DATA QUALITY RULES IN THE ANALYTIC HEALTH REPOSITORY

Susan Elizabeth Pollock

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Biomedical Informatics

The University of Utah

August 2012

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Susan Elizabeth Pollock**

has been approved by the following supervisory committee members:

Peter J. Haug	, Chair	**4/6/2012**
		Date Approved
John R. Holmen	, Member	**4/6/2012**
		Date Approved
Stanley M. Huff	, Member	**4/6/2012**
		Date Approved

and by **Joyce A. Mitchell** , Chair of the Department of **Biomedical Informatics**

and by Charles A. Wight, Dean of The Graduate School.

ABSTRACT

Data quality has become a significant issue in healthcare as large preexisting databases are integrated to provide greater depth for research and process improvement. Large scale data integration exposes and compounds data quality issues latent in source systems. Although the problems related to data quality in transactional databases have been identified and well-addressed, the application of data quality constraints to large scale data repositories has not and requires novel applications of traditional concepts and methodologies.

Despite an abundance of data quality theory, tools and software, there is no consensual technique available to guide developers in the identification of data integrity issues and the application of data quality rules in warehouse-type applications. Data quality measures are frequently developed on an ad hoc basis or methods designed to assure data quality in transactional systems are loosely applied to analytic data stores. These measures are inadequate to address the complex data quality issues in large, integrated data repositories particularly in the healthcare domain with its heterogeneous source systems.

This study derives a taxonomy of data quality rules from relational database theory. It describes the development and implementation of data quality rules in the Analytic Health Repository at Intermountain Healthcare and situates the data quality rules in the taxonomy. Further, it identifies areas in which more rigorous data quality

should be explored. This comparison demonstrates the superiority of a structured approach to data quality rule identification.

To my mother, Valerie Gray Cashin.

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

This thesis describes the data quality rules used in Intermountain Healthcare's Analytic Health Repository (AHR) and groups these rules within a data quality taxonomy derived from current research and literature.

Large integrated analytic databases present enormous opportunities for medical researchers. Data extracted from a wide variety of clinical systems can be standardized, cleansed and integrated to facilitate population-based research that can span decades. Electronic clinical data on millions of individuals can include textual data from discharge summaries, history and physical examination reports, coded billing and insurance records, lab data, genetic information, prescriptions, medication administration records and data from various specialty datamarts and disease registries. Research can include clinical analyses, treatment protocols, epidemiologic studies and clinical process improvement. Due to the large volume of data and the wide range of data inconsistencies, data quality is always an issue.

At Intermountain Healthcare, hospital information systems have been in place since the inception of the Health Evaluation through Logical Processing (HELP) system at LDS Hospital in 1967.(1) The 22 hospital system has been included on the Most Wired Hospital list by *Hospitals & Health Networks*, the journal of the American Hospital Association, for 12 out of the 13 years the survey has been in place. Intermountain Healthcare has an abundance of data. For over a decade, data from Intermountain's

clinical and operational systems has been extracted, transformed and loaded (ETL) into the Enterprise Data Warehouse (EDW). The predominant data quality standard for the EDW was, and still is, that EDW data will reflect the quality of the source system. However, there is significant variation in adherence to that philosophy. For large data sets such as the HELP system, the quality of the data in the EDW closely reflects that of the source. In other cases, such as the data that is submitted to the Society of Thoracic Surgeons Adult Cardiac National Database,(2) extensive quality validations are run against the data, errors are reported to the source and the data is corrected, re-extracted and re-examined until it meets what is considered 100% accuracy. And in other cases, downstream datasets such as the Heart Failure Patient Registry are loaded with scrubbed data from the primary EDW tables.

Development of AHR was initiated by Jason Jones in 2008 to maximize the value of Intermountain's vast stockpiles of data to support clinical process improvement, clinical analysis, and clinically-oriented research. By consolidating and validating data, the AHR insures consistency in the information provided to consumers, thereby increasing confidence in the results obtained. The AHR benefits Intermountain by making its massive data stores user-friendly for clinical analysis. It takes the most commonly used data elements (labs, vital signs, medication orders, clinical assessments, diagnoses and procedures) from a variety of sources, cleans the data and stores it in a manner that is optimized for population-based research. The AHR facilitates the development of clinical definitions. It can retain the definitions and rapidly return the defined populations and their attributes.

The AHR has already significantly reduced rework in data analysis and achieved consistency in its results. As a result of the comprehensive integration of data, fewer mistakes are made extracting information from formerly overlapping and disparate system tables. Speeding up the analysis cycle makes it easier to learn and discover iteratively, thereby allowing the researchers at Intermountain to execute the mission faster.

The AHR currently resides as a data mart within the framework of the EDW. The AHR functions as a clearing house for analytic information, presenting a clinical view of a wide variety of diseases, processes, indicators and outcomes for an entire population. Discrepancies and duplications are resolved across multiple data sources to provide "one-stop shopping" for consistent and validated data. The AHR provides research-based analytics which can be fed back into point-of-care decision support tools. It will be used to close the loop in the use of data for improving health care.

Conceptually, the AHR is composed of three layers (see Figure 1):

- Layer 1 (L1) – the preexisting source systems within the EDW (Clinical Data Repository (CDR), HELP, IDX, CaseMix, Sunquest Lab)
- Layer 2 (L2) – the building blocks of the AHR (diagnosis, procedure, lab result, problem, medication order, medication claim, vital sign, clinical assessment, encounter, patient)
- Layer 3 (L3) – the clinically-oriented, comprehensive analytic layer including the Charlson Comorbidity Index(3)

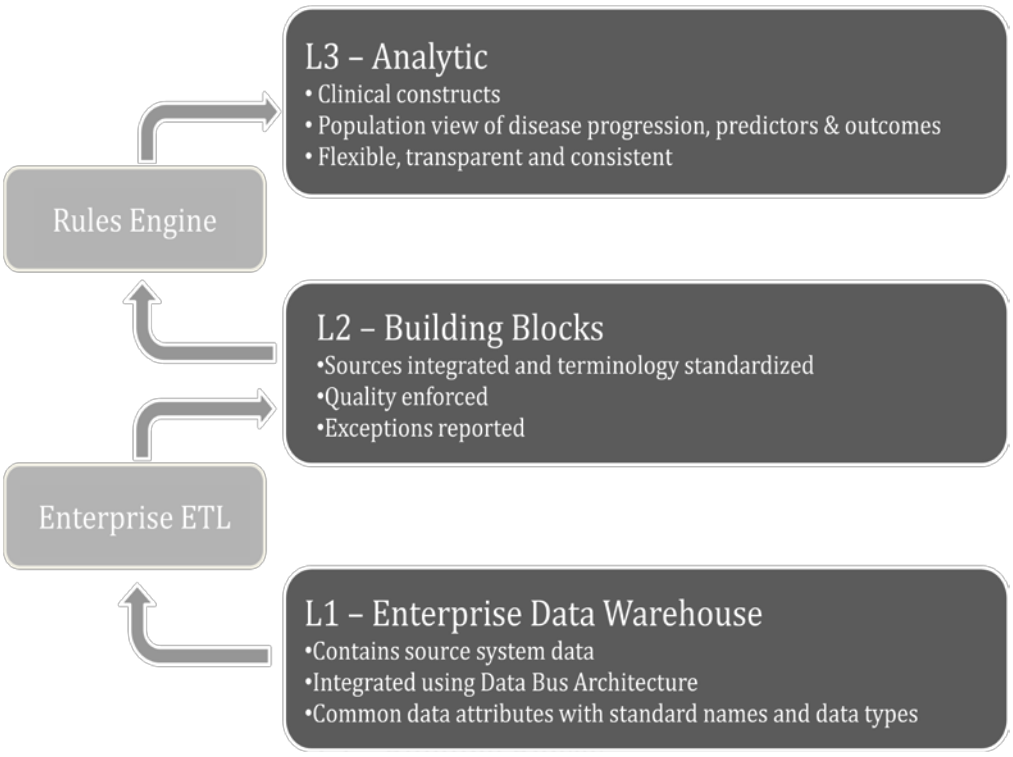

Figure 1, AHR layers

L2 is the layer in which data are imported from multiple disparate source systems, cleaned, integrated and loaded into the tables that create the building blocks for L3. At this stage in the development of the AHR, L3 contains only the Charlson Comorbidity Index and supporting data. L2 is currently used by analysts and researchers to access data via more user-friendly views over the tables that make up the blocks.

