Analysis of Data using Excel Pivot Tables

DOCUMENT SUMMARY SHEET

Document Type:	How To
Document Title:	Excel Pivot Tables
Document Summary:	This is the Adaxa HowTo Template which is used for the creation of HowTo documents. Proposals, white papers, reports, etc. will use the Adaxa Report Template
File Name:	Excel Pivot Tables
Created on:	Monday, 16 January 2012
Created by:	
Last Modified on:	Tuesday, 7 February 2012
Last Modified by:	

NOTES

1. Responsibility is disclaimed for any loss or damage (including but not limited to damage resulting from the use by the client of the document) suffered by any other person for any reason at all including but not limited to negligence by ADAXA Pty Ltd (ADAXA).

2. Whilst this document is accurate to the best of our knowledge and belief, ADAXA cannot guarantee the completeness or accuracy of any description or conclusions based on the supplied information.

3. The recommendations contained in the document are advisory and ADAXA has no responsibility for the management or operation of any recommendations that may be implemented by the client.

4. This document is licensed under the terms shown at http://creativecommons.org/licenses/by-nc-nd/3.0/au/legalcode.

Table of Contents

1 Excel Pivot Tables

1.1 Introduction

This article was written some time ago and many of the references to system sizes and capability are now outdated. However this in no way detracts from the overall content of the article.

1.2 How to use Excel as a data analysis tool for data stored in a database

Spreadsheet programs have grown in functionality, power and complexity over the years since they were introduced so that most users only ever need to use a small portion of the capabilities of their spreadsheet product of choice.

One of the newer features of Excel called *Pivot Table* is extremely powerful and not as well understood as other longer-serving Excel features. A pivot table enables a number of records in an Excel Worksheet to be summarised in many different ways. It can also be extended to operate on data external to Excel stored in a variety of databases.

The primary requirement for a Pivot Table to be useful is that the data to be analysed or reported on be generally in the following form:

table 1: Raw Data

Many readers will recognise that this form of data is typically what an accounting system delivers directly from its various accounting ledgers. The defining feature is that it is transactional in nature with each row representing a unique record. I say "typical" because some accounting systems may require two files or tables of data to be "joined" to derive all the columns of information shown above.

The data above was turned into the report of Sales by State and Sales Rep by Month in table 2 using an Excel Pivot table and about 11 mouse clicks.

Table 2 - Sales by State by Sales Rep by Month

A further 5 mouse clicks then converted Table 2 - Sales by State by Sales Rep by Month into a report of sales by Product by Sales Rep.

table 3 - Sales by Product by Sales Rep by Month

Many finance people will find the Pivot Table useful for satisfying ad-hoc requests for reports. Some of the default layouts adopted by Excel can be very simply changed by simply over-typing the words in the Pivot Table with your own preferences. This enables words like "sum of qty" and similar to replaced with better choices without changes having to be made in the underlying data.

Table 4 below is the data from table 3 with the "sum of qty" removed from the Pivot Table and other slight "tidy up" attempts.

table 4 - Sales by Product by Sales Representative by Month (tidied up)

In addition to the changes that can be made in the Pivot Table by over-typing field names and labels, the full formatting functions of Excel are available to insert headers and footers for page numbering and dating of reports.

1.2.1   Drill Down to Transactions

If you need to know what transactions make up any number within the Pivot Table just double click on the number. Excel will add a new worksheet and generate a list of the transactions which constitute the balance. This drill down function is extremely useful, especially when dealing with very large data sets.

1.2.2   Preparing Data for Use in a Pivot Table.

Unless you are lucky enough to already have the data you require in Excel you will probably need to do some work to get it into Excel. You will also probably find that you want to group and summarise your data based on criteria that you may need to extract such as a department number that is contained within an account number.

1.2.3   Getting Raw Data into Excel

Many financial systems provide the capability of exporting data from the ledgers to one the various formats supported by Excel such as comma or tab separated values or directly into an Excel Worksheet. For data which is not available in these formats - typically from legacy systems on mini-computers and mainframes it is frequently possible to get the required data in the form of a report intended for printing which has been saved as a file or perhaps e-mailed. As such reports tend to be laid out with a fixed columnar format Excel will enable the file to be opened and broken into columns at the desired location using its standard text import routines. It is worth looking at such text files using a fixed width font such as Courier which will immediately show the columnar (or otherwise!) nature of the data.

