



Chapter 6: *Database Features of a Spreadsheet*

Objectives

The exercises in this chapter introduce the database functions of Excel and builds on skills you have learned in Chapters 1 through 5. This chapter also puts much more emphasis on your ability to analyse and design your own solution to the problems presented.

This chapter will cover:

- The COUNTIF and SUMIF functions
- Sorting data and creating subtotals
- More Date functions
- The MATCH, VLOOKUP, HLOOKUP and INDEX functions

Preparation

Read the whole of the chapter before beginning the exercises.

Exercise 1 – Frequency Counts

An operation that is often performed with a large set of data is to find out how many of each value occur in the data. For example, teachers usually need to count how many students received 50 on a test, how many received 51 etc. in order to construct a frequency bar chart or distribution.

Open Exercise 1 (ch6_Ex1) in Support Files (Chapter6) of the course website. This model consists of three worksheets - the Comments worksheet, a worksheet called Numbers and a worksheet called Counts. The Numbers worksheet simply contains a list of numbers between 1 and 20 inclusive. Your task will be to calculate in the Counts worksheet how many times each number (1, 2, ... 20) occurs. There are several ways of doing this – but we will limit discussion to one that is common to most spreadsheet programs.

The COUNTIF function

The COUNTIF type of function is quite common in spreadsheet programs, although the exact name of the function might not be COUNTIF in other spreadsheets. As its name suggests there are 2 parts to the function. The COUNT will be done only IF some criterion is satisfied. In this case the criterion will be "is the value equal to 1" - in which case increment the count for 1s; or "is the value equal to 2" - in which case increment the count for 2s; and so on.



To use this function you first need to create a column containing the criteria. So in the Counts worksheet enter an appropriate column heading and then the list of possible values that might be found in the Numbers worksheet, i.e. 1 through 20. In the cell next to each of these numbers you will enter the formula which produces a count of how many times that value occurs in the Numbers worksheet.

Enter an appropriate heading for the new column and define names for the data ranges. Begin to enter the formula in the first cell (which you'll fill down the column). Use the function wizard to select the COUNTIF function from the Statistical group of functions. The arguments required are a Range and a Criteria.

The Range is simply the column of numbers in the Numbers worksheet. If you simply drag over the column in order to enter the argument in the Range text box you will see the range specified as

Numbers!A2:A301

which is going to lead to an error unless you modify it. Can you anticipate what the error will be? Do you know the best way to avoid it?

The criterion for this first cell is that the value in the Numbers worksheet be equal to 1, i.e. the value in this first row of the Counts worksheet. Look through the help pages for this function by clicking on the [Help on this function](#) link at the bottom left corner of the Insert function panel once the function has been selected or its Function argument panel. The criterion that you specify in this case is simply the reference to the cell containing the number 1 - i.e. the cell adjacent to the one containing this formula.

Finish the formula and check that the answer you get is that there are 15 1s in the Numbers worksheet. Next simply fill the formula down the column. Did you correctly enter the Range argument, or did you run into the error you were warned about above? The results for the first few counts are shown in Figure 6.1

Figure 6.1 - part of the frequency counts using the COUNTIF function

	A	B	C
1	by COUNTIF function		
2	Values	Counts	
3	1	15	
4	2	14	
5	3	16	
6	4	14	
7	5	16	
8	6	10	



Exercise 2 - Frequency Counts by Bins

You might question the usefulness of this, since very often you don't want to calculate a frequency for every unique value in a data set, but rather the frequency of values in a range or bin. For example, a teacher would probably rather count how many students received a grade between 50 and 54 (one bin), between 55 and 59 (another bin), etc.

Consider the example we used in the previous exercise. Suppose now that we want to calculate frequencies for the following bins (or ranges):

1 to 4	(bin1)
5 to 8	(bin2)
9 to 12	(bin3)
13 to 16	(bin4)
17 to 20	(bin5)

Again, there are several ways to perform such a frequency count by ranges. One way uses a LOOKUP function to actually calculate a bin number to which each of the values in the Numbers worksheet belongs and then uses the COUNTIF function to count how many times each of the bins occurs.

