The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/2059-5891.htm

Metadata-based data quality assessment

Mustafa Aljumaili, Ramin Karim and Phillip Tretten
Division of Operation, Maintenance and Acoustics Engineering, Luleå University of Technology, Luleå, Sweden

Received 2 November 2015
Revised 10 December 2015
Accepted 22 December 2015

Abstract

Purpose – The purpose of this paper is to develop data quality (DQ) assessment model based on content analysis and metadata analysis.

Design/methodology/approach – A literature review of DQ assessment models has been conducted. A study of DQ key performances (KPIs) has been done. Finally, the proposed model has been developed and applied in a case study.

Findings – The results of this study shows that the metadata data have important information about DQ in a database and can be used to assess DQ to provide decision support for decision makers.

Originality/value – There is a lot of DQ assessment in the literature; however, metadata are not considered in these models. The model developed in this study is based on metadata in addition to the content analysis, to find a quantitative DQ assessment.

Keywords Metadata, Data quality, Attributes, eMaintenance, Database

Paper type Research paper

1. Introduction

Knowledge management systems (KMSs) rely ultimately on the timely and accurate retrieval of appropriate facts and information that come in many different forms. These forms are located in the enterprise data and have different structures and attributes such as reliability, accuracy and security (Pigott and Hobbs, 2011). High-quality data make organizational data resources more reliable, increasing the business benefits gained by using them. They contribute to efficient and effective business operations, improved decision-making and increased trust in information systems (DeLone and McLean, 1992; Redman and Blanton, 1997). Advances in information systems and technology permit organizations to collect large amounts of data and to build and manage complex data resources. However, the large size and complexity make data resources vulnerable to data defects that reduce their quality (Even and Shankaranarayanan, 2009).

Although there is no consensus on the distinction between *data quality* (DQ) and *information quality* (IQ), there is a tendency to use DQ to refer to technical issues and IQ to refer to non-technical issues (Zhu *et al.*, 2014). In this study, we do not make this distinction but use DQ to refer to the full range of issues.

DQ can be defined as the data that are fit to use by data consumers. The production of high-quality statistics depends on DQ. Without a systematic assessment of DQ, there is a risk of losing control of the various statistical processes such as data collection, editing or weighting. A lack of DQ assessment assumes processes cannot be improved and problems will always be detected without systematic analysis, but without good DQ assessment, statistical departments are working blind; they can neither claim being professional nor deliver quality results (Bergdahl *et al.*, 2007).

VINE Journal of Information and Knowledge Management Systems
Vol. 46 No. 2, 2016
pp. 232-250
© Emerald Group Publishing Limited
2059-5891
DOI 10.1108/VJIKMS-11-2015-0059

Quantitative assessment of quality is critical in large data environments, as it can help set up realistic quality improvement targets, track progress, assess impacts of different solutions and prioritize improvements.

However, DQ is typically assessed along multiple quality dimensions (Even and Shankaranarayanan, 2009), and these dimensions have to be considered in relation to specific user objectives, goals and functions in a specific context. Because all users, whether human or automatic processes, have different data and information requirements, the set of attributes and the level of quality considered satisfactory vary with the user's perspective, the type of the models, algorithms and processes comprising the system. Therefore, the general ontology designed to identify possible attributes and relations between them, especially in a human–machine integrated system, will require instantiation in every particular case (Rogova and Bosse, 2010).

The literature suggests several methods for assessing DQ; the proposed quality measurements often use a scale between 0 (poor) and 1 (perfect) (Wang *et al.*, 1995a, 1995b; Redman and Blanton, 1997; Pipino *et al.*, 2002). Some methods, referred to by Ballou and Pazer (2003) as structure-based or structural, are driven by physical characteristics of the data (e.g. item counts, time tags or defect rates). Such methods are *impartial*, as they assume an objective quality standard and disregard the context in which the data are used (Even and Shankaranarayanan, 2009). Other measurement methods, referred to as content-based (Ballou and Pazer, 2003), derive measurements from data content. Such measurements typically reflect the impact of quality defects within a specific usage context and are, therefore, also called contextual assessments (Pipino *et al.*, 2002).