An Excel Worksheet can contain up to 16384 rows. A typical report format from a legacy system uses around 60 lines per page, usually with headers and footers and titles. This means that the spreadsheet could "read in" the contents of a report of up to 275 pages with no prior reformatting of the data.

1.2.4   Manipulating Raw Data to work with a Pivot Table

Another useful feature of Excel when trying to perform these data manipulation tasks is the "Text to Columns" option available from the "Data" menu drop down list. As the name indicates this

provides the capability of taking a single column and breaking it into many columns using the same method that Excel uses when opening a text file. This feature can be very useful when information has been obtained with say a 12 character account code and the report you need requires grouping based on the two characters commencing at position 3 of the account number.

1.2.5   String Manipulation to create new Columns

It is also useful to be aware of the String handling capabilities of Excel when preparing data to be used in a Pivot Table. Excel provides the capability to find characters in a cell commencing at a location and for a particular length. This can be extremely powerful when trying to create a column containing a particular piece of information such as a Department number buried in an account code or an invoice number buried in some narrative description provided by an accounting system. Remember to give each new column a name at the top of the Worksheet.

The most useful string operators are "left", "mid" and "right". Their use is well described in the Excel manuals. Table 5 shows a brief example.

Table 5 - Use of String Operators

It is sometimes useful to use the string operators to break up a column rather than using the "text to column" function because the latter will return a number where it can whereas the former always returns a text string. If a section of a column that you are seeking to break up starts with a zero, the string operator will retain the leading zeroes which can sometimes be beneficial.

After the breaking up of "account number" as shown above it would be very simple to use the Pivot Table to create reports summarised by Division, Code and Cost item. The real power of Pivot Tables then starts to become apparent since it is now possible to group and summarise automatically on any combination and in any order based on Division, Code and Cost item.

1.2.6   The Pivot Table Wizard

When Pivot Table is selected from the Data Menu an Excel Wizard guides the user through the steps necessary to create a Pivot Table. These steps are briefly described below.

Step 1 determines where the data for the Pivot Table is coming from, as you can see there are a range of options. Options 2, 3 and 4 will be commented on later. For the moment we are creating the Pivot Table from a list in an Excel Worksheet.
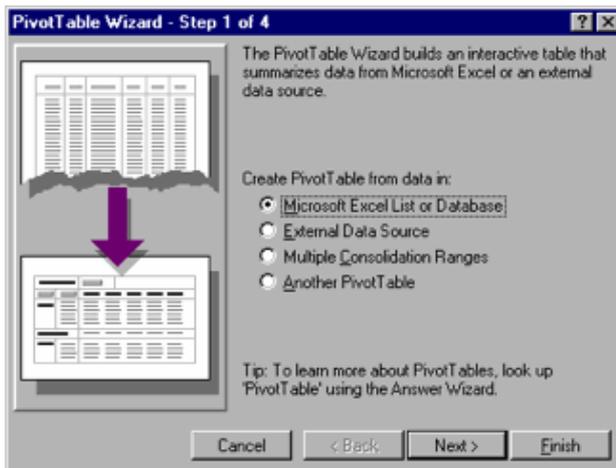
Step 2 asks for the exact range of the data to be defined. In this case the range has been defined using Excel's / Insert / Name / Define / routine with the name "raw" to predefine the range of cells that are of interest. The actual cell range could also have been typed in or selected from the current or other Worksheet in the current or other Workbook.
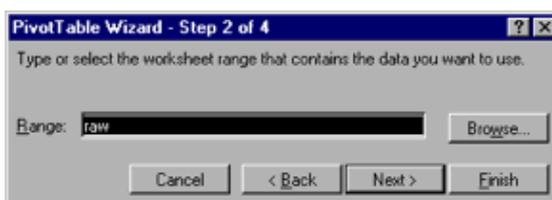
Step 3 asks how you want the data to be displayed. In the example below, the "Page" option has not been used. An appropriate use would have been to mouse drag "State" on to "Page". The "Page" option creates a drop down list of States within the Pivot Table allowing you to select only "Products" and "Reps" for a selected "State"
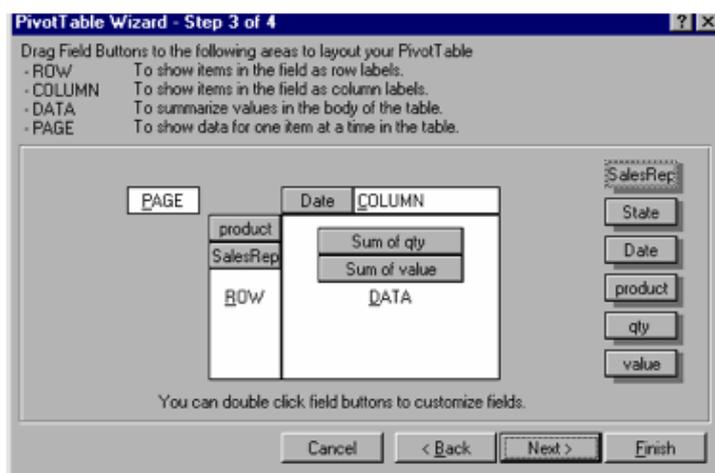
Step 4 allows further formatting options which can sometimes be helpful. In most cases it is sufficient to simply click on the finish button.
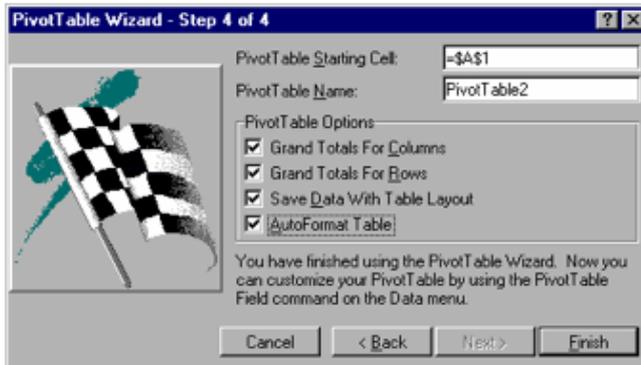
Your Pivot Table will now be created by Excel. On successful creation of the Pivot Table, the Pivot Table Toolbar should have been created on the Worksheet. (If it was not, then it is accessible from Excel's / View / Toolbars / Query and Pivot / option). It looks as follows:

