

BUILDING AN INTERACTIVE DATABASE AND RETRIEVAL SYSTEM WITH SAS/AF® V6.08 AND PROC SQL

Philip E. Friend
U.S. Department of Agriculture, Economic Research Service

Introduction

Technological developments during the last two decades invite us to re-examine old systems. Often there are more efficient, more powerful, and easier methods to perform tasks that were not available before.

Relational database design, structured query language, the personal computer, and graphical user interface techniques enabled us to redesign an old system to increase both its functionality and its user-friendliness.

Background

The Economic Research Service tracks U.S. concessionary trade and responds to researchers' and policy makers' requests for specific program information.

The original system, called PL480, maintained four SAS® files of historical yearly data. The Agency received additional monthly data on a quarterly basis from three other Federal agencies in various non-SAS formats. The new data was converted to spreadsheets and edited, and commodity and country codes were retyped to conform to the ERS coding scheme. The files were transferred back to the mainframe and converted to SAS data sets, where they were sorted, match-merged, and manipulated in other ways. Finally, the resulting files were concatenated with the existing historical data files. Thus, the data for the system existed in four files: value of goods shipped (in thousands of dollars) for fiscal year, value for calendar year, weight of goods shipped (in metric tons) for fiscal year, and weight for calendar year. Each record of historical data, in each of the four files, held the year, country code, commodity code, and data for each of several concessionary trade programs, as well as the total for all the programs. Records which contained world and regional totals, and totals for all commodities for each country and for regions, remained in the files. In addition, there were two SAS look-up tables, one for countries and one for commodities. An interactive mainframe CLIST was invoked which prompted the user for country and commodity codes. A report was then generated for those countries and commodities by a batch SAS program. The program code consisted primarily of DATA steps, Put statements, Merge statements, and PROC Sorts.

In early 1992, the agency that provided 85% of our PL480 system's data changed its system so that data would no longer be provided in the format that the old system required. At about the same time, a second agency, which provided about 10% of our data, began migrating their data

and data requests to a new system as well. The remaining data, which came from a third agency in the form of hard copy that was typed into spreadsheets by ERS personnel, was unaffected.

ERS researchers recognized the need to rewrite the old system. In addition to creating a new system to cope with the described changes, they hoped to take advantage of new technology to enable users to respond more quickly and accurately to requests. They also envisioned a system that could be expanded to include other sources and types of data and to provide different types of aggregations and outputs on demand. Finally, they hoped that other Federal Agencies could take advantage of the system to generate their own ad hoc reports.

Requirements

The new system had to build upon the existing historical annual data while adding new data which would henceforth be maintained on a monthly level. Since new concessionary trade programs would become available from time to time, the new system would have to be able to accommodate the data for these new programs. The interface should allow the user to enter countries and commodities by name rather than by code, to allow users to build reports from pre-defined groups of countries, commodities, and programs, and to allow interactive building of such groupings. World regions, for example, would be pre-defined by a researcher to contain specific countries, and then a request for a report on that region could be generated by requesting the group name rather than by specifically naming each member country.

Additional functions would need to be provided to allow the selection of years (fiscal or calendar), quarters, or even months. Totals and subtotals would be included at the users' discretion, and these should be calculated from the component data. (A real problem with including totals records in a file occurs when errors are found and corrected in the data. If totals records are not changed accordingly, reports are generated in which row and column totals appear to be unrelated to the contributing data. Editing such files requires constant vigilance and recalculation.) Different output options would have to be allowed: print, VDT display, and some kind of electronic format. Spreadsheets were the likely choice for these researchers.

The system should be expandable to include new countries, new programs, and new types of trade data, which could become available from entirely different sources and in different formats. Users would have to be able to

modify output accordingly. Other Federal agencies should be able to use the new system to generate their own ad hoc reports, rather than relying on ERS economists to respond by telephone and FAX to various requests. Current (up to the month) data should be quickly available for analysis.

Analysis

Preliminary analysis suggested a relational database design with structured query language to select the data. The elements of the main table would be country code, commodity code, program code, weight of commodity, value of commodity, month and year. Inclusion of the two time elements would enable the user to generate fiscal year and calendar year aggregations as well as monthly, quarterly, crop year and other similar reports. Tables with many short records with the preceding elements could be joined with tables containing the various codes and their translations into countries, commodities, and programs. SAS's implementation of the structured query language (PROC SQL) could perform the necessary joins, and PROC Tabulate could aggregate values by classes (countries, years, etc.) and build the output tables.

A significant difference between the structures of the old and new systems would be the breakdown of each original record with data for many programs into many records each with data for a single program. Also, the separate records for weight and value for each specific combination of commodity, country, and program would be combined into a single record. Totals, wherever they might be found, would have to be eliminated. All of these steps would be taken to normalize the data.

I was familiar with SAS/FSP® and had written screen control language (SCL) programs for several SAS/FSP applications. Since SCL is also a component of the more malleable SAS/AF, it seemed that SAS/AF could be a viable interface. The SAS macro language added another potential tool for providing data dependent, data driven code.

During the initial phases of analysis, SAS was available on all of the platforms that ERS researchers normally had available to them: MVS/XA on an IBM 3090, CMS on an IBM 4381, and both MS/DOS and OS/2 on personal computers. Soon SAS 6.07, and then SAS 6.08, became available on all platforms except DOS. This suggested an ease of portability across platforms that was rather unprecedented and therefore welcomed. The final product could be run on the platform most accessible to the various agencies requiring the system (probably the IBM 3090), yet design, programming, and testing could be done in the cheaper and more manageable PC environment.

