The CTSA Ontology Mapper and Discovery Suite: A Rules-Based Approach to Integrated Data Repository Deployment

Translational Data Warehousing Design

Strategies for Supporting the Ontology Mapper Project

Revision History

Name	Date	Reason For Changes	Version
Marco Casale	5/30/08	Initial draft	0.1
Marco Casale	7/16/08	Added Prakash's work on Mapper Database Design. Updated Core i2b2 schema design	0.2

Translational Data Warehouse Design
Strategies for Supporting the Ontology Mapper Project

Contents

1. Project Overview ... 5

1.1 Document Definitions ... 5

2. Data Warehousing Concepts ... 5

3. Database Design Concepts ... 8

3.1 Structured Database Design ... 8

3.2 Dimensional Database Design ... 9

3.3 Entity, Attribute, Value Architecture ... 11

3.4 Integrated Data Repository Design ... 11

4. Example Data Warehousing Design Approaches ... 11

4.1 Partners Healthcare RPDR ... 11

4.2 Partners Healthcare i2b2 ... 13

5. Mapper Database Design - Extending i2b2 ... 14

5.1 Encoding Dimension ... 14

5.2 Map Dimension ... 14

5.3 Observation Fact ... 15

5.4 Map Data Fact ... 15

5.5 Observation Map Link ... 16

References ... 17

Figures

Figure 1 - Data Warehouse Architecture - (Inmon et al., 2001) 6
Figure 2 - Data Warehousing Components (Kimball, 2002) 7
Figure 3 - Normalized Data Model Example 9
Figure 4 - Star Schema Example (Kimball, 2002) 10
Figure 5 - i2b2 Database Design 13
Figure 6 - Mapper Database Design 16

Translational Data Warehouse Design
Strategies for Supporting the Ontology Mapper Project

1. Project Overview

The Ontology Mapper Project is intended to provide a mechanism for enhancing translational bioinformatics. This project takes a different approach to the standard data warehousing model and enhances it with the use of ontology mapping for biomedical terminology. In order to accommodate this novel approach, a review of current data warehousing design will be compared and contrasted.

1.1 Document Definitions

Term	Definition
IDR	Integrated Data Repository
ETL	Extract, Transform and Load
HL7	Health Level 7 – Standard used for information transportation amongst disparate IT systems
i2b2	Informatics for Integrating Biology and the Bedside
ODS	Operational Data Store
CIF	Corporate Information Factory
Fact	A business measure
Dimension	Textual descriptions of the business. Descriptions of facts

2. Data Warehousing Concepts

Data warehousing is a method used to convert data into information. A clinical data warehouse can provide a structure for organizing data and render a positive impact for the purpose of patient care, biomedical research and biomedical education. However, developing a clinical data warehouse encompasses many difficult challenges. The data warehouse cannot be designed in one simple step.

The data warehouse architecture model shown in Figure 1 depicts the process of transforming operational data into information for the purpose of generating knowledge within an organization. The diagram displays data flowing from left to right in accordance with the corporate information factory (CIF) approach (Inmon et al, 2001). According to Inmon, the data enters the CIF as raw data collected by operational applications. The data is transformed through extract, transform and load processes and stored in either the data warehouse or the ODS, operational data store. “Often up to 80 percent of the work in building a data warehouse is devoted to the extraction, transformation, and load (ETL) process: locating the data; writing programs to extract, filter, and cleanse the data; transforming it into common encoding scheme; and loading it into the data warehouse." (Hobbs, Hillson & Lawande, 2003, 6)

