

The Development of Health Care Data Warehouses to Support Data Mining

Jason A. Lyman, MD, MS^{a,b,*}, Kenneth Scully, MS^{a,b},
James H. Harrison, Jr, MD, PhD^{a,c}

^a*Division of Clinical Informatics, Department of Public Health Sciences,
University of Virginia, Suite 3181 West Complex, 1335 Hospital Drive,
Charlottesville, VA 22908, USA*

^b*Clinical Data Repository, University of Virginia School of Medicine,
Suite 3181 West Complex, 1335 Hospital Drive,
Charlottesville, VA 22908, USA*

^c*Department of Pathology, University of Virginia, Suite 3181 West Complex,
1335 Hospital Drive, Charlottesville, VA 22908, USA*

Data mining requires an underlying data set and these data may be acquired, stored, and managed in multiple ways. As this data set increases in volume, data mining techniques generally become more effective and useful. Medicine is becoming well-positioned to take advantage of the capabilities of data mining; there is a tremendous wealth of largely untapped clinical data available in the operational clinical information systems of laboratories, hospitals, and clinics around the world, and the volume of these data is increasing rapidly as medical centers adopt electronic medical records. An important initial step toward the most effective mining of these data for biomedical and translational research is the development of enterprise clinical data warehouses that offload information from production systems into separate, fully integrated databases optimized for performing population-based queries. Without such systems, mining of production clinical data requires complex and time-consuming preparatory data aggregation and processing steps on a project-by-project basis. Because most health care data have potential value for multiple research endeavors, the benefits of developing and maintaining large-scale multipurpose enterprise data warehouses can be considerable. In this article, we present an introduction to clinical data warehouses, highlight examples from the literature in

* Corresponding author. Division of Clinical Informatics, Department of Public Health Sciences, Suite 3181 West Complex, 1335 Lee Street, University of Virginia Health System, Charlottesville, VA 22908.

E-mail address: lyman@virginia.edu (J.A. Lyman).

which they support data mining, and describe specific issues and challenges related to their development and use.

Introduction to data warehouses

Health care data warehouses, which integrate data from multiple operational systems and provide longitudinal, population-based views of health information, are becoming increasingly common [1–9]. One challenge in discussing these systems is the lack of consensus about the appropriate terminology. The terms data repository, data warehouse, knowledge warehouse, and information warehouse have all been used in the academic and business communities to describe databases that are distinct from production systems (eg, electronic health records) and exist to support analytic processing and strategic decision making (in the business world) and biomedical research (in the health care sector). There is ambiguity in the terminology, however, because the term “clinical data repository” is often used to describe a production clinical system designed to display integrated patient data to health care providers for the purposes of patient care [10].

A data warehouse, in general terms, can be defined as “a copy of transaction data specifically structured for query and analysis” [11]. Transaction data, as it pertains to health care organizations, may include clinical laboratory results derived directly from a laboratory information system, medication orders, textual documents (eg, discharge summaries, radiology reports, surgical pathology reports), and administrative claims data, to name a few. Although small data warehouses may serve primarily to structure production data from individual systems for analytic purposes, the most useful warehouses incorporate data from multiple traditional production systems. Integrating these disparate data into a common system that provides longitudinal views of retrospective clinical data is the essence of a clinical data warehouse. A related concept, the clinical data mart, is a more specialized version of a clinical data warehouse with a subset of data pertinent to a particular setting or topic [12,13].

Creating a copy of transaction data in a separate integrated database usually yields significant performance benefits for research queries. Production clinical systems are optimized for high speed, continuous updating of individual patient data elements, and individual patient queries in small transactions. These types of systems are generally referred to as “online transaction processing” (OLTP) systems and their optimization for processing large numbers of small transactions at high speed introduces important constraints on their internal data models [14]. In contrast to production system transactions, researchers’ queries may be complex, involving long time periods or multiple conditions and patients, and may return substantial data sets. Clinical production systems are not designed for these types of queries and often respond sluggishly, yielding poor performance with research queries and potentially compromising the speed of concurrent clinical care processes as

an unwanted side effect. Systems whose data models are structured to support data analysis are generally termed “online analytical processing” (OLAP) systems to distinguish them from OLTP systems [14]. Systems designed for OLAP, such as data warehouses, are also typically nonvolatile: data are intermittently added and never changed. Because updating performance is not critical, data additions are batch-scheduled during slow use periods and include many data types on multiple patients [15]. Consequently, the underlying data model can be optimized to maximize the performance of the types of complex queries across populations that are characteristic of research.

Data warehouses as sources for data mining initiatives

In recent years, there have been several published examples of the use of data warehouses for data mining efforts related to clinical investigation. Researchers have used these databases for factor analysis to identify patients at risk for suboptimal outcomes, to explore associations between diseases and clinical findings, for improvement of clinical information systems, and to examine disease–disease and disease–procedure associations.

