

Begin with the End in Mind: Modeling Data to Achieve Warehousing Success

Paul Gilbert & Steve Light at DataCeutics Inc
Beth Atkinson & David Reasner at Sepracor Inc

ABSTRACT

At Sepracor Inc, clinical data are entered and stored utilizing the Oracle Clinical data management system. Data are extracted from Oracle Clinical to produce analysis-ready SAS datasets through the use of Oracle views, SAS views and SAS programs. Successful and efficient implementation of the extraction process was facilitated by prior development of a comprehensive data model encompassing both source and derived data.

The goal of this paper is to describe the critical importance of a robust data model in implementation of a data warehousing solution. Specific areas to be discussed include

- Developing the data model
- Features of the data model
- Defining source data structures
- Defining derived variables
- Defining transformed data structures
- Data access methods

In the final section we will discuss the potential application of the new Clinical Data Warehouse Administrator under development by the SAS Institute PharmaHealth Technologies Group.

INTRODUCTION

The Sepracor Reporting Data Mart (SRDM) is a reporting database, (i.e. data warehouse) that offers a standard and consistent structure to address Sepracor's clinical reporting needs. The SRDM defines standard data structures for reporting clinical data in the SAS programming environment and enables development of programming code to be shared between the different drug programs and protocols. The basis of the SRDM is the Oracle Clinical data model and therefore all of the Oracle Clinical raw data items are included in the SRDM. In addition to the source data items, the SRDM defines derived data elements. Derived data can either be generated and stored within the Oracle Clinical database or created as part of the data extraction process. How data are derived depends on complexity and extent of derivations. There are three methods for deriving data:

- The first is to derive data and store the values in an Oracle Clinical question using an Oracle Clinical derivation procedure, which may use Oracle Clinical functions, SQL or PL/SQL.
- The second is to create a complex question within an Oracle Clinical data extract using the Oracle Clinical extract view builder. The value can then be derived using Oracle Clinical functions, SQL or PL/SQL.
- The third is to use the SAS System with SAS/Access to read the Oracle Clinical views, derive data and store the data as a SAS dataset.

DEVELOPING THE DATA MODEL

The SRDM was developed by a team of experienced clinical data managers, clinical programmer/analysts and clinical statisticians. Designing the SRDM started with identifying the components of the data model. The main components of the data model are the Data Files, Keys and Data Items.

There were also two overriding factors we considered when designing the data model. The first was the design requirements of the Oracle Clinical data structures to facilitate the data entry and data

management processes. The second was the data structure design requirements to facilitate reporting and analysis. Another major consideration in development of the model was compliance with current ICH guidelines.

Using the strategy of "**Beginning with the End in Mind**", we developed the reporting data model concepts, developed the reporting data model and then going backward to the Oracle Clinical data structures, developed the source data structures. It is important to note that all source data elements collected and stored in Oracle Clinical are included in the SRDM. The next section describes attributes of the SRDM. Appendix 1 contains an example of the SRDM for adverse events.

FEATURES OF THE DATA MODEL

DATA MODEL OVERVIEW

The attributes defined for each data element in the SRDM are described below.

Table 1: SRDM Attributes

Column Name	Column Description
Data File Label:	Description of the Oracle Clinical DCM and SAS Data file (40 characters max).
Data File Name:	SAS Data file Name (8 characters max).
OC DCM:	Oracle Clinical Data Collection Module (DCM) Short name (4 characters max).
OC Question Group:	Oracle Clinical Question Group Short name (4 characters max).
OC Question Name:	Oracle Clinical Question name (7 characters max).
SAS Variable Name:	SAS Variable name (8 characters max).
SAS Variable Label:	Label for the Oracle Clinical Question and SAS Variable (40 characters max).
OC DVG	Oracle Clinical Discrete Variable Group (DVG) name (7 characters max).
Key Type/Order:	Identifies the variable as being a key variable. It also defines the order of the keys in the SAS Data file. This order must also be used to sort the data in the SRDM data file. The Key Types are indicated as follows: P="primary", F="fixed", S="secondary". For example: P1,F2,F3,S4,S5 defines the key type and order in the data file.
Type:	SAS variable type attribute of NUM or CHAR.
Length:	Oracle Clinical Question and SAS variable length.
SAS Format:	SAS format name. This may be a SAS format or a user-defined format. The maximum length is 8 characters, including the \$ preceding character format names.
Source: (Source, cont'd.)	Identifies whether the SRDM variable is Oracle Clinical source data (raw), derived in Oracle Clinical (derived), created within Oracle Clinical view processing (complex question) or derived using SAS processing (external).
Description derivation	Description of the SRDM question/variable and its derivation.

