument is the first range of cells to use for the maximum criteria, and the \text{criteria1} argument is the first criterion, entered as text, that determines which cells to include. Excel applies the criterion to \text{range1}. You can enter up to 127 range/criterion pairs.

Consider the following MAXIFS function example. In this model, MAXIFS is used to calculate the maximum return rate (from the range D3:D21) just for those sales reps
whose total sales (range B3:B21) are greater than $100,000 and whose total units (range C3:C21) are greater than 10,000:

\[ =\text{MAXIFS}(D3:D21, B3:B21, " > 100000", C3:C21, " > 10000") \]

A formula that uses MAXIFS to apply a criterion to a maximum.

The MINIFS function returns the minimum value from a range that corresponds to those cells in one or more ranges that meet one or more criteria:

\[ \text{MINIFS}(\text{min\_range}, \text{range1}, \text{criteria1}[, \text{range2}, \text{criteria2}, ...]) \]

The \text{min\_range} argument is the range from which the minimum value is taken. Excel looks for the minimum only in those cells in \text{min\_range} that correspond to the cells that meet the criteria. The \text{range1} argument is the first range of cells to use for the minimum criteria, and the \text{criteria1} argument is the first criterion, entered as text, that determines which cells to include. Excel applies the criterion to \text{range1}. You can enter up to 127 range/criterion pairs.

Consider the following MINIFS function example. In this model, MINIFS is used to calculate the minimum units (from the range C3:C21) just for those sales reps whose total
Objective 3.1: Perform logical operations in formulas

sales (range B3:B21) are greater than $100,000 and whose return rate (range D3:D21) are less than 10 percent:

=MINIFS(C3:C21, B3:B21, " > 100000", D3:D21, " < 0.1")

To insert a SUMIFS function into a formula

- On the Formulas tab, in the Function Library group, click Math & Trig, click SUMIFS, and then enter the arguments, as described earlier in this topic.

To insert an AVERAGEIFS, COUNTIFS, MAXIFS, or MINIFS function into a formula

- On the Formulas tab, in the Function Library group, click More Functions, click Statistical, click the function—AVERAGEIFS, COUNTIFS, MAXIFS, or MINIFS,—and then enter the arguments, as described earlier in this topic.

Apply multiple exact-match criteria to a value

The IFS function that I describe earlier in this objective enables you to create sophisticated criteria using comparison operators such as less than (<) and greater than or equal to (>=). However, it’s often the case that your comparisons require only
Objective group 3  Create advanced formulas and macros

exact matches using the equal operator (=). In such cases, it is often easier to use the SWITCH function rather than IFS.

The SWITCH function evaluates an expression, compares the result to a list of values that you supply, and then returns a value that corresponds to the first match that Excel finds:

\[
\text{SWITCH} \left( \text{expression}, \text{value}_1, \text{value}_2, ..., \text{value}_n \right)
\]

SWITCH first evaluates \( \text{expression} \). SWITCH then compares that result to each \( \text{value}_i \) argument; when SWITCH comes across the first match, it returns the corresponding \( \text{value}_i \); if no match is found, SWITCH returns the \( \text{value}_n \) argument. You can enter up to 126 match/return value pairs.

For example, given a date, the MONTH function returns a number between 1 (for January) and 12 (for December). For a date in cell A2, you can use the following SWITCH formula to return the date's month name based on the results of the MONTH(A2) expression:

\[
\]

A formula that uses SWITCH to apply multiple exact-match criteria to an expression.

To insert a SWITCH function into a formula

➜ On the Formulas tab, in the Function Library group, click Logical, click SWITCH, and then enter the arguments, as described earlier in this topic.
Objective 3.1 practice tasks

The practice file for these tasks is located in the MOSExcelExpert2019 \Objective3 practice file folder. The folder also contains a result file that you can use to check your work.

➤ Open the ExcelExpert_3-1 workbook and do the following:

- On the Gross Margin worksheet, add formulas to the Gross Margin field (H7:H14) that calculate the gross margin by subtracting Cost from Retail and then dividing by Cost. To avoid #DIV/0! (division by 0) errors, wrap the calculation inside an IF function that returns the gross margin if Cost is not 0, or the message Cost is 0! otherwise.

