Database Design Strategies in CANDAs

Sunil Kumar Gupta
Gupta Programming, Simi Valley, CA

ABSTRACT

Developing a productive CANDA system starts with an optimal database design. Traditional objectives and standards of normalization are not appropriate. A focus on function, ease of operation and performance drive the data file structure design to become more integrated. The regulatory approval process defines the system functions and features of the CANDA.

This paper will outline the strategies of clinical data structures and their role in the regulatory approval process.

INTRODUCTION

Developing a CANDA system is a major project that is time consuming and expensive. Because it is often a critical element of the drug approval process, there should be strategies established for the information delivery system.

Standardization in SAS programs and reports helps to build the foundation for a more productive environment. As a prerequisite, a well-designed database structure is essential for the success of all other tools built around these datasets. The goal would be to build SAS libraries based on the CANDA's system functions.

Correctly applying SQL to the relational database model will assure complete and accurate results.

TRADITIONAL METHODS OF NORMALIZATION

Traditional methods of normalization are not enough. Although many of the objectives of a normalized database are desirable, they are not always ideal for a CANDA system. A tremendous benefit of a normalized database system is the prevention of duplicate entries of the same data point. Not only does this minimize data maintenance costs, it also minimizes the risk of data integrity. All the tables in the database system should be optimally normalized with the key fields protected.

Concept

Normalization is the process of reducing a complex data structure to its simplest, most stable form by removing redundancies and assigning attributes to each entity. The primary objective of normalization is to reduce data storage overheads while managing the information contained in the data files. This assures that any given fact about the drug's performance is being recorded in one and only one place. This is a critical issue considering that a CANDA often contains large files of clinical data and crf graphic images.

Specifically, the aims of normalization are:

- To identify and isolate the entities involved. An entity is something about the drug which the reviewer needs to know.
- To remove redundant information. The same non-key attribute may not be recorded more than once. This reduces the potential for inconsistent data values due to record creation, update, and deletion.
- To determine the identifying attributes for records. An attribute is a fact that describes an entity. The unique attributes of the record define the primary key for each entity.

Data Model

In many clinical studies, the relational model provides the closest fit to the functional requirements of the protocol. Storing data in files that resemble the case report forms allows for efficient storage and retrieval of information.

The interaction between case report form design and database design is fundamental and unavoidable: the manner in which data is collected influences both the meaning of what is collected and the structure in which it is stored. The relational database model will allow the programmer to create many tables with defined relationships. In general, the case report forms dictate the storage format of the data items in the database.

Although having a strict normalized database system facilitates the development of a data entry system, it is not recommended. It would be easy to establish a single dataset containing all the fields in the case report forms. All information collected on patients would be stored in each observation. On the case report form for example, there may be five events per page that can be stored in event1 to event5 fields. The trade-off comes in the more time required to generate reports and analysis. This is because of the loss of flexibility in the information stored.

Generous allowance should be made for the possibility of additions and revisions to data collection requirements or a change in the number of participants to be enrolled in the study. Overall storage estimates should include database overhead, storage of necessary administrative data and space for multiple copies of the database for statistical analysis.

The architecture of the relational database is what allows the programmer to utilize SQL as a tool to relate tables of information for reporting and analysis. Additional datasets or views can be generated from these relationships.

The table below describes the three different types of relationships between entities:

Relationship	Entity 1	Entity 2
one-to-one	Patient	Medical History
one-to-many	Patient	Vital Signs
one-to-many	Patient	Adverse Events

Outcome

With SAS , it is possible to define a relational database model to meet the objectives of normalization while having the powerful tools needed to manage and analyze the information.

Typical common datasets in clinical studies include Patient, Medical History, Efficacy and Adverse Events.

Normalization will allow for the following system characteristics:

1) Unlimited number of patients, centers and doctors; unlimited number of follow-ups and complications.

2) Consolidation of similar data points to allow more efficient reporting and analysis. This consolidation will also remove any possibility of duplicate data points, which otherwise jeopardize data integrity.

3) Consistency between studies, reducing the need for specific training for each study.

OPTIMAL DATA BASE DESIGN

Designing an optimal database takes time and effort. This critical process however, will save more time than it takes. A carefully constructed plan provides a blueprint for the database structure. The key to a successful database layout is to present the clinical data as valuable information to the user. Each file of the database system must be well defined and logically associated with each other. The end user of the CANDA system should not be required to know any technical knowledge needed to combine files.

One of the most important features of the design is that data consistency and quality are enforced in all data views, analysis and reports generated. This will assure credibility in the system and in the clinical information.

The elements of a robust database design are the following: complete datasets, flexible and integrated system, system standards, and data structured for ease of data retrieval.