The LOOKUP/COUNTIF Approach

Reopen the Exercise 1 (ch6_Ex1) in Support Files (Chapter 6) of the course website. (Don't use the Excel file that you created in the last exercise - start with the original.)

Create a lookup table in the Counts worksheet based on the table of bins above. Refer back to Chapter 3 if you have difficulty with this.

Use the lookup table to create a new column in the Numbers worksheet, which contains the bin to which each number belongs. Figure 6.2 shows the first few rows resulting from this.

Figure 6.2 - bins for the first few numbers

	A	B
1	Numbers	Bin
2	9	3
3	12	3
4	9	3
5	18	5
6	18	5
7	10	3
8	6	2
9	10	3
10	9	3
11	18	5
12	17	5
13	16	4



Return to the Counts worksheet and enter a new column in which the values are calculated by the COUNTIF function and which count how many times each bin number occurs in the Bins column in the Numbers worksheet. Check that your results match those shown in Figure 6.3

Figure 6.3 - calculating frequencies by Bins

	A	B	C
	Lower		
1	Bound	Bin	Frequency
2	1	1	59
3	5	2	54
4	9	3	72
5	13	4	61
6	17	5	54

Exercise 3 - Simple Statistical Summaries

This next exercise illustrates how a spreadsheet can be used for summarising data and for producing simple statistics for a large set of data. The frequency counts produced in the previous exercise are also a simple form of statistical summary, and this exercise extends these ideas.

Open Exercise 3 (ch6_Ex3) in the Support files (Chapter 6) of the course website. Besides the Comments worksheet there is a worksheet called Sales_Transactions, which contains a lengthy list of sales transactions that took place over some period of time. It includes information about the transaction such as the sales person's name, the sale price, the region and a commission rate. These could be real estate transactions although they could equally well be sales of something else.

There are certain obvious statistical summaries that we might like to derive from the data. For example:

- What is the total sales and total commission for each sales person?
- What are the total sales in each region?
- What is the average commission for each sales person?
- What is the average sale price in each region?

We'll look at answering the questions involving totaling (or summing) in the next section and questions involving averaging in the section after that.



Ex 3.1 – Selective Summing

Totals by Sales Person

Apart from the total sales you also want the total commission for each sales person - so first create a new column and the formula to calculate the commission (sales amount multiplied by commission rate). Note that names have been defined for the data ranges. Verify this by examining the Name Manager in the Formulas tab.

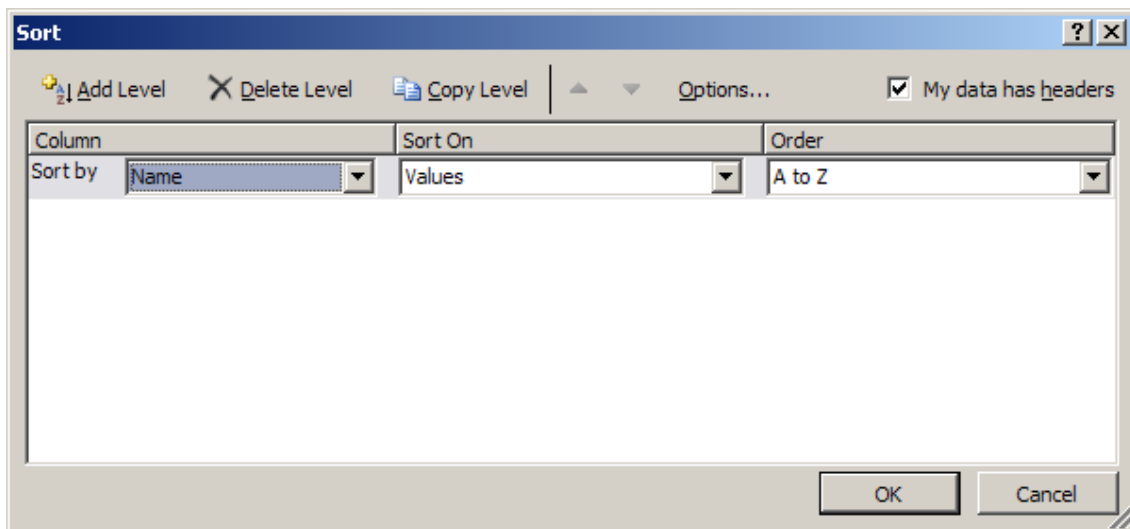
It is possible to calculate these simple totals and averages by modifying the Sales_Transactions worksheet itself. Before you do this save your work and open a new version of the ch6_Ex3 file so that you start afresh.