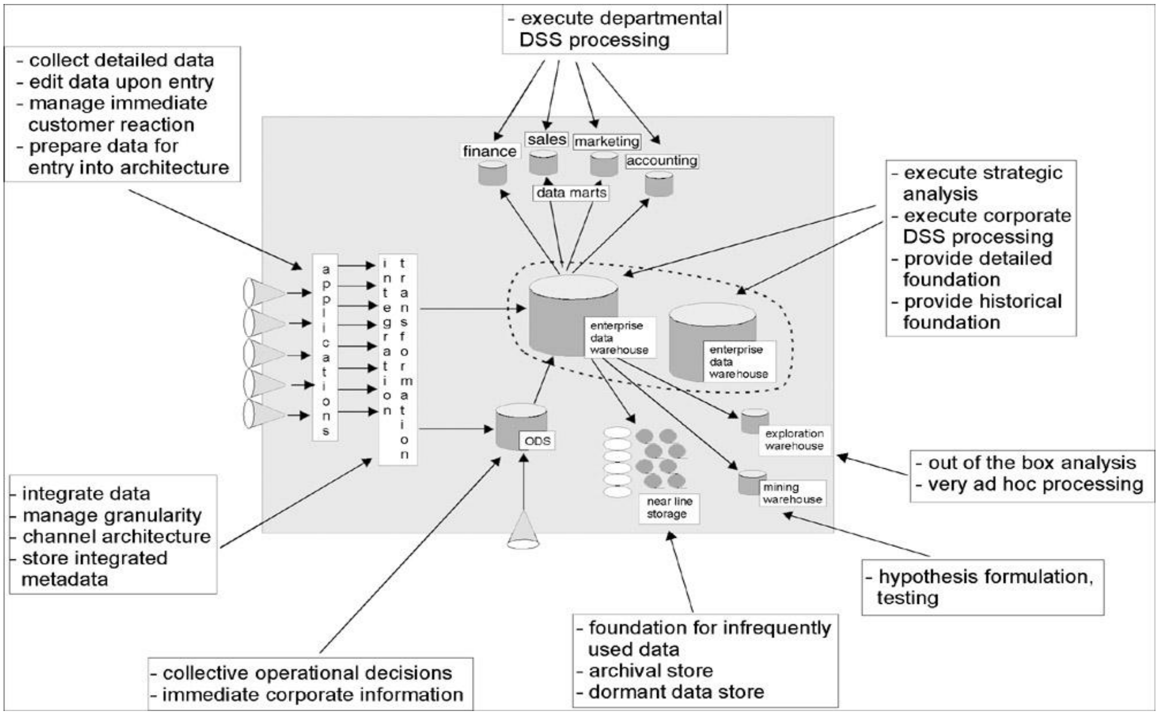

- execute departmental DSS processing
- collect detailed data
- edit data upon entry
- manage immediate customer reaction
- prepare data for entry into architecture
finance
sales
marketing
accounting
data marts
- execute strategic analysis
- execute corporate DSS processing
- provide detailed foundation
- provide historical foundation
applications
integration
transformation
enterprise data warehouse
enterprise data warehouse
exploration warehouse
ODS
near line storage
mining warehouse
- out of the box analysis
- very ad hoc processing
- integrate data
- manage granularity
- channel architecture
- store integrated metadata
- hypothesis formulation, testing
- foundation for infrequently used data
- archival store
- dormant data store
- collective operational decisions
- immediate corporate information

Figure 1 - Data Warehouse Architecture - (Inmon et al., 2001)

According to Kimball (2002), the data warehousing architecture can further been delineated into four categories:

- Source Systems
- Data Staging Area
- Data Presentation Area
- Data Access Tools

Source Systems may be composed of operational systems, legacy based systems or other data repositories. Data is transmitted or extracted from the source systems into the data staging area where it is processed. It should be noted that the staging area is not intended for direct query access. The purpose of data processing is to normalize the data, using standard data types, data lengths, precision of data and data naming conventions amongst the inbound extractions.

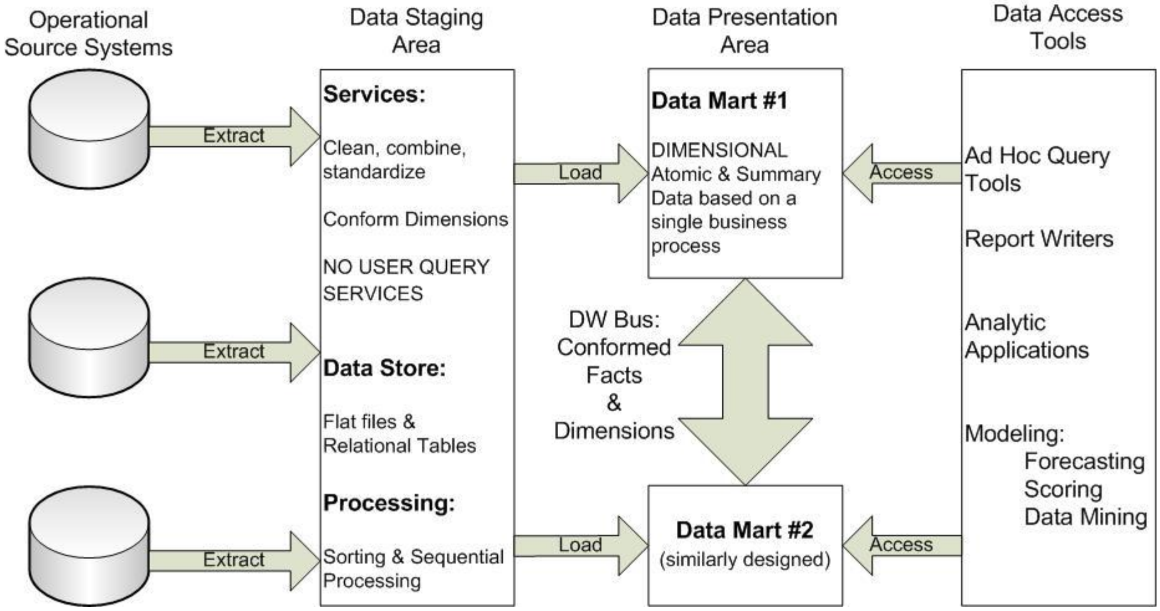


Figure 2 - Data Warehousing Components (Kimball, 2002)

Furthermore, after the data has been prepared, it is loaded into the de-normalized schema of the data warehouse or data marts and resides there in a fine grain level of detail. The logical design of a data warehouse is usually composed of the star schema. “A star schema is a simple database design (particularly suited to ad-hoc queries) in which dimensional data (describing how data are commonly aggregated) are separated from fact or event data (describing individual business transactions).” (Hoffer, Prescott & McFadden, 2002, 421) The data mart design is intended to support the data presentation layer.