SAS/ACCESS® became a distant possibility too; it could potentially enable us to view data from many different sources, perhaps in their original different formats, as though they were one SAS data set with the precise elements required to produce the requested reports!

Thus, analysis suggested a solution consisting of a relational database built of SAS tables with simple and compound indices, PROC SQL, PROC Tabulate, and a GUI written in SAS/AF, all developed under OS/2 on a high performance personal computer for eventual porting to a mini, mainframe, or network.

Several problems also suggested themselves: Would PROC Tabulate bog down if a request drew large amounts of data? Would PROC SQL handle the number of simultaneous joins and unions that would be necessary? How much memory and other computer resources would be required for each of these procedures? In a Personal Computer environment, we have begun to expect fancy, intuitive memory intensive interfaces. How would the interface play on a mainframe? How expensive would it be? How quickly would SAS tables grow, and how would their size affect performance on any platform? Would they become unwieldy despite the use of indices?

In fact, during the two years of development, several changes did occur: SAS/AF was removed from the mini computer; SAS/AF was enhanced; PC processing, speed, and memory improved; processors on both the mini and mainframe were upgraded; graphical user interfaces became increasingly popular; wide area and local area networks significantly expanded functionality and reliability; new countries emerged and new tradeable commodities were developed.

Application Development

The two calendar year historical data files (weight and value) were downloaded, decomposed, and re-combined into a single file with the structure described earlier. Codes and a decoding table for trade programs were developed. Indices on the files were created and a SAS PROC SQL program was written to perform the appropriate joins and summaries. This system of four SAS files and a compact (but dense!) PROC SQL program followed by a simple PROC Tabulate program succeeded in generating a report similar to one produced by the old system.

Macro variables replaced the hard-coded list of example countries, commodities and programs, and several %Let statements were included at the top of the program to simulate data coming from an interface. The results demonstrated the flexibility of the relational scheme and the functionality of SQL.

Extensive discussions with the researchers to define the interface needs and enhancements followed. We determined what information would be required by the system programs, how the user would supply the information, and what aspects of the requested output would be controlled by the user. I designed, on paper, several screens and a draft of the flow of their interaction.

At the end of 1992 the SAS/AF entry type "program" appeared to be the most flexible, so that was what was

used almost exclusively during development. One program entry was written to push the %LET statements, PROC SQL program, and PROC Tabulate program to the SAS processor with the Submit block. The other program entries provided the user interface menus and screens. A keys entry was created to provide hot keys for the system.

The initial tests passed the SAS/AF screen variables and SCL variables to the macro variables in the %LET statements; eventually the %LET statements were dropped and the screen and SCL variables were passed directly to the Submit block.

It happens that the sources of data (ERS and three other Federal Agencies) have used different coding schemes for their data. A code that for one agency meant Argentina, for another agency stood for the Philippines; a code for dry milk in one was soybeans for another. (These are not actual examples.) A translation table was therefore required for each of the four sources for countries, commodities, and trade programs. These three "input translation tables" convert the codes in the input data to the appropriate ERS codes. The original country and commodity look-up tables from the old system, plus a new table of programs and program codes that was created during the decomposition of the old PL480 files, became the "output translation tables." These three output translation tables convert the ERS codes into the country, commodity, and program names.

One agency might report their value of shipments in dollars and cents; another might report the information rounded to thousands of dollars. Thus two additional tables were needed to convert weight and value to some standard units of measure based upon a units code that would be provided with the data.

A major aspect of the entire process, incidentally, involved working with the resource agencies to determine precisely how our data needs could be met within the confines of their new systems. Two of the Agencies' systems, which together provided 95% of our data, were in flux; their data were being collected and entered differently from before; their systems were being altered to meet their needs, not ours; coding schemes changed even as both we and they were designing and implementing our systems. At least 75% of our work consisted in dealing with these critical inter-agency and data issues that were outside of our immediate control.

There would now be four data files (one from each of the Agencies), three input translation tables, the value and weight units tables, and three output translation tables. The PROC SQL program was supposed to create virtual subsets of the input translation tables based on the results of queries upon the output translation tables. A view was then to be created by performing the union on a series of intermediate views. Structured Query language was created to perform this type of processing, but PROC SQL would not handle the number of virtual joins in a single statement that was necessary. An inelegant solution was to create

views for each of the data files and concatenate them with a DATA step. This method worked.

The interface screens were improved as new versions of SAS/AF became available with new capabilities. An important feature was added with extended tables - pick lists that allowed the user to select names of countries, commodities, and programs from live SAS data sets, and to have the codes of the selections passed directly to the SAS program in the submit block.

In the final (Beta) version of the interface, the user selects the countries, commodities, and programs from pick-lists, or chooses from a selection of previously defined groups of these classes. The user tells the program whether to create a table of value or weight or both, and at what levels totals are to be generated. He or she enters the beginning and ending years for the report (which are verified by SAS/AF), chooses the time series (fiscal year, calendar year, etcetera), and then pushes a button to begin the report generation program.

Conclusion

The structure of SAS data sets lends itself to the creation of relationships among them. PROC SQL allows the building of virtual data views based upon links among different tables and other SQL views. The PROC Tabulate procedure is a powerful aggregation tool which generates reports with minimal code. SAS/AF and PROC Build permit us to create true graphical user interfaces with interactive screens, pick lists, push-buttons, and input parameter checking. Together, these facilities of the SAS system enabled us to construct a robust interactive database and report system.

SAS, SAS/ACCESS, SAS/AF, and SAS/FSP are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. * indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.