A Database Model for Studies of Cocaine-Dependent Pregnant Women and Their Families

Peter A. Charpentier and Richard S. Schottenfeld

INTRODUCTION

Collecting, organizing, and analyzing data for clinical research projects that involve families of cocaine-dependent mothers present significant data management challenges. Depending on the specific administrative task or analysis, a "record" or unit of analysis might be a mother, a pregnancy, or a child. Data must be organized to permit such multiple views while preserving the family structure, and controlling access to personal identifying information is a higher priority for these data than for other clinical research database systems. This chapter describes the authors' efforts to design and implement a database system for the Mothers Project, a Perinatal-20 project established in New Haven, Connecticut. A brief description of the Mothers Project follows.

THE MOTHERS PROJECT

The Mothers Project is both a clinical epidemiological study of the correlates of cocaine and other drug abuse during pregnancy and a controlled clinical trial in which cocaine-dependent women are randomized into one of two treatment programs: an enhanced primary care treatment program or a comprehensive day treatment program that provides family support services and child care.

Routine drug abuse screening in a hospital-based prenatal clinic uses structured interviews and urine testing to identify cocaine users and enroll them in the clinical trial. Assessments of the mother and child are made at 3, 6, 9, 12, 18, and 24 months. It is permissible for women to be enrolled twice for two pregnancies.

SPECIFICATIONS OF THE DATABASE SYSTEM

The database system described in this chapter addresses the data management needs of the clinical trial component of the Mothers Project.

The design of a database system can be described in terms of *functional* and *performance* specifications (Stevens 1987, pp. 35-36); researchers interested in the fundamentals of relational database theory should see Date (1990, pp. 3-24). Functional specifications state what a database system must do, whereas performance specifications describe how often and how fast specific functions must be carried out, as well as such "real world" issues as constraints on the computing environment and the skill levels of the system operators. The performance and functional specifications of the Mothers Project database are shown in the following lists.

Performance Specifications

1. All demographic, identifying, and tracking data will be available for inspection or modification on an as-needed, per-record basis.

2. The number of data elements associated with demographic, identifying, and tracking data will be fairly small (fewer than 50 fields for a given mother, pregnancy, or child).

3. Demographic, identifying, and tracking data will be continually updated.

4. All subject tracking and contact scheduling functions will be performed by operators who have good computer skills but no programming training.

5. Interview data will be processed in "batches"; interactive record retrieval systems are not necessary.

6. The volume of the interview data, in terms of record length, will be large (e.g., several hundred variables per record).

7. The system must run on inexpensive desktop computers in either a stand-alone or networked, multiuser setting.

8. The Mothers Project staff will include data analysts and managers who are proficient in a statistical package with an excellent flat-file data management programming language (e.g., SAS) but will not need to include staff members who are proficient in relational database methods and software.

9. Although much of the database system will be developed by consultants, all software systems will be *maintainable* by project staff members who can make minor changes or enhancements to the database system.

Functional Specifications

1. Provide for the entry, storage, and modification of basic demographic, clinical, and personal identifying data for mothers, pregnancies, and children.

2. Provide a high level of security for subject identifiers such as names, addresses, and hospital unit numbers. Make it possible for all approved staff members to access the database and encrypt certain sensitive data to all but a few selected operators.

3. Track the current participation status of each subject.

4. Maintain the contact schedule of each subject.

5. Generate lists, cover sheets, and mail/merge data files for contacts falling within a specified date range. These lists assist field staff members in arranging for interviews; cover sheets are used by interviewers as "snapshots" of subjects' current status and addresses; and mail/merge data files are used for labels and followup letters.

6. Track basic information, such as dates and outcomes, for all contacts with subjects.

7. Keep track of and report missed contacts with subjects to arrange for special retrospective interviews that attempt to recover selected data.

8. Provide a framework or basis for additional planned and unplanned subsystems, such as collecting health care utilization information.

9. Provide "read only" access for ad hoc queries and for user-designed reports.

10. Provide for the entry and storage of data from baseline and followup interview data collection forms.

11. Check interview data for errors and allow operators to make appropriate changes to the data files while maintaining a complete audit of all changes.