Four physical schemas currently make up the AHR:

- AHR_REF – the definitions and reference data
- AHR_L2 – multi-source data aggregated into clinical blocks
- AHR_RULE – the logic and rule sets required for L3
- AHR_L3 –patient-level analytic data

One of the biggest challenges facing the AHR is identifying, correcting and logging the myriad of data quality issues found in each source system. According to a Gartner study, “More than 50 percent of business intelligence and customer relationship management deployments will suffer limited acceptance, if not outright failure due to lack of attention to data quality issues.”(4) And Dasu, Vesonder and Wright report that operational databases commonly have 60% to 90% bad data.(5)

It is obviously better to prevent errors at their source. Data integrity is considered by most to be synonymous with data quality and refers to the overall correctness and accuracy of the data. Data quality in operational—frequently known as transactional systems—can be addressed through multiple methods that have been described since the inception of databases. The father of relational database theory, Edgar F Codd, designed the relational model and a process called data normalization to reduce data redundancy and subsequently improve data integrity. Data can be, and often are, denormalized to

optimize the performance of a database. As a rule, data warehouses store their data in denormalized table structures. The process of denormalization, by its very definition, is going to obscure many data quality problems.

Operational databases frequently come into existence secondary to the applications they support without addressing the overall structure of the data or its existence independent of the initial application. When the initial application gets upgraded, converted to a new system, or the data is mapped and interfaced to another system, unseen and frequently unanticipated data quality issues will surface. When strategic initiatives mandate a move towards data-driven decision making for planning and management, preexisting data is loaded into data warehouses or repositories. As these data generally support analysis, data quality is of overriding importance.

Data quality is an important aspect of any database but when aggregating and integrating primary health care data for clinical research, data quality management is critically important. There is no simple answer or "one size fits all" solution available. For over a decade, researchers have struggled to develop objective and subjective data quality measures.(6) Significant research has led to long lists of "candidate data quality attributes" or focused on "stimulating thinking by the design team"(7) but a specific methodology for deriving basic data quality rules for a data set remains vague.

In the book "*Data quality assessment*", Arkady Maydanchik (8) begins by grouping the causes of data problems into three broad categories.

- *External processes* include manual data entry, batch feeds, real-time interfaces, data conversions, and system consolidations—all of which

either bring flawed data in from the outside or introduce errors during the process.

- *Internal processes* such as data processing, cleansing and purging can inadvertently corrupt data.
- *Data decay* results over time when experts leave a business taking the domain knowledge with them, system upgrades rely on out-dated data models and metadata, old data is inadequate for new uses, semantic changes are not captured by the data, or process automation replaces the human overseer of data quality.

In large analytic data stores, data usage is much less predictable, compounding the challenge of sustaining a satisfactory level of data quality. Experienced data architects and analysts know anecdotally what data quality problems are common and it frequently falls upon those implementing the data repository to identify and implement the appropriate quality checks. Frequently these observations are organized into some sort of data quality classification schema and applied to data as it is extracted, transformed and loaded from the source into the analytic system. Unless there is current metadata or mapping-specifications available for the source system, the data quality specifications will focus on applying the basic syntactic and key constraints of the source system. There is seldom master reference data available to provide domain standards and the overall content can be even further obscured if the data set contains free text.

This thesis reviews current literature on data quality, focusing on the taxonomies that have been suggested for categorizing data quality rules. Background information is presented on the data sources, relational database theory, and data integrity rules. A data

quality taxonomy is derived from relational database theory. In the results section, the data quality processes and rules currently in place in the AHR are described and subsequently categorized into the taxonomy. This taxonomy is reviewed for its strengths and weakness and a discussion of next steps for improving data quality in the AHR follows.

LITERATURE REVIEW

As in other fields, poor data quality affects all levels of operational, tactical and strategic decision-making in healthcare. Anecdotal reports of the extent of the problem vary but its existence is pervasive throughout all industries.(9) Poor data quality in clinical data repositories includes data entry errors, inconsistent data types, multiple semantic representations and missing data elements.(10) Bad data impede clinical research, quality measurement and process improvement initiatives and inhibit the development of federated networks of electronic health databases.(11)

Concerns regarding data quality emerged almost immediately after the first commercial database was developed at General Electric in 1964. Based on the network data model developed by C W Bachman, the primary function of the database was to isolate data from the logic of applications and to make it available to more than one program.(12) In 1970, E F Codd, the inventor of the relational model for database management, proposed applying his relational theory to “large banks of formatted data” as a solution to the problems of data independence and inconsistency that were surfacing from such systems. He described how the relational view of data provided better insight into the limitations of a system and a firm foundation for dealing with the data quality issues of redundancy, inconsistency, and the confusion surrounding derived data.(13)

Throughout the past four decades, there has been a plethora of literature surrounding the topic of data quality. Because of the correlation between quality issues in

information systems and product quality in manufacturing, researchers in the field have suggested that a review of data quality literature be broken down into the following seven sections: 1) management responsibilities, 2) operation and assurance costs, 3) research and development, 4) production, 5) distribution, 6) personnel management, and 7) legal functions. The third section, research and development, is further divided into three subsections: the definition and characteristics of quality dimensions, the analysis and design of quality features, and the design of the systems that implement these features.(14)

This literature review focuses primarily on the first subsection under research and development, the definition of data quality dimensions and their characteristics. Throughout the literature, quality dimensions have been defined and described in many different ways. Tu and Wang (15) suggest modeling data quality and context through an extension of the Entity-Relationship (ER) model. The ER model is used by database designers to capture the semantics and business rules related to a database. Chen, who first suggested the ER model, proposed that the model could be extended to incorporate quality aspects. However, there is no consensus on what constitutes a good set of data quality rules. Even the most commonly mentioned dimensions, accuracy and completeness do not have clear, unambiguous definitions. For example, Ballou and Pazer (16) define accuracy as when “the recorded value is in conformity with the actual value” and Loshin describes accuracy as “the degree to which data values agree with an identified source of correct information.”(17) Although the term appears synonymous with correctness, there is little structure or rigor to such definitions.