KEYS

Keys are fundamental to the SRDM structure. One or more keys uniquely identify a particular row in a SRDM data file. Primary keys uniquely identify every subject in a subject-level data file and must be present on all subject-level data files. Fixed keys exist in many SRDM subject-level data files for the purpose of merging/joining data files in a consistent way. Secondary keys are usually data file-specific and are used to uniquely identify a particular row in the SRDM file. The fixed and secondary keys can vary between data files. For example, the demographics data file will only contain primary keys as it is structured as a one row per subject table. The adverse event data file may contain four additional fixed and secondary keys that are used to uniquely identify a row.

- Primary keys are keys that uniquely identify individual subjects; this key(s) is present in every SRDM subject-level data file. The primary key(s) must be identical across each of the subject-level data files, i.e. the variables have the same length and type. Examples of primary keys are protocol and patient.
- Fixed keys are keys that have the same meaning across the SRDM data files and are used to describe how data files can be joined. These keys have a fixed relative position to the other data file keys. For example, if the keys named “interval” and “visit” are used in five data files, then they must always be the second and third key in all of these data files.
- Secondary keys are keys that are unique to a data file and are used to uniquely identify a row within the data table. Examples of these may be body system, preferred term and onset date contained within the adverse event data file or assessment dates and times within the vital signs data file.

Data files that are not subject-specific may use a different key structure than the subject-level data files. An example of a non-subject-specific data file is a lab range data file, which might contain keys such as analyte, gender, age and range.

NAMING CONVENTIONS

A convention was established such that Oracle Clinical Questions and SAS variables are named using up to seven character; whenever possible Oracle Clinical Question and SAS variable names are the same. All variables, except primary and fixed secondary keys, are prefixed by the abbreviated data file name (e.g. **dm_age**, **ae_act**). The prefix (**dm_**, **ae_**) assists in grouping variables from a specific data file. The prefix will also help to identify the source data file. Primary key variable names and fixed secondary key variable names are easily identified because they are common to many SRDM data files and are not prefixed.

In the SRDM data files that utilize dates there is often an associated time variable for each date variable. Date and time variables are typically used in one of two ways. First, some data files contain a single date, such as an evaluation date; second, other data files contain begin and end dates for individual observations. The date and time variables use a standard suffix, described below. Additionally, the data file may contain a derived “study day” variable that represents the number of days relative to a specified time point in the study. Each entered date and time is stored in the database as two separate questions/variables. The first question contains the date or time as collected on the CRF; these are stored as text and extracted to SAS as character. The second question contains the derived date or time; these are stored as an Oracle Clinical date type and extracted to SAS as a numeric with a date or time format.

Table 2: Date/time types and standard names

variable	suffix	example
actual date	_da	lb_da
derived date	_dd	lb_dd
actual time	_ta	lb_ta
derived time	_td	lb_td
actual begin date	_bda	ae_bda
derived begin date	_bdd	ae_bdd
actual begin time	_bta	ae_bta
derived begin time	_btd	ae_btd
actual end date	_eda	ae_eda
derived end date	_edd	ae_edd
actual end time	_eta	ae_eta
derived end time	_etd	ae_etd
study day	_dy	ae_dy

DEFINING SOURCE DATA STRUCTURES

The Oracle Clinical (source) data structure must be optimized to facilitate the data entry and data management process. The most important factor to consider when designing the source data structures is data normalization. Oracle Clinical is designed to, and works best with, denormalized data structures. The following situation illustrates the difference between a denormalized and normalized data structure for vital signs data.

