**Lesson: INDEX/MATCH Formulas in Excel**

With INDEX/MATCH you can automate your invoices, quotation workbooks, business models. You can also develop reports with data coming from many different sources. With these formulas you can virtually create relational databases in Excel and develop reports downstream. They are more powerful and versatile than **LOOKUP** formulas.

The quantities sold are in one data set, the name and the address of the client is in another set of data and then the product description is in a third set of data...no problem, a single INDEX/MATCH formula copied all the way down your column solves this problem. You now have a data set including sales, products and clients.

Stop entering data manually in your workbooks and reports use the easy to master INDEX/MATCH formula.

The INDEX/MATCH formula is a formula using 2 functions INDEX and MATCH.

With the example below, the formula =INDEX(A2:E5,1,2) would return " Tiger Auto" because the formula translated in plain English says, what is the value found in the rage " A2:E5" in the first row and the second column.

The formula =MATCH(B11,A2:A5,0) in cell B12 would return 2 because the value in B11 (86598) is on the second row of the range A2:A5. The zero at the end of the formula tells Excel that you want an exact match. If you were looking for a number, a 1 instead of a zero would tell Excel to use the next higher value as a -1 would mean use the next lower value.

With MATCH  you don't have to sort the range or use FALSE or TRUE.

Now let us replace the row number in the INDEX formula by the MATCH formula:  
**=INDEX(A2:E5,MATCH(B11,A2:A5,0),2)**  
and whenever you change the value in cell B11, a new value appears in cell B12.

Here are some formulas combining INDEX and MATCH functions.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | D | D | E |
| 1 | Client # | Name | Address | City | State & ZIP |
| 2 | 36596 | Tiger Auto | 33 Woods | Miami | FL 10230 |
| 3 | 86598 | Phil Lumber | 555 Makes | Boston | MA 34567 |
| 4 | 58971 | David Eng. | 1200 Duvall | Charleston | SC 10004 |
| 5 | 87456 | Stewart Inc. | 5673 Payne | San Francisco | CA 27002 |

Tired of typing names, addresses, cities and states on invoices?

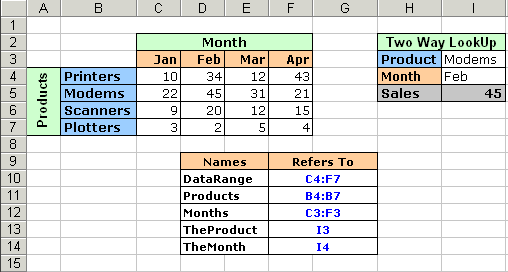
|  |  |  |
| --- | --- | --- |
|  | A | B |
| 11 | Client #: | 86598 |
| 12 | Name: | **=INDEX(A2:E5,MATCH(B11,A2:A5,0),2)** |
| 13 | Address: | **=INDEX(A2:E5,MATCH(B11,A2:A5,0),3)** |
| 14 | City: | **=INDEX(A2:E5,MATCH(B11,A2:A5,0),4)** |
| 16 | State & ZIP | **=INDEX(A2:E5,MATCH(B11,A2:A5,0),5)** |

Each time you enter a Client #, the name, address, city and State are automatically modified. Imagine the time you save and the number of errors that you avoid when you complete your invoices. You can do the same with your products entering the name and extracting the unit price, the product number from a table.

If you are a Excel user then invariably you must've have used the lookup functions of Excel, namely HLookUp() and VLookup(). For those who haven't: A lookup function is used to return a value from a given table by looking up another value in the same table. A simple example could be a discount table consisting of two columns, Purchase Amount & Discount. You can write a formula that uses **VLookUp()** to determine the discount rate for a given purchase amount.

Unfortunately the lookup functions in Excel are only appropriate for one-way lookups. There is no inbuilt worksheet function for performing a two-way lookup. Let me share my 2-way lookup formula.

Consider the table (refer to the figure) which displays the rather dismal monthly sales figures of various computer peripherals.



**Objective:**

To arrive at the Sales figure for Modems sold in Feb

**Overview:**

We shall make use of two functions **Match()** and **Index()** to create our two way lookup. Let me discuss briefly how these functions are used. For more detailed info look up the Excel help file.

* **Match(value\_to\_lookup, lookup\_array, match\_type):** Returns the relative position of an item in an array that matches a specified value in a specified order. Value\_to\_lookup is the value you want to match in lookup\_array. Lookup\_array   is a contiguous range of cells containing possible lookup values. Match\_type specifies the matching criteria to be used. If match\_type is 0, MATCH finds the first value that is exactly equal to value\_to\_lookup.
* **Index(array, row\_num, column\_num):** Returns the value of an element in a table or an array, selected by the row and column number indexes. Array is a range of cells. Row\_num selects the row in array from which to return a value. Column\_num selects the column in array from which to return a value. INDEX returns the value in the array (cell) at the intersection of row\_num and column\_num.

**My 2-Way lookup formula:**

**=INDEX(DataRange, MATCH(TheProduct,Products,0), MATCH(TheMonth,Months,0))**

This is the formula in Cell **I5**.

**Explanation:**

**MATCH(TheProduct,Products,0)** will look up the value of the range **TheProduct** in the array **Products** and will return the position of the first occurrence of the value. If **TheProduct** is Feb then the above call will return 2 as Feb occurs at the second position in our array (Jan, Feb, Mar, Apr)

Similarly **MATCH(TheMonth,Months,0)** will look up the value of the range **TheMonth** in the array **Months** and will return the position of the first occurrence of the value. If **TheMonth** is Modems then the above call will return 2 as Modems occurs at the second position in our array (Printer, Modems, Scanners, Plotters)

Hence the above formula reduces to: **=INDEX(DataRange, 2, 2)**which returns the value of cell in the Range **DataRange** which occurs at the intersection of the 2nd Row and 2nd Column i.e **45**. Yahoooooo!!!

**For Advanced users:**

Now the formula will return an error in case the Product or Month does not exist in the array. So let us use **IsError()** and **If()** to arrive at a more refined formula which will return 0 if either Product or Month does not exist in the Array.

**=IF(ISERROR(INDEX(DataRange, MATCH(TheProduct,Products,0), MATCH(TheMonth,Months,0))),0,INDEX(DataRange, MATCH(TheProduct,Products,0), MATCH(TheMonth,Months,0)))**