Journal of Official Statistics
Vol. 1, No. 1, 1985, pp. 65–73

Statistical Database Management

The Relational Approach Meets the Special Needs and Basic Requirements of a Statistical Office

T. Michael Jeays[1]

1. Introduction

The needs of a national statistical office in the field of database management are significantly different from those of commercial organizations and other government departments. Statistical data is both the primary 'stock-in-trade' and the product of such an office. It is present in large quantities and in very diverse forms, and careful management of it can make the work of the office more effective and efficient. There is a need to document and store data over long periods of time, and to balance the problem of evolving standards with the need for consistency from year to year.

Statistical offices need to be able to analyze data in complex ways that are not determined in detail when the data is collected. To reduce the need for scarce programming resources, it must be possible for analysts to work with minimal need for programming in languages such as COBOL and PL/I. They should be able to obtain most results by the use of packaged software operating against an organized and well documented database that contains most of the data known to the office.

Data must be stored efficiently, to minimize the costs of the storage media and the use of machine time when reading it. The organization of the database must use effective compression algorithms, and should avoid redundant copies. Retrieval processes should read only the data they really need. As most data files are created once and then ready many times, the database should be optimized for retrieval at the expense of features such as random access update with concurrency control that are often major features of commercial software.

The relational approach to database design satisfies most of the criteria listed. It avoids redundancy, and permits analysts to process files in ways that were not anticipated during the design and collection phases. Network relationships, built in during design, are common in commercial packages; they provide efficiency at the cost of flexibility. They are less appropriate to the needs of a statistical office. Very high level languages are coming into use that permit staff without programming skills to combine and process files in complex ways at short notice. It becomes feasible to experiment with the data. Normalized files can easily be interfaced with statistical packages, and the combination of a relational database package with modern statistical, graphical and tabulation software provides analysts with a set of extremely powerful tools.

In a modern statistical office, most of the functional steps needed in processing censuses and surveys can be built around a relational database, as a set of software packages that may be thought of as processing tools. These functions include frame maintenance, sampling, data capture, editing, imputation,

[1] Assistant Director SRS, Informatics Services Division, Statistics Canada.

estimation, tabulation, statistical analysis, graphics and confidentiality analysis. Some of these functions are common to other types of organizations, and are likely to be available from software houses. Others are special to national statistical offices, and will need to be designed and implemented there.

Future work will concentrate on the closer integration of the range of software in use, centered around a relational database system. Extension of stored metadata to provide full documentation of the data held by an office will be a priority. Storage of tables and time-series data within this framework will receive more attention.

2. Why is a statistical office different?

Modern statistical offices differ from large industrial corporations in many significant ways. The data they produce is in a very real sense their final end product, whereas in an industrial corporation it is a means to an end. Statistical offices tend to build up a large amount of historical data over a long period, and much of this data needs to be preserved indefinitely. It may be considered as part of the national heritage, and efforts must be made to ensure that it does not become lost due to failure of the materials on which it is stored, or because the techniques used to record it are no longer available to read it at a later date.

The volume of data collected by a statistical office presents a very real problem. Even in a small country, the files from the census of population will contain millions of records, and may be beyond the capacity of many commercial database management systems. A typical statistical office will collect tens of thousands of files over a period of a few years, and no commercial software is capable of organizing such a large collection of data into a single integrated database, even if such an objective were desirable. Files will be created with an enormous variety of formats, some with hundreds or even thousands of different variables, and some with many millions of records. The technique known as network organization used in some database software depends on pre-defining the relationships between the many files and variables, and is far too demanding and quite inappropriate for statistical data that is this varied and this voluminous.

There is a need for a method of designing files that is simple and easy to understand, and which will be stable over long periods of time. Files must be documented very clearly so that they can be used effectively when the person who designed them is no longer available for consultation, and this documentation must include not only a physical and logical description of the file in EDP terms, but must also describe methods by which the data was collected, measures of its quality, applications for which it is suitable, and many other items which are not normally part of the technical description of a file.

The technique of file normalization covers many of the requirements listed above. Primarily, it is a simple principle which will tend to avoid misinterpretation. Relational database systems support a collection of normalized files as an essential feature of their design, and they usually provide means of documenting, within the file itself, the set of variables that are defined within the file. This feature is important for data that is to be preserved over long periods, as it eliminates the risk that the documentation and the data to which it corresponds may become separated, and therefore unusable. However, current DBMS packages usually limit this internal file documentation to a set of technical items, and may not permit a user to store as much detailed descriptive information as is really required.