Prather and colleagues [16] describe the development of a clinical data warehouse at Duke University Medical Center and its subsequent use for a data mining project intended to identify factors associated with increased risk for preterm birth. Data on more than 45,000 patients were transferred from the perinatal database of their organization’s electronic medical record system to a separate relational database designed for research. In a preliminary analysis restricted to a 2-year time period, multiple queries against the data warehouse were used to create a data set of 3902 patients. During this process, variables were cleansed to correct erroneous or discrepant values and formatting inconsistencies. Once the final data set was created, factor analysis was performed. For patients who met study criteria, more than 91% of the data values were usable, and three latent factors were identified that explained 48.7% of the variance in the data. In a subsequent larger analysis of 19,970 patients with 1622 variables per patient, 7 predictor variables for preterm birth risk were identified, yielding 0.72 area under the curve for receiver operating characteristics [17].

In another example, researchers at Columbia University used co-occurrence statistics to discover associations between the presence of diseases and clinical findings [18]. Such associations, they believed, could improve the usefulness of an automated problem list summarization tool used in their electronic medical record system by allowing the removal of redundant, clinically non-informative findings (eg, chest pain in a patient who had myocardial infarction). They compared two methods, Chi-square and the proportion confidence interval (PCI), assessing their respective abilities to detect clinically recognized disease-finding associations in discharge summaries. Using the former technique, 94% of the associations that were identified were believed to be true associations as judged by expert physicians,

whereas 77% of the associations found using the PCI approach were believed to be correct. Although the purpose of their knowledge discovery effort was to improve the usefulness and accuracy of an automated problem list generator, they acknowledged the potential usefulness of their approach for identifying novel associations between disease–disease, disease–medication, disease–procedure, and so forth.

The identification of such associations was the focus of another data mining effort using the University of Virginia’s Clinical Data Repository, an enterprise-wide data warehouse developed to support clinical investigation. Mullins and colleagues [19] described a collaborative endeavor between researchers at the University of Virginia, the Virginia Commonwealth University, and IBM Life Sciences, in which a data set with 667,000 patients was mined using three different unsupervised methods to identify potentially interesting disease associations. Results were compared with automated searches of the biomedical literature in an effort to distinguish between associations that were well established versus those that might represent previously unknown relationships. The analysis identified multiple associations of both types, including “congestive heart failure–valvular disease–hypertension” (a well-known association) and “albuterol–tracheostomy–magnesium” (an association not found in the biomedical literature).

Developing a health care data warehouse

The design and implementation of an integrated data warehouse is a resource-intensive process requiring a multidisciplinary approach and substantial investments of time and energy. Third-party systems are available, but fully functional warehouses can be developed using freely available open-source tools. A common limitation of systems currently offered by the vendor community is that they are often focused on the business aspects of health care (eg, finances, use) rather than needs of biomedical investigators, which may require different types of queries with different optimal underlying data models and analysis techniques.

The development of a functional data warehouse involves tasks in three areas: (1) design and implementation of the data warehouse structure, (2) population of the warehouse, including data acquisition, processing, loading, and linking, and (3) data access and query management with appropriate security and user interface services. Data acquisition typically includes the initial collection of historic data and the development of mechanisms for prospective, real-time, or batch feeds for new data. Each of these task areas requires a significant effort with careful planning.

Database design and implementation

Database design typically begins with requirements—gathering activities in which the desires and needs of potential users are elicited and

documented. In the context of building a data warehouse for clinical investigation and data mining, this can be challenging because of the diverse, ever-changing nature of biomedical research. It is difficult to anticipate at the outset the breadth and depth of information that will ultimately be required. This is in contrast to business-oriented applications, in which there are typically a small number of focused questions that a data warehouse is intended to support.

Murphy and colleagues [7] approached the requirements process by retrospectively examining queries of an existing clinical information system to identify particular data types that were of most value. In their study, coded diagnoses and medications accounted for more than 90% of queries, and this information was used to optimize their data model for these types of queries. This approach has benefits but is also limited because historical queries can only use information that was readily available at the time, and by definition cannot represent data or types of queries that may become available in a new system.

By practical necessity, the development of a clinical data warehouse is often largely driven by the data that are electronically available at the time of its creation. Universally collected data, at least in the United States, include administrative information necessary to support billing and governmental reporting requirements. These include coded diagnoses, procedures, demographics, discharge disposition, and payer information. Although there is debate about whether the accuracy of such administrative data is sufficient for clinical research or quality assessment [20,21], this type of information has been used successfully for scores of clinical and health services research projects. Because clinical laboratory results are typically stored in electronic format at most institutions and are important indicators of diagnosis, disease progression, and response to therapy, this information along with administrative data often form the core around which a clinical data warehouse is constructed. Other data, including medications (ordered or administered), vital signs, monitoring data (eg, EKG), or textual reports, are often useful for research studies but are less commonly available electronically. Patient identifiers, although perhaps not necessary for many research projects, are still important for linking data from multiple systems over time, and their inclusion has significant ramifications for the security requirements of the data warehouse.