- On the Inventory worksheet, populate the Reorder Now? column with formulas that use nested logical functions to determine whether a product should be reordered based on two conditions: The Qty Available is less than or equal to the Reorder Level, and the Qty on Order is 0. Each function should return “Yes” if a reorder is required, and nothing otherwise.

- On the Inventory worksheet, create a formula in cell K1 that uses the SUMIF function to sum the Value range for products with a non-zero Qty Available value.

- On the Inventory worksheet, create a formula in cell K2 that uses the SUMIFS function to sum the Qty On Hand range for products with a Product Name value that includes Soup and a Qty On Hold value of zero. (Hint: To match cells that include some text, surround that text with the * wildcard character.)

- On the Parts worksheet, create a formula in cell F16 that uses the AVERAGEIF function to calculate the average of the Gross Margin values for the parts that cost less than $10. Use structured table references in your formula (the table name is Parts).
- On the **Customers** worksheet, create a formula in cell L1 that uses the COUNTIFS function to return the number of customers with the Country value *United States* and the Region value *OR* (the abbreviation for *Oregon*, not the OR function).

- On the **Orders** worksheet, populate the Weekday column with formulas that use the SWITCH function to return the weekday of each date in the Date column.

➤ Save the workbook.

➤ Open the **ExcelExpert_3-1_results** workbook. Compare the two workbooks to check your work, and then close the open workbooks.
Objective 3.2: Look up data by using functions

The table—more properly referred to as a lookup table—is the key to performing lookup operations in Excel. The most straightforward lookup table structure is one that consists of two columns (or two rows):

- **Lookup column** This column contains the values that you look up. For example, if you were constructing a lookup table for a dictionary, this column would contain the words.
- **Data column** This column contains the data associated with each lookup value. In the dictionary example, this column would contain the definitions.

In most lookup operations, you supply a value that the function locates in the designated lookup column. It then retrieves the corresponding value in the data column.

The lookup table theme has many variations. The lookup table can be one of these:

- **A single column or row** In this case, the lookup operation consists of finding the nth value in the column.
- **A range with multiple data columns** For instance, in the dictionary example, you might have a second column for each word’s part of speech (noun or verb, for example), and perhaps a third column for its pronunciation. In this case, the lookup operation must also specify which of the data columns contains the value required.
- **An array** In this case, the table doesn’t exist on a worksheet but is either an array of literal values or the result of a function that returns an array. The lookup operation finds a particular position within the array and returns the data value at that position.

The VLOOKUP function works by looking in the first column of a table for the value you specify. (The V in VLOOKUP stands for vertical.) It then looks across to the column that you specify and returns whatever value it finds there.

Here’s the full syntax for VLOOKUP:

\[
\text{VLOOKUP(lookup\_value, table\_array, col\_index\_num, range\_lookup)}
\]
The following table describes the VLOOKUP function arguments.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lookup_value</td>
<td>The value you want to find in the first column of table_array. You can enter a number, string, or reference.</td>
</tr>
<tr>
<td>table_array</td>
<td>The cell range or named table to use for the lookup.</td>
</tr>
<tr>
<td>col_index_num</td>
<td>The column number in table_array that contains the data you want the formula to return (the first column—that is, the lookup column—is 1, the second column is 2, and so on).</td>
</tr>
<tr>
<td>range_lookup</td>
<td>A Boolean value that determines how Excel searches for lookup_value in the first column. If FALSE, VLOOKUP returns the first exact match for lookup_value. If TRUE (the default), if no exact match is found, the function returns the largest value that is less than lookup_value.</td>
</tr>
</tbody>
</table>

Here are some notes to keep in mind when you work with VLOOKUP:

- If range_lookup is TRUE or omitted, you must sort the values in the first column in ascending order.
- If the first column of the table is text, you can use the standard wildcard characters in the lookup_value argument (use ? to substitute for individual characters; use * to substitute for multiple characters).
- If lookup_value is less than any value in the lookup column, VLOOKUP returns the #N/A error value.
- If VLOOKUP doesn’t find a match in the lookup column, it returns #N/A.
- If col_index_num is less than 1, VLOOKUP returns #VALUE!; if col_index_num is greater than the number of columns in table_array, VLOOKUP returns #REF!.