There are two principal ways of determining data quality dimensions: a pragmatic methodology and a scientific approach. Both processes present data quality as a “multidimensional” concept and can be implemented with varying degrees of structure and rigor.(4,18)

The pragmatic approach bases the choice of dimensions on intuitive understanding, experience or a review of the domain literature in which the dimensions are user-defined, either by committee, expert consensus or the government, and depend heavily on the context of the data.(19)

In a two-stage survey, Strong and Wang collected and analyzed a list of 118 data quality descriptors from data consumers and taxonomically grouped them into fifteen dimensions and four categories: intrinsic quality, contextual quality, representational quality, and accessibility. In the Strong and Wang study, accuracy is considered a dimension of intrinsic quality, completeness is a dimension of contextual quality, and consistency belongs to the representational quality class.(20)

In 1998, Wang characterized data as a product.(21) He used a manufacturing-based task cycle (define-measure-analyze-improve), with the data quality specifications defined by end-users, to create a process to generate a better quality information product. Wang’s conceptualization of data as a product was a significant departure from the traditional view of data as a system by-product and presented theoretically-grounded methodologies for Total Data Quality Management. This methodology relies heavily on end-user interaction with the data to define the data quality dimensions specific to each data set.

A large body of the literature seems to concur that business value is actually reflected more by the content and business use of data and therefore, a context-based assessment of quality is more appropriate for determining data quality dimensions. In a five-year study reported by Lee in 2003,(22) experienced practitioners solved data quality problems by analyzing the context of the data and defining data quality dimensions. This study found that the practitioners were able to solve data quality problems by "reflecting on and explicating knowledge about contexts embedded in, or missing from, data." This arduous process required years of expert review of the data to derive the quality dimensions for a single data set.

A second, more objective way of defining data quality dimensions is the scientific approach. The scientific approach is design-oriented and attempts to align the quality dimensions of data with the data structures themselves. A design-oriented definition of data quality can reveal the expected use of the information and because it is rooted in database theory, it has the potential to provide guidance to system developers.(11) The following four methods indicate efforts toward a more scientific methodology.

An ontology-based approach to data quality focuses on system design without being source specific and identifies data deficiencies in terms of the difference between the real-world representation (identified by direct observation), and the inferred information system representation. This analysis generates four data quality dimensions: complete, unambiguous, meaningful and correct.(12) The ontology-based approach is similar to the conceptual guidelines presented by Codd and the presentation of the definitions is rigorous. But the derivation of the dimensions is less than scientific and fails to give any concrete guidelines to developers.

A semiotic framework for data quality uses the linguistic theory of sign-based communications as a theoretical basis for defining data quality dimensions and deriving the quality criteria in each category. This framework generates three intrinsic data quality dimensions: syntactic, semantic and pragmatic.(23) The quality criteria for the syntactic and semantic groups are derived from business integrity rules and Wand and Wang's ontology-based theories.(12) The pragmatic dimension is based on subjective measures. Unfortunately, the first two dimensions rely heavily on the preexistence of database metadata which is frequently unavailable or nonexistent, particularly for legacy systems. Early information systems generally did not consider metadata a part of the development process and more recent systems frequently only include it as an afterthought. Even when metadata can be found, they are often outdated and incorrect.

Oliveira, Rodrigues and Henrigues present a taxonomy of data quality problems, organized by level of database granularity from the lowest level of a single attribute to the highest level of multiple data sources. The significance of this taxonomy is that it is based on the fundamental relational model for database management and is expressed using relational algebraic notation. At the four levels of granularity—attribute, table, database and multiple source—this approach details specific data quality problems related to the data quality dimensions of completeness, consistency and accuracy.(24) This is a data-centric methodology and comes closest to using a scientific approach to provide tangible data quality dimensions. However, it strays from scientific rigor in an attempt to be all inclusive of a wide variety of data quality problems.

Finally, the amount of data available on the World Wide Web (WWW) has lead to the development of semantic web technologies to manage the quality of data published in

online networks. Fürber and Hepp propose the use of a “domain-independent machine-readable conceptual model for data quality management in the form of an ontology.”(25) The requirements for this ontology are based on a series of questions which require the informant to know the content and context of the target database. The answers to the questions are then used to define the conceptual elements of the model, their properties and the associated rules. Currently, this approach has only been tested on a very small dataset and lacks a sufficient vocabulary to provide a tool for data quality management on the WWW.

Despite the growing volumes and complexity of large analytic data repositories, there is very little consensus on usable data quality dimensions and much of the research literature on data quality rules remains theoretical. While most authors identify similar problems, there is no general agreement on a pragmatic, design-oriented methodology for identifying data quality dimensions and rules.(26)

A recently published book, “*Data quality assessment*,” by Maydanchik (7) fills a void for those in need of a practical guide for the definition of data quality rules and the development and implementation of a data quality assessment process. Although rhetorically unassuming, the content is technically complex. *“Data quality assessment”* provides clear guidelines for identifying and implementing data quality rules in a large database. Based on decades of experience with large data sets, Arkady’s book focuses on data quality assessment, “the process of identifying data problems and measuring their magnitude and impact on various data-driven business processes.” He devotes five chapters to data quality rules and groups them into the following categories:

- Attribute domain constraints

- Relational integrity rules
- Rules for historical data
- Rules for state-dependent objects
- Attribute dependency rules

These rules provide the foundation for a data quality assessment process, an integrated metadata repository and a data quality scorecard. The data quality scorecard is the end result that merges the various components of the data quality assessment.

The question remains as to whether this type of data quality assessment is applicable to medical data. In a study examining the feasibility of using the data available from a computerized patient record to support point-of-care decision support for patients with community-acquired pneumonia, it was determined that the overwhelming source of error is free-text nurse charting. The study also notes that the dearth of computerized guidelines is related to the difficulty of retrieving clinically relevant data in a computable form and the need to validate the quality of data prior to the implementation of clinical guidelines. The authors mention that few studies focus on how to assess data quality in clinical data.(27)

While the data quality of free-text remains a significant problem of its own, Intermountain's AHR addresses the second obstacle to computerized guidelines by making clinically relevant data more available to researchers and clinicians. Many of the quality rules suggested by Maydanchik are directly applicable to data extracted from the source systems and the relational structures of the AHR.

Hasan et al.(28) provide a quantitative analysis of the impact of poor data quality on clinical decision support systems. They propose future research to design controls to

detect and minimize data quality problems. One of their concluding goals is to define "realistic distributional and structural assumptions about the nature of patient data and errors." This goal is consistent with the data quality rules suggested by Maydanchik and the taxonomy suggested in the following section of this paper.

BACKGROUND

Data Sources

Intermountain Healthcare is a nonprofit, integrated healthcare delivery system that includes 22 hospitals, multispecialty clinics, InstaCare centers, lab services, homecare, hospice, a physician division and a health insurance service. It is a community-oriented organization with over 32,000 employees that serves the healthcare needs of Utah and southeastern Idaho.

The EDW at Intermountain is comprised of data from most of Intermountain's computer systems incorporating clinical, operational and financial systems. It is a relational database that includes over a terabyte of clinical data that is organized into source-specific data stores and clinical program datamarts. The AHR is contained within the EDW and includes data from the major clinical systems. The AHR is designed to be the trusted source of clinical data for research and quality improvement studies leveraging all of Intermountain's data assets and expert knowledge. AHR data are integrated across source systems. It is cleaned, standardized and optimized for population-based analysis. The multifaceted nature of clinical data and its complex representations combine to make data integration a challenge.

Four types of data quality violations are tracked and reported in the AHR:

- structural constraint violations, e.g., a letter value where there should be a number,

- single-source rule violations, e.g., missing patient identifier,
- multisource rule violations, e.g., BP on a deceased patient,
- expectation violations, e.g., trending in lab values.

The specific data quality rules are established on a case by case basis.