One final step was required to produce the report referred to as table 2 above.

This required the use of the Pivot Table "Group By" function. This is accessed most easily by the Query and Pivot toolbar by clicking on the right pointing arrow which then brings up the "Grouping" options box below. The "Months" button was clicked on to summarise the report down to one column per month.
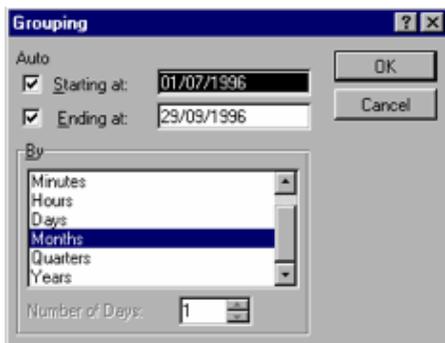
Where data exceeds one year in duration it is necessary to click on both years and months or the months will be aggregated. Excel seems to have no concept of Financial Years only Calendar Years.

1.2.7 More about Data Manipulation

Reference was previously made to the reporting and analytical power of Pivot Tables which was able to be exploited by creating separate columns of information which could be used for

summarising information. The example used above showed the splitting up of an account code into its components, Division, Code and Cost item.

When a Pivot Table is created from the data, report labels are created using the numbers for each of Division, Code and Cost item. These labels may then be over-typed to put meaningful names on the report. This is not very helpful if you have say 100 Divisions and you don't wish to type each label.

A way to overcome this problem is to create a Worksheet which contains the Division Code and the Division name. This information can usually be printed from somewhere and then imported into Excel as described above. Such a table may look as follows.

Table 6: List of Division codes

	A	B
1	Div	Div Name
2	123	NSW - Sales
3	124	NSW - Marketing
4	125	NSW - Accounts
5	126	NSW - Premises
6	127	NSW - Computer Facilities

If a column is then inserted in the Worksheet containing your Pivot Table data beside the column containing Division, the Division name can then be made available to your Pivot Table with little effort and in a self maintaining fashion.

1.2.8   The Vlookup function

The Vlookup function can be used to read the Division code in your Pivot Table data, look this value up in the table of Division Codes and Division Names and then insert the Division Name back into your Pivot Table data. The name will then be available to your Pivot Table based report with no further effort.