As an example, assume that a worksheet uses a VLOOKUP formula in cell B4 to take the account number entered in cell B2, locate the exact match in the first column of the range D3:E15, and return the account name from the second column in that range. The formula would look like this:

=VLOOKUP(B2, D3:E15, 2, FALSE)

The HLOOKUP function is similar to VLOOKUP except that it searches for the lookup value in the first row of a table. (The H in HLOOKUP stands for horizontal.) If successful, this function then looks down to the specified row and returns the value it finds there. Here’s the syntax for HLOOKUP:

HLOOKUP(lookup_value, table_array, row_index_num[, range_lookup])
A formula that uses VLOOKUP to look up an account name given an account number.

The following table describes the HLOOKUP function arguments.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lookup_value</td>
<td>The value you want to find in the first row of table_array. You can enter a number, string, or reference.</td>
</tr>
<tr>
<td>table_array</td>
<td>The cell range or named table to use for the lookup.</td>
</tr>
<tr>
<td>row_index_num</td>
<td>The row number in the table that contains the data you want the formula to return (the first row—that is, the lookup row—is 1, the second row is 2, and so on).</td>
</tr>
<tr>
<td>range_lookup</td>
<td>A Boolean value that determines how Excel searches for lookup_value in the first row. If FALSE, HLOOKUP returns the first exact match for lookup_value. If TRUE (the default), if no exact match is found, the function returns the largest value that is less than lookup_value.</td>
</tr>
</tbody>
</table>

As an example, assume that a worksheet uses an HLOOKUP formula in cell C10 to take the month entered in cell C9, locate the exact match in the first row of the range C1:N7, and return the value from the TOTAL row (the seventh row in that range).

The formula would look like this:

`=HLOOKUP(C9, C1:N7, 7, FALSE)"
Objective group 3  Create advanced formulas and macros

A formula that uses HLOOKUP to look up a monthly expenses total given a month name.

The basic lookup procedure—finding a value in a column or row and then returning an offset value—will satisfy most of your needs. However, a few operations require a more sophisticated approach that makes use of two more lookup functions: MATCH and INDEX.

The MATCH function looks through a row or column of cells for a value. If MATCH finds that value, it returns the relative position of the match in the row or column. Here’s the syntax:

\[ \text{MATCH(lookup\_value, lookup\_array[, match\_type])} \]

The following table describes the MATCH function arguments.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lookup_value</td>
<td>The value you want to find. You can use a number, string, reference, or logical value.</td>
</tr>
<tr>
<td>lookup_array</td>
<td>The row or column of cells you want to use for the lookup.</td>
</tr>
</tbody>
</table>
| match\_type  | How you want Excel to match the lookup\_value with the entries in the lookup\_array. You have three choices:  
  - 0 finds the first value that exactly matches lookup\_value. The lookup\_array can be in any order.  
  - 1 finds the largest value that is less than or equal to lookup\_value (this is the default value). The lookup\_array must be in ascending order.  
  - −1 finds the smallest value that is greater than or equal to lookup\_value. The lookup\_array must be in descending order. |
Objective 3.2: Look up data by using functions

Tip You can use wildcard characters within the lookup_value argument (provided that match_type is 0 and lookup_value is text). You can use the question mark (?) for single characters and the asterisk (*) for multiple characters.

Normally, you don’t use the MATCH function by itself; you combine it with the INDEX function. INDEX returns the value of a cell at the intersection of a row and column inside a reference. Here’s the syntax for INDEX:

\[
\text{INDEX}(\text{reference}, \text{row\_num},[, \text{column\_num}],[, \text{area\_num}])
\]

The following table describes the INDEX function arguments.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>A reference to one or more cell ranges.</td>
</tr>
<tr>
<td>row_num</td>
<td>The number of the row in reference from which to return a value.</td>
</tr>
<tr>
<td>column_num</td>
<td>The number of the column in reference from which to return a value.</td>
</tr>
<tr>
<td></td>
<td>You can omit column_num if reference is a single column.</td>
</tr>
<tr>
<td>area_num</td>
<td>If you entered more than one range for reference, area_num is the range you want to use. The first range you entered is 1 (this is the default), the second is 2, and so on.</td>
</tr>
</tbody>
</table>

The idea is that you use MATCH to get row_num and/or column_num (depending on how your table is laid out) and then use INDEX to return the value you need.