3. Reduction of conventional programming

While the cost of computing equipment has dropped rapidly relative to performance in the

last two decades, there has not been a corresponding decrease in the cost of skilled data processing staff. The result has been a tendency to create and use software that concentrates on making it possible for non-EDP staff to obtain useful results from computers, rather than to rely on detailed programming for every problem. In the field of statistical analysis, integrated software packages such as SAS and SPSS have made it possible for staff with virtually no programming experience to perform sophisticated calculations. The results are usually more reliable, as most of the work is done by routines that have been very thoroughly tested by people in many different organizations. By contrast, a routine written in a language such as PL/I to perform a simple calculation such as finding the mean and standard deviation of a set of observations may contain subtle numerical errors that will not appear in simple tests.

An analogy with engineering methods is appropriate. The designer of, say, a new clothes washer does not design new switches, pumps and electric motors. Instead, he uses standard components that can be obtained from several different manufactures, which have been tested and proven in practice. The result is a product that will be reliable, can be maintained easily, and which can be repaired by someone with relatively limited skills. While this concept has been standard in engineering practice for many years, it has not yet been universally applied in the software field. There is still a strong tendency to 're-invent the wheel', with the result that software is less reliable and less easily maintained than might otherwise be the case.

Frequently, processing steps can be implemented by using general purpose software, without the need for programming in high-level languages such as PL/I or COBOL, with a substantial increase in productivity of the development team. In many cases, it is possible to buy or lease software that will carry out the required functions. In other cases, especially for types of processing that are only needed in statistical offices, it will be cost-effective to implement new software packages within the office. But this is becoming relatively unusual, as more software becomes available in the marketplace.

Even for people with programming experience, modern software can improve productivity significantly. The use of integrated graphics procedures, for example, makes it possible to produce results in a few hours that might take weeks to program in a conventional language, and it becomes feasible to experiment with different techniques at low cost.

4. Requirements for the analysis of data

Statistical analysis usually means the exploratory analysis of data that has already been obtained from some suitable source. Most statistical offices have a large collection of data files extending over many years, and valuable results can be produced without the need for collecting new data. The work is demanding and sophisticated, and requires a background in social sciences as well as in statistical techniques. Finding people with these skills in addition to advanced programming skills is nearly impossible. Fortunately, modern software packages help to overcome this problem, as they can be used effectively by people with limited EDP training, and without programming experience.

In recent years, desktop computers have become an important tool for processing statistical data. This may be attributed to their low cost, and to the availability of a wide range of powerful software that is both easy to learn and flexible. They have been particularly useful for functions such as word processing and business graphics. Statistical software of high quality has not yet become widely available (there are a few exceptions), but may be expected very soon. SPSS has already

been announced for some small machines, and it is likely that SAS will become available on desktop machines in the near future. There is a need for software that combines support for relational files with statistical and graphical functions in a package that is highly interactive and easy to learn and use. It will need to link easily with mainframe databases, so that subsets of mainframe data can be 'downloaded' for local processing.

5. Storage requirements in a statistical office

A number of requirements for storage of data in a statistical office can be derived from the discussion in the previous sections, as follows:

i. The very large volumes of data make it essential that data be stored in the most efficient manner possible. Techniques for compressing data are desirable, both to reduce the amount of the storage medium (tape or disk) that is required, and also to reduce the processing time needed to read this data when analyzing or tabulating it.

ii. Storage media should be low in cost, while preserving both reliability and fast processing. Magnetic tape remains the principal medium for storage of large volumes of data that is used relatively infrequently, with magnetic disks for data that is much more active. At least one office has acquired the IBM mass storage device, which holds very large amounts of data on a special form of magnetic tape, and automatically stages this data to random access devices when needed. It may be expected that optical disks, which are about to appear on the market, will be highly suitable for use in statistical offices.

iii. For many purposes, there is no need to update 'permanent' data files once they have been created and verified. Much of the analytical work done in a statistical office requires that data files be processed many times in 'ready-only' mode.

iv. A form of database organization such as the use of transposed files[2] is likely to be of value, as it becomes possible for analytical programs to read only the data for variables that they actually need to process. This contrasts with the need to read the values of every variable into high-speed memory for each record processed, as occurs in most other file organizations.