Before using the Vlookup function it is beneficial to use Excel's / Insert / Name / Define / routine in the Worksheet that contains your equivalent of "table 6: List of Divisions" to declare the fixed range that will contain the Division names to be looked up. This prevents errors arising when copying formulae in the spreadsheet if you forget to make the cell references absolute by some other method. I will use the name "Lookup" as the defined range.

You should also sort the list of Division names by their code to prevent spurious results which otherwise can occur. (Excel's Help covers the issues in some detail)

Step 1, Click on the "fx" (function wizard) which will be visible if you have the standard toolbar selected. (/ View / Toolbars / Standard / to switch on).

Step 2, Click on Vlookup and the following Function Wizard will appear will appear.
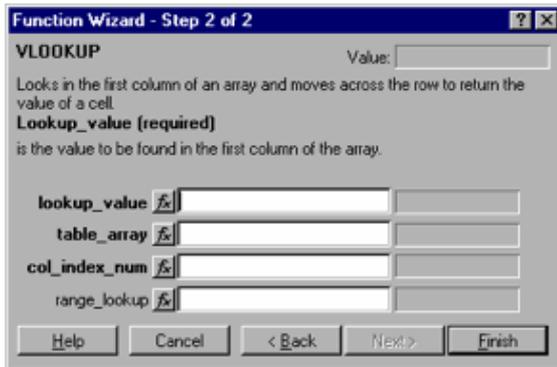
In the cell **Lookup_value**, enter the cell reference for the fist Division in your Pivot Table data sheet.

In the **table_array**, type 'lookup" being the previously named range.

In the **col_index_num**, type "2" since the value you want is in the second column of the table (see table 6 above).

In the **range_lookup**, type "false" otherwise any missing Div Names in your "lookup" table will be given the next closest approximation.

Now copy the formula from the cell just entered down the full range of the data in your Pivot Table data worksheet and the Div Name will be inserted beside every Div Code in the table.

1.2.9   Redefine your Data Range and Refresh your Data

To be accessible to the Pivot Table, the "Div Name" must be within the range of data that the Pivot Table looks at. If it is not then go back to the Pivot Table wizard and declare the new range with the name "raw" then click on "finish".

Note that a column added within the constraints of an existing range reference will be automatically included, one added to the right hand side will not.

Mouse-click **!** on the Pivot Table toolbar to force a re-reading of the data if the column you created did not require a range redefinition. The Pivot Table will not know about the new column till this has been done. Your mouse must have selected a cell within the Pivot Table bounds for this to work.

The Pivot Table will now have available a list of Div Names that can be used in the Pivot Table without requiring any over-typing.

1.2.10   Table Joins

The above section describes a method by which Excel can be used to perform a "join" or a "relational cross" on two tables. We started with a table of transactions, one column of which

---

contained a "Div Code" and we looked up the name associated with "Div Code" in a separate table and joined it to the data in the first table. This is the equivalent of a left outer join in SQL jargon.

I am sure you have formed the view that there is nothing very clever in this but that it could be pretty useful sometimes. If you have files of data available to you containing information of interest you can usually access the information using Excel's Microsoft Query tool. Providing the data is in the format of a table.

1.2.11   Microsoft Query

Microsoft Query is a tool that comes bundled with Excel. It enables you to extract information from tables of data held in many forms back into an Excel Worksheet. When so used, the maximum number of rows of information that can be dealt with is 16384 since this is the maximum number of rows that can be fitted in a worksheet.

(If you can't find the "Get External Data" in the / Data / menu in Excel it probably has not been included in your install routine. Rerun the MSoffice install process and select the appropriate option.)

MsQuery is also used by a Pivot Table to attach to an External Data Source (Pivot Table Wizard, Step 1 of 4, second radio button). When used within a Pivot Table the data is not actually written into a Worksheet in the normal way and the 16384 row limitation of Excel does not apply. It seems that the limitation becomes memory, disk speed and patience.

A Pentium 100 with 32 mb of memory seems happy enough with data sets of 80,000 records. It may seem that 32mb of memory is excessive but with memory costing around $150 for 16mb, it is cheaper to buy some memory for your senior finance people than to get your IT department to write programs to achieve the same purpose.