For example, consider a worksheet that uses a MATCH and INDEX formula in cell B2 to take the part number entered in cell B1, locate the match in the range H6:H13, and return the quantity from the range C6:C13.

The formula would look like this:

\[
=\text{INDEX}(C6:C13, \text{MATCH}(B1, H6:H13, 0))
\]
To insert a VLOOKUP or an HLOOKUP function into a formula

- On the Formulas tab, in the Function Library group, click Lookup & Reference, click VLOOKUP or HLOOKUP, and then enter the arguments, as described earlier in this topic.

To insert the MATCH and INDEX functions into a formula

1. On the Formulas tab, in the Function Library group, click Lookup & Reference, and then click INDEX.

2. In the Select Arguments dialog box, click reference, row_num, column_num, area_num, then click OK.

3. In the Function Arguments dialog box, enter the INDEX function arguments, including a MATCH function for the row_num and/or column_num arguments, and then click OK.
Objective 3.2 practice tasks

The practice file for these tasks is located in the MOSExcelExpert2019 \Objective3 practice file folder. The folder also contains a result file that you can use to check your work.

➤ Open the ExcelExpert_3-2 workbook and do the following:

❑ On the Tax Rate worksheet, add a formula to cell B18 that uses the tax table in the range C9:F15 to look up the income entered in cell B17 and return the applicable tax rate.

❑ On the Discount Schedule worksheet, create formulas in the range D3:D10 that use the discount schedule in the range B13:G14 to look up the units ordered from the range A3:A10 and return the applicable discount percentage.

❑ On the Parts worksheet, add a formula to cell B3 that uses the range A7:H14 to look up the part number entered in cell B1 and then return the corresponding value from the field entered in cell B2.

➤ Save the workbook.

➤ Open the ExcelExpert_3-2_results workbook. Compare the two workbooks to check your work, and then close the open workbooks.
Index

SYMBOLS

* (asterisk) in number formats, 47
@ (at sign) in number formats, 47
\ (backslash) in number formats, 47
: (colon) in number formats, 46
, (comma) in number formats, 46
$ (dollar sign) in number formats, 46
- (minus sign) in number formats, 46
# (number sign) in number formats, 46
() (parentheses) in number formats, 46
% (percent sign) in number formats, 46
. (period) in number formats, 46
+ (plus sign) in number formats, 46
? (question mark) in number formats, 46
" " (quotation marks) in number formats, 47
/ (slash) in number formats, 46
[] (square brackets) in number formats, 47
_ (underscore) in number formats, 47
0 (zero) in number formats, 46

A

activating. See enabling
adding
  authoring languages, 31–32
  calculated fields, 159–160
  calculated items, 160
  scenarios, 113
  text in another language, 34
  watches, 122
AND function, 73, 74–75
applying PivotChart styles, 167–168
asterisk (*) in number formats, 47
at sign (@) in number formats, 47
authoring languages
  adding, 31–32
  removing, 33
automatic subtotals, inserting, 56–58
AutoRecover
  configuring, 9–10
  previous versions, reverting to, 10
AutoSave
  enabling, 11
  previous versions, reverting to, 11–12
AVERAGE function, 79, 157
AVERAGEIF function, 81–82
AVERAGEIFS function, 83–84