One of the primary sources of inpatient data for the AHR is the HELP system. One of the first computerized health information systems, HELP was installed at LDS Hospital in the late 1960s. It is currently in place at all Intermountain facilities and has evolved to become a complete knowledge-based patient care system. HELP data are used for the Lab, Vital Sign, Clinical Assessment, Problem and Encounter Blocks in the AHR. The hospital admission, transfer and discharge (ADT) data that originate in the HELP system are extracted from the EDW and transformed and loaded into the AHR Encounter Block. In the AHR, this dataset is formatted to include the date and time each patient entered a room or department (Lab, ED, OR) during their hospital stay. This supports research involving questions such as, what is the average time a patient spent in the emergency department prior to admission for pneumonia. The ADT component of HELP has evolved over the years to meet many needs in many different ways. It not only records when a patient is admitted, transferred and discharged, but it has been used to preadmit patients, create fictitious beds to report overflow, and track miscellaneous charges. There is over a terabyte of patient data in HELP. Therefore, it is a formidable challenge to insure that the data are correct.

A second significant data source for the AHR is the Clinical Data Repository (CDR). The CDR was developed by 3M in conjunction with Intermountain Healthcare and serves as a longitudinal patient record incorporating clinical data interfaced from

both inpatient and outpatient systems. CDR data are used for the Lab, Rx Order and Problem Blocks in the AHR.

The Sunquest Lab System is the third major source of AHR data. Sunquest consolidates the lab data for all of Intermountain Healthcare. The terminology is standardized across the system and extensive quality checks are in place to verify the validity of the data.

Relational Databases

The relational model was developed in 1970 by Edgar F Codd who implemented the model using relational calculus. Codd used set theory to represent data as mathematical relations. By using relational calculus, Codd was able to develop a flexible and succinct model.(29)

In the relational model, a database consists of several tables each representing an entity. An entity is similar to a noun. It can be a person, place or thing. It is something that is distinct from other aspects of the real world. Examples of an entity include Employee, Patient, Room, or Discharge Order.

A table consists of one or more columns that uniquely describe the entity. A column is referred to as an attribute and an attribute is populated from a domain. A domain describes the set of possible values for a specific attribute. A table is also made up of rows, sometimes called tuples. Each row holds a database record.

A relationship describes how two entities relate to each other. A relationship can be thought of as a verb. An example of a relationship is an Occupies relationship between the Patient and Room entities.

Entities and relationships can both have attributes. The attributes of a Patient entity could include gender, birth date, death date and blood type. The attributes of the relationship Performs between the entities Physician and Procedure could be start time, end time, patient name and room.(30)

A candidate key is any attribute or set of attributes that uniquely defines a record. For example, in a Patient table both the Enterprise Master Patient Identifier (EMPI) and the Social Security Number (SSN) may be candidate keys. However, if a candidate key is composed of more than one attribute, any subset of the candidate key cannot also be a candidate key. Therefore, if the Account table has a compound candidate key of hospital and account number, then the account number by itself should not be a candidate key.

A primary key is a special case of a candidate key. One candidate key can be selected as a primary key and the other candidate keys become alternate keys. No two rows can have the same primary key value. A primary key may be a surrogate key or a natural key. A natural key is a candidate key and as such, has some sort of meaning, such as SSN or EMPI. A surrogate key is assigned by the database system and is used solely to identify a record. Surrogate keys frequently mask otherwise redundant records.

The concept of functional dependency is related to the primary key and is important to data quality. Functional dependency describes the relationship between attributes in a table. In the statement of functional dependency X —> Y, attribute Y can be said to be functionally dependent on the attribute X (the candidate key), if each value of X is associated with one, and only one, value of Y.(31)

A foreign key uses the primary key of one table to link it to the data in another table to allow cross-referencing. A foreign key can be self-referencing causing a recursive relationship within the table. This generally allows for greater flexibility in design.

There is a specific relationship between tables based on cardinality. Cardinality specifies how many rows of an entity relate to one instance of another entity. For example, to be in the Patient table a person must have at least one hospital encounter. It is also possible that the same patient could have many encounters. Therefore, one row in the Patient table is related to one or many rows in the Encounter table. In this case, the cardinality between the Patient and Encounter tables is one-to-many.

The process of organizing data in relational databases to minimize redundancy and enforce data integrity is called database normalization. To identify and eliminate anomalies, normalization decomposes relations to produce smaller, well-structured relations. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications to a field can be made in just one table and then propagated throughout the rest of the database via the defined relationships. This is done to avoid redundancy and inconsistencies. The process of normalization evokes specific levels of data integrity constraints aligned with the progressive levels of normalization. Each consecutive form is dependent on the validity of the prior form so a table that is in Second Normal Form is by definition, also in First Normal Form. A database is generally considered normalized, if it is in Boyce-Codd Normal Form.(32)

- First Normal Form (1NF) – Multi-valued attributes are removed. All domains must contain only scalar values. The intersection of each column

and row should contain only one value. Consider the attribute, Diagnosis. If the field contains a single diagnosis, it is in 1NF. If it can contain the string, “HF, COPD, MI” it is not in 1NF.

- Second Normal Form (2NF) – In 2NF, every non-key attribute must be a fact about the entire key. 2NF is relevant only if the candidate key is a composite key. Partial dependencies are removed. Partial dependencies are when one attribute can be removed from the candidate key and the dependency still exists. Consider a table of patient lab results. If the candidate key for the table is (EMPI, Collection Date, LOINC®,(33) Lab Name) —> Lab Result, there is a partial dependency. Lab Name is dependent on LOINC®. When Lab Name is removed from the candidate key, it becomes (EMPI, Collection Date, LOINC®) —> Lab Result, the partial dependency is removed, and the table is in 2NF.
- Third Normal Form (3NF) – This form removes transitive dependencies. No non-key attribute is dependent on another non-key attribute. For example, if EMPI —> (City, State, Postal Code) it is not in 3NF. (City, State) is transitively dependent on Postal Code. However, EMPI —> Postal Code is in 3NF.
- Boyce-Codd Normal Form (BCNF) – This form expands upon the previous forms to include tables with compound keys, in which all attributes contribute to some candidate key. BCNF requires that the only determinants are candidate keys. Table 1 is an example of a table in 3NF that is not in BCNF.

Table 1, OR Schedule

Room	Start Time	End Time	Surgeon
OR1	0700	1000	Dr Jones
OR1	1000	1200	Dr Jones
OR2	0700	0930	Dr Brown
OR2	1000	1130	Dr Brown

- Fourth Normal Form (4NF) – In 4NF there are no multivalued dependencies. Therefore, all attributes are functionally dependent on the candidate key.

Designing for Data Quality in an Integrated Data Repository

When designing and implementing an integrated data repository, there are three broad categories to consider in defining data quality constraints.

- The data imported from the source systems
- The structure of the destination repository
- The data loaded into destination repository

The constraints placed on data imported from source systems are defined by the preexisting table structures and primary keys. If a table does not have a primary key, it must be identified by analyzing the data. Ideally, an initial data quality assessment of the sources would include a data profile of the interdependencies within the data and an

assessment revealing preexisting data errors and their causes. In reality, this is an enormous task and seldom done, especially on very large historical datasets.(6) However, the most basic review of the source tables should identify the primary keys, foreign keys, attribute constraints, and update timestamps for the tables that contain the data of interest. Primary keys may be the source system's surrogate key or a key that has been identified by analysis as defining unique records. Foreign keys define the table joins and the validity of these keys should be evaluated. The source system probably has preexisting rules defining constraints on the attributes of interest but additional constraints can be developed to minimize the import of unusable data. Tangentially related to data quality are the update timestamps. Not all sources have update timestamps and the validity of those that do cannot be guaranteed. However, if the repository is going to be incrementally updated, it is essential to be able to identify new and modified records.