12. Prepare files for analysis using a standard statistical package.

13. Facilitate data analysis by providing detailed data documentation.

Functional specifications fall into two major groups that might be called the *administrative* and *analytic* function groups. In practical terms,

different database and statistical software and file handling procedures, as well as staff skills, were optimally suited to each function group. Therefore, separate administrative and analytic database systems were developed to meet the functional specifications of the Mothers Project. These two database systems are described below.

Administrative Database System. The administrative database system was associated with the necessity to monitor and contact subjects. Administrative services provided by the Mothers Project database system included rapid access to records (e.g., mothers, pregnancies, children) to ascertain or change current status, addresses, and other information; periodic conduct of study reports (e.g., those concerning screening, enrollment, and participation rates); contact scheduling; and mail/merge applications for contact and followup letters.

Key requirements of the administrative database system were that data be retrievable on an individual record basis and through a variety of views. Furthermore, as the performance specifications below indicate, the administrative database system was operated by field staff members who were competent in the use of computers but were not computer programmers. As in most clinical research projects, there were not inexhaustible resources for equipment or for exotic database implementations. Thus, these factors led to considering a commercial relational database package designed for inexpensive office computers. Any of several modern, business-oriented database products would have been suitable for the Mothers Project administrative database system. A quasi-relational database package was chosen that featured sophisticated application development tools and a reasonably "user-friendly" interface for ad hoc queries and reports.

Structure of the Administrative Database. The administrative data for the Mothers Project were stored in three data sets, or tables, called the *mother*, *pregnancy*, and *child* tables. The mother table includes mother and family descriptors, such as mother's name and address and a contact person's name and address. The pregnancy table includes not only information relevant to each pregnancy (e.g., date, outcome) but also enrollment, treatment assignment, and selected followup information. Because enrollments were based on pregnancies rather than mothers, it was possible for the same mother to be included twice in the enrollment list, but for different pregnancies. The child table contains descriptors for each child, including name, birth statistics, and vital status.

The three tables that make up the foundation of the administrative database were designed to store data in an efficient and "normalized" manner. Each piece of information about a mother, pregnancy, or child is stored in only

one place (i.e., in a record within the appropriate table) so that updating information is straightforward. This structure can accommodate any number of pregnancies per mother, or children per pregnancy, without the loss of data storage efficiency. To illustrate, consider a family consisting of a mother and three children, two of whom are twins. The record structure for such a family (one mother record, two pregnancy records, and three child records) is diagramed in figure 1. Mother records are identified uniquely by a *mother ID code*, a number assigned to each new mother who is enrolled. Pregnancy records are identified uniquely by the mother ID code and a *pregnancy number*. Finally, child records are identified by the mother ID code, the pregnancy number, and a *child number*. The codes used to identify records are in a sense arbitrary and are used only for that purpose. Assembling a family's records, or those associated with a pregnancy or child, is a matter of arranging the "relationships" among the tables to construct the proper view or "relation."

Views Into the Data. Three main "views" arose from the administrative database structure: the mother, or family, view; the pregnancy, or enrollment, view; and the child view. These views are diagramed in figure 2 using the same family structure illustrated in figure 1. For most applications, the pregnancy view is used because it was the basis for enrollment. However, ancillary studies that focus on mothers or children make use of the other views.