1.2.12   Data Sources

MsQuery uses predefined data sources to attach to external data tables on local and remote computers. These data sources are accessible from the Control Panel, ODBC section. The screen shot is as below:
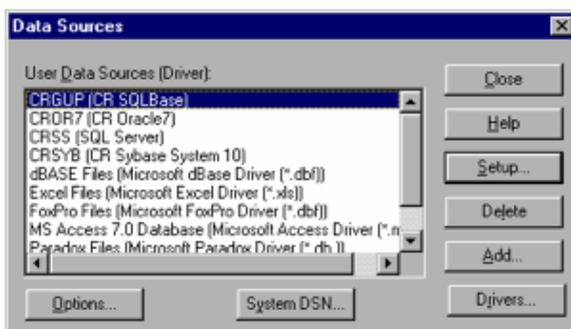
As you can see, MsQuery allows you to define connections to many different types of data files ranging from simple flat text files to sophisticated databases.

Defining database sources is beyond the scope of this article and readers with an interest are referred to the Excel manual. It is not particularly difficult or complex but when you try to connect to these data sources you should be prevented from doing so by the security around the database. As a result you will almost certainly need to involve your IT staff in the process.

1.2.13   Using MsQuery to access data

After you choose the External Data option in the Pivot Table wizard, MsQuery will start up. It looks as follows:
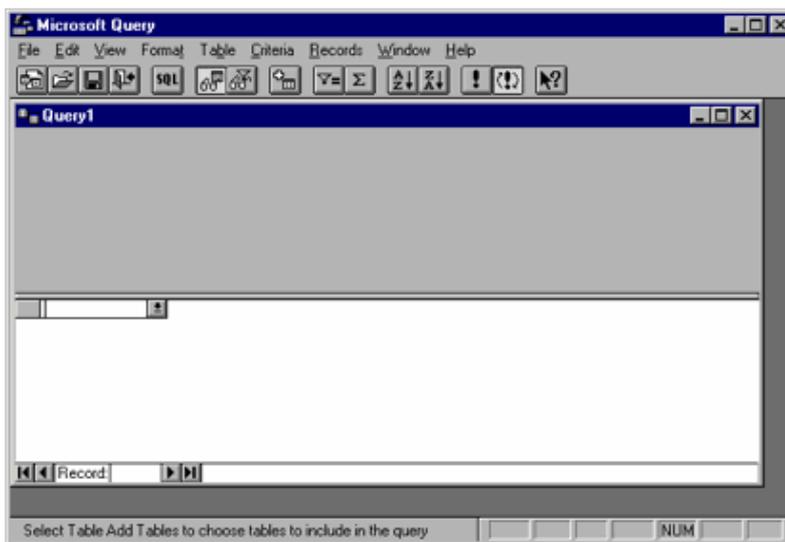
You will then be asked to choose a Data Source: In the example below a data source has been defined to point to the MSAccess Northwind demonstration database.
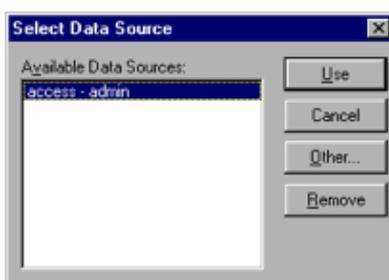
You will then be asked to add the tables you are interested in to the query:
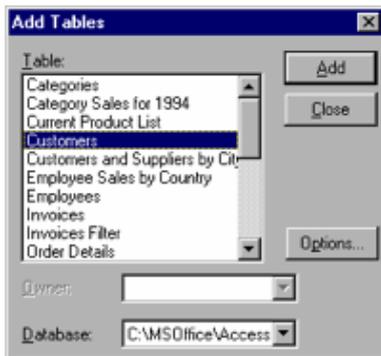
You will then be shown the tables in the query and if there is a link between the tables in the database this will also be shown by a line connecting a field in each table.