Complete Datasets

In order to have complete datasets, the programmer needs to identify all the data collected in the study. Next, the programmer needs to categorize the data into files that reflect the organization of the case report form. In general, each file contains the information on only one case report form page.

A dataset is complete if it contains all the information that must be collected and stored. This step is achieved by entering the data contained on the case report forms. The information requested on the case report form will be used to support the safety and effectiveness of the drug. It will be required by the FDA as defined in the protocol.

The basic core questions that the FDA reviewer will ask need to be identified and addressed in the database design. This will make the system more productive by being more efficient in the review process. This determines the purpose and direction of the database.

Flexible and Integrated System

The optimal database design is flexible and powerful enough to adapt to the changes forced upon it by the environment. This includes adding new fields and data entry codes as well as performing complicated queries.

In addition, this “super” dataset would represent a collection of all the clinical studies needed to support the drug approval application. This may include all pilot studies and phase I, II, & III studies. Analysis can be performed on individual studies as well as across all the studies as required. This integrated dataset approach makes pooling the data together much easier. Where possible, differences between studies should be accounted for to prevent loss of information.

System Standards

Standards in field name convention, coding system, and field type and size across all studies help to assure consistency and quality in programming. By assigning a patient number by it’s clinical study, the patient is uniquely identified in the “super” study. This will prevent having duplicate patient numbers when pooling the patient data from all clinical studies.

For all follow-up data, the visit date should be stored as an identifier in the dataset. If the data is repeatedly collected on the same day, then the time of collection should also be recorded. It is only through standardization across studies that makes it possible to combine data from these studies into a single report or analysis.

The system standard incorporates quality control measures that enable time-saving programming. In addition, these standards impact the ease of the CANDA development and use.

Efficient database maintenance is achieved through standards, quality control, and proper documentation. The macro programming language helps to centralize the code for standard reports and analysis of similar studies.

Data File Structure Design

The new data file structure design enhances the basic principles of normalization to be more specific for the pharmaceutical industry. The relationship between patients, visits, and events is usually consistent in all clinical studies. Because of this, several strategies can be applied to the storage and analysis of the information to increase the performance of the system.

As a requirement, there should be several components in each record - Study Number, Site/Investigator Number, and Patient

Number. If applicable - visit number, visit date, and time should be included for follow-up information.

The critical question that must be addressed is the following: for multiple occurrence of data items at different time points, is it easier to retrieve multiple records or one record with repeated data items? A vertical file structure defines the multiple records concept and the horizontal file structure defines the repeated data items structure.

The two options to consider are vertical and horizontal file structures.

A. Vertical File Structure

To utilize a vertical file structure, there must be a method to distinguish between the multiple occurrences of the data. This is accomplished with the visit date and visit time fields. In general, the single record contains information on the patient's measurements at a defined date and time.

The advantages of having this structure include the utilization of cross-tabulation features for reporting and analysis and the minimization of the amount of missing data. Many SAS procedures require this structure for processing. In addition, programming efforts are reduced due to processing multiple records instead of multiple fields.

ex. Vital sign

Study	Site	Patient	Visit Date	Visit	Temp	BP
01	1	01	2/1/93	**1**	97.1	120/ 70
01	1	01	2/8/93	**2**	97.8	115/ 70

B. Horizontal File Structure

To utilize a horizontal file structure, the information collected must be fixed. The field names should have suffixes that help identify the visit date. This could be a problem if data for additional visits is collected and there is a limitation on the number of fields or the record length in a single dataset.

Although having this structure makes it easier to compare similar fields over time, it is not a recommended approach. This structure requires more programming effort without any benefit in program flexibility. Most of the programming would result in hard-coded programs with limited purpose and scope.

For example, it would be difficult to determine if the patient's temperature reached 100 from 10 temperature fields. The vertical file approach is much better.

Example: - Horizontal File Approach -

```
if temp1=100 then count = count + 1;
if temp2=100 then count = count + 1;
...
if temp10=100 then count = count + 1;
```

- Vertical File Approach -

```
if temp=100 then count = count + 1;
```

ex. Vital sign

Study	Site	Patient	Temp1	Temp2	BP1	BP2
01	1	01	97.1	97.8	120/ 70	115/ 70

Dataset Relationships

There are essentially three different levels of relationships between patients, visits, and events. These levels are defined by time-based relationships among the different events. By designing the CANDA system with these three different levels incorporated, the system becomes more focused for the FDA reviewer.