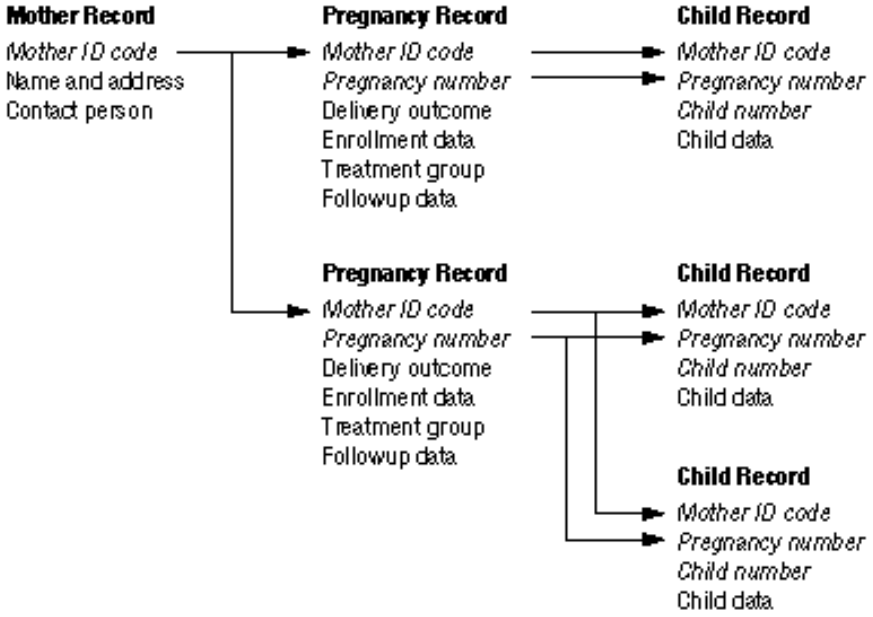

FIGURE 1. Administrative data structure for a family

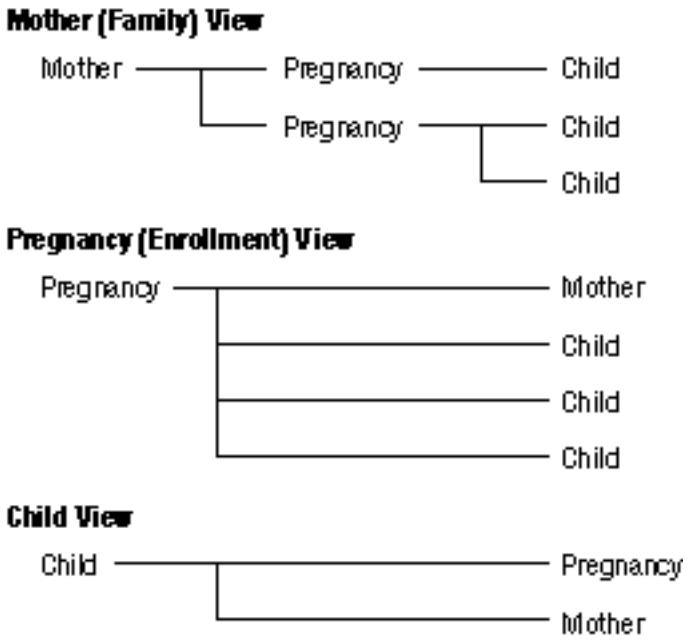

FIGURE 2. The three administrative views

Function and Implementation. As indicated above, the administrative database system supports critical subject management functions and is operated by staff members who are not computer specialists, which makes the interface a major consideration. The software application is designed to isolate inexpert users from the underlying complexity of the database and the command language by controlling access to functions and data through menus and control panels.

Figure 3 illustrates a "screen shot" of the main data access screen. A control panel at the left provides the basic record location functions, and the data are available (in read-only mode) in three windows corresponding to the three main tables. What is shown in these windows depends on the view, which is selectable on the control panel. To obtain more detailed information on a mother, pregnancy, or child record, the user activates the appropriate window by "clicking" it on with a mouse, moves the cursor to the desired record, and presses the "Enter" key. The view is then expanded to a full screen showing all the information in that record, and the data are available for inspection or modification.