You must then decide what selection criteria if any you wish to apply to the data. This is done by clicking on / Criteria / Add Criteria / to open up the dialogue box below. In this instance a field of "OrderDate" has been selected from the table "Invoices" and criteria limiting the data to invoices with a date between 01/7/93 and 31/3/93 applied.
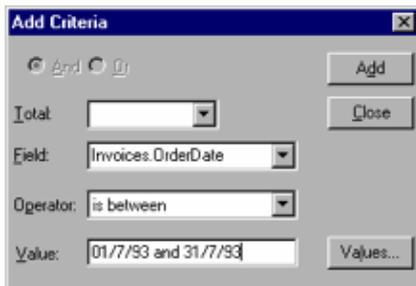
After closing the Add Criteria Box, select the fields from the table "Invoices" and the table "Customers" by clicking on those that you wish to appear in your Pivot Table. A subset of the data will appear in the MsQuery window to show what you data you will receive.
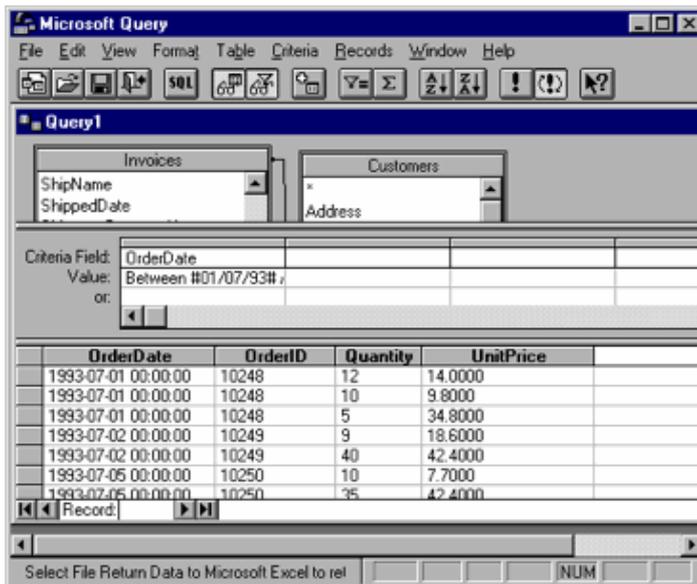
Now click on the arrow pointing through the door or select / File / Return Data to Excel / and then step 3 of the Pivot Table wizard will appear. Click on "Next" and the actual query will run and then return the data to Excel and step 3 of the wizard will then prompt you to choose the layout of your Pivot Table.
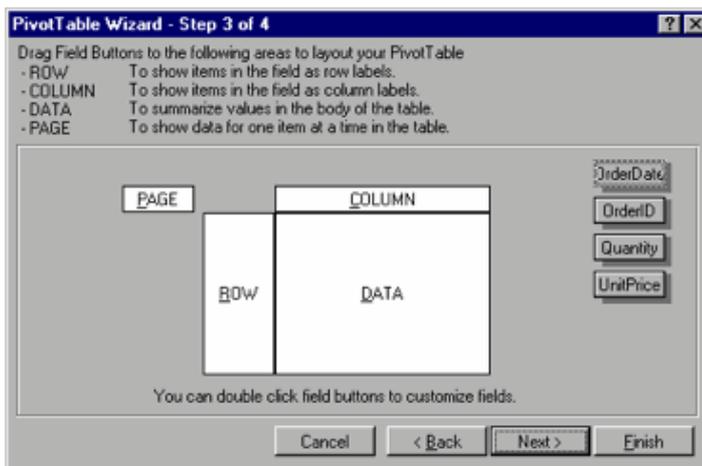
Lay out your pivot table fields as desired then click on "Finish" and your report is available.

1.2.14   Changing your Pivot Table Data

Once data has been returned to the Pivot Table it can be refreshed by re-running the Query by simply pressing the "**!**" button on the Pivot and Query toolbar.

If you need to change the Query because (say) you have just passed a month end and are interested in the current months sales then click on the "Back" button in the Pivot Table wizard until you see "Get Data" in step 2.

Click on the "Get Data" button and you will be returned to the underlying query. Click the button on the Toolbar which shows a filter and a pair of binoculars to display the query criteria and then edit the criteria as required. (Tip - it sometimes easier use "Add Criteria" and add a new criteria and then delete the superseded criteria than it is to edit the existing criteria)

1.2.15 Summary to Date

I hope that the reader will accept that it is possible using the Pivot Table in Excel in combination with data stored in an external database to simply develop reports that are:

- of instant use,
- are capable of being easily re-arranged,
- that provide drill down to the underlying transactions, and
- that can easily have their selection criteria changed by the user.

Furthermore, these features are available with no additional expense for existing Excel users and with little additional training. I would also contend that this functionality starts to move us into the bottom end of the benefits that vendors selling Data Warehousing applications would suggest as reasons why we should consider their products.