The structures of the destination data repository should ideally be in Fourth Normal Form. This insures the consistency and integrity of the incoming data and illicits additional data quality constraints that should be applied to the incoming data. The normalization process used to decompose the source tables is a fundamental methodology that helps to insure data consistency.(34) Normalized tables provide a framework that facilitates the application of relational data integrity rules.

By adhering to a semantic methodology in conceptually defining the tables and their primary keys, the structure of the database can provide another layer of quality assurance. For example, when bringing encounter data into the AHR, the source system defined the primary key for each encounter as the compound key, (Facility, Account Number). Account numbers could be duplicated in different facilities but not in the same

facility. This was validated in the source system (Table 2). In the AHR, the base Encounter table had a unique key constraint assigned to the candidate key, (EMPI, Entry Timestamp). This candidate key represents the concept that a person can only be one place at a time and all other attributes in the table should be functionally dependent on that key. The key constraint failed. The root cause was a batch update in the source system that had inadvertently duplicated rows and assigned another facility to the duplicate (Table 3). This demonstrated the importance of using a semantic model and normalization to identify the data integrity rules in the analytic repository.

Data Quality Constraints

In relational databases, data quality and consistency are maintained by the use of constraints. Constraints restrict the data that can populate an attribute and which rows are allowed within a table and database. Constraints can be applied to the attribute, table and

Table 2, Source System Primary Key

Facility ID	Account No
128	10028557
132	10028557
128	23244412
144	23244412
132	98843398
118	98843398

Table 3, AHR Candidate Key Constraint Reveals Data Quality Problem

EMPI	Entry DTS	Account No	Facility ID
40312311	01/03/2008 12:53	10028557	128
40312311	01/03/2008 12:53	10028557	132
47822312	11/12/2004 04:11	23244412	128
47822312	11/12/2004 04:11	23244412	144
32106629	01/03/2008 12:53	98843398	132
32106629	01/03/2008 12:53	98843398	118

database levels, and upon the integration of disparate sources. The integrity constraints for an integrated data repository can be loosely categorized into four groups: domain constraints, entity constraints, database dependency constraints, and integration constraints. Domain-level data quality problems are related to inconsistencies and errors in the actual data content. Entity level problems are also reflected in the attributes but can best be addressed by improved database design.

- Domain constraints refer to constraints on the values of single attributes.
    - *Optionality* constraints prevent attributes from being null or using defaults as a substitute.
    - *Acceptability* constrains domain values to atomic, nondecomposable values.
    - Domain dependency constraints limit the values that can populate an attribute based on the domain definition. This can include:

        - *Format constraints*, which define a specific format and limit the values to that format.
        - *Business domain constraints*, which represent a business decision about the values that are acceptable to a specific domain.
        - *Precision constraints*, which require that all numeric values use the prescribed number of decimals and that all date/time values are carried out to the same level of granularity.(7)

- Entity constraints refer to constraints between attributes and rows within a single table and are related to relational integrity. There are four basic types of entity constraints:
    - *Entity integrity* concerns the concept of a primary key and states that no primary key can be null. A primary key insures that each record in a table corresponds to one and only one real world entity. Surrogate keys are conceptually meaningless and frequently mask real-world duplicates.
    - *Functional dependency* (FD) describes the relationship between attributes in the same table. FD is when one attribute in a relation uniquely determines the value of another attribute. A candidate key, and therefore primary key, should uniquely determine the values of the attributes in that record. For example, if the Encounter Room entity has a candidate key (EMPI, Entry Timestamp), it is assumed

that a person can only be in one place at a time. It would therefore be a FD constraint violation for two records with the same (EMPI, Entry Timestamp) key to have different values for the Facility attribute.

  - *Business rule constraints* define the relationship rules between attributes within the same record. Within a record, a business rule constraint might dictate that a patient's admit date must precede their discharge date.

- Database constraints refer to constraints related to multiple tables within a database.
  - *Referential integrity* refers to the relationship between tables. Every reference between tables must be successfully resolved. When the primary key from one table is included in the attributes of another table, it is called a foreign key. Foreign keys join tables and create dependencies between them. Foreign key constraints insure referential integrity.
  - *Cardinality* rules constrain the number of related occurrences between entities, e.g., all Patients will have at least one Encounter.
  - *Inheritance* rules constrain the data that is involved in sub-type relationships, e.g., Person, Patient and Provider entities.(6)
- Multisource integration constraints refer to rules related to the integration of multiple sources. The integration of multiple sources compounds the data quality problems present in a single system. Multiple sources may

contain similar data but in semantically and syntactically different formats. Data can overlap and disagree. Functional dependency constraints must be resolved when multiple sources contribute attributes for the same candidate key. Data quality rules must manage three types of issues that can occur with overlapping data:

- When multiple sources contribute the same attribute and the values agree, which will be the system of record?
- When multiple sources contribute the same attribute and the values disagree, how will the situation be resolved?
- When multiple sources contribute conflicting attributes for the same record, e.g., death date precedes hospital admit date, how will the situation be resolved?

Data Quality Taxonomy

According to the Montague Institute, “A taxonomy is a system for naming and organizing things into groups that share similar characteristics."(35) In their article, Oliveira et al. describe the benefits of a data quality taxonomy as including the ability to identify data quality problems that deserve further attention.(36)

Table 4 presents a taxonomy specifically for an integrated data repository derived from the data quality constraints described previously in this paper. In the results section, this taxonomy will be compared to the data quality checks actually in place in the AHR.

Table 4, Constraints by Type and Granularity

Type	ID	Constraint	Source System	Repository Structure	Repository Data
Domain	D1	Optionality	X	X	
	D2	Acceptability	X	X	
	D3	Format	X	X	
	D4	Business domain constraints	X	X	X
	D5	Precision	X	X	
Entity	E1	Entity integrity	X	X	
	E2	Functional dependency		X	
	E3	Business rule constraints	X	X	X
Database	DB1	Referential Integrity	X	X	
	DB2	Cardinality		X	
	DB3	Inheritance		X	
Multisource	MS1	Functional dependency			X
	MS2	Inconsistent duplicate values			X
	MS3	Conflicting attributes			X

Constraints against the source systems' data are usually imposed by means of filters and transformations in the ETL process. The domain constraint of optionality is utilized to filter out records in which the attribute of interest is null. The acceptability constraint can be met by parsing strings of concatenated values into scalar elements. Nonstandard formats can be transformed during the ETL process. Business domain constraints are enforced in a number of ways such as requiring that numeric values be in

a certain range or text values in a specific domain. Precision constraints need to be reviewed by business users in order to develop an acceptable management strategy. Entity integrity can initially be maintained by filtering out any records which don't contain valid primary key values as defined in the source system. Functional dependencies can be identified in the source systems by complicated queries or data profiling tools but is best enforced by the table structures and constraints of the destination repository. Business constraints can be enforced on source data with simple queries as can the referential integrity violations. Both cardinality and inheritance violations are more easily identified and enforced in the destination repository.

The normalized design of the destination repository should enable most all of the subsequent domain, entity and database constraints. The remaining business rule and domain constraints can be evaluated in the repository. Multi-source constraints are the final step in this simplified methodology.