You are going to explore the Subtotals command in the Outline group of the Data tab. This feature operates on the table of data itself - rather than using that data (but not altering it) to create a separate summary table as the pivot table approach did.

First you have to sort the data. If you want to create summaries by sales person you need to first sort by sales person. Later you would have to sort by region in order to perform summaries by region.

Select any cell in one of the columns of data and then choose Sort in the Sort & Filter group of the Data tab. The window shown in Figure 6.4 should appear. Click on the down pointing arrow for the Sort by specification to inspect the pop-up list of possible choices. All of the column headings should be in the list. We want to sort by Name so make sure it remains selected. Click on Add Level and check out the other secondary sort specifications indicated by the Then by text boxes. Before you click OK examine Options... as well.

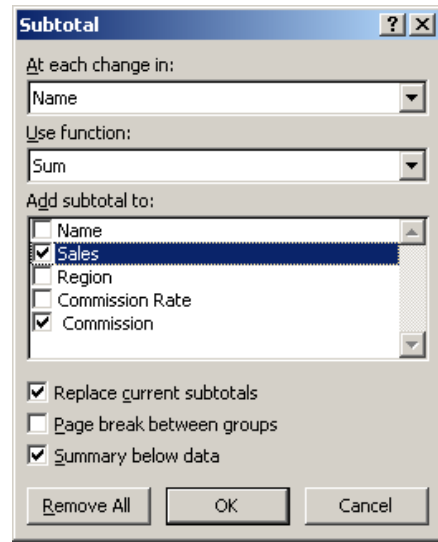
Figure 6.4 - the sort specification window





The data should be ordered starting with BERNIER, then BINCE, then BUSHBY, etc. It is now possible to summarise these groupings and to collapse the groups in order to see only the summaries. Select any cell within the data and choose Subtotal from the Outline group in the Data tab. The window shown in Figure 6.5 appears.

Figure 6.5 - the Subtotal specification window



Most of the specifications this window allows are straightforward. Examine the Use function: pop-up list in order to see what types of summaries are possible, but make sure that Sum remains selected. Also make sure that Sales and Commission are selected in the Add subtotal to: window. The three check boxes at the bottom of the window may stay at their defaults although you should make sure you understand what their effects are. When you click on the OK button the Sales_Transactions worksheet is modified as shown (in part) in Figure 6.6.

Figure 6.6 - part of the Sales Transactions worksheet with subtotals

	A	B	C	D
1	Name	Sales	Region	Commission Rate
2	BERNIER	\$171,904	west	4.50%
3	BERNIER	\$171,904	central	4.80%
4	BERNIER	\$171,904	north	4.90%
5	BERNIER	\$171,904	south	4.40%
6	BERNIER	\$171,904	north	6.30%
7	BERNIER	\$171,904	north	6.00%
8	BERNIER	\$171,904	west	4.50%
9	BERNIER Total	\$1,203,328		
10	BINCE	\$162,605	east	4.60%
11	BINCE	\$162,605	central	5.80%
12	BINCE	\$162,605	south	4.10%
13	BINCE	\$162,605	east	4.60%
14	BINCE Total	\$650,420		
15	BUSHBY	\$200,448	south	6.20%



The controls on the left of this worksheet allow you to collapse (to see only the subtotals) and expand the groupings again. You can do this either collectively or one group at a time. There are three little boxes labeled 1, 2, and 3 in the top left corner, which represent the level of subtotaling. Currently all levels (down to the details of each sales persons transactions) are displayed. Click on the box labeled 2 and you'll find that only the subtotals for each person are displayed (Figure 6.7).