Database design includes not only requirements gathering and the identification of desired data elements but also the adoption or development of the data model that will be used. The data model serves as the blueprint for database construction and ultimately determines how data will be organized by the database management system. Data models for data warehouses tend to be fundamentally different from those developed for transactional (OLTP) databases that support, for example, electronic medical records and other production systems. Analytic (OLAP) systems typically use a multidimensional data model that is implemented in one of two

ways: specialized multidimensional databases (multidimensional OLAP, or MOLAP) or relational databases with a multidimensional schema (relational OLAP, or ROLAP) [14,22]. A multidimensional data model is one in which the item under analysis is often some type of event (a time-stamped retail purchase or a patient care event, for example) that can be characterized by several features. Each feature represents a dimension. Common dimensions in the clinical setting might include patient, physician, location, month of year, day of week, type of event, associated diagnosis, result or observation, treatment type, dose, and many more. For a set of events, dimensions can be selected and plotted against each other or otherwise compared to determine their relationships. These types of databases are often called “data cubes” because they share a set of common dimensions, although the number of dimensions is usually more than the three that define a physical cube.

MOLAP systems are implemented as a set of specialized arrays containing the dimensional data, with precalculated aggregates and summaries derived from the data that are generated at load time. They are typically completely loaded once from an external data store and are designed to support specific analyses that are particularly useful for business purposes. To update the data, old cubes are discarded and new cubes are reloaded from scratch. MOLAP systems generally yield fast query response because of their highly optimized (nonrelational) internal design and the substantial amount of precalculation that is done on loading. This design allows efficient data summarization (“roll-up”) or inspection of detailed data contributing to a summary (“drill-down”). Their analytic capabilities may not be flexible enough for evolving biomedical research needs, however. MOLAP systems also tend to perform poorly in highly dimensional settings (ie, those with more than 10 dimensions) because of the processing, memory, and storage requirements for precomputing and storing summaries and relationships between many dimensions [23]. Medical environments often yield 50 or more dimensions per event.

ROLAP systems are standard relational databases that implement multidimensional data models [14,22]. The relational database schemata that support these data models generally have a central “fact table” that contains an entry for each item (eg, a clinical event) along with a limited number of data, such as a time stamp. The remaining characteristics of the item—the dimensions—are contained in a set of dimension tables that are linked to the fact table. The linkage of dimension tables to each other through the fact table allows flexible and efficient correlation analyses between the dimension tables. For example, if values are specified for one or more dimensions, it is easy to look up the associated values in an additional dimension. This table structure consisting of multiple data tables surrounding a central table that contains primarily linkage data is referred to as a “star” or “snowflake” schema depending on implementation details and is characteristic of a relational data warehouse [7–9,11,14,24].

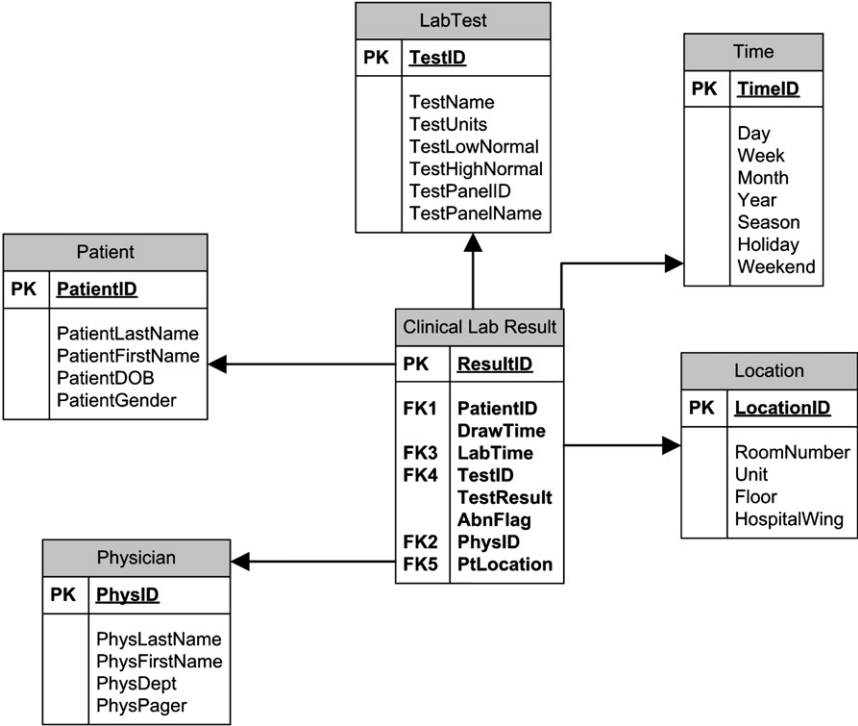


Fig. 1. Star schema for clinical laboratory data. Star schemata are characterized by a central fact table (Clinical Lab Result) containing a minimal representation of an event, such as a laboratory test, with links to multiple surrounding dimension tables holding the detailed data on the characteristics of the events. FK, foreign key, which points to a primary key in another table; PK, primary key.

A fact table focused on laboratory test results might contain the following fields: patient identifier, date/time identifier, laboratory test identifier, specimen type, and numeric result (see Fig. 1). If a patient had multiple serum glucose tests, each result would be stored in a row in the table. In this example, a single “fact” refers to a specific result for a specific laboratory test that was obtained for a specific patient at a specific point in time. The identifiers function as links to the dimension tables: a patient dimension, time dimension, and laboratory test dimension, where more detailed information about each of those elements would be stored. Star schemata make it easy to add new dimensional characteristics. For example, if one of the dimensions is time, then the dimension table would allow you to group the time of events recorded in the fact table in different ways: by year, quarter, month, day of week, season, weekend indicator, holiday indicator, and so forth. A patient age dimension might be constructed to easily allow data to be queried or analyzed by adult versus pediatric, age decade, or some other locally defined age group.

