The Software Pro®

Microsoft Excel:

Top Essential Functions

Learning Guide

Dawn Bjork, MCT, MOSM
The Software Pro®

Technology Speaker
Microsoft Certified Trainer
Certified Excel Expert
Microsoft Office Specialist Master

Dawn@TheSoftwarePro.com
TheSoftwarePro.com

Bonus Resources

As an extra bonus, additional Microsoft Excel resources, a keyboard shortcuts handout, and other helpful tips are waiting for you at:

TheSoftwarePro.com/Excel

The Software Pro®

Foundation of Excel Functions

Excel lets you easily perform powerful calculations using formulas and functions. More than 400 functions are built into Microsoft Excel for simple to complex calculations.

First, we'll quickly review some of the basics needed to build your foundation of knowledge and then we'll move on to explore more about other essential functions in Excel.

Creating Functions in Excel

The basic structure of a function is:

`=FunctionName(arguments)`

`Arguments` are the pieces that the function needs to complete the calculation.

To create a function in Excel, you can:

- Type the function directly into a cell.

- Use the **Insert Function** tool found under the **Formulas** tab or to the right of the formula bar.

- Work with the **Formula AutoComplete** feature to build a function. After you type an = (equal sign) and beginning letters or a display trigger, Microsoft Excel displays a dynamic drop-down list of valid functions, range names, and text strings that match the letters or trigger.

- Choose a function from the **Functions Library** under the **Formulas** tab.

The Software Pro®
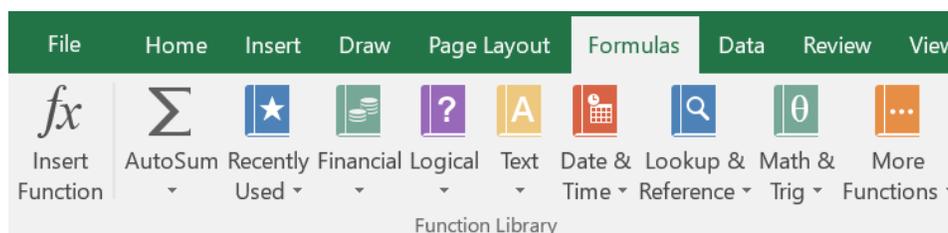
Review: The SUM Function

The most common function in Excel is **Sum** which automatically creates a total of a range or ranges of cells in a worksheet. The structure of the **SUM** function is:

`SUM(range)`

For example, **`=SUM(B2:B10)`**

To enter the SUM function directly:

1. Move to the location where you want to see the answer, such as cell **A5** (below).

2. Type **=sum**.

3. You will see a drop-down list of possible functions to match your entry.

   To pick one, highlight and press `Tab` to accept without entering the entire formula. Or, just keep typing to create the function on your own.

4. With your mouse, highlight the range you want to total. You can also type in the range which is **`first cell:last cell`** such as **`B4:B25`**.

5. Finish with a closed parentheses **)** or just press `Enter` and Excel will add the parentheses for you.

To enter the SUM function with AutoSum:

1. Select the **Sum** or **AutoSum** command button found in the **Editing** group of the **Home** tab or the **Function Library** in the **Formulas** tab.

Excel will automatically highlight the range of cells that it closest to the active location. Re-define the range to total if needed and then press `Enter` to add the **Sum** of your values.

To enter the SUM function with a keyboard shortcut:

1. Press `Alt` + =

AVERAGE, COUNT, MAX, MIN

The **AutoSum** button also includes other commonly used functions you can perform on consecutively filled cells in a column or row. In addition to **Sum**, these functions include **Average**, **Count**, **Max**, and **Min**.

AutoSum automatically uses the active cell for the suggested range. If this suggested range is incorrect, you can drag to select the cells containing the values you want to calculate.

To add other functions with the Sum button:

1. Select the cell into which you want to enter a formula.
2. Click the arrow on the **AutoSum** button.
3. Select the desired function.
4. If necessary, drag to select the range you want to calculate.
5. Press [Enter] to create the formula.
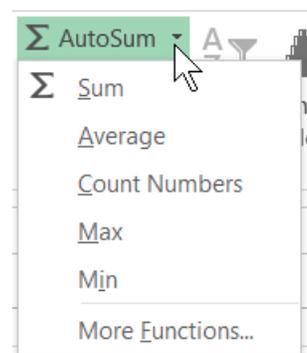
COUNT Functions

One of the options in the AutoSum drop-down is Count Numbers or `COUNT`. This function counts how many numbers are in a range. The `COUNTA` function counts how many alphanumeric entries are in a range.

`COUNT(range)`
Counts how many numbers are in the list of arguments; numeric

`COUNTA(range)`
Counts how many values are in the list of arguments; alphanumeric

ROUND Functions

When you select a popular number formatting option in Excel like currency or comma, Excel will round the displayed results to the specified number of decimal places such as 2 or 0. The obstacle is these formatting actions do not change or round the value behind the result to the same number of decimal places. Instead, Excel retains the full value as it is passed on to other formulas in the worksheet.