The final stage is in the process of converting data into information through reporting, analysis and data mining techniques. (Inmon et al, 2001) Kimball describes this stage as the data access tools area or in modern day vernacular, the business intelligence layer.

3. Database Design Concepts

There are a variety of data modeling approaches for data warehousing design. These approaches can range from highly structured database design, dimensional database design, entity, attribute, value design, object oriented design to name a few.

3.1 Structured Database Design

Operational transaction processing systems are commonly created using a highly structured or normalized data model. This model is intended to reduce several anomalies (insert, update and delete) that may occur during data transaction processing. The goal of normalization theory, developed by E.F. Codd and CJ Date, is to determine where the functional dependencies exist within and amongst relations. It is supported by mathematical set theory, relational algebra and relational calculus. The end result is to produce a data model which reduces the opportunity for data redundancy, and provides the highest level of data consistency. Figure 3 is a depiction of a third normal form data model of patient visits and their associated observations.

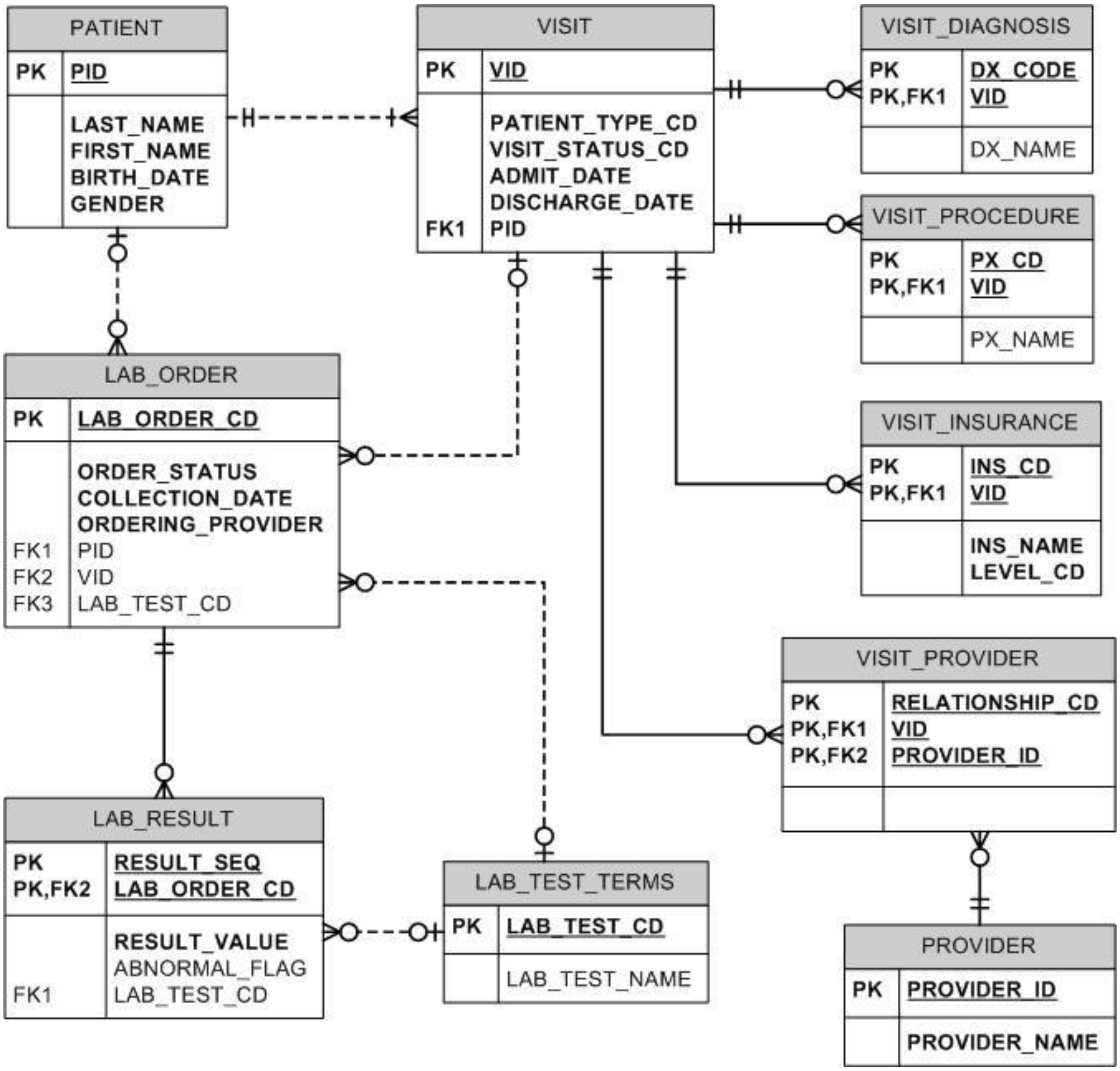


Figure 3 - Normalized Data Model Example

3.2 Dimensional Database Design

In contrast, Kimball proposed a dimensional modeling approach for data modeling design in support of high performance queries rather than high volume transaction processing. The dimensional model is composed of a fact table joined to several dimensions. A fact table is the primary table of the dimensional model. Each fact table contains measurable facts. "A row in a fact table corresponds to a measurement. A measurement is a row in a fact table. All the measurements in a fact table must be at the same grain." (Kimball, 2002). Figure 4 is an example of a dimensional model that illustrates a fact table of 'Daily Sales' related to dimensions of Date, Product and Store.