RESULTS

The AHR currently resides within the framework of the EDW. Layer 1 of the AHR consists of supporting schemas within the EDW that function as operational data stores (ODS). These schemas contains data from CaseMix (the billing, coding and financial data system), the Clinical Data Repository (CDR), HELP, the outpatient billing system (IDX), lab (Sunquest) and claims systems. Data from L1 are used to populate the L2 Blocks as described in Table 5.

Table 5, AHR Sources and Blocks

		AHR BLOCK									
SOURCE		ICD9 DX	CPT4	LAB TEST	PROBLEM	RX ORDER	RX CLAIM	VITAL SIGN	ASSESSMENT	ENCNTR	PATIENT
	Case Mix	X									
	CDR			X	X	X					
	Sunquest			X							
	HELP			X				X	X	X	
	PHXDBA						X				
	IDX	X	X								
	CLAIMS	X	X								
	PATIENT										X

Source System Data Quality Rules

There is a multipronged approach to data quality in L2 of the AHR. Data quality rules are specified in a declarative fashion so as to be reusable for multiple sources and various stages. A workflow infrastructure exists to execute all data validation steps in a traceable, flexible and robust manner. There are three distinct sections for quality checks within the AHR:

- Filters applied during the extracts from the source systems,
- The semi-normalized L2 data structures that minimize redundancy and enforce data integrity, and
- The checks run against the attributes that populate the L2 tables.

The AHR updates its SRCTRACK entities daily. SRCTRACK was developed by Steven Catmull to keep track of changes (insert, update and delete) in the source systems. The SRCTRACK table has a subtype table for each source system. A complete inheritance relationship exists between SRCTRACK and the subtype tables, requiring that each supertype record has an associated record in one of the subtype tables. Global problems that frequently affect data quality in the source systems have included:

1. No primary key.
2. Unenforced foreign key constraints.
3. Unindicated changes to the source system's structure and content.
    a. Structural changes, such as adding a new field, are usually picked up when they cause the downstream ETL to fail.
    b. Content changes, such as adding a new discharge code, frequently slip through unnoticed.

4. No field indicating when the data was last inserted or updated.

Filters exist on the extract from the source systems to exclude records that are missing the primary key (E1), the EMPI (D1), or have out-of-range event dates (D4). After the update to the SRCTRACK system, a final set of quality checks verifies that all records that are logically deleted have a quality check recorded that instantiates the reason for the delete (DB1).

Specific data quality checks done against the daily updates to SRCTRACK are table driven (see Appendix A). Table 6 contains a summary of data quality issues identified in the daily SRCTRACK load.

Table 6, SRCTRACK Quality Checks

SRCTRACK Quality Checks (Constraint ID)
Patient's first name is null (D1)
Patient's last name is null (D1)
Patient's last name is unknown and is not null (D3)
Birth date after today (D4)
EMPI for test patient (D4)
SRCTRACK event date is too early or after today[a] (D4)
Birth date prior to 1850 (D4)
Birth date is after death date (E3)
EMPI not in PATIENT table (DB1)
EMPI not used in a billing or clinical system[b] (DB1)
EMPI reconciled to a new identifier[c] (MS3)
Event date after the death date of the patient (MS3)
Event date prior to the birth date of the patient (MS3)

[a]Too early is defined as prior to 1994.
[b]The EMPI has been reconciled to a new EMPI but the source system uses the old EMPI.
[c]EMPI is found in the PATIENT table but not in any of the source systems. These records are logically deleted so the EMPI will not be used in clinical cohorts.

AHR Block Quality Rules

The AHR runs a weekly full refresh of the data that populates its L2 Blocks. The data quality checks done during the weekly full refresh of the L2 Blocks are hard-coded, in-stream quality checks (see Table 7). The greatest proportion of quality checks involves domain constraints, functional dependency violations, business rule violations and integration inconsistencies.

Table 8 presents a comparison of the number of data quality constraints that are currently in place in L2 of the AHR with the Data Quality Taxonomy presented earlier. There are no AHR rules listed for domain acceptability (value is constrained to atomic, non-decomposable values), domain precision, cardinality or inheritance rules for data that are involved in sub-type relationships.

Table 7, Block Quality Checks

Domain Constraint Violations (Constraint ID)
Provider ID is null (D1)
Reported value is null (D1)
Lab result date and time null (D1)
Problem ID is null (D1)
Medication ID is null (D1)
Room Code is null (D1)
Place of service is not numeric (D3)
Lab result is not numeric (D3)
Numeric result is outside of allowable range(D4)
Account is marked as a junk account (D4)
Entity Constraint Violations (Constraint ID)
More than one final result for a lab test (E2)
More than one result for vital sign (E2)
More than one service type in the same claim (E2)
More than Provider ID on a claim (E2)
Patient in two or more rooms at the same time (E2)
Multiple entry/exit for a room encounter (E2)
Multiple results (by LOINC) for same test from single source and values differ (E2)
Excessive time elapsed from ADT occurrence to data entry (E3)
Database Constraint Violations (Constraint ID)
Provider ID not found in reference data (DB1)
Unrecognized CPT4 Code Used (DB1)
Unrecognized ICD9 DX Code Used (DB1)
Multi-Source Violations (Constraint ID)
Multiple results for same test from multiple sources and the values agree (MS1)
Multiple results for same vital from multiple sources and the values agree (MS1)
Multiple results for same test from multiple sources and the values differ (MS2)
Multiple results for same vital from multiple sources and the values differ (MS2)
Patient not alive when transferred into room (MS3)

Table 8, Count of Quality Rules in the AHR

Type	ID	Rule Count	Constraint
Domain Level	D1	8	Optionality
	D2	0	Acceptability
	D3	3	Format
	D4	6	Business domain constraints
	D5	0	Precision
Entity Level	E1	1	Entity integrity
	E2	7	Functional dependency
	E3	4	Business rule constraints
Database Level	DB1	5	Referential Integrity
	DB2	0	Cardinality
	DB3	0	Inheritance
Multisource	MS1	2	Functional dependency
	MS2	2	Inconsistent duplicate values
	MS3	4	Conflicting attributes

DISCUSSION

In their article, *An Ontology-Based Approach for Data Cleaning*, Oliveira et al. conclude that "a completely automated system that receives dirty data, detects and corrects the problems, and produces clean data without user intervention, is impossible to achieve."(36) However true this may be, the data quality taxonomy presented in this thesis illustrates that a significant number of data quality constraints can be addressed by a well-designed conceptual data model. Business rules will always require user input. But many business rules can be captured in the data models and translated into data integrity constraints. A hybrid approach to the development and implementation of a data quality program will most likely involve a mixture of user input and automated rule generation.

AHR Data Quality Shortcomings

The data quality of the AHR is still far from perfect and it is a challenge to anticipate the different ways in which end users may need to query the data. The specific quality areas that are not adequately addressed by the AHR constraints include domain level acceptability, precision, cardinality and inheritance.

- Domain level acceptability is defined by 1NF as the requirement that attributes be atomic and scalar. The blocks in L2 of the AHR were designed with that basic requirement in mind and a majority of the data currently imported is scalar. However, there are non-scalar values and free

text records in a few fields (e.g., room code and problem) that have been loaded into the AHR and rules should be developed to deal with this.