When a formula performs calculations, Excel usually uses the values stored in cells referenced by the formula. For example, if two cells each contain the value 100.005 and the cells are formatted to display values in currency format, the value $100.01 is displayed in each cell. If you add the two cells together, the result is $200.01 because Excel adds the stored values 100.005 and 100.005, not the displayed values.

The ROUND Function

To change your Excel formulas so the displayed answers match the values behind the results, add the ROUND function to your formulas. Using the ROUND function, you can calculate your values using the same number of decimal places you want to display in your worksheet.

The basic structure for the ROUND function is:

```
ROUND(number, num_digits)
```

The ROUND function can be easily applied to formulas with multiple expressions. Just add it as the outermost function to any formula, such as:

```
ROUND(S5*(H5+J5),2)
```

With Excel functions and calculations, the ROUND function may also be nested within other functions.

The number of decimal places in the ROUND function should always match the number of decimal places chosen for the formatting of the cell. Even though some formulas may create correct results without the ROUND function, be consistent so you can automatically rule out rounding problems when auditing worksheets.

ROUNDUP & ROUNDDOWN

The ROUNDUP function and its buddy ROUNDDOWN are specialized variations of the ROUND function. The assumption with ROUND is anything 5 or higher will round up to the next number.

As the names imply, the ROUNDUP function always rounds up and the ROUNDDOWN function always rounds down regardless of the value.

Create the ROUNDUP function with the formula structure or syntax of:

```
ROUNDUP(number, num_digits)
```

Create the ROUNDDOWN function with:

```
ROUNDDOWN(number, num_digits)
```

The Software Pro®

Understanding Cell References

When you are creating formulas in Excel, you may work with more than one type of cell reference, that is, how you refer to the cell address or location. In it's simplest expression, a cell address is just the label for the location, such as A1, which is the intersection of Column A and Row 1. In Excel there are 3 types of cell references:

- Relative
- Absolute
- Circular (which you DO NOT want to have in a worksheet)

Relative References

The default structure of a formula in Excel is that cell references <u>adjust</u> automatically when you copy a formula from one cell to another. This adjustment is known as *relative referencing*, that is, that the contents of a formula adjust relative (or based on) the new location of the formula.

Absolute References

When you don't want the cell references to change when you copy the formula to a different cell, you must designate one or more values to be constant. This is called *absolute referencing*.

You can create an *absolute reference* for a constant by placing a dollar sign before the parts of the reference that do not change. You can also designate a part a formula as absolute by highlighting it and pressing the F4 function key.

Here are some examples of absolute cell references:

- **$G$5**
- **B$3**
- **$B5**

Circular References

A circular reference occurs when a cell in an Excel worksheet refers to itself, whether directly or indirectly. For example, if `=100+A2` is entered into cell A2, then a direct circular reference has been created. An indirect circular reference is when the formula in a given cell refers to one or more other cells that in return refer back to the original cell. For example a formula in A1 refers to cell A2, A2 refers to A3, and A3 refers back to A1.

When Excel encounters a circular reference in a worksheet, a Circular Reference Warning displays in a dialog box. You can either ignore the circular reference or you can locate it to edit the worksheet. Ignoring circular references risks that your data is incorrect and may lead to bad decisions.

If you see the Circular Reference dialog box, click **OK**, and then track down the problem references so you can correct the formula logic:

- Look in the status bar. If you see the message ***Circular Reference*** followed by a cell then the error is on the active worksheet. If not, move to other worksheets in the workbook until you find the worksheet with one or more cell references. -OR-

- Pick **Formulas > Error Checking > Circular References** which shows you the circular references in the open workbooks.
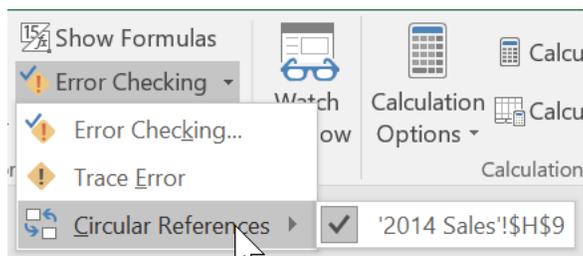
Understanding Range Names

In much of your work with formulas and functions in Excel, you might find it faster and easier to refer to cells and blocks of selected cells by name rather than cell address. Excel makes it easy to name individual cells or cell ranges so you can find data by a logical name and refer to cell ranges in formulas. Common uses for a named range include:

- Navigation

- Printing

- Formulas and Functions

- Advanced Filters

- Macros

Using range names in a workbook is one of the keys to simpler and more accurate ways to refer to cells for navigation and use in formulas.

Naming Ranges

Regardless of how you create a range name, these naming conventions should be used:

- Range names cannot begin with a number, contain spaces or punctuation, or resemble a cell address. For example, the range name, *FY2018*, would not be allowed.

- Instead of spaces, you can use underscores between words. Excel will automatically add an underscore in front of any range name that is entered with a number at the beginning.

- Range names are not case sensitive but using underscores and/or *TitleCase* will help make the range name more readable and descriptive.

- Although Excel will allow a name up to 255 characters, shorter names are easier to remember and to use in formulas.

- Be consistent. If you are naming a number of similar ranges, applying a consistent structure such as *RegionSW*, *RegionNE*, *RegionMW*, will make it easier for users to locate the names.