Figure 6.7 - the Sales Transaction worksheet collapsed to show subtotals

1	2	3	A	B	C	D
			1 Name	Sales	Region	Commission Rate
	+		9 BERNIER Total	\$1,203,328		
	+		14 BINCE Total	\$650,420		
	+		21 BUSHBY Total	\$1,202,688		
	+		29 CAMPBELL Total	\$1,816,500		
	+		36 CORREIA Total	\$2,049,768		
	+		40 CUKIER Total	\$624,666		
	+		44 DA-COSTA Total	\$817,563		

Click on the + symbol to the left of one of the rows and you'll see just that grouping expanded again. Click on the box labeled 1 and you'll find only the top level of subtotaling - the grand total displayed. Experiment with shifting between the levels of subtotaling.

Ex 3.2 – Summing by Region

To obtain subtotals by region you need to remove this subtotaling by sales person (observe that the Subtotal window, Figure 6.5, has a Remove All button), sort the data by region and apply the subtotaling again.

Ex 3.3 – Alternative Approaches

It is generally preferable NOT to manipulate the data in this way, but rather to extract such summaries to another worksheet. Create a new worksheet and name it Summaries. To compute the subtotals we use a conditional function - the SUMIF function - to perform the selective totals. As with the COUNTIF function in the first exercise you need to be able to enter a list of the names (all of them, because if you leave out a name that sales person will not appear in the summary results).

An easy way to create the list of names is to click the Remove All button again. The Sales_Transactions worksheet will be sorted by name, without subtotals. Select and copy the Name column. Copy it to the Summaries worksheet. Now delete all the extra occurrences of each name. There should be 34 names when you've finished.



Create a second column for Sales Totals and select the first cell. The SUMIF function can use either 2 or 3 arguments. The first 2 are the same as used in COUNTIF – range (where to look), and criterion (the condition to look for). But there is no way in which one can SUM names. The third argument specifies the range in which to find the values to be SUMmed.

Implement the SUMIF function to produce the summary. The first several lines are shown in Figure 6.8.

Figure 6.8 - the Summaries showing subtotals for each Name

	A	B
1	Names	Sales Total
2	BERNIER	\$1,203,328
3	BINCE	\$650,420
4	BUSHBY	\$1,202,688
5	CAMPBELL	\$1,816,500
6	CORREIA	\$2,049,768
7	CUKIER	\$624,666

Selective Averages

We also want to calculate the average sales for each sales person and the average sales amount for each region.

Add a column for Average Sales to the Summaries worksheet. This value is easily calculated using the Sales Total just computed and dividing it by the number of sales. The number of sales is computed by COUNTIF. You should be able to create this column yourself.

To perform a summary based on Region it will be advisable to insert another worksheet. Since this one will also contain summaries, it could be named Summaries by Region. To complete it you will need to create a list of all the Regions in one column, and a column of formulas to compute the Average Sales for each. You should be able to do this by relying on the example above.



Exercise 4 - Accounts Receivable Data Analysis

This next model brings together some of the features we have been examining in previous chapters as well as further emphasizing the design of a model as interacting worksheets. In this exercise the worksheets in the model resemble the tables in a relational database in which records are selectively combined by means of a key field.

The context is one in which a company has sold various items to other companies (its customers), has sent an invoice to the customer asking for payment, and is now waiting to be paid - i.e. the account is receivable.

The company allows a discount for prompt payment. The percent amount of the discount and the number of days during which it applies may be different for each customer. Also the number of days by which the full amount is meant to be paid may be different for each customer. If an account is not paid by the due date the customer is charged interest and the interest rate may vary for each customer.

Open Exercise 4 (Lab6_Ex4) in the Support Files (Chapter 6) of the course website. The information just described is contained in the Customers worksheet. There are six customer companies. Discount Days is the number of days after issuing the invoice for which the discount applies. This means that if the customer pays within that period of time they will get the discount indicated in the Discount column. Days Due is the number of days within which the invoice should be paid and if it is not then the Late Percent is applied.