- Precision has been a problem in the AHR and some convoluted strategies have evolved to identify and correct this issue. With end user input, these approaches can be standardized and applied to all numeric attributes to identify and, if possible, correct the imprecision.
- Cardinality is used in loading the Patient table to exclude persons for whom no clinical data exists. Cardinality rules can be elicited from business users or identified by an analysis of the relationships in the source data. Generally, each relationship will have two cardinal rules.(6)
- Inheritance refers to the requirement that each sub-type record must have a record in the super-type table and conversely each super-type record must participate in at least one sub-type entity. The classic example of this in the healthcare setting is between the entities Person, Patient and Provider. Every person is a patient, a provider, or both. Any Person record not found in either the Patient or Provider entity is either incorrect or indicates a missing Patient or Provider record. Although the AHR has not used cardinality constraints, their applicability should be evaluated.

Data Quality Taxonomy Shortcomings

The data quality checks listed in the taxonomy do not always address the causes of the constraint violations. Frequently, it is not obvious which data element has triggered the violation. Knowing the exact location of the error is important regardless of whether

the record is eliminated or corrected. The taxonomy is only a starting point. Techniques should be developed to identify false positives (records that prove to be correct), false negatives (errors that are missed by the data quality constraints) and uncertainties in which the exact location of the error is not clear.(6)

Other data quality areas not covered by the taxonomy include quality issues related to historical data, the complex topic of state-dependent objects,(7) and trending violations for specific attribute values. All of these areas can contribute significantly to problematic data especially in datasets that are too large for manual overview.

Data Quality Technologies

"Every data quality management guru will tell you that data profiling is the first step towards better information quality. Every data warehousing professional knows that you must profile the source data before implementing a new BI application. A data migration consultant will place data profiling on the first page of the project plan. Master data management starts with data profiling and it is a cornerstone of any metadata repository."(37) Data profiling is a series of techniques used to analyze a data source to understand its content. It generates information that can be used to evaluate the quality of the data, update preexisting metadata and models that describe the data, and understand the risks inherent in the data source. Basic types of data profiling include attribute profiling, relationship cardinality profiling, and dependency profiling. Data profiling can be done manually but many vended tools are currently on the market and should be evaluated for use in conjunction with the AHR.

Master Data Management (MDM) technology combined with a robust data quality program could significantly help in the maintenance of the domain constituents of the AHR. In an Internet article on MDM, Dan Power recommends, “profiling the individual source systems” and using “the selected data quality tool as a staging area and filter for loading the MDM hub.”(38) Terminology and domain management are areas that could be addressed within the context of MDM technology.

A data quality taxonomy could evolve into an ontology and be used in conjunction with a rules engine to semi-automate the detection and resolution of data quality problems. The application of a table-driven rules engine to the data validation process would increase its flexibility, facilitate the identification and implementation of new rules, and support a more sophisticated data quality program.

CONCLUSION

Secondary use of Electronic Health Records (EHR) is increasingly important for clinical research. As more institutions port this information into clinical warehousing environments and federated database systems, clear and effective rules for insuring adequate data quality become essential. In a recent article on the data quality issues encountered in the use of secondary EHR data for a survival analysis of pancreatic cancers, the authors note that, “Effective strategies for secondary use of EHR data could also be accumulated from case studies and shared with the research community as the best practices.”(39)

Experience gained in identifying data quality issues in the AHR indicates that many errors can be identified by the application of relational data integrity constraints both to the raw data as it is imported from the source systems and to the new data constructs within the integrated data repository.

In this paper, a data quality taxonomy for an integrated data repository is presented. The taxonomy is based on relational database theory and provides a foundation for identifying and grouping data constraint violations. It builds in granularity from the attribute level on up to the multisource database level and aligns the described violations with the concepts explicated by normal form theory.

The data quality rules applied to data coming into the AHR were originally designed in a heuristic manner by developers who had significant experience with the

source systems. The original AHR rules are listed in this paper and subsequently grouped within the taxonomy. It is noted that a significant amount of problematic data is excluded from the AHR at the onset by filtering out records that do not meet basic entity integrity requirements (i.e., they do not have a primary key). Good coverage by the AHR rules can be noted in several areas including domain optionality, business rules, functional dependency and multi-source inconsistencies. The areas that are not well covered by the AHR rules include domain level acceptability, precision, cardinality and inheritance. In those areas, a more rigorous data quality methodology should be explored.

This comparison between a data quality taxonomy designed for an integrated data repository and the data quality rules developed for the AHR demonstrates the benefits of a structured approach to data quality rule identification.

APPENDIX

The data structures (see Figure 2) of the AHR Quality Checks revolve around a primary table, the Quality Check table that contains the information relevant to specific data quality constraints (see Table 9). The Quality Level table (see Table 10) contains information on the original Quality Levels recommended by Jason Jones for inclusion with the Quality Checks. Quality Message records are generated by the data constraints run during the ETL and identify the quality violation, the source of the error and the date the error was detected.

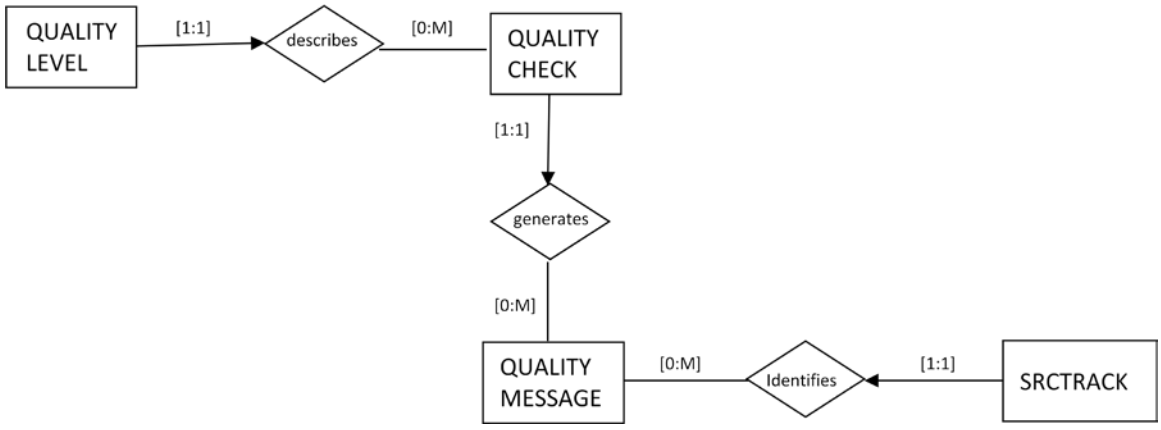

Figure 2, Entity-Relationship Diagram for the AHR Quality Check

Table 9, Quality Check Table

Attribute	Description
Quality_Id	Primary Key
Stage_Nm	The ETL stage in which the quality check runs
Stage_Seq_No	Within the ETL stage, the order the quality check occurs
Schema_Nm	The schema from which the ETL job originates
Table_Nm	The primary table for the quality check
Prmry_Field_Nm	The primary field for the quality check
Quality_Level_No	Foreign key from the Quality Level table (see Table 13)
Action_Cd	Action triggered by the quality violation (Warn, Reject)
Message_Txt	The reporting message associated with this quality check
Status_Cd	Status of the Quality Rule (Active, Inactive, In Dev)
Key_Column	The primary key on the primary table with the violation
Sql_Txt	SQL for the quality check
Last_Update_Dts	Most recent date and time the quality check was edited

Table 10, Quality Levels

Quality Level	Description
QL1	Structural Constraint Violations - violate basic database constraints
QL2	Single-Source Context/Rule Violations - violate expected rules in a single source
QL3	Multiple-Source Violations - cross-source integrity violations
QL4	Expectation violations - trending violations

Table 11, Quality Message Table

Attribute	Description
Quality_Id	Primary Key
Srctrack_Id	Primary Key
Insert_Dts	The date and time the quality violation was identified

REFERENCES

1. Gardner RM, Pryor TA, Warner HR. The HELP hospital information system: Update 1998. Int J Med Inform. 1999;54(3):169-182.