Naming ranges offers an easy alternative to working with absolute references in formulas. Instead of needing to remember where to place dollar signs, etc., just use the appropriate range name in your formulas.

Name a Cell or Range Directly

To name a cell or range of cells directly:

1. Select the cell(s) you want to reference.

2. Click the **Name Box** at the left end of the Formula bar.

3. Type a name (see naming rules on Page 7), and press ⌨Enter.

<div style="border:1px solid #000; display:inline-block; padding:4px 8px;">TripsGrandTotal ▾</div>
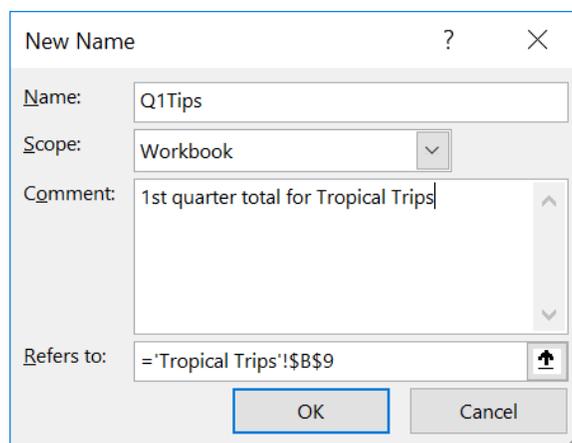
Name a Cell or Range in a Dialog Box

Select the cells you want to reference.

1. From the **Formulas** tab, pick **Define Name** in the **Defined Names** group.
    -OR-
    Right-click the selection and click **Define Name**.

    In either case, the **New Name** dialog box appears.

New Name	? ✕
Name:	Q1Tips
Scope:	Workbook ▾
Comment:	1st quarter total for Tropical Trips
Refers to:	='Tropical Trips'!$B$9 ⬆
	OK Cancel

2. Type a name for the cell or range.

3. Click the **Scope** down arrow, and select whether the name applies to the entire workbook or to one of its worksheets.

4. If desired, type a comment that more fully explains the meaning of the named cells. Comments can be up to 1,000 characters and will appear as a tooltip when the name is used in formulas and functions.

5. If you want to modify the cell or cells to be named, pick the **Refers To** text box, and type the reference (starting with the **=** sign), or reselect the cells from the worksheet.

6. Click **OK** when finished.

Range Names from a Selection

Range names can also be created directly from the selected cells.

To build range names from a selection:

1. To quickly name multiple related ranges, highlight a selection of cells with the text labels at one or more of the outside edges of the selection.

2. From the **Formulas** tab, pick **Create from Selection** in the **Defined Names** group.

3. Choose which of the values to use for creating the names and pick **OK.**

Go to a Named Cell

Named cells are quickly found and selected for you.

- Click the **Name Box** down arrow to open the drop-down list, and click the named cell or range you want to go to.
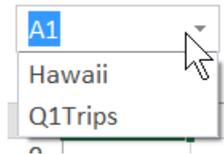
  -OR-

- Press Ctrl + G to open the **Go To** dialog box. Select the range name that you want to move to and **OK**.
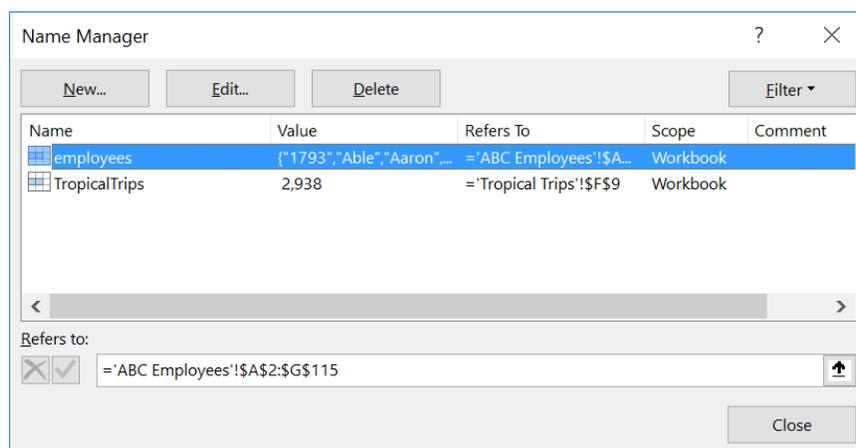
Working with Range Names

Excel provides a **Name Manager** to help you manage and organize your named cells.

To open the Name Manager:

1. From the **Formulas** tab, pick **Name Manager** in the **Defined Names** group.

The Name Manager window opens listing all named cells in the workbook.

From here, you can edit, delete, or even create a new range name.