Schedules and Tracking Reports. Once the enrollment information is entered for a pregnancy, the administrative database system generates an optimal contact schedule. Several reports are implemented that use the

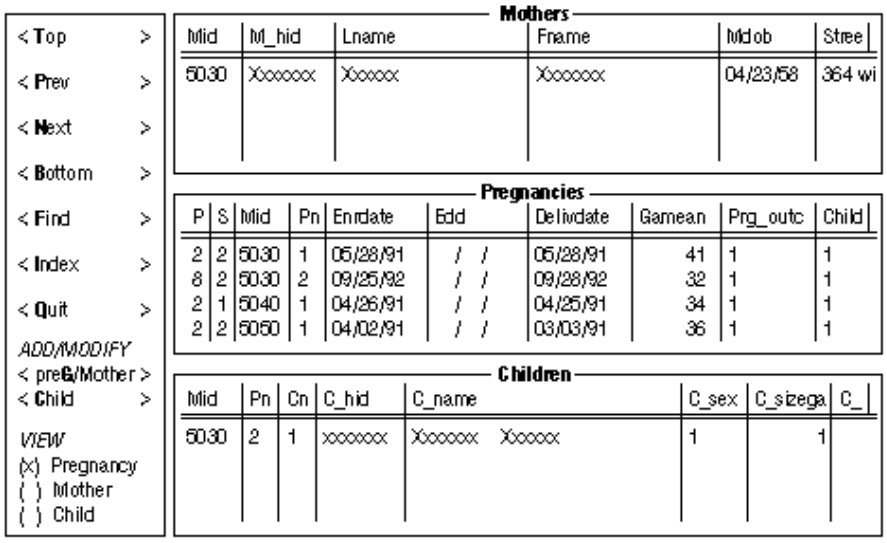

FIGURE 3. The main view screen

SOURCE: APT Foundation (New Haven, CT). Copyright 1991.

optimal and actual contact information to provide various tracking reports. For example, a sweep of the database is made periodically to identify pending and overdue contacts for all participants. Two reports are then printed: a summary list of contacts for that week and a set of "cover sheets" that report detailed information helpful in arranging the contact. The cover sheet includes the name and address of the mother and a contact person, the optimal contact schedule, and dates and outcomes of actual contacts with a special note of missed followups. (Interviewers recover some data from missed contacts through specifically designed questions.) A mail/merge data file is automatically generated when any report is printed and can be used with any word processing program to generate contact letters or other correspondence.

Security. Protecting the identities of cocaine-dependent women and their families is a requirement of the administrative database system. Therefore, a considerable effort is made to make these data secure. All information that might be used to identify any individuals, including names, addresses, Social Security numbers, and hospital unit numbers, is stored in encrypted form. The data are displayed or reported in decrypted form only for users who enter the correct encryption key when the application is loaded. The encryption key can be changed periodically by a single staff member who has access to the software. It is also possible to use the database system without providing the encryption key; the sensitive data appear garbled, but the application functions otherwise.

Additions to the Administrative Database System. The administrative tasks have grown in complexity, particularly as the Mothers Project expanded its coordination with other drug abuse studies based in New Haven. There are ongoing studies, for example, on children of cocaine-dependent mothers. The flexible database structure makes it possible to switch among mothers, pregnancies, and children as units of analysis for the purposes of new projects and substudies that become attached to the Mothers Project. It is also a simple matter to add other tables to the database and incorporate them into any of the views. More detailed information on health care utilization by mothers and children has been added to the tracking system in this manner and is immediately accessible through the interface.

Because the underlying software package features strong reporting and querying tools and has sophisticated tools that can attach such modules to existing applications, advanced users can add significant functions to the database system without additional programming.

Analytic Database System. In contrast to the administrative functions that require record-based processing, questionnaire and clinical data intended for analysis are processed at the *data set* level, that is, in *batches*. Easy access to individual records is not a priority. Furthermore, whereas administrative records include at most a few dozen fields (i.e., variables), questionnaire records typically include several hundred. Because complex relations among questionnaire records are not required, nonrelational or flat-file data management methods are suitable, such as those provided by statistical packages.

The analytic database consists of a series of data sets, one for each kind of contact or interview, all linked by a single key field (the subject identification code) in a one-to-one relationship. Each data set contains many hundreds of fields. Reports required by even the simplest analyses include statistical functions not found in business-oriented relational database packages. All this argues for using a statistical package with strong flat-file handling and reporting capabilities to manage the analytic data.