The Invoices worksheet lists a few accounts that are active. It contains the customer number, the amount owed, and the date on which the invoice was issued (usually the date the customer made the purchase).

(The invoice date is calculated whereas normally it would be input as data. It is calculated only in order to keep the dates current. A side effect is that the dates will keep changing as you work with this model, so don't be alarmed by those changes. The formulas you create should work regardless of the actual values for the dates.)

Your first task is to add a column to the worksheet that contains the number of days since the invoice was issued. This will simply be today's date minus the invoice date. Use the date function TODAY() to obtain today's date (or you can use the NOW() function to produce date and time) and don't forget to format the column as numbers rather than dates if it is necessary.

Next you would like to know the status of each invoice. Is it within the discount period, is it open (that is past the discount period but not yet late) or is it late? But this depends on who the customer is so somehow it will be necessary to match the Customer_Number for



each invoice in the Invoices worksheet with a Cust_No in the Customers worksheet in order to find out the appropriate discount and late dates for that customer.

For this Status column you need a formula that will implement the following:

*If days open is less than discount days, status is discount,
else if days open is less than due days, status is open, else
status is late.*

Or writing this using more Excel-like syntax you would have:

```
IF (Days_Open <= discount days for this customer, "discount",  
    IF(Days_Open <= days due for this customer, "open", "late"))
```

In this formula it is assumed that a name, Days_Open, has been defined for that data.

The quantities in italics are the remaining difficulty. You need a function that will use the customer number from a row in the Invoices worksheet in order to find the correct row in the Customers worksheet and return as its result the value from a different column (the Discount Days or Days Due columns) as its result.

A straight forward way to do this is simply name the columns of the Customers worksheet and perform 2 LOOKUPS.

```
=IF(Days_Open<=LOOKUP(Customer_Number,Cust_No,Discount_Days),"Discount",  
    IF(Days_Open<=LOOKUP(Customer_Number,Cust_No,Days_Due),"Open", "Late"))
```

There are some advantages, however, in building tables of data and using a single function call to perform 2-way selection. In our LOOKUP example we specified different explicit names for the range of returned values. Dealing with the table as a whole, and specifying which column to use is accomplished by the VLOOKUP function.

Enter a column heading for the status column and format it. Before beginning to create the formula for the first cell you should define various names - one for the whole table of data in the Customer worksheet (you haven't defined a name for a table of data before, but it's no different than defining a name for a column, except that you select the table), and others for Cust_No, Amount and Days_Open in the Invoices worksheet.

The VLOOKUP function requires as arguments a Lookup_value (this will be a value from the Cust_No column), a Table_array (this will be the table of data from the Customers worksheet), and a Col(umn)_index_num(ber) representing which column to use for the value returned by the function.



In one case the Col_index_num should be 3 (for the Discount Days column) and in the other case it should be 4 (for the Days Due column). The formula will look like this, assuming you have similar defined names:

```
=IF(Days_Open <= VLOOKUP(Customer_Number,lookup_table,4), "Discount",  
    IF(Days_Open <= VLOOKUP(Customer_Number,lookup_table,5), "Open",  
    "Late"))
```

The next task is to calculate whether a discount is applicable, or whether a late fee applies, or neither (in which case the amount in the Amount column is owed). Enter a heading for the new column and create the formula to do this. The formula is the implementation of the following:

*IF status is Discount, calculate discount amount (using discount rate for this customer),
OTHERWISE IF status is late, calculate late fee (using late fee rate for this customer),
OTHERWISE do nothing*

The discount amount is meant to be subtracted from the amount owing, whereas the late fee is meant to be added, so you should force it to be negative by multiplying by -1.

In Excel syntax and assuming names have been defined similarly you end up with

```
=IF(Status="Discount", -1*VLOOKUP(Customer_Number,lookup_table,3)*Amount,  
    IF(Status="Late",VLOOKUP(Customer_Number,lookup_table,6)*Amount,""))
```

The final empty double quotes represent the *do nothing* calculation when the account is still in the open period.