Depending on the intended application, a clinical data warehouse fact table may contain multiple different types of events or there may be several fact tables, each with a narrower focus. For example, the RPDR data warehouse serving Partners Healthcare in Boston, Massachusetts, contains at least four separate fact tables [7]. Some investigators have described the importance of using different fact tables for different data types [25]. Clinical laboratory results, for example, may be numeric (further broken into integer or decimal types) or textual, and these data types should be indexed separately for best query performance.

ROLAP data warehouses may also have other structural differences from relational OLTP systems. Because data warehouse queries may be executed across millions of lines of data for thousands of patients at a time, storing redundant or precomputed data that are used commonly in queries can substantially improve query performance. Because this redundancy is designed in and programmatically evaluated at load time and during maintenance, it does not create a data integrity problem. For example, given a patient's date of birth and a date/time-stamped event, it is theoretically redundant to store both a patient's date of birth and age with the time-stamped event because the age at the event can be calculated from the date of birth and time stamp. Precalculating age, however, can dramatically improve the performance of queries that specify an age condition or carry out mathematical calculations based on age. In aggregate, the characteristics of ROLAP systems described above yield data warehouses that can provide reasonable performance with larger data sets and more dimensions than MOLAP systems can effectively manage, while also allowing greater database design and analytic flexibility. Most large-scale health care data warehouses thus are based on a ROLAP design.

Implementation of the data warehouse requires selection of the underlying relational database management system (RDBMS) from among a wide variety of commercial and open-source systems. Clinical data warehouses have been successfully implemented using Microsoft SQL Server [16, 26, 27], Oracle [8, 28, 29], Sybase [4], and the open-source systems Postgres (www.postgresql.com) [30] and MySQL (www.mysql.org) [31], to name just a few examples. Although practically any fully functional RDBMS can be used for a data warehouse, the ultimate decision depends on multiple factors, including consideration of other institutional information systems, available local expertise, database size, performance requirements, interface requirements, and functionality.

Data acquisition and processing

Clinical data typically are entered and stored in multiple operational systems distributed throughout a health care organization. It is not unusual for data mining to require data that are stored in several of these operational systems. Because each data source typically has its own format and

idiosyncrasies, mining the data is difficult unless it is processed into an appropriate form and collected into a single database. One of the most important functions of data warehouses is to aggregate these data in a form that is appropriate for mining. Several steps are required to adequately organize and process the data before loading them into the data warehouse: extraction, filtering and transformation (sometimes referred to as “data cleansing”), classification, and aggregation. The ultimate goal is to standardize the information from disparate systems and transform it so that it is useful for mining.

Extraction

Data may be extracted from production systems as files or may be received in real time through existing hospital interfaces (Fig. 2). Real-time data are often communicated through an interface engine using the HL7 messaging standard [32], and there are general purpose HL7 parsers (commercial and open source) that can extract data from HL7 feeds. Nonstandard real-time formats require custom programming. Alternatively, some data sources (particularly those focused on administrative requirements, such as billing) may deliver data as batch files 1 to 2 months after hospital discharge. These batch files can be parsed and processed using standard text parsing or database import libraries.

Filtering

Not all of the data provided by source systems are desirable or accurate. Some of the information may have no potential clinical research benefit and may be removed (eg, inventory data or clerical information about how reports were transcribed). Data related to patient identity (eg, social security number, insurance policy numbers), which are usually required for accurate linkage of records from multiple systems, need special handling. Data errors are a perpetual challenge and the optimal approach to them is to identify errors at the time of data acquisition and filter them out. For example, invalid data or data with related missing critical values may need to be eliminated. Decisions must be made on which data will be filtered versus which will be transformed into missing or unknown values. For example, if a lab result is received with a date that is in the future, should the record be eliminated, should it be loaded with the bad date, or should the date be changed to a best guess or unknown value? Each approach has its advantages and disadvantages, and the answer should be determined by the goals and requirements of the data warehouse.

Transformation

It may be useful to force some data to fit predetermined standard values. For example, a patient’s gender may be represented in one data source as “M” or “F” but in another as “1” or “2.” Some systems provide patient names in a non-atomic format that combines first, middle, and last name all into one field. Transformation should yield consistent data