Administrative data are always processed by project staff members, but analytic data sometimes are made available to outside collaborators who have different computing environments. Thus, the “cross-platform” capability of a statistical package is a major factor. The package used includes a complete flat-file database programming language in addition to a standard set of statistical functions and procedures. This particular package is in wide use on virtually every computing system used by research projects and has a “data transport” utility that makes it possible to transfer data between different operating systems.

The flow of analytic data is diagramed in figure 4 and explained in the sections that follow.

Data Entry. As noted, two characteristics of the Mothers Project analytic database system are that records can be processed in batches and that records from questionnaires can be fairly large. Therefore, the authors chose to implement a traditional keypunching procedure for analytic files. Data collection instruments are designed so that data entry fields are of fixed length, column specified, and located in the right margin of each page. Completed forms undergo a preliminary review where they are checked for completeness and obvious coding errors. Forms with errors are returned immediately to the field staff for correction. Approved forms are assembled in batches and transferred to an in-house data entry professional. Using a keypunch emulation program, data are keypunched

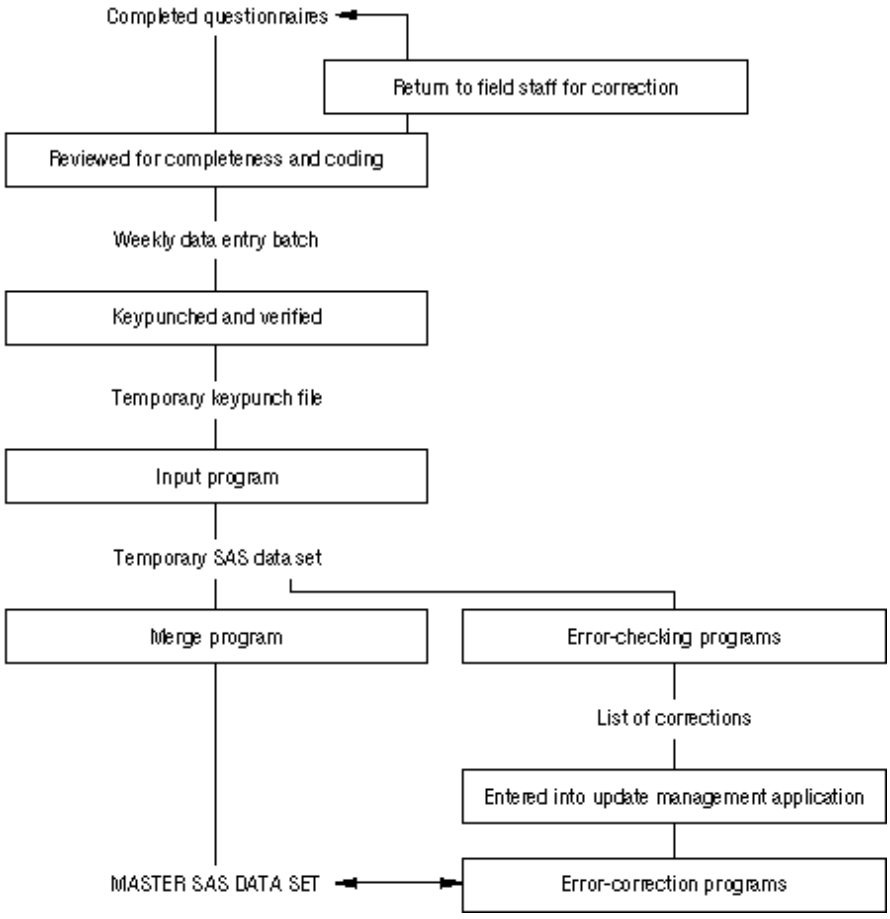

FIGURE 4. *Data flow diagram for analytic (questionnaire) data*

and verified (i.e., keyed twice). The data entry operator then returns a file of newly entered records to the data manager.

With well-designed data collection forms and appropriately trained data entry operators, data entry rates of more than 10,000 keystrokes per hour have been achieved. This style of data entry appears well suited to projects that do not have in-house data entry capability because outside data entry service bureaus usually can keypunch traditional, column-specified data.