Level	Relationship type	Entities
1	Simple/ Non-Time dependent	Patient
2	Time dependent (Visit)	Patient - Vital Signs
3	Non-Time dependent	Patient - Adverse Events

A. Level 1 - This is a basic and simple relationship that is not time-dependent. There is one observation per patient.

The Patient dataset is related to the Vital Sign and Adverse Events datasets.

ex. Patient

Study	Site	Patient	Age	Weight	Race	Sex
01	1	**01**	22	155	white	F
01	1	**02**	56	170	black	M

B. Level 2 - The next level is the patient-visit relationship where data is collected as one observation per patient visit. Usually, similar information is collected over time. There is one observation per patient per visit.

ex. Vital Sign

Study	Site	Patient	Visit Date	Visit	Temp	BP
01	1	**01**	**2/1/93**	**1**	97.1	120/ 70
01	1	**01**	**2/8/93**	**2**	97.8	115/ 70

C. Level 3 - The highest level is the relationship where events are independent of visits. There could be a variable number of events per visit. There is one observation per event for each patient or for each patient and visit.

ex. Adverse Events

Study	Site	Patient	Visit Date	Event
01	1	**01**	**2/1/93**	**Fever**
01	1	**01**	**2/1/93**	**Migraine**
01	1	**02**	**2/12/93**	**Cold**

CLINICAL REPORTS WITH SQL

The end result of storing clinical information is to generate reports and analysis. Powerful tools such as SQL (Structured Query Language) help to combine datasets into meaningful information.

SQL is a non-procedural language with unique features that allow programmers to write compact code to create SQL views and to obtain data summaries. Often it takes fewer steps to summarize the data with the SQL than with the data step. SAS's SQL provides a simple and resourceful programming tool.

Two important components of any drug approval application are table listings that contain all patient information and table summaries that describe the safety and efficacy of the drug. SAS's SQL procedure provides many complex operations and options for generating these results.

Because data gets updated on a frequent basis, a system should be in place to prevent the user from viewing and analyzing out-dated information. Utilizing derived data and the reporting warehouse system developed through SQL will help to minimize this.

Derived data is an attribute whose values can be determined by applying an algorithm to other base attributes. An example would be to calculate the patient's current age or to calculate the total number of days on drug.

The reporting warehouse system consists of data or SQL views, permanent datasets and temporary datasets. They represent a collection of summary & item level datasets by key fields. All data views would be a dynamic representation of the actual dataset. In addition, there could be integrated data views defined for the most likely information requested by the FDA reviewer. For analysis of a patient group, a temporary dataset can be created for subsequent analysis.

Dataset Type	Dataset Information
Actual	Individual Clinical Studies
View	Integrated Studies - summary & item level
Temporary Dataset	Integrated Studies - patient group

SUMMARY

SAS offers the features of a relational database model needed to create an efficient CANDA system. In addition, there are numerous tools including PROC SQL to facilitate the generation of reports and analysis.

In summary, several techniques are available to take advantage of the database structure of typical clinical studies. These techniques facilitate the development and operation of a CANDA system. As the clinical protocols and regulatory review process become more focused, the database structure design becomes more standardized.

REFERENCES

Optimum Clinical Data Structures for Use with SAS/PH-Clinical ® Software, Sandra D. Schlotzhauer, Andrew T. Fagan, SAS Institute Inc.

SAS/PH-Clinical ® Software: CANDA Implementation Strategies, J. Meimei Ma, Quintiles Inc., Sandra D. Schlotzhauer, SAS Institute Inc.

Building Dictionary Technology with the SAS System to Improve the Clinical Data Review Process, Martin F. Michael, SAS Institute Inc.

Taking the First Steps with CANDAs, Joel Dobbs and Michele Hardy, Applied Clinical Trials - May 1992, Volume 1, No.1

CANDA 1995: An International Regulatory and Strategy Report, Speeding the CANDA Development Process: CANDAs and Clinical Data Base Design, Lofton Harris, Research Data Worldwide

Simplifying NDA Programming with PROC SQL, Aileen L. Yam, Besselaar Associates, Princeton, NJ

TRADEMARK INFORMATION

ABOUT THE AUTHOR

The author welcomes your comments & suggestions.

Sunil Gupta
Gupta Programming
213 Goldenwood Circle
Simi Valley, CA 93065-6772
(805) 577-8877
Sunil@GuptaProgramming.com
http://www.GuptaProgramming.com

ACKNOWLEDGMENTS

The author would like to thank Kirk Paul Lafler of Software Intelligence Corporation, Spring Valley, CA, Dr. Fred Hoehler of Data Management Center, Santa Ana, CA, Karen Walker of

Walker Consulting, San Diego, CA, and Don Carver of Don Systems Inc. for their invaluable assistance in the preparation of this paper.