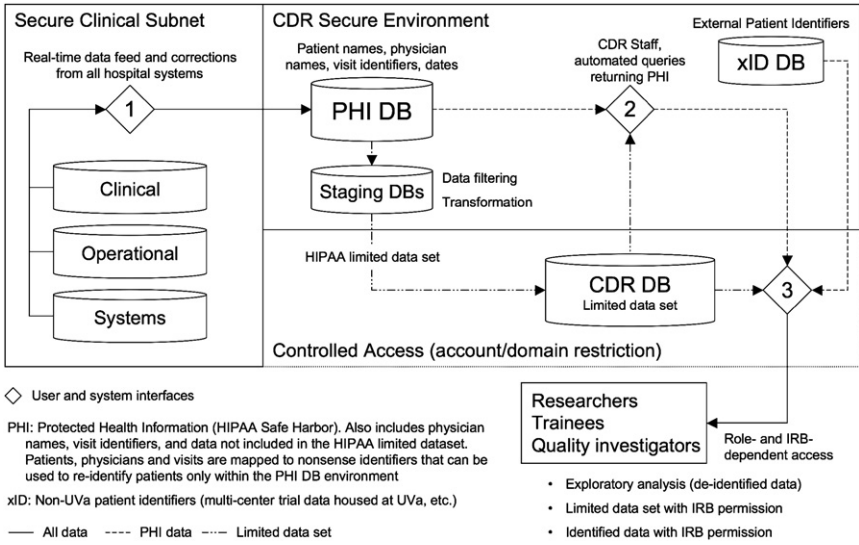


Fig. 2. Data flow associated with the University of Virginia Clinical Data Repository (CDR). Data are derived from clinical production systems (*left*) as HL7 messages or batch files. Messages pass through an HL7 interface engine (1), to the CDR PHI database in a secure environment (*top*). PHI (HIPAA Safe Harbor data elements) are stripped and stored there, and the remaining limited data set is passed to a set of staging databases (*middle*) where data are held between batch updates of the main CDR database (CDR DB). During this time, data corrections are passed from the production system to the PHI and staging databases, and the data are filtered and transformed as necessary. An interface (2) is available in the secure CDR environment that allows CDR staff access to the PHI and de-identified databases, and also executes prebuilt queries for quality assurance studies or researchers who have permission to access patient identity data (see below). CDR users (*bottom right*) working with de-identified data or limited data sets with IRB permission query the database directly through an SSL-secured Web interface (3) located with the de-identified databases in a controlled-access environment (*middle*) that is restricted by user IP address and account name/password. When users who have permission to view PHI extract data, a call is passed from interface 3 to interface 2 to run an appropriate prebuilt query, and combined data from the de-identified and PHI databases are returned. There is no direct external user access into the secure CDR environment. The system also provides a separate database in the secure environment for PHI for external patients (xID DB, *upper right*) with capabilities similar to the primary PHI database, to accommodate, for example, outside data from multicenter trials. The CDR is implemented as MySQL databases running in Linux, and Web access is provided by way of JDBC from custom Java servlets running in the Apache Tomcat environment.

representation across the warehouse and individual data elements should be separately accessible for query. Certain invalid data may also need to be transformed: a gender received as “G” or empty may be changed and stored in the database as “U” (for Unknown). This step should also transform data containing confidential information into acceptable forms. For example, a patient’s medical record number may be converted to a disguised number. Transformation of local data representations into standard coding schemes or ontologies may have substantial benefit for simplifying queries and

supporting data sharing, and should be considered strongly where possible [31,33]. With respect to pathology data, local lab codes may be converted to LOINC [34] or text strings may be identified in pathology reports and encoded using a standard vocabulary [35].

Classification

Data may be grouped and classified into new schemes designed to support particular data warehouse or mining requirements. For example, a person's age may be classified into discrete age groups (younger than 18 years, 18–40 years, and so forth). Many data mining software techniques require that data be submitted in a limited number of discrete categories rather than as a data element with an unlimited number of possible values, so pre-classifying may aid in readying a database for extraction and input into a data mining software tool. Classifying data can itself be a data mining operation and it may be desirable to feed results of this data mining back into the database to help with other data mining activities. Although classification can have advantages, it is important to maintain the raw, detailed data in the database to retain flexibility for multiple uses and future reclassification as needed.

Aggregation

It may be useful from a query formulation and performance standpoint to precalculate certain commonly used values (eg, totals, averages) and store these in the database. The use of precalculated patient age was discussed previously. Another candidate for precalculation might be the total number of days for a hospitalization, calculated from admission and discharge dates. This particular variable is often used as an outcome in health services research and precalculation improves the speed of the many queries that make use of it. It may also be beneficial to aggregate hospital stay data by patient diagnosis, location, or service, and precompute average, maximum, minimum, and variance values, if there is sufficient local interest in these data. As local needs change, the type of values that are precalculated and stored can also change as long as the raw data on which the calculations are based remain in the data warehouse.

Security and confidentiality

The Health Insurance Portability and Accountability Act (HIPAA) that went into effect in April 2001 has significant implications for the use of clinical data in general and their use in biomedical research in particular [36]. Although the conduct of clinical research frequently does not require identifiable data for analysis, patient identifiers are needed when (1) patients must be contacted for study recruitment, or (2) the research requires existing data from multiple, disparate systems to be linked together by patient. From the standpoint of developing a data warehouse, identifiable data are

essential for linking records derived from disparate systems and for maintaining longitudinal records over time. At the University of Virginia, our data warehouse, the Clinical Data Repository (CDR, see Fig. 2), addresses this task by separating direct identifiers, such as name, medical record number, and social security number, and storing this information in a distinct database on its own highly protected secure server, accessible only to members of the CDR project team [4]. Date of birth, gender, and race are also stored on this server because they are sometimes useful for resolving ambiguous matches when linking records. Our system assigns a disguised identifier for each new patient, which links clinical records for that patient within the database that is accessible to our research users. This disguised identifier is mapped to the identifiable data in the highly secure database and serves as the master link between the two systems. The names, dates of birth, medical record numbers, and so forth can thus be omitted from the accessible research server but are available when necessary for record linking and for research purposes with appropriate Institutional Review Board permission.