6. Characteristics of the relational approach

The basic idea behind the relational approach to designing databases is one of simplicity. A relational database consists of one or more 'flat' or 'normalized' files. These two terms, which are considered to be synonymous, are used to describe files that have the following properties:

i. Every logical record in the file contains the same set of variables, and they have the same meanings throughout the file. Files where one record denotes the 'type' of entity being described, which then determines the precise meaning of other variables within that record, are not permitted. As an example, a record containing a variable 'sex', coded male or female, followed by a field 'date', which carries the date of conscription for males, but the date of birth of first child for females, would not be acceptable.

ii. No repeating groups of variables are permitted. For example, a record may not contain a variable-length list of telephone numbers for the person represented.

iii. The meaning of the file should not be affected by arranging the records in a different order. For example, consider a file in which the records represent people in an organization, and where the supervisor of each department appears before

[2] A file in which the values of a particular variable are stored physically together, as opposed to the technique of storing all values for a tuple together.

the members of his staff. The information about reporting relationships is destroyed if the file is sorted into some other order.

These problems might be solved as follows. In the first example, an additional variable should be defined, so that one date is used for the date of conscription, and the second for the birth of the first child. A suitable 'null' value should be used when necessary. The second problem could best be solved by creating a new relation with two columns; one to hold a personal identification number and a second to hold a single telephone number. (The identification number must also appear in the original file, and the relations can be processed together by the JOIN function when needed.) The third problem could also be solved by creating a new file with two variables, the first of which would be the supervisor's identification number, and the second would be the number of a person reporting to him.

As a simple test, one should consider whether performing standard mathematical operations on one or more variables within a file will give correct results. Calculation of the mean and standard deviation on an unnormalized file will often give incorrect results, and such operations must not be carried out. As an additional benefit of normalization, the fact that such operations can be carried out by standard software routines results in increased utility of the files.

The major statistical packages all expect the files that they are to process to be in normalized form. It is interesting to note that most non-EDP professionals are surprised that anyone would design unnormalized files, and regard them as unnecessarily complex.

Much has been written in the literature on relational databases about this topic, and detailed guidance can be found on how one should reduce a set of unnormalized files into the normalized form. (See for example Date (1981).)

7. Benefits of the relational approach

The relational approach has special benefits when applied in a statistical office, for the following reasons:

i. It provides a consistent design methodology for producing and storing a large number of independent files, when the relationships between them are complex, and usually not known in advance.

ii. A well defined set of operators is available in many software packages for manipulating normalized files. (These are the SELECT, PROJECT, JOIN, (Date (1981)) etc. operators from relational database theory.) They provide powerful and standardized tools for manipulating a collection of relations into a form suitable for further processing.

iii. The need for detailed programming is reduced, due to the availability of general purpose software packages. A large part of the work done in a statistical analysis project can be performed by the statisticians themselves, without the need for programming skills, or for writing detailed specifications for a professional programmer to follow. This can result in a large increase in productivity, the opportunity to experiment with new techniques and to reject them quickly if they are not successful, and much greater 'job satisfaction' for the analysts.

iv. The creation of new software becomes essentially independent of the data. A very wide variety of statistical algorithms, graphics functions and so on may be implemented on the understanding that they will take their input from one or more relations. (This point is illustrated once more by the success of the commercial statistical packages.)

Some of the early commercial database management systems require users to define linkages between files in great detail at the time they are designed. This is quite practical

for systems with a small number of well-defined files, but is impossible for the very large collection of files maintained in a typical statistical office. An important feature of analytical work is that files from very different sources are processed together, to obtain data in new forms.

A major benefit of the relational approach is that the definition of these linkages can be deferred until it becomes necessary to link the files. Effort put into the standardization of variables within the files will make future linkages even easier. Examples are the use of common coding systems (such as industrial, commodity and geographic codes) that, if used in many different files, will make the task of linking them much more simple. The risk of misinterpretation of data will also be reduced, with a resulting increase in productivity.

Record linkage work performed in one office provides an interesting example. Files of people who had worked in circumstances where they might have been exposed to carcinogenic materials were matched against death records, and against hospital admission records. These files did not carry a personal identification number, and the linkage was made on the basis of a number of items of information, using a statistical matching technique. This activity was not considered at the time the files were designed, and was simplified by the fact that the files were in an essentially normalized form.