Table3: Denormalized vs. Normalized structures

	<u>Denormalized data structure</u>		
<u>Visit</u>	<u>systolic</u>	<u>diastolic</u>	<u>respiration</u>
1	120	80	20
2	118	82	21
	<u>Normalized data structure</u>		
<u>Visit</u>	<u>vsparm</u>	<u>vsvalue</u>	
1	systolic	120	
1	diastolic	80	
1	resp	20	
2	systolic	118	
2	diastolic	82	
2	resp	21	

Another factor considered when defining and naming the source data structures is how to extract the values of an Oracle Clinical data item to SAS. There are several ways of extracting OC question information for questions that are associated with a DVG. The methods for data extraction have an impact on how SAS user-defined formats are utilized. OC questions that are associated with DVGs can be extracted in several formats, three are listed below.

- The default extraction method provides the DVG short value (DVG_SHORT_VALUE), this value usually represents the value which is stored in the OC data table.
- DVG Number (DVG_NUMBER), which is the DVG sequence number.
- DVG Long Value (DVG_LONG_VALUE), which is the long description for the DVG value.

We decided that as default, all questions that are associated with DVGs would be extracted using all three methods. Extracting all three values for one question allows the SAS

programmer to dynamically create user-defined SAS formats when needed. Additionally, it eliminates the need for creating a SAS format library for DVG values. The OC default question/SAS variable is extracted with the DVG short value. The other two extractions produce variables with the same OC question name and a modified SAS variable name. The variables extracted using the DVG Number format are named by appending a "N" to the SAS variable name. The variables extracted using the DVG Long Value are named by appending a "L" to the SAS variable name. An example is listed below.

Table 4: DVG Extraction methods

Extract method	SAS Variable name	Extracted value
DVG Short value	DM_RACE (Race)	DVG_SHORT_VALUE attribute
DVG number	DM_RACEN (Race)	DVG_NUMBER attribute
DVG long value	DM_RACEL (Race)	DVG_LONG_VALUE attribute

DEFINING DERIVED VARIABLES

Some derived data items are stored in the Oracle Clinical database and some derived data may be created during the data extraction process. Reasons for storing derived data items in the Oracle Clinical database include availability of these derived items when running Oracle Clinical edit checks, and for data browsing. The mechanism for deriving data will depend on complexity and extent of derivations. There are three methods for deriving data:

- The first is to derive data and store the values in an Oracle Clinical question using an Oracle Clinical derivation procedure, which may use Oracle Clinical functions, SQL or PL/SQL.
- The second is to create a complex question within an Oracle Clinical data extract using the Oracle Clinical extract view builder. The value can then be derived using Oracle Clinical functions, SQL or PL/SQL. Complex questions become part of extract views but are not stored in OC questions.
- The third is to use the SAS System with SAS/Access to read the Oracle Clinical views, derive data and store the data as a SAS dataset.

Most of the derived data items are created using the second method. The third method is discussed in the following section.

DEFINING TRANSFORMED DATA STRUCTURES

The reporting data structure must be optimized to facilitate reporting and analysis process. The most important factor to consider when designing the analysis and reporting data structures is data normalization. For our needs, the normalized data structure is preferred for analysis and reporting. The normalized data structure preference is in conflict with the denormalized Oracle Clinical data structure intended use. This situation is rectified by creating transformed data structures external to Oracle Clinical for data files where normalized structures are preferred. These are data files such as electrocardiograms, laboratories and vital signs where data are collected for several time points (i.e. visits, dates, times) and several parameters.

In this case the SAS System with SAS/Access is used to read the Oracle Clinical views, derive data and create the transformed data structure, and store the data as a static SAS dataset.

DATA ACCESS METHODS

Surfacing the final SRDM is a three level process where default views are created within Oracle Clinical, the default views are modified, and

SAS/Access is used to access the Oracle Views.

- Within Oracle Clinical, data are collected and stored in logical groupings of data items. These groupings are named Data Collection Modules (DCM). An example of a DCM is Demographics or Adverse Events. By default, Oracle Clinical creates one or several Oracle views for each DCM. The default Oracle views contain the source data items and the derived questions which are derived using Oracle Clinical derivation procedures.
- The default views are then modified using the Oracle Clinical extract view builder. In this step the extract view builder is used to generate data items as complex questions.
- Oracle Clinical generates SAS programs used to create SAS views linked to the Oracle views.
- The SAS System with SAS/Access is used to read the Oracle Clinical views. In this step SAS coding is used to derive additional data items or create the appropriate transformed data structure. All of the data are then stored as static SAS datasets.