2.1   Data Warehousing for Beginners

Data Warehousing is a means by which an organisation's transaction processing and its analysis of data is separated typically onto different machines and frequently on to different database products. It is argued that the database structures that suit transaction processing are inappropriate for data analysis where very quick roll-up of information on lots of different criteria will provide information for fast and effective decision making. It is also argued that the information in the data warehouse will probably come from many sources not just the financial system.

There also seems to be a significant amount of cynicism about the benefits that can be derived from data warehouse projects compared to their fairly significant costs. There does not seem to be an accepted method of quantifying the benefits to be derived from data warehousing projects. Readers may care to refer to the article in CFO in July 1996 titled "Don't be fooled again" which discusses this topic in some detail.

Readers contemplating moving their organisations down the data warehousing route might well contemplate trying to do a very quick and simple prototype using tools that they probably have already purchased. It may not deliver the best of all possible outcomes but it won't cause a lot of unnecessary waste of time or money and much will be learned in the process that may assist in making further plans on a "real" data warehouse.

The following information outlines the steps that were undertaken in one such proto-typing project. This project has been partly delivered and is now generating a significant amount of feedback from users who are seeking to use the system in ways that were not contemplated when the project commenced.

The project was started in response to requests that information be provided to managers in a more convenient fashion than paper based reports and in a way that did not require managers to have to develop a lot of familiarity with a legacy accounting system in order to get the information they needed. Performance also was a significant issue. The methodology chosen was as follows.

2.1.1   The Prototype Data Warehouse

Tables of data were extracted from the VAX based financial system every night as tab delimited files using a mixture of Cobol programs and Datatrieve a DEC language used for manipulating databases.

These tables were automatically transferred by the FTP file transfer utility each night from the VAX to a Pentium PC running Windows NT and Microsoft's SQLServer.

A batch process is run on the PC each night which reads the data from the tab delimited text files into SQL Server. SQL Server then indexes the files and also creates some summary level tables.

The process re-creates the whole database from scratch every night because the whole process only takes about 20 minutes and therefore has not warranted the programming effort involved in creating a process to append new transaction to the existing database.

The tables in the database created contains:

- 500,000 job related transactions (being every job related transaction since 1989)
- 10,000 job header files
- 100,000 job component
- 2000 Customer files
- 3000 Accounts Receivable transactions
- 3000 Vendor files
- 4000 Accounts Payable Transactions
- about a dozen other small tables of data from the legacy system plus some calculated summary tables.

Job budgets are being added to the system at present which will add another 200,000 records.

The system at this stage enabled users of Excel and MsQuery to generate their own queries joining multiple tables and applying relevant criteria. They also had to understand one-to-many and many-to-one type relationships and the impact that encountering a "null" in the database would cause to their information. This meant that a reasonable level of technical skill was necessary to generate reliable reports. The perception was that this skill level was more than that which could be expected of most people without a large training program. This problem was overcome using pre-packaged "Views" of the data.

2.1.2 'Views' of Data

A feature of SQL Server (and other database products) that is used extensively in the prototype is what is known as a 'View'. A View is a predefined query that appears to the users of a system as if it were a table and saves them from having to go through the steps of selecting multiple tables and then defining the basis on which the tables are joined. From the user's perspective it is just another table which just happens to contain everything they need to know to answer a particular question. Views can also have data security rules applied to them.

A View is constructed from SQL code which says which tables are to be joined and how, which fields are to be displayed and what these are to be named. It can also do all kinds of pre-filtering and apply calculations to predetermine values which are not expressly stored in the database itself.

An example of this from the current prototype is a requirement to list every sales transaction on every job which is classified as being associated with a government purchasing scheme and on which a commission is payable to the government authority.

Another example is a sales commission calculation using very convoluted rules for the calculation.

Both views took less than an hour to construct and solved long standing organisational problems which had arisen from the inflexibility of the legacy accounting system.

2.1.3   What is the Data Warehouse used for?

The system is presently used to by managers and admin staff to monitor and report on the progress of jobs, it is being used by sales managers to monitor sales and order received values.

Access to the system is exclusively by Excel Pivot Table. Staff have adapted well to accessing the system by using Excel. There has not been a higher training requirement to have people comfortable with using Excel rather than printing out endless reams of paper reports.

The ability of the Pivot Table to take the data and present it in a particular fashion to answer a question without having to define and print a whole new set of reports is now regarded as essential.