Documentation of normalized files is relatively easy, at least in a technical sense. Most relational DBMS software, and the major statistical packages, support a form of self-documenting file. This usually means that the files contain an internal description of the variables that have been defined, in such a way that the software can interpret the file automatically. This internal documentation is limited to technical details of the variables and their coding sequences; it does not include the much more voluminous descriptions that are needed if the file is to be used by many different people. Documentation at this level is much harder to prepare. It should include detailed notes about how the file was obtained, information about its accuracy and reliability, any warnings about inappropriate use, and many other items. Ideally, this information should be prepared in machine readable form, and linked to the file it describes in such a way that they will not become separated.

8. Processing techniques

Most of the functions required in processing a survey or census may be built around a relational database, as a set of software packages that may be thought of as processing tools. These functions are usually some or all of the following: frame maintenance, sample selection, data capture, edit and imputation, estimation, tabulation, statistical analysis and confidentiality analysis.

There is an important change in the philosophy of the design of database applications. In older systems based on magnetic tapes, the design consisted of a number of processing steps, each of which accepted one or more tapes as input, and produced one or more tapes as output for use in later steps. Very often, the record format on each tape was different, and it was not possible to run the processing steps in different orders. In a database system, by contrast, the data should be seen as the central point of the system. It will be supported by a dictionary that describes the (normalized) files and the variables defined within each one. Each processing step (for example, tabulation) should be seen as an independent process that operates against the database.

The order of the processing steps is not entirely independent – data capture must obviously be done before tabulation, for example. However, there are many degrees of freedom; statistical analysis programs could be run on 'raw' data when required, before

the edit and imputation processing has been done, and tabulation of partial results is often useful.

Processing steps can also be run on a subset of the data, with significant savings in computer resources. Subsets may be defined either in terms of rows (logical records) or columns (variables) within the database. Provided the internal file design supports this kind of subsetting effectively, the savings may be very substantial. The use of primary and secondary B-trees (Martin (1977)) designed to enable subsets of the rows to be processed is an important technique. It should be possible, for example, to perform a tabulation on all the records for a single province or industry without the need for moving other records into high-speed memory. Similarly, a transposed file organization enables a retrieval program to process all the values for a given subset of the columns defined within the file, without the need for accessing other columns. These techniques contrast sharply with the use of sequential files, in which every value held in every record is usually moved from the secondary storage device into memory even when only a small subset is needed.

Design of the database is fundamental to successful operation of any application, and should be undertaken first with great care. It should then be possible to design most of the processing steps independently, and these become tasks that can be done by different groups of people if the job must be completed quickly. The file design will be the main document of agreement between them.

It is rarely possible to anticipate all the requirements for a system during the early design stages. One of the major benefits of the relational method is its flexibility. New relations, and new attributes for existing relations, can be added without disturbing earlier parts of the design and implementation. The VIEW concept[3] is important here, in which the DBMS software is designed to make applications as independent as possible of the specific file design.

9. The need for concurrent update software

A database management system for statistical use needs to be optimized for retrieval, at the expense of sophisticated update facilities. There is rarely a need for concurrent update methods, which are an expensive feature (in terms of complexity) of many commercial packages. In a typical application, there will be a substantial amount of update activity during the data capture, editing and imputation phases, to the point that a clean microdata file has been prepared. Subsequently, most activity will be for retrieval purposes, and the file will no longer be changed. It may even be appropriate to use different software for the two phases, provided it is easy and automatic to move data from one environment to the other.

Requirements during the early stages of survey processing may differ significantly. Frame maintenance (such as updating a register of businesses) could be implemented as an interactive system, in which records can be displayed on a CRT device, and updated directly. Concurrent update facilities would be needed, to enable a number of clerks to work simultaneously, and the system would typically need to access all the data in a given logical record, or group of related records. This type of processing is handled well by many commercial database packages, and the requirements differ substantially from those of statistical analysis and tabulation.

By contrast, the later stages in a typical survey or census do not usually need concurrent update. Imputation is usually done as a

[3] A method of enabling application programs to access subsets of a relation (both by row and column) as if they were the entire relation. A view mechanism will usually provide automatic data conversion to the mode requested by the application. See also Ullman (1979).

batch process by a single task, and the files being processed should not be available to any other task at that time. Once a 'clean' set of microdata has been loaded into a suitable set of files, most statistical analysis and tabulation can be done in 'read-only' mode. In many computer systems, such files can be shared by many different users or jobs simultaneously. Analytical users may need to create modified copies of subsets of the data for their own purposes, and they may make these available to other analysts in 'read-only' mode.