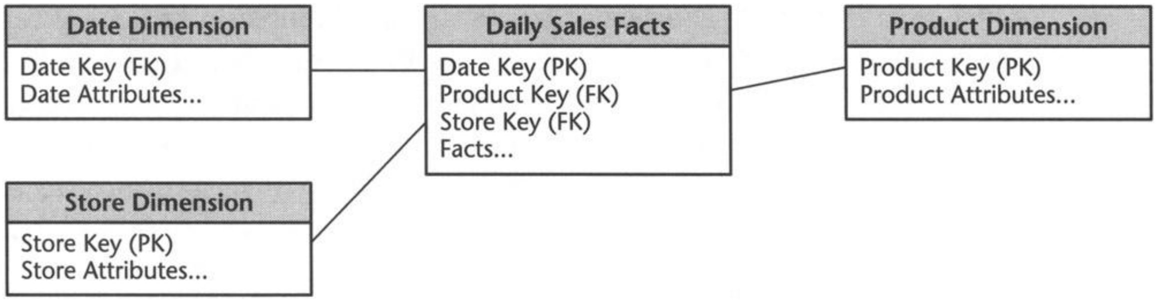


Figure 4 - Star Schema Example (Kimball, 2002)

"The dimension tables contain the textual descriptors of the business." (Kimball, 2002). Each dimension table may be composed of many attributes. The dimension attributes are used to help qualify the criteria used in generating queries from the star schema. Every dimension table contains a primary key. Furthermore, the fact table contains a foreign key to every dimension table and typically, the primary key of the fact table is a composition of all of the foreign keys in the fact table, a composite primary key.

Now that the transformed data resides in the underlying fact and dimension tables of the data warehouse, the next step entails deriving the necessary aggregates that will define the associated data marts. "An aggregate is simply the rollup of an existing fact table along one of its dimensions, which more often than not is time." (Scalzo, 2003, 147) For example, if a fact table holds daily automobile fatalities, then location-based aggregates might be town, county, and state. Thus, since the overall purpose of a data warehouse is to support the online analytical processing (OLAP) capabilities associated with data mining, the data in the warehouse must be propagated, as referred to in Figure 1, into information residing within data marts.

Data marts serve to hold specialized, codified, dimensional, aggregated data roll-ups in support of the OLAP tools that will be used to discover business relationships and organization insights. However, slicing a cube does not always yield the desired knowledge. As a result, there are times when exploratory data mining processes must be initiated against the data warehouse itself, as opposed to a specialized multidimensional array. This drill down technique can also be accomplished with OLAP tools, specialized data mining software or even customized ad-hoc programmatic queries running against the finer level of detail. The data access tools in Figure 2, processes all of these requests and coordinates the communication exchange between the warehouse or marts and the end user's tools.

3.3 Entity, Attribute, Value Architecture

3.4 Integrated Data Repository Design

Inferenced Based Approach
Harvest Table Design

4. Example Data Warehousing Design Approaches

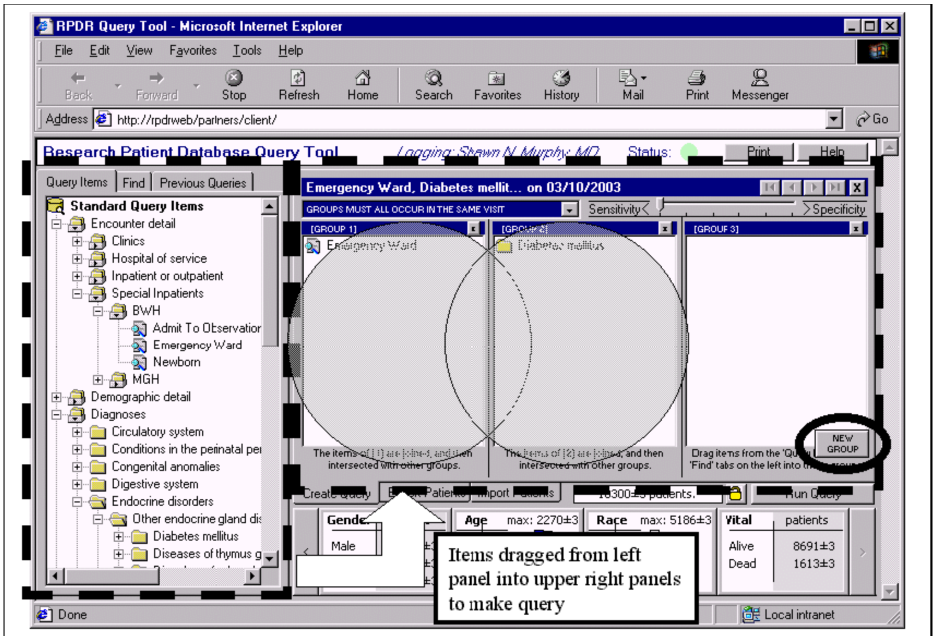
4.1 Partners Healthcare RPDR

A significant challenge for applications of clinical data warehousing has been the ability to make them facile for non-database experts. Typically, database experts are required to write complex query statements in order to extract data. The following example depicts how one organization dealt with this difficult dilemma.