2.1.4   Performance

Performance of the system has been impressive. The hardware used has only recently been replaced by a dual Pentium 120 machine with 64mb of memory. A query such as "give me every transaction associated with job number nnnnnn" takes about 20 seconds to return about 4000 rows of data. The principal delay seeming to come from the 10 megabit Ethernet network rather than server itself. The same query run from the server seems almost instantaneous.

2.1.5   Future Plans

The prototype will soon have the whole financial and time sheet databases for the last 6 years added to it which will increase the number of transaction records to around 2,000,000. Extra hardware is expected to be required to cope with the volume and additional users - although the cost will be quite small. Another 256mb of memory is sub $3000 and an extra 10 gigabytes of disk space is around $2,500. An upgrade to two 200mh Pentiums in lieu of the two existing 120mh Pentiums will cost whatever it costs when we decide to do it. Currently it would be around $1000 each and the two 120mh would be used elsewhere.

Clearly the number of transactions to be recorded in the data warehouse is of significant importance in deciding whether a relatively modest prototype such as that described above can be of value in an organisation. The organisation described is primarily a seller of computer hardware, software and services and has a turnover in excess of $60m and some 300 staff. The data kept on line is every transaction since 1989, not because it is needed, but because it was easier to just leave it all there. (You can justify a lot of shortcuts in a prototype!)

2.1.6   Was the Prototype a Success?

Some indicators as to whether the system is perceived as having added value are as follows:

2.1.6.1   Successes

The system stopped recently due to a hardware failure. There was much grumbling from staff about how they could be expected to perform their jobs properly if the information was either unavailable or not up to date as at 4:00am of that day.

There are requests for new views of the existing data to be made available to simplify some existing tasks and to facilitate reviews of work in hand. The following is an example. A sales forecasting system generates a forecast of services sales and margin by month and skill type. The difference between sales and margin is cost. The group responsible for delivery of each type of service in each state is known, as is the average cost rate for each delivery group. With this information it was relatively simple to create a View of services staff requirement in hours automatically as a result of the sales forecast and make that available via the corporate data warehouse where it can be analysed in depth by the services and resource managers.

Finance people now automatically use the data warehouse to get the current position on any matter relating to sales and cost of sales even though they are the owners of and experts in the company's core financial system. The reason is flexibility in the analysis of the data and stunning performance compared to the legacy system.

2.1.6.2   Shortcomings

The system's major shortcoming derives from its success. Staff rapidly adopted the system as their main source of information about the current position on any matter relating to sales and cost of sales. Complaints are now emerging that the data needs to be able to be updated more than once per day because people detect errors in the underlying data when performing reviews and normal management tasks, they arrange for correcting entries to be processed and then have to wait to the following day to see the results of the adjustments.

Excel Pivot Tables are not flexible in allowing reports which show both monthly and year to date figures on the same report. Essentially you can have one or the other unless you pre-process the data to create a view with separate columns for (say) Month Actual, Month Budget, YTD Actual and YTD Budget. The variances can be computed using "calculated Fields" in the Pivot Table.

2.1.7   Summary

The prototype is regarded as a success and worthy of further development. It is already regarded as a core system.

Investigation is occurring into the use of a report writer such as Crystal Report Writer or perhaps Microsoft Access to address some of the reporting inflexibility in the Pivot Table to do with month

and year to date type reports however Excel Pivot Tables will continue to be the primary reporting and analysis tool for accessing the prototype data warehouse.

3 Adaxa's Offices and Contacts

Information of a general nature about Adaxa and its services can be found at www.adaxa.com or obtained by sending an email to info@adaxa.com with a description of the information that you would like to receive. If you are an existing client and wish to initiate a request for software support please send an email to helpdesk@adaxa.com with as much detail as possible about the nature of your support request.

For all other information please contact the Adaxa office nearest to you.

3.1   Australia

Address: 10 Kylie Place, Melbourne, Victoria, 3192, Australia

Contacts: Office

- 1300 990 120 (Within Australia)
- +613 9510 4788 (Outside of Australia)

3.2   New Zealand

Address: 73 Boston Road, Mt Eden, Auckland, 1023, New Zealand

Contacts: Office

- 0800 232 922 (Within New Zealand)
- +649 9744 600 (Outside of New Zealand)

3.3   United States of America

Address: 4400 NE 77th Ave, Suite 275, Vancouver, WA 98662, USA

Contacts: Office

- +1 760 576 5115