Functions for Data Analysis

Useful functions for data analysis include **IF**, **SUMIF**, **COUNTIF**, **AVERAGEIF**, **RANK**, and **VLOOKUP**.
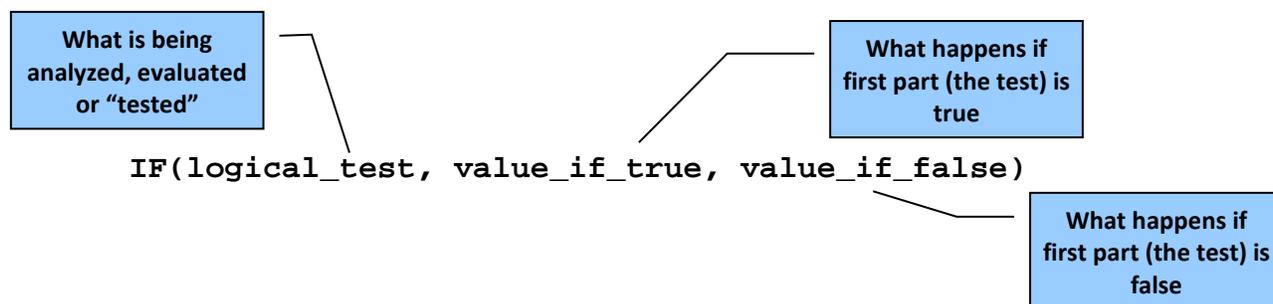
The IF Function

You can use conditional logic in a formula to return a specific result depending on whether a certain test, or condition, is met. If the condition is true, one result will be calculated. If the condition is false, a difference result will be displayed.

The most common logical function is the **IF** function which performs a calculation or other entry only if a certain condition is true, and performs a different calculation or entry if that condition is false.

The cell or entry that is evaluated may be a formula, value, or text; the displayed result may also be a formula, value, or text answer. For example, *if amount is more than 5% over budgeted amount, then display "OVER", otherwise show "OK."*

The structure or syntax for the IF function is:

What is being analyzed, evaluated or "tested"		What happens if first part (the test) is true

`IF(logical_test, value_if_true, value_if_false)`

		What happens if first part (the test) is false

- `logical_test` is any value or expression resulting in TRUE or FALSE.

- `value_if_true` is the value (or formula result) that will be returned if `logical_test` is TRUE.

- `value_if_false` is the value (or formula result) that will be returned if `logical_test` is FALSE.

The result values can display text with an argument such as `"OK"`, calculate a formula such as `B12*5`, or display the contents of a cell. Text results must be entered into an IF function as a character string, that is, in double quotations. Spaces after the commas in the IF function are optional.

Examples:

Simple IF function formulas	Explanation
=IF(B2>100, B2*1.25, B2)	If the value in cell B2 is greater than 100, multiply that value by 1.25, otherwise just display the value in B2 in the cell where the formula is entered.
=IF(C34>=B34, C34, "Entry not valid")	If the value in cell C34 is greater than or equal to the value in cell B34, display the value in C34, otherwise display the text "Entry not valid".
=IF(D10<2500, "No", "Yes")	If the value in cell D10 is less than 2500, display "No", otherwise display "Yes".

Creating Nested Functions

You can use nested functions to perform more complex calculations. A *nested function* serves as an argument of another function; in other words, it's contained within another function. With Excel, you can nest up to 64 levels of functions in a formula! Although many of Excel's functions can be nested, this is a common practice with the IF function. Use of the AND, OR, and NOT functions within the IF function are examples of nested functions.

The OR, AND, and NOT Functions

OR, AND, and NOT are also functions you can use to conditionally evaluate a formula. The syntax for these functions, when nested within an IF function, is described below.

The OR function

With the OR function, only *one* of the conditions, `logical1, logical2, etc.,` named needs to be true for the specified result to be displayed:

```
IF(OR(logical1, logical2), value_if_true, value_if_false)
```

The AND function

In the AND function, *all* conditions specified must be met for the `value_if_true` result to be returned.

```
IF(AND(logical1, logical2), value_if_true, value_if_false)
```

NOT Returns Opposite Condition

The NOT function reverses the result of the logical argument from true to false or from false to true. Use this function to return the opposite condition of the `logical_test` in an IF statement.

Examples:

Complex IF() function formulas	Explanation
=IF(AND(B12>5, B12<20), "Good", "Invalid")	If the contents of cell B12 is more than 5 and less than 20, the text "Good" displays, otherwise the text "Invalid" appears.
=IF(OR(C5="ABC", C5="DEF", C5="GHI"), "Correct entry", "Wrong entry")	If cell C5 contains "ABC", "DEF", or "GHI" (in upper or lower case), display "Correct entry" in the formula cell, otherwise display "Wrong entry".
=IF(NOT(OR(B36=12, B36=20)), "Not a 12 or 20", "Is a 12 or 20")	If B36 contains neither 12 or 20, displays the text "Not a 12 or 20", otherwise displays "Is a 12 or 20".

When troubleshooting a nested function or other IF function that is not working, count the number of open and close parentheses ( ) to make sure that every function has a matching set of parentheses. Excel also adds color to each parenthesis to make it easier to match up the sets of parentheses.

When troubleshooting a nested function or other IF function that is not working, count the number of open and close quotations " " to make sure that every text entry (character string) has a matching set of quotations.

<div align="center">

Helpful Functions with IF

</div>

In addition to OR, AND, NOT, these functions are valuable options to evaluate the `logical_test`.