The Case for Keypunching

As an alternative to keypunching the analytic data, the statistical package's built-in data entry facility could have been used. Using this facility, an interactive system with point-of-entry error checks and branching could have been developed. Mothers Project staff members found the keypunching procedure more efficient for three reasons. The first is data entry speed: No screen-oriented, interactive system can match the speed attained by a professional keypunch data entry operator. Second, trapping errors during data entry may sound appealing, but this practice slows down data entry. Keypunch errors, made by the data entry operator, can be detected by double-pass data entry or verification. Errors detected by range and consistency check routines are almost always *coding* errors that were made by the person who filled out the form and were not detected in the coding review. Those persons usually are not available to resolve coding errors that are detected during data entry. Thus, the data entry operator has to stop and decide whether a coding error can be resolved immediately or must be deferred; even a brief pause significantly interrupts the flow of a data entry session. The third reason for traditional keypunching of long records concerns development costs. It would have taken time to develop a responsive, interactive data entry system for the Mothers Project data collection instruments. Even simple changes to the data collection instrument would have caused delays while the data entry system was being modified. Thus a keypunch style of data entry for long records is used that permits delayed processing of batches.

Error Checking and Master File Updating. Each data collection form has an associated *input* program, one or more *error-checking* programs, and a master file *update* program. These programs are run on batches of newly keypunched records. The input program converts the keypunched ASCII data to the "native" format of the statistical package. The temporary file thus created is then passed to the error-checking program and to the update program. New records are checked for errors and simultaneously added to master data sets.

The printout from the error-checking program(s) is checked by the data manager and passed on to the field staff for resolution. Corrections always are made to master data sets (i.e., not to keypunch files), and all changes are completely documented. Auditing the changes to analytic data is considered to be so important that a specialized database application is employed solely for this purpose. As corrections are identified, they are entered into the system as "data value update" records. Each data value update record consists of a record ID, a variable name, and the correct value. The changes are made to the master data files using the error-correction program written by the specialized software. These programs generate printouts that clearly document each change that is made.

Data Documentation. The primary function of the analytic database system is to prepare files for analysis. A major part of this task is to develop and distribute comprehensive data documentation. The specialized application is also used for this purpose. In addition to managing data *value* changes, the software is designed to manage data *descriptors* for the databases. The software also provides for the development of descriptors at the database, data set, data element (i.e., variable), and scale (i.e., aggregate data element) levels and includes extensive reporting capabilities.

SUMMARY

The database management functions for the Mothers Project are arranged into administrative and analytic task groups, and separate systems are devised for each. The task groups can be distinguished not only by differences in data structure but also by interface requirements. The administrative database system uses a relational database technology, whereas the analytic database system employs more traditional flat-file methods. Although the database management systems are complex, they are based on standard database practices, used in widely available software packages, and run on inexpensive desktop computing equipment.

REFERENCES

APT Foundation. Mothers Project database system. New Haven, CT: APT Foundation, 1991.

Date, C.J. *Introduction to Database Systems, Vol. 1*. 5th ed. Reading, MA: Addison-Wesley, 1990.

Stevens, A.C. *Data Base Development*. Portland, OR: Management Information Source, 1987.

AUTHORS

Peter A. Charpentier, M.P.H.
Research Affiliate in Epidemiology
Yale University School of Public Health
60 College Street
New Haven, CT 06520-8034
(203) 785-7178 or (203) 785-6386 (Tel)
(203) 737-4260 (Fax)
charpentpa@maspo2.mas.yale.edu (Internet)

Richard S. Schottenfeld, M.D.
Associate Professor of Psychiatry
Yale University School of Medicine
Director
Substance Abuse Treatment Unit
Connecticut Mental Health Center
34 Park Street
New Haven, CT 06519
(203) 789-7079, ext. 308 (Tel)
(203) 789-7087 or (203) 789-7088 (Fax)
schottenrs@maspo3.mas.yale.edu (Internet)

Click here to go to page 254