B

backslash (\) in number formats, 47
Box & Whisker charts, creating, 136–137

C

calculated fields in PivotTables, 157–160
calculated items, 159
  adding, 160
calculation options, configuring, 20–25
cells
  filling
    with Fill Series options, 41–42
    with Flash Fill, 38–40
  locking, 16
  unlocking, 16
changing cells
  in Goal Seek, 110
  in Scenario Manager, 113
charts, creating. See also PivotCharts
  Box & Whisker, 136–137
dual-axis, 134–135
  Funnel, 137–138
  Histogram, 138–139
  Map, 139
  Pareto, 138–139
  Sunburst, 140–141
  Treemap, 141–142
  Waterfall, 142
client workbooks, 6
collapsing PivotCharts, 168
colon (:) in number formats, 46
color scales, 64
colors in number formats, 47
comma (,) in number formats, 46
command macros. See macros
comments. See also threads (comments)
deleting, 28
editing, 27
managing, 25–28
reading, 28
responding to, 28
comparison operators, 73
conditional formatting rules
creating custom, 65–66
with formulas, 66–67
managing, 67–68
types of, 62–65
configuring
AutoRecover, 9–10
calculation options, 20–25
data validation, 50–53
language options, 31–33
preferred languages, 33
consolidating data, 106–109
copying macros between workbooks, 2–4
COUNT function, 79, 157
Count Numbers function, 158
COUNTIF function, 80–81
COUNTIFS function, 84–85
criteria
exact-match, 87–88
logical operations with, 79–87
CUMIPMT function, 115
CUMPRINC function, 115
current date/time, referencing, 98–99

data analysis. See also PivotTables
consolidation, 106–109
financial functions, 114–116
NPER, 116–117
PMT, 117–118
what-if analysis, 109
with Goal Seek, 109–111
with Scenario Manager, 111–114
data bars, 63–64
data fields, 144
data groups, 53–56
data validation, configuring, 50–53
date data, grouping in PivotTables, 156–157
date formats, creating custom, 47–48
DATE function, 100
date functions
referencing current date, 98–99
serial numbers for, 98, 100–103
DAY function, 100–101
DB function, 115
DDB function, 115
deleting. See also removing
comment threads, 28
comments, 28
conditional formatting rules, 68
macros, 131
number formats, 49–50
dependence, tracing, 120–121
dependency trees, 21–22
dependent workbooks, 6
destination ranges, 106
Developer tab, viewing, 3
display languages
installing, 32
priority order, changing, 32–33
displaying. See viewing
dollar sign ($) in number formats, 46
drilling down PivotCharts, 168
dual-axis charts, creating, 134–135
duplicate records, removing, 58–59
E
E+/e+/E-/e- in number formats, 46
editing
comments, 27
conditional formatting rules, 67–68
macros, 130
PivotCharts, 166
PivotTables, 148–151
restricting
formatting protection, 15–16
range passwords, 17–19
text in another language, 34
EFFECT function, 115
embedding PivotCharts, 165
enabling
AutoSave, 11
error indicators, 123
macros, 4–6
worksheet protection, 18–19
error indicators, 123
error-checking rules, 123–126
Evaluate Formula, 126
Evaluating formulas, 126
Exact-match criteria, logical operations with, 87–88
External data sources, creating PivotTables from, 147–148
External references, creating, 6–9

F
Fields (PivotTables), moving, 148, 151
Fill Series options, 41–42
Filled Map charts, creating, 139
Filling cells
With Fill Series options, 41–42
With Flash Fill, 38–40
Filtering
PivotTable data, 149–150, 151
With slicers, 151–153
Financial functions, 114–116
NPER, 116–117
PMT, 117–118
Flash Fill, 38–40
Format Cells dialog box, opening, 48
Formatting
PivotChart styles, 167–168
PivotTable data, 160–161
Formulas
Calculation options, configuring, 20–25
Conditional formatting rules with, 66–67
Evaluating, 126
Functions, inserting, 72
Hiding, 16
Troubleshooting, 120
Error-checking rules, 123–126
With Evaluate Formula, 126
Tracing precedence and dependence, 120–121
With Watch Window, 121–123
Validating, 123–126
Viewing, 16
Frequency distribution, 138
Functions
date/time
Referencing current date/time, 98–99
Serial numbers for, 98, 100–104
Financial, 114–116
NPER, 116–117
PMT, 117–118
In formulas, inserting, 72
Logical, 72–77
Lookup, 91–96
Nested, 77–78
Summary in PivotTables, 157–158
Volatile, 21
Funnel charts, creating, 137–138
Future value, 114
FV function, 115

G
GETPIVOTDATA function, 158
Goal Seek, 109–111
Grouping
data, 53–56
PivotTable data, 153–157