> **`ISERROR(value)`**
> `Tests if the value is an error`
>
> **`ISNA(value)`**
> `Tests if the value is an error, specifically #N/A`
>
> **`ISBLANK(value)`**
> `Tests if the value is blank`
>
> **`ISNONTEXT(value)`**
> `Tests if the value is not text`
>
> **`ISNUMBER(value)`**
> `Tests if the value is a number`

For example:

> **`=IF(ISBLANK(G4),"Data Missing", G4*J4)`**

IFERROR Function

The **`IFERROR`** function is related to the **`IF`** function but is intended to trap errors in calculations.

> **`=IFERROR(value, value_if_error)`**
> `The `**`value`**` is often a calculation which may include one or more functions.`

For example:

> **`=IFERROR(VLOOKUP(A4, employees, 6, FALSE), "Emp ID not found")`**

This example handles errors with the **`VLOOKUP`** function (see page 18) when the matching data is not found.

SUMIF, COUNTIF & AVERAGEIF Functions

You can conditionally summarize, count, and average data with math and statistical functions including SUMIF, COUNTIF, AVERAGEIF as well as SUMIFS, COUNTIFS, and AVERAGEIFS which allow for multiple criteria.

SUMIF Function

Similar to the SUM function, you use the SUMIF function when you want to add values within a range of cells based on the evaluation of a criterion in another range. The syntax of the SUMIF function is:

```
SUMIF(range, criteria, sum_range)
```

For this function, range is the range of cells where the function will test the criterion specified in `criteria`. The argument `sum_range` specifies the actual cells whose values are to be added. `Sum_range` is optional; if it's omitted, the cells specified in `range` are evaluated by `criteria` and are added if they match `criteria`.

COUNTIF Function

Similar to COUNT which counts everything in a range, you use the COUNTIF function to count the number of cells in a range which meet your specified criteria. The syntax for the COUNTIF function is:

```
COUNTIF(range, criteria)
```

`Range` is the cell or range of cells to count that meet the stated criteria. Text values and blank cells are ignored. `Criteria` can be text, number, expressions, or cell references that identify the cells to be counted. Any criteria other than just a number must be entered as a character string, that is, with quotations around it.

AVERAGEIF Function

The AVERAGEIF is, like SUMIF and COUNTIF, a conditional math function. Use AVERAGEIF to conditionally average a range of numbers. The syntax for the AVERAGEIF function is:

```
AVERAGEIF(range, criteria, average_range)
```

`Range` is the cell or range of cells you want to average, `criteria` is the number, expression, cell reference, or text that identifies which cells should be averaged. `Average_range` is the corresponding set of cells you want to average. If you omit this, range will be used instead.

RANK Functions

The **RANK** function returns the rank of a number within a set of numbers. It can give you information on where a particular value falls in a list such as a vendor score or price.

In recent versions of Excel, RANK has actually been split into two different functions: RANK.AVG and RANK.EQ. They provide similar results, but how they differ is important.

RANK & RANK.EQ

RANK and **RANK.EQ** returns the highest rank of multiple results. The syntax of the RANK or RANK.EQ function is:

```
RANK.EQ(number, ref, [order])
```

For this function, `number` is the value to be checked against a range of values. The argument `ref` specifies the range of cells to be compared for ranking. The argument `[order]` is optional; if it's omitted or 0 (zero), the order is descending, that is, the highest value will be 1. For ascending order, enter any non-zero value.

4	=RANK.EQ(B2,$B$2:$B$6)
3	2
3	2
2	4
1	5

With RANK.EQ, in this reference list of 4, 3, 3, 2, 1, this function returns 2 for the duplicates, because that's the highest rank of the number 3.

RANK.AVG

RANK.AVG returns the average position of multiple results. The syntax of the RANK or RANK.AVG function is:

```
RANK.EQ(number, ref, [order])
```

For this function, `number` is the value to be checked against a range of values. The argument `ref` specifies the range of cells to be compared for ranking. The argument `[order]` is optional; if it's omitted or 0 (zero), the order is descending, that is, the highest value will be 1. For ascending order, enter any non-zero value.

4	=RANK.AVG(B2,$B$2:$B$6)
3	2.5
3	2.5
2	4
1	5

With RANK.EQ, in this reference list of 4, 3, 3, 2, 1, the average rank of the duplicates of 3 would be 2.5.

You can still use the RANK function without adding AVG or EQ, but RANK might not work in future versions of Excel. While it does work, it functions like RANK.EQ.

Breaking a Tie with RANK

While the RANK function is useful to determine ranking or importance, you might need to refine results when there are ties or duplicates.

To simplify working with the RANK functions, create range names for the ranges you want the function to evaluate such as the range names **Scores** and **Times** below.