Summary

What has been achieved by organising the worksheets in this way? Certainly you had to write some quite complex formulas and that might be considered a disadvantage. The advantage, however, is that customer information is localised to the Customer worksheet where each customer occurs only once. There may be many occurrences of a customer in the Invoices worksheet (since each customer may have made many purchases), but in each case the information that is required to perform calculations can be selected from the Customer worksheet. This is a great convenience if you decide to change the discount rate for a customer for example. With this design you only have to change it in one place and the change is immediately reflected in the calculations.

One important thing to note, however, is that using literal values to specify the column index number is really a misuse of VLOOKUP. The whole point of this argument being a number is that it can be calculated!



Exercise 5 – Reading a Schedule with VLOOKUP

Open Exercise 5 (ch6_Ex5) in the Support Files (Chapter 6) of the course website. This model implements a schedule for a sports league. There are 3 worksheets in this model – the Comments worksheet; the Master Schedule (figure 6.9); and a worksheet that acts as the interface. Here a user can enter their team number in the cell named My_Team and the week in the cell named Week, and the sheet will return the number of their opponent for that week. All that's required is to write the formula.

Figure 6.9 - the Master Schedule worksheet

	A	B	C	D	E	F	G	H	I	J	K
1	Team	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10
2	1	2	3	4	5	6	7	8	9	10	
3	2	1		3	4	5	6	7	8	9	10
4	3	10	1	2		4	5	6	7	8	9
5	4	9	10	1	2	3		5	6	7	8
6	5	8	9	10	1	2	3	4		6	7
7	6	7	8	9	10	1	2	3	4	5	
8	7	6		8	9	10	1	2	3	4	5
9	8	5	6	7		9	10	1	2	3	4
10	9	4	5	6	7	8		10	1	2	3
11	10	3	4	5	6	7	8	9		1	2

The Master Schedule is arranged with team numbers ordered down the first column and the weeks ordered across the first row. Having two indices allows 2 dimensional lookups. VLOOKUP is ideally suited for this kind of task. The Lookup_value is My_Team. Excel will look for this value in column 1 of the table (called Schedule). So if the user enters 8 as their team number VLOOKUP will locate this value in row 9 of Schedule. So it will return one of the values in row 9 – but which one?

This question is answered by the Col_index_num. The value entered in Week can be used, but notice that data for Week4 is actually in column 5 of the table. So the formula required is:

=VLOOKUP (My_Team, Schedule, Week+1)

The function HLOOKUP can be used to perform the same sort of 2 dimensional operation except that it searches in the top row of the table to find the right column and uses the Row_index_num to locate the appropriate row.

It will frequently happen that the data we need in our table won't be as easy to access. For example, the titles of our columns might not be numeric (Week1, Week2, ...) Exercise 6 demonstrates the use of nested functions to handle such a case.



Exercise 6 – Using VLOOKUP and MATCH

Open Exercise 6 (ch6_Ex6) in Support Files (Chapter 6) of the course website. This model starts an analysis of the results of a campus-wide literacy test. It contains a short list of student names, the faculty with which each is associated, and their score on the test.

The faculties, however, have different ways of assigning letter grades to the scores. The 3 different systems are detailed in GradeTable on the Table worksheet (Figure 6.10).

Figure 6.10 - the GradeTable

	A	B	C	D
1		Arts	Science	Graduate
2	0	F	F	F
3	40	E	F	F
4	50	D	Pass	F
5	55	D+	C-	F
6	60	C	C	Pass
7	65	C+	C+	Pass
8	70	B	B-	C
9	75	B+	B	C
10	80	A	B+	B
11	85	A	A-	B
12	90	A+	A	A
13	95	A+	A+	A

Your task is to create a formula to produce the Grade column on the Class worksheet. It requires the lookup of the Score in the GradeTable and returning the letter from the column that contains grades for the student's faculty. One way to attempt this is with a nested IF, like this:

```
=IF (Fac="Arts", LOOKUP (Score, name of the first column of the table, Arts) ,  
    IF (Fac="Science", LOOKUP (Score, name of the first column of the table, Science),  
        LOOKUP (Score, name of the first column of the table, Graduate)))
```

But there's an easier way! The MATCH function will return the position in a list at which it finds a search item. In a way it does the first part of a LOOKUP, i.e. it finds lookup_value in the lookup_array and simply reports the position. So we can use MATCH to find the student's faculty in the list of faculties and use that as the Col_index_num in a VLOOKUP.