H
Hiding formulas, 16
Highlight cells rules, 62
Histogram charts, creating, 138–139
HLOOKUP function, 92–94
HOUR function, 103–104

I
Icon sets, 65
IF function, 73, 74
IFS function, 78
INDEX function, 95–96
Inserting
Functions in formulas, 72
Subtotals and totals, 56–58
Installing
Display languages, 32
Language packs, 32
IPMT function, 115
IRR function, 115
Iterative calculations, 22, 25

L
Language options
Configuring, 31–33
Usage examples, 33–35
Language packs, installing, 32
linking workbooks, 6–9
links
  creating, 7–8
  updating, 8–9
locking cells, 16
logical expressions, comparison operators in, 73
logical functions, 72–77
logical operations
  with criteria, 79–82
  with logical functions, 72–77
  with multiple exact-match criteria, 87–88
  with multiple ranges and criteria, 82–87
  multiple tests with, 78
  with nested functions, 77–78
lookup functions, 91–96
lookup tables, 91

M
Macro Recorder, 128
macros, 128–131
  copying between workbooks, 2–4
  deleting, 131
  editing, 130
  enabling, 4–6
  recording, 129–130
  running, 131
managing
  comments, 25–28
  conditional formatting rules, 67–68
  workbook versions, 9–12
manual recalculations, 21–22, 23–24
Map charts, creating, 139
MATCH function, 94–96
MAX function, 158
MAXIFS function, 85–86
MIN function, 158
MINIFS function, 86–87
minus sign (-) in number formats, 46
MINUTE function, 103–104
MIRR function, 115
modifying. See editing
MONTH function, 100–101
moving PivotTable fields, 148, 151
multiple criteria/ranges, logical operations
  with, 82–87
  multiple logical tests, 78
multithreaded calculations, 24

N
nested functions, 77–78
NETWORKDAYS function, 102–103
NOMINAL function, 116
NOT function, 73, 76–77
notes, 26
NOW function, 99
NPER function, 116, 117
NPV function, 116
number formats
  creating, 45–50
  deleting, 49–50
  number sign (#) in number formats, 46
numeric data, grouping in PivotTables, 155–156

O
OneDrive files, enabling AutoSave, 11
opening
  Format Cells dialog box, 48
  Record Macro dialog box, 128–129
OR function, 73, 75–76
outlines
  creating, 53–55
  removing, 55–56

P
parentheses (()) in number formats, 46
Pareto charts, creating, 138–139
passwords for ranges, 17–19
payments, 114
percent sign (%) in number formats, 46
period (.) in number formats, 46
periods, 114
Personal Macro Workbook, 2
  creating, 4
  viewing, 2
PivotCharts, 164
  collapsing, 168
  creating, 164–165
  drilling down, 168
  editing, 166
  embedding, 165
  styles, applying, 167–168
pivoting PivotTable data, 148, 151
PivotTables, 144
  calculated fields, 157–160
creating, 145–148
editing, 148–151
filtering data, 149–150, 151
formatting data, 160–161
grouping data, 153–157
pivoting data, 148, 151
slicers, creating, 151–153
sorting data, 148–149, 151
plus sign (+) in number formats, 46
PMT function, 116, 117–118
PPMT function, 116
precedence, tracing, 120–121
preferred languages, configuring, 33
present value, 114
priority order of display languages, changing, 32–33
PRODUCT function, 158
Project Explorer, viewing, 3
proofing text in another language, 34
protecting
workbook structure, 19–20
worksheets
enabling protection, 18–19
with formatting, 15–16
range passwords, 17–19
reopening comment threads, 28
reordering conditional formatting rules, 68
resolving comment threads, 28
responding to comments, 28
restricting editing
formatting protection, 15–16
range passwords, 17–19
reverting to previous versions
with AutoRecover, 10
with AutoSave, 11–12
row fields, 144
rules. See conditional formatting rules
running macros, 131
Scenario Manager, 111–114
scenarios, 111–112
adding, 113
viewing, 113–114
SECOND function, 103–104
secondary axis, 134
serial numbers, 98, 100–104
Series command, 41–42
server workbooks, 6
shortcut keys, 129
slash (/) in number formats, 46
slicers, creating, 151–153
SLN function, 116
sorting PivotTable data, 148–149, 151
source ranges, 106
source workbooks, 6
square brackets ([]) in number formats, 47
STDDEV function, 158
STDEV function, 158
structure of workbooks, protecting, 19–20
styles in PivotCharts, applying, 167–168
subtotals, inserting, 56–58
SUM function, 79, 157
SUMIF function, 79–80
SUMIFS function, 83
summary calculations
changing, 58, 159
in PivotTables, 157–158
Sunburst charts, creating, 140–141
SWITCH function, 87–88
SYD function, 116