The HIPAA privacy rule specifies a list of 18 data elements that must be removed for a database to be considered de-identified [37]. A data warehouse that maintains a de-identified database for direct user access has security and convenience advantages that may be preferable to requiring various users to obtain permission to access a system that contains patient identifiers. Removal of HIPAA identifiers may not prevent identification for patients who have rare conditions or rare combinations of conditions, however. In these cases, queries may return very small groups or single patients [38]. For this reason, most clinical data warehouses that implement investigator access to de-identified data return only summary data if the data set resulting from a query contains fewer than a defined minimum number of patients.

Although many of the data elements that HIPAA includes as potentially identifiable are not needed at the user query stage, temporal information is often important. The HIPAA privacy rule prohibits specific health-related dates (other than year) in a de-identified data set, so disguising dates in a way that preserves their research usefulness is beneficial. The use of date offsets, the number of days (or hours, minutes, and so forth) between events, is one way to address researchers' needs for temporal constraints on query conditions. Typically the absolute date is not required for research, but it is necessary to determine whether events occurred within some specified time period (eg, readmission to the hospital within 30 days of discharge, or the use of a medication within 24 hours of hospital admission). The use of offsets requires the identification of a "time zero" for each patient; our data warehouse uses the date/time of the first event for a patient as a starting point and all offsets are calculated based on that reference. The raw dates can be stored in the highly secure database containing the identifiable data set. This approach, although having the benefit of improved confidentiality, does have important limitations. Strictly adhering to this method means that a table storing outpatient clinic visits could not include the

date of the visit, just the year and the number of days between that visit and the patient's first event in the database. A user query aimed at identifying specific seasonal trends in hospitalized patients, for example, would require links back to the identifiable data to find the correct dates of origin for the offset calculations.

The challenges of textual data

Some of the richest, most clinically detailed information in the medical record is stored as text. This information includes consultative reports (eg, pathology, radiology), operative notes, progress notes, discharge summaries, and also some clinical laboratory results, including microbiology. These data are often valued by researchers but present the data warehouse developer with multiple challenges, including ensuring confidentiality, classifying data correctly, and providing useful query methods.

Confidentiality and textual data

As opposed to coded clinical/administrative data (eg, diagnoses, procedures, medications, and so forth) or numeric clinical laboratory tests, data in textual fields are much more susceptible to the inclusion of identifiable data, because the data are entered in a free text format, and the identifiers may be clinically and operationally important for communication between members of the health care team. A surgical pathology report typically includes phrases that describe how specimens are labeled, and may explicitly identify patient name and medical record number. A textual field in a laboratory result might include the phrase, "Result called at 3/23/2006 9:12 AM to Dr. Smith" or a reference to a particular health care facility. Although access to these reports might be allowable for researchers who have IRB approval to review identifiable data, the ultimate goal is to provide as much de-identified information as possible so that researchers may work without identities unless they are truly necessary. The automated de-identification, or scrubbing, of textual reports is an active area of medical informatics research, and there are increasing numbers of available tools to accomplish this [35,39,40]. A detailed discussion is beyond the scope of this article but two approaches described in the literature include (1) automated extraction of accepted medical terms, which are stored in the de-identified database in lieu of the textual report [35], and (2) removal of identifiable data from the corpus of the text, leaving behind a report that is ostensibly clinically detailed but free from any information that might make the patient's identity known [39,40].

Querying textual data

A common query into a clinical data warehouse might ask for all newly diagnosed patients who have a certain form of cancer. Surgical pathology

reports are often a valuable source of information for this type of question, but locating these cases means successfully searching potentially hundreds of thousands of documents that are rife with abbreviations, homonyms, synonyms, and misspellings. The query challenges are, in essence, similar to those that users face when searching the biomedical literature or even the World Wide Web, and consequently similar approaches can be used to facilitate successful queries. One of the commonly studied methods for addressing this challenge is to use a computer-based auto-coding approach in which the text is parsed and codes are assigned from a standard terminology, such as the Unified Medical Language System Metathesaurus [35,41,42]. Currently, though, the large-scale use of standard terminologies for linkage to, or replacement of, textual reports remains a future goal.

Performance issues

Except for very small databases, performance should be a major focus when designing a relational clinical data warehouse. If the database will be large, performance will be a significant issue and good database design will be a particularly critical requirement for success.

One of the most significant features of a relational database system that can enhance performance is the ability to place indexes on any data element (or set of data elements). When searching the database or joining tables together, tremendous performance gains can be achieved by strategically adding indexes on those columns that will be used in database queries. Indexes are separate database files that are maintained internally and contain a copy of the indexed columns in sorted order along with pointers to corresponding rows in the original data table. Indexes allow the database system to use a fast binary search algorithm to find data and the original table rows that contain it quickly.

Indexing, however, cannot usually be placed on every data element. It is only useful if the ratio of unique data values to the number of rows is relatively high. Creating indexes on data elements that contain few distinct values (such as gender or race) typically offers little benefit, and can actually be detrimental to query performance. There are times, therefore, when a query must find rows by doing a sequential read through the entire table. In these cases, query speed is related to table size, so tables should be designed to be as lean as possible.