By now you should be able to construct this formula yourself. You will need to use the optional argument in MATCH to enable Excel to search the unsorted list of faculties.



Exercise 7 – The INDEX Function

We frequently encounter situations in which we need to search a table of data on different keys. For example, a Student Records System locates information by using Student ID, but will also allow searches by surname or even Social Insurance Number (SIN). Figure 6.11 shows part of such a data table. The fictitious data is in Exercise 7 (ch6_Ex6.xlsx).

Figure 6.11 – a Student Records data table

	A	B	C	D
1	Student #	Last Name	First Name	SIN
2	200599035	Leung	Afshine	554386021
3	200629527	Leung	Alex	563240108
4	200668145	Lewis	Alex	465637569
5	200705004	Acerbi	Alexandria	311099586
6	200735910	Li	Alice	593572927
7	201197201	Lee	Alison	364251231
8	201207497	Lindsay	Amanda	595878125
9	201227297	Liu	Amankumar	203604506
10	201228923	Lee	Ambrose	484385421
11	201235506	Lau	Ana-Mariea	433759404
12	201239359	Lai	Anca	435640730
13	201256585	Kwan	Anh	512142776
14	201257260	Kupczynski	Anna	238734027
15	201308022	Liu	Anna	291411734
16	201325216	Mah	Arale	439178856
17	201345602	Kohl	Arie	460084012
18	201352541	Mao	Arlene	569614562
19	201417351	Mariani	Bart	269147056
20	201419282	Kidd	Brandon	577769336
21	201438621	Martin	Brandon	357187626
22	201453224	Massiah	Carlos	424213315
23	201518182	Amin	Caroline	283507081
24	201520113	Khona	Catherine	407582879
25	201596626	Andrews	Chi	555985721
26	201616168	Mattei	Ching	420300594
27	201820380	Mayer	Ching	256670754
28	201829241	Meyers	Christine	448009858
29	201844927	Mohabir	Chucheng	389352985
30	201861954	Atefi	Cindy	320196785

When the ID is known, LOOKUP can be used to retrieve data from a specific column or VLOOKUP can retrieve data from a column whose index number can be calculated, but



since VLOOKUP always searches in column 1 and LOOKUP requires a sorted search vector, how could data be retrieved for a particular SIN?

The solution is the INDEX function which returns the value in a specified cell within an array. This is the syntax of the function:

INDEX (array, row_num, column_num)

- array is a range of cells or an array constant
- row_num is an integer that selects the row in array from which to return a value
- column_num is an integer that selects the column in array from which to return a value

As with VLOOKUP, the ability to calculate the row_number and column_number gives this function great flexibility, and it's also true that MATCH will frequently be the best way to perform such calculations.

Examining the names in ch6_Ex7.xlsx reveals that each column has been named, as have the table itself, and the row of titles at the top. The object is to create a search page that allows the user to enter a student's SIN and retrieves the data from the other 3 columns. Figure 6.12 shows how such a sheet might look.

Figure 6.12 – a Search

	A	B	C	D
1	Criterion			
2	SIN	435640730		
3	Student #	201239359		
4	Last Name	Lai		
5	First Name	Anca		

Write ONE formula that will calculate all three other fields based on the SIN entered. That is, create a (the ONE) formula and enter it into cell B3. This formula will then be copied and pasted to cells B4 and B5. During the copy/paste process the formulas will be "adjusted" by the way Excel handles relative cell references. Exploit this in the design of your original formula.

This is the general goal of designing spreadsheets. Plan the layout of your spreadsheet carefully and exploit the patterns so that the formulas are adjusted by Excel as they are copied and pasted, rather than you typing minor variations over and over or editing the formulas afterwards. Remember, your goal is to be as lazy as possible and let the computer (Excel) do the work!