In the example below, the rank was calculated on the test scores with 1 representing the highest score. Because two of the entries have the same ranking, the test timing is used to break the tie. The formula for this solution is:

```
=IF(COUNTIF(Scores,B2)>1, RANK.EQ(C2, Times,1)/100,0)
```

	A	B	C	D	E	F	G	H	I	J
1	**Student**	**Scores**	**Time**	**Rank**	**Tie Break**	**Final Rank**				
2	Apple, Anna	75	22	14	=IF(COUNTIF(Scores,B2)>1,RANK.EQ(C2,Times,1)/100,0)					
3	Broccoli, Betty	83	25	12	0	12				
4	Carrot, Charlie	72	28	15	0	15				
5	Cauliflower, Cathy	100	28	1	0	1				
6	Cilantro, Carrie	93	27	5	0	5				
7	Garlic, George	87	21	10	0	10				
8	Grape, Gina	99	29	2	0	2				
9	Kiwi, Ken	91	20	7	0.01	7.01				
10	Lemon, Lola	86	29	11	0	11				
11	Lime, Larry	95	27	4	0	4				
12	Mango, Marty	98	30	3	0	3				
13	Melon, Mary	78	27	13	0	13				
14	Orange, Oscar	88	21	9	0	9				
15	Pepper, Paula	92	29	6	0	6				
16	Plum, Peter	91	23	7	0.05	7.05				
17	Tomato, Tommy	68	24	16	0	16				

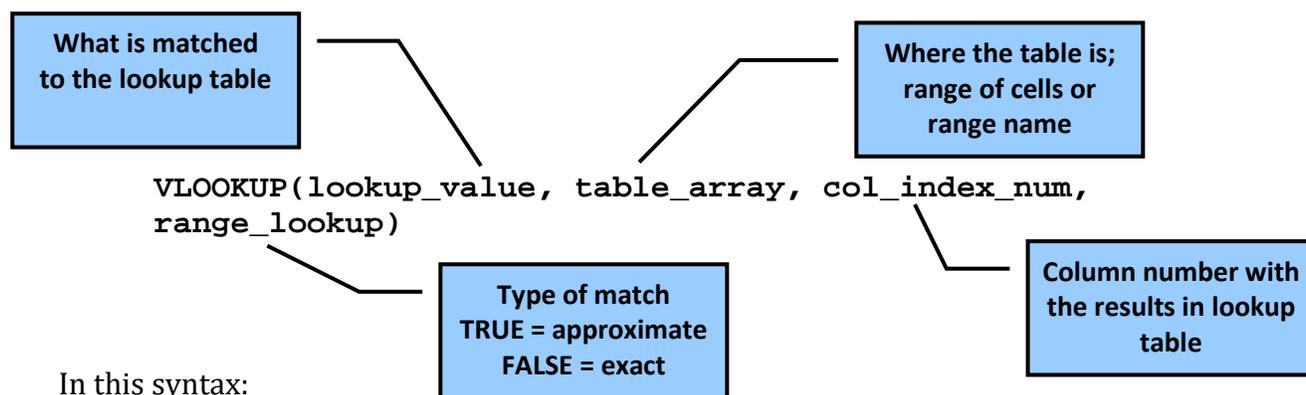## *How the Tie Break Formula Works*

In the example above, the Tie Break formula uses the **COUNTIF** and **RANK.EQ** functions, wrapped with an **IF** function, to see if a tie breaking decimal amount should be added to the original Rank.

1. First, the TieBreak formula checks to see **IF** there is more than one instance of the number in the entire list; **Scores** is a created name for the range B2:B17.

   ```
   IF(COUNTIF(Scores,B2)>1
   ```

2. If there is more than one instance, it ranks the Time column in ascending order, because a lower time is better; **Times** is a created name for the range C2:C17.

   ```
   RANK.EQ(C2,Times,1)
   ```

3. Finally, it divides that amount by 100, to get a decimal amount. In another calculation, you'll add this decimal amount to the original Rank. The divisor, 100, could be changed to another number if you are working with a longer list. To complete the IF function, if there is only one instance a of a rank, the result for the Tie Break is zero.

   ```
   /100, 0)
   ```

VLOOKUP Function

Use VLOOKUP to look up a value you want to find in an Excel list or table. Using VLOOKUP is similar to looking up a person's name in a telephone book (old school ☺) to get a telephone number. VLOOKUP looks at a value in one column and finds its corresponding value on the same row in another column. Lookup functions include HLOOKUP, LOOKUP, and VLOOKUP; VLOOKUP is the most commonly used of these functions.

The VLOOKUP function performs a *vertical* lookup. It searches for the lookup value in the first column of the lookup table and returns a value in the same row from the specified column of the table. The syntax is:

What is matched to the lookup table		Where the table is; range of cells or range name

```
VLOOKUP(lookup_value, table_array, col_index_num,
range_lookup)
```

	Type of match TRUE = approximate FALSE = exact	Column number with the results in lookup table

In this syntax:

- **lookup_value** is the value to search for in the first column of the lookup table or range. This doesn't have to be the first column of a worksheet if you apply a range name to the table.

- **table_array** is the name of the lookup table range. Although you don't have to add a range name to the lookup table, it is **highly** recommended.

- **col_index_num** is the number of the column from which a value will be returned. Count the number from the first column in the specified range or range name. This is always a number and not the column heading. If the number is greater than the number of columns in **table_array**, VLOOKUP returns the **#REF** error value.

- **range_lookup** is an optional argument that specifies whether you want VLOOKUP to find an exact or approximate match. Use FALSE for an exact match; use TRUE for an approximate match. If you don't include this argument, VLOOKUP assumes the value is TRUE.

  Although optional, I recommend always including this argument as a best practice so it isn't inadvertently left out.

Another way to remember the structure of the VLOOKUP function is with *What, Where, Which, What*:

- **What** are you looking up?