At this point the analysts and statisticians can access the static copy of SRDM. Additionally, they can access the live source data using the Oracle Clinical views.

CONCLUSION

Using the strategy of "**Beginning with the End in Mind**", we were able to quickly design a Warehouse to fit our Analysis and Reporting needs, and to facilitate our Oracle Clinical data model design. The Warehousing "tools" in our case are a combination of the Oracle Clinical extract view builder and the SAS System.

The SAS Institute PharmaHealth Technologies Group is currently developing a clinical data warehouse tool set named SAS/PH.DataWare. It is possible that this tool can be used to replace our custom SAS code at several points in the data extraction process.

- Automate creation of the SAS/Access views to the Oracle Clinical views.
- Streamline creation of derived data and transformed data structures.
- Automate scheduling for data extractions to refresh the SRDM.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Paul Gilbert
 DataCeutics Inc.
 244 High Street, Suite 203
 Pottstown, PA 19464
 Phone: 610-970-2333
 Email: gilbertp@dataceutics.com

DataCeutics Inc. is a SAS Institute Quality Partner and SAS Institute PharmaHealth Technologies Quality Partner.



BIOGRAPHY

Paul Gilbert is Vice President of DataCeutics, Inc., a consulting/outsourcing company specializing in software solutions, system integration, programming and support in the areas of clinical data management and statistical reporting. His seventeen years' experience includes clinical data management, implementing and maintaining Domain Clintrial, designing and developing SAS-based biostatistics reporting systems, managing SAS based NDA programming support, SAS/PH-Clinical implementation and support, and clinical systems integration. He has a BS in Science.

Steve Light is Director of Clinical Reporting at DataCeutics, Inc. He is responsible for SAS report macro development and biostatistics/clinical SAS programming support. His fourteen years of experience with clinical information systems include SAS systems development and validation, NDA driven project management, SAS/PH-Clinical implementation and support, clinical programming and clinical data management. He has a BS in Biological Sciences.

Beth Atkinson is the Associate Director of Clinical Data Management at Sepracor, Inc. where she is responsible for the planning and implementation of an internal data management function as well as overseeing outsourced data management projects. Prior to Sepracor, Ms. Atkinson spent five years at Parexel International Corporation as a line and project manager. She was a member of an international task force whose mission was to implement corporate worldwide data management standards and procedures. Ms. Atkinson received a BS in Medical Technology from the University of Wisconsin - Madison and an MBA from Boston University.

David Reasner Ph.D. is Vice President, Clinical Operations and Data Analysis, at Sepracor, Inc. located in Marlborough, Massachusetts. Responsibilities include outsourcing clinical trials in five drug development programs while managing a growing department including clinical personnel, medical writers, data managers, programmers and biostatisticians. Previously Reasner worked as Senior Research Associate at Statistics Unlimited in Wellesley, Massachusetts with responsibilities in protocol development, analysis, and reporting for clinical and preclinical studies to support drug registration.