In the laboratory test result tables of our CDR, the lab test description and unit of measure were extracted to separate tables. Some space was saved because of reduction in redundancy, but there was a larger reward in performance. Descriptions tend to be large so they can significantly increase the size of a table, which in turn adds to sequential search times. By extracting the descriptions, we decreased the size of the table yielding faster sequential search times. The added time required to extract the description from

a separate table is comparatively minimal. Performance of complex queries, particularly those using sub-queries, may also be improved by judicious use of temporary database tables, “pre-fetched” data, and similar techniques, depending on the design and capabilities of the particular database management software.

User interfaces

A flexible, easy-to-use user interface for a clinical data warehouse increases the accessibility of data, promotes the use of the system by researchers, and helps moderate the demand on the warehouse staff for support of individual projects. Direct access to the data may also improve query results by allowing users to successively refine their queries as results are returned to yield the most valuable data set for a project. User interfaces for data warehouses may include several different components: query formulation (in which the user is insulated from manual SQL scripting), presentation of standard reports, creation of custom reports, downloading data sets, and statistical analysis. Although query formulation is a requisite for any data warehouse interface, the other components are optional. There is a wide range in the extent to which these other functions are supported in existing clinical data warehouses. For example, the RPDR at Partners Healthcare, Inc. offers an interface that has been developed using MS Access, allowing researchers to refine queries until the desired data set is identified [27]. At that point, researchers with the appropriate authorization can request these data, which are then compiled by the RPDR staff. Investigators at Indiana University/Regenstrief have developed a powerful query tool for users as part of the Shared Pathology Information Networks project that supports online statistical analysis and summarization of large data sets using open source tools [43]. Finally, our Clinical Data Repository at the University of Virginia includes a flexible Web-based query and reporting interface that permits users to create, review, and refine queries online, and then directly download these data sets if they have appropriate authorization [4].

Summary

Clinical data warehouses are becoming increasingly common at academic health centers in the United States, and their value will continue to grow as electronic medical records gain further adoption and the volume of clinical data in electronic form continues to increase. High-quality commercial and open-source software is available for the construction of the warehouse itself along with query and reporting interfaces and statistical analysis tools. Development of these systems is a significant undertaking that requires multidisciplinary expertise, but the benefits to clinical research are substantial. A robust clinical data warehouse offers a rich source of integrated, longitudinal patient data, and it has the potential to greatly facilitate all forms of

data mining. Even in cases in which data are exported from a data warehouse so that they can be restructured into an appropriate format for mining, such tasks are orders of magnitude less time-consuming and resource-intensive than de novo collection and processing of data from multiple clinical operational systems. Furthermore, data mining software developers often incorporate relational database connectivity into their applications, allowing mining directly against warehouses without the need for data export and transformation. These developments indicate that the nature of biomedical research is evolving: we are entering an era in which large amounts of clinical data in electronic form will be accessible to researchers using well-designed analysis tools to pursue biomedical and translational knowledge discovery.

References

- [1] Dewitt JG, Hampton PM. Development of a data warehouse at an academic health system: knowing a place for the first time. *Acad Med* 2005;80:1019–25.
- [2] Kamal J, Pasuparthi K, Rogers P, et al. Using an information warehouse to screen patients for clinical trials: a prototype. *Proc AMIA Symp* 2005;1004.
- [3] Bock BJ, Dolan CT, Miller GC, et al. The data warehouse as a foundation for population-based reference intervals. *Am J Clin Pathol* 2003;120:662–70.
- [4] Einbinder JS, Scully KW, Pates RD, et al. Case study: a data warehouse for an academic medical center. *J Healthc Inf Manag* 2001;15:165–75.
- [5] Tusch G, Muller M, Rohwer-Mensching K, et al. Data warehouse and data mining in a surgical clinic. *Stud Health Technol Inform* 2000;77:784–9.
- [6] Wisniewski MF, Kieszkowski P, Zagorski BM, et al. Development of a clinical data warehouse for hospital infection control. *J Am Med Inform Assoc* 2003;10:454–62.
- [7] Murphy SN, Morgan MM, Barnett GO, et al. Optimizing healthcare research data warehouse design through past COSTAR query analysis. *Proc AMIA Symp* 1999;892–6.
- [8] Verma R, Harper J. Life cycle of a data warehousing project in healthcare. *J Healthc Inf Manag* 2001;15:107–17.
- [9] Berndt DJ, Hevner AR, Studnicki J. The catch data warehouse: support for community health care decision-making. *Decision Support Systems* 2003;35(3):367–84.
- [10] Sittig DF, Pappas J, Rubalcaba P. Building and using a clinical data repository. Available at: <http://www.informatics-review.com/thoughts/cdr.html>. Accessed April 23, 2007.
- [11] Kimball R. The data warehouse toolkit. New York, NY: John Wiley & Sons, Inc.; 1996.
- [12] McNamee LA, Launsby BD, Frisse ME, et al. Scaling an expert system data mart: more facilities in real-time. *Proc AMIA Symp* 1998;498–502.
- [13] Brandt CA, Morse R, Matthews K, et al. Metadata-driven creation of data marts from an EAV-modeled clinical research database. *Int J Med Inform* 2002;65:225–41.
- [14] Rob P, Coronel C. Database systems: design, implementation, and management. 7th edition. Boston: Thomson/Course Technology; 2007.
- [15] Inmon WH. Building the data warehouse. 4th edition. Indianapolis (IN): Wiley; 2005.
- [16] Prather JC, Lobach DF, Goodwin LK, et al. Medical data mining: knowledge discovery in a clinical data warehouse. *Proc AMIA Symp* 1997;101–5.
- [17] Goodwin LK, Iannacchione MA. Data mining methods for improving birth outcomes prediction. *Outcomes Manag* 2002;6:80–5.
- [18] Cao H, Markatou M, Melton GB, et al. Mining a clinical data warehouse to discover disease-finding associations using co-occurrence statistics. *Proc AMIA Symp* 2005;106–10.