Partners Healthcare Inc. created a research patient data repository (RPDR) in support of clinical research. Partners recognized the need for several database experts writing complex queries, when they started to receive 100-200 query requests per week. Partners decided to develop a graphical analytic tool to help novice database users create effective mining algorithms. (Murphy et al, 2003) Partners set out to develop a Querytool for the RPDR shown in Figure 5. However, they were confronted with many challenges.

Murphy described the first challenge as allowing novice population of computer users to navigate the metadata and formulate the queries without computer expert intervention. The Querytool leverages medical vocabularies such as ICD9, SNOMED-CT, LOINC, NDC and CPT4. Unfortunately, each vocabulary presents its own obstacles. For instance, ICD9 codes contain categories that can be used respectively for coding. On the other hand, NDC drug codes can be reused so that their meanings change from year to year. (Murphy et al).

The second challenge involved construction logic. “Patient inclusion and exclusion criteria for most research studies are jumbles of different types of information.” (Murphy et al). For instance, a researcher might request all patients who have been diagnosed with a certain disease and are in a particular age group but do not take a certain medication. Therefore, it was the Querytool’s objective to disguise the complex construction logic with an intuitive interface (Figure 5).


RPDR Query Tool - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Messenger
Address http://rpdrweb/partners/client/
Go
Research Patient Database Query Tool
Logging: Shawn N Murphy, MD
Status:
Print
Help
Query Items
Find
Previous Queries
Standard Query Items
Encounter detail
Clinics
Hospital of service
Inpatient or outpatient
Special Inpatients
BWH
Admit To Observatior
Emergency Ward
Newborn
MGH
Demographic detail
Diagnoses
Circulatory system
Conditions in the perinatal per
Congenital anomalies
Digestive system
Endocrine disorders
Other endocrine gland dis
Diabetes mellitus
Diseases of thymus g
Emergency Ward, Diabetes mellit... on 03/10/2003
GROUPS MUST ALL OCCUR IN THE SAME VISIT
Sensitivity
Specificity
[GROUP 1]
Emergency Ward
[GROUP 2]
Diabetes mellitus
[GROUP 3]
The items of [1] are joined, and then intersected with other groups.
The items of [2] are joined, and then intersected with other groups.
Drag items from the 'Query' 'Find' tabs on the left into th
NEW GROUP
Create Query
Run Query
patients.
Gender
Male
Age max: 2270±3
Race max: 5186±3
Vital
patients
Alive 8691±3
Dead 1613±3
Done
Local intranet

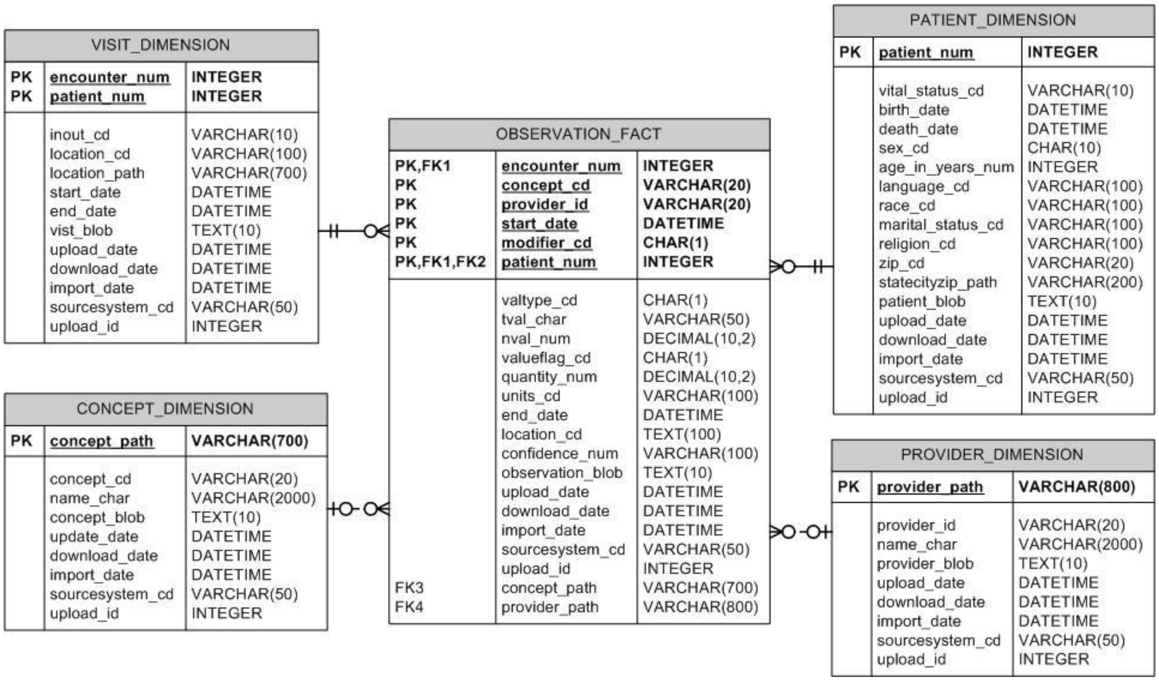
Items dragged from left panel into upper right panels to make query