Appendix 1
Sample SRDM for Adverse Events

DF_LABEL	DF_NAME	OC_DCM	OC_QG	OC_Q	SAS_NAME	SAS_LABEL	OC_DVG	KEY	TYPE	LENGTH	SAS_FMT	SOURCE	Descrip
Adverse Events	ADVERSE				PATID	Subject ID		P1	CHAR	14		complex key	Subject Identifier derived from study, invsite and pt.
Adverse Events	ADVERSE	AE	AE	AE_CBS	AE_CBS	AE: Costart Body System		S2	CHAR	50		Derived	Costart Body System
Adverse Events	ADVERSE	AE	AE	AE_CPT	AE_CPT	AE: Costart Preferred Term		S3	CHAR	60		Derived	Costart Preferred Term
Adverse Events	ADVERSE	AE	AE	AE_BDD	AE_BDD	AE: Begin Date		S4	DATE	8	DATETIME	Derived	AE Onset Date
Adverse Events	ADVERSE	AE	AE	AE_BTD	AE_BTD	AE: Begin Time		S5	TIME	6	DATETIME	Derived	AE Onset Time
Adverse Events	ADVERSE	AE	AE	AE_EDD	AE_EDD	AE: End Date		S6	DATE	8	DATETIME	Derived	AE End Date
Adverse Events	ADVERSE	AE	AE	AE_ETD	AE_ETD	AE: End Time		S7	TIME	6	DATETIME	Derived	AE End Time
Adverse Events	ADVERSE	AE	AE	AE_ANY	AE_ANY	AE: AE Present?	YES_NO		CHAR	3		Raw	Indicator Question- Any AEs?
Adverse Events	ADVERSE	AE	AE	AE_NUM	AE_NUM	AE: AE Number	XX_NUM		CHAR	4		Raw	AE Number
Adverse Events	ADVERSE	AE	AE	AE_TEXT	AE_TEXT	AE: Reported Term			CHAR	200		Raw	AE Verbatim Description
Adverse Events	ADVERSE	AE	AE	AE_CBSC	AE_CBSC	AE: Costart Body System Code			CHAR	20		Derived	Costart Body System Code
Adverse Events	ADVERSE	AE	AE	AE_CPTC	AE_CPTC	AE: Costart Preferred Term Code			CHAR	30		Derived	Costart Preferred Term Code
Adverse Events	ADVERSE	AE	AE	AE_BDA	AE_BDA	AE: Actual Begin Date			DATE	8		Raw	AE Actual Onset Date
Adverse Events	ADVERSE	AE	AE	AE_BTA	AE_BTA	AE: Actual Begin Time			TIME	6		Raw	AE Actual Onset Time
Adverse Events	ADVERSE	AE	AE	AE_EDA	AE_EDA	AE: Actual End Date			DATE	8		Raw	AE Actual End Date
Adverse Events	ADVERSE	AE	AE	AE_ETA	AE_ETA	AE: Actual End Time			TIME	6		Raw	AE Actual End Time
Adverse Events	ADVERSE	AE	AE	AE_ACT	AE_ACT	AE: Action	AE_ACT		CHAR	2		Raw	AE Action Taken with Study Drug
Adverse Events	ADVERSE	AE	AE	AE_FRQ	AE_FRQ	AE: Frequency	AE_FRQ		CHAR	2		Raw	AE Frequency
Adverse Events	ADVERSE	AE	AE	AE_SEV	AE_SEV	AE: Severity	AE_SEV		CHAR	2		Raw	AE Severity
Adverse Events	ADVERSE	AE	AE	AE_OUT	AE_OUT	AE: Outcome	AE_OUT		CHAR	2		Raw	AE Outcome
Adverse Events	ADVERSE	AE	AE	AE_REL	AE_REL	AE: Relationship	AE_REL		CHAR	2		Raw	AE Relationship to Study Drug
Adverse Events	ADVERSE	AE	AE	AE_SER	AE_SER	AE: Serious	YES_NO		CHAR	3		Raw	AE- Is Event Serious?
Adverse Events	ADVERSE	AE	AE	AE_TX	AE_TX	AE: Treatment	AE_TX		CHAR	2		Raw	AE Treatment
Adverse Events	ADVERSE	AE	AE	AE_COM	AE_COM	AE: Comment			CHAR	200		Raw	AE Comment
Adverse Events	ADVERSE				AE_TXE	AE: Treatment Emergent			CHAR	3		complex question	Is AE Treatment Emergent?
Adverse Events	ADVERSE				AE_DY	AE: Study Day			NUM	8		complex question	Subtract onset date (AE_BDD) from first treatment date (SD_FTDD) + 1
Adverse Events	ADVERSE				AE_DUR	AE: Duration			NUM	8		complex	AE Duration of event

Events								
Adverse Events	ADVERSE	AE_DURU	AE: Duration Unit	CHAR	10	question complex question external	AE Duration of event units	
Adverse Events	ADVERSE	AE_ODOS	AE: Dose at Onset	CHAR	20	external	Dose of study drug at AE onset	
Adverse Events	ADVERSE	AE_RTF	AE: Rel Time First Dose	NUM	8	complex question	Elapsed Time from First Dose of Study Therapy. Subtract the date of first dose from the date of the evaluation.	
Adverse Events	ADVERSE	AE_RTL	AE: Rel Time Last Dose	NUM	8	complex question	Elapsed Time from Last Dose of Study Therapy. Subtract the date of last dose from the date of the evaluation.	
Adverse Events	ADVERSE	AE_RTU	AE: Rel Time Unit	CHAR	10	complex question	Relative Time Units.	