Q

question mark (?) in number formats, 46
quotation marks (""") in number formats, 47
R

ranges
creating PivotCharts from, 165
creating PivotTables from, 146–147
logical operations with, 82–87
passwords, 17–19
RANGE function, 116
reading comments, 28
Record Macro dialog box, opening, 128–129
recording macros, 129–130
referencing current date/time, 98–99
removing. See also deleting
authoring languages, 33
duplicate records, 58–59
outlines, 55–56
watches, 123
reopening comment threads, 28
reordering conditional formatting rules, 68
resolving comment threads, 28
responding to comments, 28
restricting editing
formatting protection, 15–16
range passwords, 17–19
reverting to previous versions
with AutoRecover, 10
with AutoSave, 11–12
row fields, 144
rules. See conditional formatting rules
running macros, 131
Scenario Manager, 111–114
scenarios, 111–112
adding, 113
viewing, 113–114
SECOND function, 103–104
secondary axis, 134
serial numbers, 98, 100–104
Series command, 41–42
server workbooks, 6
shortcut keys, 129
slash (/) in number formats, 46
slicers, creating, 151–153
SLN function, 116
sorting PivotTable data, 148–149, 151
source ranges, 106
source workbooks, 6
square brackets ([]) in number formats, 47
STDDEV function, 158
STDEV function, 158
structure of workbooks, protecting, 19–20
styles in PivotCharts, applying, 167–168
subtotals, inserting, 56–58
SUM function, 79, 157
SUMIF function, 79–80
SUMIFS function, 83
summary calculations
changing, 58, 159
in PivotTables, 157–158
Sunburst charts, creating, 140–141
SWITCH function, 87–88
SYD function, 116
tables

creating PivotCharts from, 165
creating PivotTables from, 146–147
text
adding in another language, 34
inge in another language, 34
proofing in another language, 34
translating, 34–35
text data, grouping in PivotTables, 157
threads (comments), 26. See also comments
creating, 26
deleting, 28
navigating, 26–27
reopening, 28
resolving, 28
time data, grouping in PivotTables, 156–157
time formats, creating custom, 47–48
TIME function, 103
time functions
referencing current time, 98–99
serial numbers for, 98, 103–104
timeline slicers, 152, 153
TODAY function, 98–99
top/bottom rules, 63
totals, inserting, 56–58
tracing precedence/dependence, 120–121
Translate command, 33
translating text, 34–35
Treemap charts, creating, 141–142
troubleshooting formulas, 120
error-checking rules, 123–126
with Evaluate Formula, 126
tracing precedence and
dependence, 120–121
with Watch Window, 121–123
two-digit years, 100

V
validating
data, 50–53
formulas, 123–126
VAR function, 158
VARP function, 158
versions of workbooks, managing, 9–12
viewing
Developer tab, 3
formulas, 16
Personal Macro Workbook, 2
Project Explorer, 3
scenarios, 113–114
VLOOKUP function, 91–93
volatile functions, 21

W
Watch Window, 121–123
watches
adding, 122
removing, 123
Waterfall charts, creating, 142
WEEKDAY function, 101–102
what-if analysis, 109
with Goal Seek, 109–111
with Scenario Manager, 111–114
workbooks
linking, 6–9
macros
copying between, 2–4
enabling, 4–6
structure, protecting, 19–20
versions, managing, 9–12
WORKDAY function, 102
worksheets, protecting
enabling protection, 18–19
with formatting, 15–16
range passwords, 17–19

Y
YEAR function, 100–101
years, two-digit, 100

Z
zero (0) in number formats, 46