Figure 5 – RPDR Querytool used for selected cohorts from a tree structure into a Venn Diagram (Murphy et al, 2003)

In short, the Querytool is an excellent example of a data warehouse graphical user interface to novice database users for clinical research. The end user is easily able to narrow down the 1.8 million patients at Partners to a manageable number between 100-1000 patients. Finally, Murphy states "The Querytool has achieved excellent acceptance at Partners Healthcare Inc. and is the most prevalent way of obtaining research cohorts."

4.2 Partners Healthcare i2b2

The following figure depicts the core i2b2 database schema. It conforms to a star schema design based on Ralph Kimball's initial work.



VISIT_DIMENSION

PK	**encounter_num**	**INTEGER**
PK	**patient_num**	**INTEGER**
	inout_cd	VARCHAR(10)
	location_cd	VARCHAR(100)
	location_path	VARCHAR(700)
	start_date	DATETIME
	end_date	DATETIME
	vist_blob	TEXT(10)
	upload_date	DATETIME
	download_date	DATETIME
	import_date	DATETIME
	sourcesystem_cd	VARCHAR(50)
	upload_id	INTEGER

CONCEPT_DIMENSION

PK	**concept_path**	**VARCHAR(700)**
	concept_cd	VARCHAR(20)
	name_char	VARCHAR(2000)
	concept_blob	TEXT(10)
	update_date	DATETIME
	download_date	DATETIME
	import_date	DATETIME
	sourcesystem_cd	VARCHAR(50)
	upload_id	INTEGER

OBSERVATION_FACT

PK,FK1	**encounter_num**	**INTEGER**
PK	**concept_cd**	**VARCHAR(20)**
PK	**provider_id**	**VARCHAR(20)**
PK	**start_date**	**DATETIME**
PK	**modifier_cd**	**CHAR(1)**
PK,FK1,FK2	**patient_num**	**INTEGER**
	valtype_cd	CHAR(1)
	tval_char	VARCHAR(50)
	nval_num	DECIMAL(10,2)
	valueflag_cd	CHAR(1)
	quantity_num	DECIMAL(10,2)
	units_cd	VARCHAR(100)
	end_date	DATETIME
	location_cd	TEXT(100)
	confidence_num	VARCHAR(100)
	observation_blob	TEXT(10)
	upload_date	DATETIME
	download_date	DATETIME
	import_date	DATETIME
	sourcesystem_cd	VARCHAR(50)
	upload_id	INTEGER
FK3	concept_path	VARCHAR(700)
FK4	provider_path	VARCHAR(800)

PATIENT_DIMENSION

PK	**patient_num**	**INTEGER**
	vital_status_cd	VARCHAR(10)
	birth_date	DATETIME
	death_date	DATETIME
	sex_cd	CHAR(10)
	age_in_years_num	INTEGER
	language_cd	VARCHAR(100)
	race_cd	VARCHAR(100)
	marital_status_cd	VARCHAR(100)
	religion_cd	VARCHAR(100)
	zip_cd	VARCHAR(20)
	statecityzip_path	VARCHAR(200)
	patient_blob	TEXT(10)
	upload_date	DATETIME
	download_date	DATETIME
	import_date	DATETIME
	sourcesystem_cd	VARCHAR(50)
	upload_id	INTEGER

PROVIDER_DIMENSION

PK	**provider_path**	**VARCHAR(800)**
	provider_id	VARCHAR(20)
	name_char	VARCHAR(2000)
	provider_blob	TEXT(10)
	upload_date	DATETIME
	download_date	DATETIME
	import_date	DATETIME
	sourcesystem_cd	VARCHAR(50)
	upload_id	INTEGER

Figure 5 - i2b2 Database Design

5. Mapper Database Design - Extending i2b2

The proposed Mapper functionality that enables creation of maps for transformation of data from source encoding to target encoding required several changes on the original i2b2 schema. These changes that were effected on the i2b2 schema are outlined herewith on a table by table basis.

5.1 Encoding Dimension

This is a new table created for the purpose of storing the various encodings used in the Mapper system. ENCODING_CD serves as the primary key for the table. Other columns of the table are enlisted below:

- ENCODING_NAME – the name of the encoding
- ENCODING_DESC – description of the encoding
- CREATE_USER – the user that created the encoding
- CREATE_DATE – the date of creation of the encoding
- UPDATE_USER – the user that last updated the encoding
- UPDATE_DATE – the date of last update of the encoding

5.2 Map Dimension

This table has been created to store info on the mapper xml instances uploaded onto the system. MAP_ID which is a running sequence no. serves as the primary key for this table. Other columns of this table are enlisted below:

- MAP_NAME – the name assigned for this map
- CONCEPT_CD – the concept code for which the map has been written
- CONCEPT_PATH – the concept path for which this map applies
- SOURCE_ENCODING – the source encoding of the map
- TARGET_ENCODING – the target encoding of the map
- MAP_PATH – the physical path of the mapper xml instance in the system
- MAP_DESC – description of the map
- IMPORT_DATE – the date of import of the map into the system
- UPDATE_DATE – the date of last update of the map
- UPLOAD_USER – the user that uploaded the map onto the system
- UPDATE_USER – the user that last updated the map
- LAST_RUN_DATE – the last execution date of the map

5.3 Observation Fact

- Added column CONCEPT_PATH referenced from table CONCEPT_DIMENSION for storing the concept path pertaining to the concept code used in this table. This is required since the original i2b2 design supports only the CONCEPT_CD (this can be the same for multiple concept paths) and resolving the concept path from the same is difficult.
- Added column ENCODING_CD referenced from table ENCODING_DIMENSION to denote the encoding type this record is encoded with.
- Column OBSERVATION_FACT_ID added to store a running sequence no. for uniquely identifying each record in the table.

5.4 Map Data Fact

- This is a new table that closely resembles the OBSERVATION_FACT and is designed to store the records created as a result of map execution. All the transformed records in target encodings will be housed in this table.
- This table has been created to segregate the transformed data from the source data and also to keep the i2b2 fact table (OBSERVATION_FACT) design intact. Mapper functionality required several design changes on the OBSERVATION_FACT which might have rendered this table incompatible with future i2b2 releases. This new table resolves the aforesaid issue by providing a similar structure as the i2b2 fact table with the flexibility of effecting new design changes required for the Mapper functionality.
- In accordance with Mapper functionality the new table is designed to have a new primary key (rather a unique key) with the columns CONCEPT_PATH, ENCODING_CD, MAP_ID being added to the existing primary key combination of the i2b2 fact table (OBSERVATION_FACT).
- MAP_ID column of this table is referenced from MAP_DIMENSION and stores info on the map that created this record.
- MAP_DATA_FACT_ID column is a running sequence no. that uniquely identifies each record in the table and is similar to the OBSERVATION_FACT_ID column of the OBSERVATION_FACT.
- All other columns in this table resemble those in the OBSERVATION_FACT both in name and purpose.

5.5 Observation Map Link

This table provides the link between the i2b2 fact table (OBSERVATION_FACT) and the mapper fact table (MAP_DATA_FACT). In other words it links the transformed records with the source records. The combination of OBSERVATION_FACT_ID (referenced from OBSERVATION_FACT) and MAP_DATA_FACT_ID (referenced from MAP_DATA_FACT) serve as the primary key of the table.