- **Where** are you looking?

- **Which** column has the result you want?

- **What** type of match?

For example, suppose you have a list of employees in a table with a range called *employees*. The employee's ID numbers are in the first column of the range as shown below.

	A	B	C	D	E	F	G
1	Emp ID	Last Name	First Name	Position	Department	Division	Supervisor
2	1696	Abdul	Cathy	Mechanical Engineer	Engineering	Green	Melon, Mary
3	1793	Able	Aaron	Admin. Assist.	Admin.	Red	Broccoli, Betty
4	1055	Albert	Maxine	Group Admin. Assist.	Marketing	Green	Carrot, Charlie
5	1759	Alexi	Stephanie	Lead Engineer	Engineering	Blue	Lettuce, Lola
6	1814	Al-Sabah	Daoud	Technician Assist.	Engineering	Red	Plum, Peter
7	1723	Alstain	Isolde	Technician	Engineering	Red	Plum, Peter
8	1967	Aruda	Felice	Admin. Assist.	Admin.	Green	Tomato, Tommy
9	1168	Asonte	Toni	Group Admin. Assist.	Engineering	Red	Plum, Peter
10	1906	Bankler	Rowena	Admin. Assist.	Marketing	Blue	Orange, Oscar
11	1923	Barber	Lisa	Product Marketer	Marketing	Red	Lemon, Lisa

If you know the employee's ID number, you can use the VLOOKUP function to return the first or last name of the employee, their department or division, or even their supervisor.

Emp ID	Sales	Division	Last Name	Supervisor
1759	$17,530	=VLOOKUP(A4,employees,6,FALSE)		
1969	$25,980	Blue		
1949	$30,760	Green		
1302	$27,910	Red		
1696	$32,530	Green		

The *What, Where, Which* for this example is:

- **What** = The **Emp ID** in Column A

- **Where** = **employees** range

- **Which** = Column 6 for the **Division**

- **What** = The type of match is exact or **FALSE**. This lookup is an exact match, that is, you don't want a match only similar to the **Emp ID**.

Use VLOOKUP for Exact Matches

When you use the VLOOKUP function and require exact matches, remember the following:

- The lookup value must always be located in the first column of the lookup table; this doesn't have to be the first column of the worksheet. These values don't need to be sorted.

- The **range_lookup** argument is FALSE or 0 (zero). The sort order of the lookup table isn't important but only exact matches will return a result. If the first column includes duplicates, the row of the first entry matched will be used to return the answer.

- If an exact match is not found, an error (**#N/A**) is returned.

- Uppercase and lowercase text are equivalent.

Use VLOOKUP for Approximate Matches

You can also use the VLOOKUP function to return an approximate match, that is, to lookup an entry that may fall between entries in the lookup table. To use the VLOOKUP function for approximate matches:

- Specify the **range_lookup** argument as TRUE or 1.

- The first column in the lookup table must be in ascending order.

- If the function doesn't find an exact match, it looks for the largest value less than the lookup value and returns its corresponding data. This is the default value if you leave the argument blank.

- If the lookup value is less than the smallest value in the table, an error (**#N/A**) is returned.

- If the lookup value is more than the largest value in the table, the value will be matched to the largest value in the table.

POWER TIP: Most Excel resources indicate the lookup value must always be found in the first column of a worksheet. That isn't actually true. If you want to look up in the first column sometimes but also want to look up in the second column, you do not have to create multiple versions or move data around. Just create more than one range name for separate lookup calculations with different tables.

Date & Text Functions

Other valuable functions include options to manipulate date and text data.

Functions for the Current Date

Two of the most common date functions are:

>**TODAY()** for the current date
>
>**NOW()** for the current date and time depending on formatting

These date functions are frequently used to display a date within a worksheet and for calculations between a date and the current date. Because all date have a numeric value, you can also create formulas to calculate the differences between two or more dates.

Functions for Working with Dates

To more effectively sort and filter dates, you can extract key elements from date entries and even perform built-in calculations.

>**MONTH(serial_number)** to show the month in numeric format from 1 (January) to 12 (December)
>
>**YEAR(serial_number)** to display the year in numeric format such as 2015 or 2018
>
>**EOMONTH(start_date, months)** to calculate the end of the month of a date based on a number of months
>
>**EDATE(start_date, months)** to calculate a future date based on a number of months

Text Functions to Change and Extract

In Excel, you can use a function to format text in a cell, and copy the result to a range of cells. This is more efficient than manually formatting each cell individually. For example, let's say you wanted to make all the names in a list uppercase. You could manually edit each name, but it would be much easier to use the UPPER function and copy the result to the cells you want to change.

Text Functions to Change Case

What if you have text in cell A1 (or all of column A) that has a mismatch of upper- and lower-case characters such as **`chris Edwards`** and you want all entries to be consistent?

> **`PROPER(text)`** `to change the data to initial caps/title case`
> *`Chris Edwards`*
>
> **`UPPER(text)`** `to change to all upper case`
> *`CHRIS EDWARDS`*
>
> **`LOWER(text)`** `to change to all lower case`
> *`chris edwards`*

Where **`text`** is the text you want to convert. It can refer to a cell or a text string.