Consideration may be given to the use of two different database packages. A package designed to provide concurrent update and record oriented retrieval might be used for frame maintenance, sample selection, mailing and data capture. The data would then be transferred to a second package that was oriented towards efficient retrieval of subsets of rows and/or columns, and which provided tabulation and analytical functions. Provided both packages support relational files, and have compatible data dictionary facilities, it should be easy to transfer the data and associated metadata without the need for clerical intervention.

10. Future activities

It is expected that future work will proceed in the same general direction. The integration of software functions into a coherent set of tools that can be applied by relatively unskilled staff will continue. The extension of metadata concepts, and in particular the level of detail that can be carried about any file, will help to organize the work of statistical offices.

Techniques for storing summarized information within a relational database will also be investigated. At present, the use of summary files as defined within the SAS package is a popular method for conserving machine resources, and techniques for documenting and distributing these must be worked out.

Wider adoption of a standard language for expressing relational queries will bring flexibility in the use of different data storage methods. The SQL (Date (1981)) language promoted by IBM as part of the System R project is a likely candidate, and has been implemented for several different machines.

Advanced desktop computers will become able to provide users with more processing power than is currently supplied by time sharing systems. Software for these machines will improve in terms of functional capability and ease of use. Relational database packages with integrated statistical and graphics functions will make it possible to process all but very large files on these machines. Communications with mainframes will make it easy to transfer data quickly, so that subsets of the mainframe files may be processed locally. Large centralized computers will continue to be used as the central repository for data.

11. Appendix: Experiences at Statistics Canada

The design and implementation of RAPID (See Schmidt and Brodie (1983) and also Statistics Canada (not dated) and other documentation, available from Statistics Canada) has strongly influenced the processing of large applications at Statistics Canada during the last few years. RAPID is a relational database package, whose major features are as follows:

i. A transposed file architecture, in which all the values of a given variable are stored together physically. This ensures that a subset of a relation by columns can be processed with high efficiency.
ii. The ability to create B-trees on any column or set of columns, enabling subsets of the relation by row to be obtained efficiently.
iii. The ability to add or delete either rows or columns dynamically, without the need for file reorganization.

iv. Very compact storage, especially for variables that have a limited range of possible values.

v. A range of utilities, including a data dictionary subsystem, that reduces the need for conventional programming.

Application development at Statistics Canada has concentrated around four major software packages in the last few years. RAPID has been used for several very large projects, including the Census of Population and the Consumer Price Index. Its transposed file architecture and B-tree retrieval mechanism for obtaining subsets of rows are important features, and there is no need for concurrent update facilities. By contrast, ADABAS (a popular and well-designed commercial DBMS package) has been used for applications that require record-oriented retrieval and concurrent update capability. This package includes a high-level language package known as NATURAL, which enables users of a database to produce reports and perform retrievals without the need to resort to programming languages. TPL (Table Producing Language, developed by the US Bureau of Labor Statistics) has been widely used for generating tabulations. Finally, the SAS statistical package has become extremely popular with data analysts as well as with many professional EDP staff, due to its excellent statistical, reporting and graphical functions.

In such an environment, the ability to transfer data between these packages is vital. At the time of writing, software for moving data files between SAS and RAPID is in place, and a commercially-marketed interface that will enable SAS to read ADABAS files is planned for acquisition. Interfaces between RAPID and TPL and between TPL and SAS are also in use.

Statistics Canada is in the early stages of preparing a directory of files that are available for general use within the organization. It will attempt to cover the requirements listed above, and it is hoped that it will become a valuable tool in the hands of the data analysts.

In a second stage of the project, it is expected that this directory will become a component of the Statistical Data Documentation System (SDDS), which is a machine readable catalogue of the surveys and publications operated and produced by the organization. This will then complement the CANSIM system, which makes a large collection of time-series data available in machine readable form to the public. When the file directory has been integrated with other documentation about surveys and publications, the bureau will be in much better control of its data, and will be able to pursue more complex data analysis projects at lower cost than before.

12. References

Date, C. J. (1981): An Introduction to Database Systems. Addison Wesley, Cop. Reading, Mass.

Martin, J. (1977): Computer Data-Base Organization. Prentice-Hall, Inc., Englewood Cliffs, N. J.

Ullman, J. D. (1979): Principles of Database Systems. Computer Science Press, Washington, D. C.

Schmidt, J.W. and Brodie, M.L., Eds. (1983): Relational Database Systems: Analysis and Comparison. Springer Verlag, New York.

Statistics Canada (not dated): RAPID DBMS – General Introduction.

Received November 1984
Revised January 1985