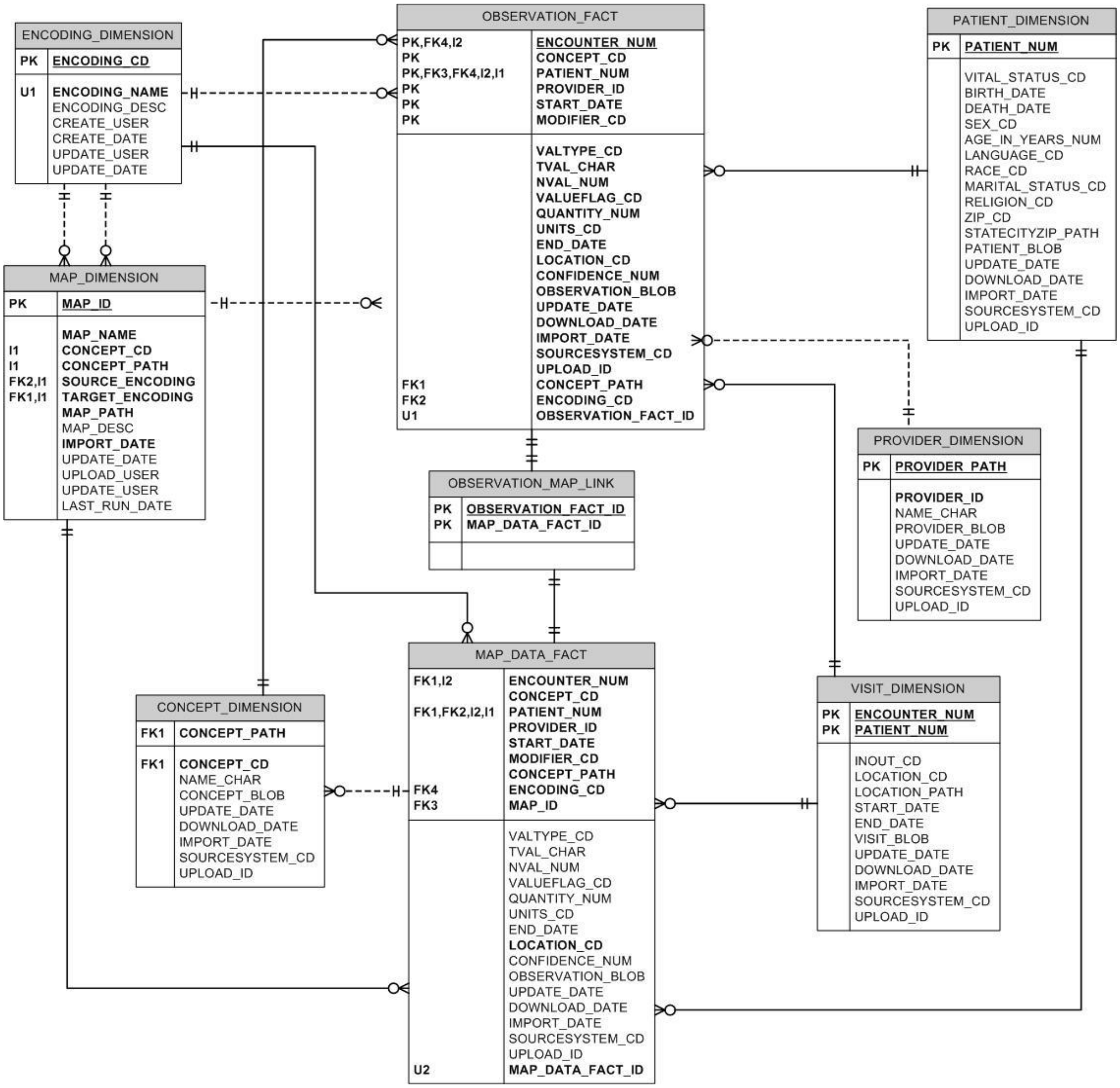


Figure 6 - Mapper Database Design

References

Adelman, Sid & Terpeluk-Moss, Larissa (2000). *Data Warehouse Project Management*. Boston: Addison Wesley.

Berry, M. J. A. & Linoff, G. (2000). *Mastering Data Mining: The Art and Science of Customer Relationship Management*. Hoboken: John Wiley & Sons.

Berson, A. & Smith, S. & Thearling, K. (2000). *Building Data Mining Applications for CRM.* New York: Computing McGraw-Hill.

Bray, T. (2001). *What is RDF*
Referenced July 26, 2003 from
http://www.xml.com/pub/a/2001/01/24/rdf.html?page=2

Delmater, R. & Hancock, M. (2001). *Data Mining Explained: A Manager's Guide to Customer-Centric Business Intelligence.* St. Louis: Digital Press.

English, Larry P. (1999). *Improving Data Warehouse and Business Information Quality.* New York: Johm Wiley & Sons.

Han, Jiawei & Kamber, Micheline (2001). *Data Mining: Concepts and Techniques.* USA: San Francisco: Morgan Kaufmann.

Hobbs, Lilian; Hillson, Susan & Lawande, Shilpa (2003). *Oracle9iR2 Data Warehousing.* Burlington: Digital Press.

Hoffer, Jeffrey A.; Prescott, Mary B. & McFadden, Fred R. (2002). *Modern Database Management.* Upper Saddle River: Prentice Hall.

Inmon, W. H. (2001)(a). *A Brief History of Metadata*
Retrieved July 20, 2003 from
http://www.inmoncif.com//library/whiteprs/wp.asp#Metadata

Inmon, W. H. (2001)(b). *An Illustrated Taxonomy of Metadata*
Retrieved July 20, 2003 from
http://www.inmoncif.com//library/whiteprs/wp.asp#Metadata

Inmon, W. H. (2002). *Building the Data Warehouse.*
New York: John Wiley & Sons, Inc.

Inmon, W. H., Imhoff, C., Sousa, R. (2001). *Introducing the Corporate Information Factory 2nd Edition* New York: John Wiley & Sons, Inc.

Inmon, W. H. (2000). *Metadata in the Data Warehouse*

Retrieved July 20, 2003 from http://www.inmoncif.com//library/whiteprs/wp.asp#Metadata

Inmon, W. H. (1997). *Metadata in the Data Warehouse: A Statement of Vision* Retrieved July 20, 2003 from http://www.inmoncif.com//library/whiteprs/wp.asp#Metadata

Jarke, Matthias; Maurizio, Lenzerini; Vassiliou, Yannis & Vassiliadis, Panos (2003). *Fundamentals of Data Warehouses.* (2nd. Edition). Berlin: Springer.

Johnson, Diane (1999). Implementing a Hybrid Online Analytical Processing (HOLAP) solution. In Purba, Sanjiv (Ed.). *Data Management Handbook.* (3rd. edition, pp. 747-753). Boca Raton: Auerbach.

Lieberman, M., Ricciardi, T., Masarie, F., Spackman, K. (2003), *The Use of SNOMED-CT Simplifies Querying of a Clinical Data Warehouse* **.** AMIA Annu Symp Proc. 2003; 2003. 910. Retrieved November 16, 2006 from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1480260

Meyer, Don, Cannon, Casey (1998). *Building a Better Data Warehouse.* Upper Saddle River: Prentice Hall.

Murphy, S., Gainer, V., Chueh, H.C. (2003), *A Visual Interface Designed for Novice Users to find Research Patient Cohorts in a Large Biomedical Database***.** AMIA Annu Symp Proc. 2003;
2003. 489–493. Retrieved November 15, 2006 from http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pmcentrez&artid=1480150

Scalzo, Bert (2003). *Oracle DBA Guide to Data Warehousing and Star Schemas.* Upper Saddle River: Prentice Hall PTR.