Combining Two or More Cells (Sounds Like Cat)

Existing data may not be formatted exactly the way you need. For example, if employee names are in 2 columns, Last Name and First Name, but you want to display a full name for reporting as in Last Name, First Name (Smith, Joe) there are several ways to do this. One helpful function is CONCATENATE:

> **`CONCATENATE(text1, text2,...)`**

In the following examples, Column B is Last Name and Column C is First Name. To create the Full Name in Column D, the formula would be similar to (either example works):

> **`CONCATENATE(B2,", ",C2)`**

CONCATENATE is a function that "combines"; additional text like the comma is included in quotations to create a character string.

The & sign also combines the values in cells B2 and C2 without the CONCATENATE function.

> **`=B2 & ", " & C2`**

Removing Extra Spaces

When working with data originating from other sources, such as downloads from a different system, it is common for the data to have leading or trailing spaces which will often affect the success of searches and sorting. To display a text entry without any extra spaces at the beginning or end, use the TRIM function as in:

```
TRIM(text)
```

TRIM removes all leading and trailing spaces and replaces internal strings of multiple spaces by a single space. You can also nest other functions inside of a TRIM function such as:

```
TRIM(PROPER(A1))
```

Another function, **CLEAN(text)**, removes all non-printable characters from text.

Extracting Only What You Want

Another text entry problem is when you only want to work with a portion of a data entry. What do you do when you to sort or search on characters at the beginning, middle, or end of a text entry?

```
LEFT(text, num_chars)

RIGHT(text, num_chars)

MID(text, start_num, num_chars)
```

Your results might be combined with other functions or you could use Paste Special to copy only the results into another location in the worksheet.

Pulling Text Apart by Bringing Functions Together

Sometimes Excel data includes names, addresses or other entries you might want to split up for easier sorting and searching. The following examples work with 2 more functions:

```
LEN(text)

FIND(find_text, within_text, [start_num])
```

To split up a text entry such as **"Chris Edwards"** in cell A1, use a combination of functions including FIND which is used to locate specific characters such as a space or comma:

```
LEFT(A1, FIND(" ",A1)-1)
The result is "Chris"-all of the characters to the left of
the space

RIGHT(A1, LEN(A1)-FIND(" ",A1))
The result is "Edwards"-all of the characters after the
space
```

Extracting Data with Flash Fill

Perhaps the closest thing to magic in a workbook is the **Flash Fill** feature (new in Excel 2013 and Excel 2016/365). With Flash Fill, you may not even need to write formulas using text functions to automatically extract the data you want. As soon as Flash Fill detects a pattern in your data, it enters the rest of your information all at once.

For example, if you have a column of contacts with a first and last name and you want to create a separate column for the first name, simply type the first name in an adjacent column. Repeat typing the next first name in the row below and Flash Fill kicks in and duplicates the pattern and formatting for the rest of the data. Its predictive data entry recognizes text, dates, and even values.

G	H
Supervisor	**Supervisor Last**
Broccoli, Betty	Broccoli
Grape, Gina	Grape
Cilantro, Carrie	Cilantro
Melon, Mary	Melon
Banana, Bruce	Banana
Lettuce, Lola	Lettuce
Lime, Larry	Lime
Melon, Mary	Melon
Lettuce, Lola	Lettuce
Lettuce, Lola	Lettuce
Apple, Anna	Apple
Mango, Marty	Mango
Plum, Peter	Plum
Melon, Mary	Melon

Put these tips and techniques to practice and become more productive with Excel!

Bonus Resources

As an extra bonus, additional Microsoft Excel resources, a keyboard shortcuts handout, and other helpful tips are waiting for you at:

TheSoftwarePro.com/Excel

Meet Dawn

Dawn Bjork, MCT – The Software Pro®

Software Productivity is Only a Click Away™

Dawn Bjork, MCT, is The Software Pro® with 25+ years' experience as a *Software Navigator*.

A professional speaker, trainer, consultant, author, and business owner, Dawn simplifies complex processes with an informative and entertaining approach and delivers content that is immediately useful. Dawn has helped more than 100,000 individuals from Fortune 500 companies, government, education, and other organizations to learn smart, easy, and productive ways to use software and technology more effectively.

Microsoft has awarded Dawn with the certifications of Microsoft Certified Trainer (MCT), Microsoft Office Specialist Master (MOSM), and individual certifications in Microsoft Windows, Excel, Word, PowerPoint, Outlook, Access, OneNote, and Visio. In addition, The Software Pro is certified as a Women's Business Enterprise (WBE) and Woman Owned Small Business (WOSB).

Dawn is the Immediate Past President of the National Speakers Association-Colorado, the largest chapter in the country. When she's not immersed in the world of software productivity, Dawn enjoys the Colorado mountains, hiking, biking, golf, yoga, playing poker, and spending time with her family.

Dawn Bjork, MCT ∙ The Software Pro®

(303) 699-6868

Dawn@TheSoftwarePro.com

TheSoftwarePro.com

Facebook.com/TheSoftwarePro

Twitter.com/TheSoftwarePro

LinkedIn.com/in/DawnBjork

YouTube.com/TheSoftwarePro