2. Grover FL, Shroyer AL, Edwards FH, et al. Data quality review program: The Society of Thoracic Surgeons Adult Cardiac National Database. Ann Thorac Surg. 1996;62(4):1229–31.

3. Charlson ME, Pompei P, Ales KL, MacKensie CR. A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation. J Chronoc Disease. 1987; 40(5):373-83.

4. Informatica Corporation. Data quality in data warehouse and business intelligence. c2008 [cited 2011 Mar 3]. Available from: http://www.informatica.com/it/Images/03030_6702_dq-dw-bi.pdf

5. Dasu T, Vesonder GT, Wright JR. Data quality through knowledge engineering. Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining; 2003 Aug 24-27; Washington, DC. New York: ACM; 2003. P.705-710.

6. Pipino LL, Lee YW, Wang RY. Data quality assessment. Commun ACM. 2002 Apr; 45(4):211-218.

7. Wang RY, Kon HB, Madnick SE. Data quality requirements analysis and modeling. Proceedings of the Ninth International Conference on Data Engineering; 1993 Apr 19-23; Vienna, Austria. Cambridge, MA: Massachusetts Institute of Technology, 1992.

8. Maydanchik A. Data Quality Assessment. Bradley Beach, New Jersey: Technics Publications; 2007.

9. Wand Y, Wang RY. Anchoring data quality dimensions in ontological foundations. Commun ACM. 1996 Nov;39(11):86-95.

10. Lin J, Haug PJ. Data preparation framework for preprocessing clinical data in data mining [Internet]. AMIA Annu Symp Proc 2006: 489-493 [cited 2012 Mar24]. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839316

11. AHIMA e-HIM®. Workgroup on Assessing and Improving Healthcare Data Quality in the EHR. Assessing and improving EHR data quality. Journal of AHIMA. 2007 Mar; 78(3):69-72.

12. Danielsen A. The evolution of data models and approaches to persistence in database systems. 1998 May 6 [cited 2011 Oct 17]. Available from: http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2000/DAP_DisAvDB/documentacion/OO/Evol_DataModels.html

13. Codd EF. A relational model of data for large shared data banks. Commun ACM. 1970 Jun; 13(6):377-387.

14. International Organization for Standardization. ISO 9000: International standards for quality management. Geneva, Switzerland: International Organization for Standardization; 1992.

15. Tu SY, Wang RY. Modeling Data Quality and Context Through Extension of the ER Model. Proceedings of the WITS-93 Conference. 1993 Dec; Orlando, Florida.

16. Ballou D, Pazer H. Modeling data and process quality in multi-input, multi-output information systems. Manage Sci. 1985 Feb; 31(2):150–162.

17. Loshin D. The Practitioner's Guide to Data Quality Improvement. Burlington, MA: Elsevier; 2010.

18. Wang RY, Storey VC, Firth CF. A framework for analysis of data quality research. IEEE Transactions on Knowledge and Data Engineering. 1995; 7(4): 623-640.

19. Wand Y, Wang RY. Anchoring data quality dimensions in ontological foundations. Commun ACM. 1996; 39(11).

20. Wang RY, Strong D. Beyond accuracy: what data quality means to data consumers. J Manage Inform Syst. 1996 Spring; 12(4): p 5.

21. Wang R. A product perspective on total data quality management. Commun ACM. 1998; 41.

22. Lee Y. Crafting rules: Context-reflective data quality problem solving. J Manage Inform Syst. 2003 Winter; 20(3): 93-119.

23. Price RJ, Shanks G. A semiotic information quality framework [Internet]. In Decision Support in an Uncertain and Complex World IFIP TC8/WG8.3 International Conference 2004 [cited 2011 Oct 2]. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.9817&rep=rep1&type=pdf

24. Oliveira P, Rodrigues F, Henrigues P. A formal definition of data quality problems [Internet]. In International Conference on Information Quality. MIT, Cambridge, MA, 2005 [cited 2011 Mar 1]. Available from: http://mitiq.mit.edu/ICIQ/Documents/IQ%20Conference%202005/Papers/AFormalDefinitionofDQProblems.pdf

25. Fürber C, Hepp M. Towards a vocabulary for data quality management in semantic web architectures. In 1st International Workshop on Linked Web Data Management. Uppsala, Sweden, 2011 [cited 2012 Feb3]. Available from: http://www.heppnetz.de/files/dataquality-vocab-lwdm2011.pdf

26. Arts DG, de Keizer NF, SchefferG. Defining and improving data quality in medical registries: a literature review, case study, and generic framework. J Am Med Inform Assoc. 2002 Nov-Dec; 9(6): 600-611 [cited 2012 Apr 6]. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC349377/?tool=pubmed

27. Aronsky D, Haug PJ. Assessing the quality of clinical data in a computer-based record for calculating the pneumonia severity index. J Am Med Inform Assoc. 2000; 7(1):55–65.

28. Hasan S, Padman R. Analyzing the effect of data quality on the accuracy of clinical decision support systems: a computer simulation approach. AMIA Annu Symp Proc. 2006; 2006: 324–328.

29. Mooney RJ. First-Order Logic: First-Order Predicate Calculus. Available from: http://www.cs.utexas.edu/~mooney/cs343/slide-handouts/fopc.4.pdf

30. Storey VC, Wang RY. Modeling quality requirements in conceptual database design. In Proceedings of IQ. 1998; 64-87.

31. Padigela, Mahipal. Database design [Internet]. 2004-2006 [cited 2012 Jan 3]. Available from: http://www.mahipalreddy.com/dbdesign/dbqa.htm

32. Date CJ. An Introduction to Database Systems. 6th ed. Reading, MA: Addison-Wesley, Reading; 1995.

33. Logical Observation Identifiers Names and Codes (LOINC®). [cited 2012 Mar 15]. Available from http://loinc.org/

34. Lu DF, Street WN, Currim F, Hylock R, Delaney C. A data modeling process for decomposing healthcare patient data sets [Internet]. Online Journal of Nursing Informatics. 2009 February; 13(1) [cited 10 Oct 2011]. Available from http://ojni.org/13_1/Lu.pdf

35. Graef JL. Ten taxonomy myths. The Montague Institute. 2002 27 Nov [cited 01 Mar 2012]. Available from http://www.montague.com/review/myths.html

36. Oliveira P, Rodrigues F, Henrigues P. An ontology-base approach for data cleaning [Internet]. In International Conference on Information Quality. MIT, Cambridge, MA, 2006 [cited 2011 Oct 11]. Available from: citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.8811

37. Maydanchik A. Data profiling: myth and reality [Internet]. 1 Jul 2009 [cited 2011 21 Jun]. Available from http://www.dataqualitypro.com/?data_profiling_ark1

38. Power, Dan. The relationship between master data management and data quality [Internet]. 19 Aug 2008 [cited 2012 Mar18]. Available from http://www.information-management.com/news/10001823-1.html?zkPrintable=1&nopagination=1

39. Botsis T, Hartvigsen G, Chen F, Weng C. Secondary use of EHR: data quality issues and informatics opportunities. In AMIA Summits Transl Sci Proc. 2010; 2010: 1-5 [cited 2011 Oct10]. Available from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041534/?tool=pmcentrez