- [19] Mullins IM, Siadaty MS, Lyman J, et al. Data mining and clinical data repositories: insights from a 667,000 patient data set. *Comput Biol Med* 2006;36:1351–77.
- [20] Humphries KH, Rankin JM, Carere RG, et al. Co-morbidity data in outcomes research: are clinical data derived from administrative databases a reliable alternative to chart review? *J Clin Epidemiol* 2000;53:343–9.
- [21] Iezzoni LI. Assessing quality using administrative data. *Ann Intern Med* 1997;127:666–74.
- [22] Gorla N. Features to consider in a data warehousing system. *Commun ACM* 2003;46(11):111–5.
- [23] Weber R, Schek H, Blott S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *VLDB'98, Proceedings of the twenty fourth International Conference on Very Large Data Bases*; 1998. p. 194–205.
- [24] Levene M, Loizou G. Why is the snowflake schema a good data warehouse design? *Information Systems* 2003;28(3):225–40.
- [25] Nadkarni PM, Brandt C. Data extraction and ad hoc query of an entity-attribute-value database. *J Am Med Inform Assoc* 1998;5:511–27.
- [26] Breen C, Rodrigues LM. Implementing a data warehouse at Inglis Innovative Services. *J Healthc Inf Manag* 2001;15:87–97.
- [27] Murphy SN, Gainer V, Chueh HC. A visual interface designed for novice users to find research patient cohorts in a large biomedical database. *Proc AMIA Symp* 2003;489–93.
- [28] Ledbetter CS, Morgan MW. Toward best practice: leveraging the electronic patient record as a clinical data warehouse. *J Healthc Inf Manag* 2001;15:119–31.
- [29] Nigrin DJ, Kohane IS. Scaling a data retrieval and mining application to the enterprise-wide level. *Proc AMIA Symp* 1999;901–5.
- [30] Corwin J, Silberschatz A, Miller PL, et al. Dynamic tables: an architecture for managing evolving, heterogeneous biomedical data in relational database management systems. *J Am Med Inform Assoc* 2007;14:86–93.
- [31] Lyman JA, Scully K, Tropello S, et al. Mapping from a clinical data warehouse to the HL7 reference information model. *Proc AMIA Symp* 2003;920.
- [32] HL7. Health level 7. Available at: <http://www.hl7.org>. Accessed June 15, 2007.
- [33] Nardon FB, Moura LA. Knowledge sharing and information integration in healthcare using ontologies and deductive databases. *Medinfo* 2004;11(Pt 1):62–6.
- [34] Khan AN, Griffith SP, Moore C, et al. Standardizing laboratory data by mapping to LOINC. *J Am Med Inform Assoc* 2006;13(3):353–5.
- [35] Berman JJ. Concept-match medical data scrubbing. How pathology text can be used in research. *Arch Pathol Lab Med* 2003;127:680–6.
- [36] National Institutes of Health. Clinical research and the HIPAA privacy rule. Available at: http://privacyruleandresearch.nih.gov/clin_research.asp. Accessed June 18, 2007.
- [37] Schell SR. Creation of clinical research databases in the 21st century: a practical algorithm for HIPAA compliance. *Surg Infect (Larchmt)* 2006;7(1):37–44.
- [38] El Emam K, Jabbouri S, Sams S, et al. Evaluating common de-identification heuristics for personal health information. *J Med Internet Res* 2006;8(4):E28.
- [39] Gupta D, Saul M, Gilbertson J. Evaluation of a deidentification (de-id) software engine to share pathology reports and clinical documents for research. *Am J Clin Pathol* 2004;121:176–86.
- [40] Beckwith BA, Mahaadevan R, Balis UJ, et al. Development and evaluation of an open source software tool for deidentification of pathology reports. *BMC Med Inform Decis Mak* 2006;6:12–21.
- [41] Nadkarni P, Chen R, Brandt C. UMLS concept indexing for production databases: a feasibility study. *J Am Med Inform Assoc* 2001;8:80–91.
- [42] Hazlehurst B, Frost HR, Sittig DF, et al. MediClass: a system for detecting and classifying encounter-based clinical events in any electronic medical record. *J Am Med Inform Assoc* 2005;12:517–29.
- [43] McDonald CJ, Dexter P, Schadow G, et al. SPIN query tools for de-identified research on a humongous database. *Proc AMIA Symp* 2005;515–9.