IQ can be assessed on three levels: information content, information source and information system quality. Major attributes of the quality of information content are *accessibility*, *availability*, *relevance*, *timeliness* and *integrity*. Information sources can be subjective or objective. Subjective sources include human observers, experts and decision makers. Objective information sources include sensors, models and automated processes; these are free of the biases inherent to human judgment and depend only on how well sensors are calibrated (Rogova and Bosse, 2010).

Information systems should be well designed to ensure high-quality data. The database design includes tables and metadata.

Metadata are crucial for information systems, and the past 30 years has witnessed a tremendous growth in the use of metadata (Lee *et al.*, 2006). However, metadata are not yet used for DQ assessment. Therefore, this study proposes a methodology to assess quality, considering both content and database metadata. By merging and comparing the two, it seeks to improve the assessment of DQ and facilitate better decision-making.

2. Types of data

Data can be considered an asset. An asset is a useful item that is a product or byproduct of an application development process. An asset can be tangible, such as data, designs or software code; or intangible, such as knowledge and methodologies (Lee *et al.*, 2006). In general, three types of data should be considered when determining DQ: structured, unstructured and semi-structured data.

Fully structured data follows a predefined schema, conforming to certain specifications (Sint *et al.*, 2009). A typical example of fully structured data is a relational database system. Structured data are often managed using Structured Query Language

(SQL) – a programming language created specifically for managing and querying data in relational database management systems (RDBMS).

Unstructured data have no identifiable structure. These data cannot be stored in rows and columns in a relational database; examples include photos and graphic images, videos, streaming data and web pages. The advantage of unstructured data is that no additional effort is necessary to classify them. A limitation is that no controlled navigation is possible within unstructured content (Sint *et al.*, 2009). Therefore, DQ techniques become increasingly complex as data lose structure (Batini *et al.*, 2009).

Semi-structured data are a cross between the other two. They represent a type of structured data but lack the strict data model structure. They are often explained as schema-less or self-describing terms, with no separate description of the type or structure of the data. Semi-structured data do not require a schema definition. With semi-structured data, tags or other types of markers are used to identify certain elements within the data, but the data do not have a rigid structure. Therefore, XML and other mark-up languages are often used to manage semi-structured data. One of the strengths of semi-structured data is their ability to accommodate variations in structure; data may be created according to a specification or based on a type (Sint *et al.*, 2009).

The quality dimension has different metrics depending on the type of data (Batini *et al.*, 2009). In maintenance, computerized maintenance management systems deal with all types of data, but the large majority of contributions in the DQ literature focus on structured and semi-structured data (Batini *et al.*, 2009). This study considers structured data stored in RDBMSs. In RDBMS, the data content is described by metadata schema.

As noted above, in this study, data assessment is based on both content and metadata analysis.

3. Relational database system

Though database work has not traditionally focused on DQ management, many of the tools developed have relevance for managing DQ. For example, research has considered how to prevent data inconsistencies (integrity constraints and normalization theory) and how to prevent data corruption (transaction management) (Wang *et al.*, 1993). The most mature and widely used database systems in production today are RDBMSs (Hellerstein *et al.*, 2007). A relational database stores information about the data and how they are related. The concept was proposed by Edgar (Ted) Codd in 1970 at IBM (Date, 2003). Data and relationships are represented in a flat, two-dimensional table that preserves relational structuring; see Figure 1. Relational systems serve as the repositories behind nearly all online transactions and most online content management systems (blogs, wikis, social networks, etc.; Hellerstein *et al.*, 2007).

Features of modern relational systems include powerful query facilities, data and device independence, concurrency control and recovery. These are useful in applications such as engineering design, office automation and graphics (Haskin and Lorie, 1982). An RDBMS is the physical and logical implementation of a relational database (hardware and software). An RDBMS controls reading, writing, modifying and processing the information stored in the databases. The data are formally described and organized according to each database's relational model (database schema), based on the design.

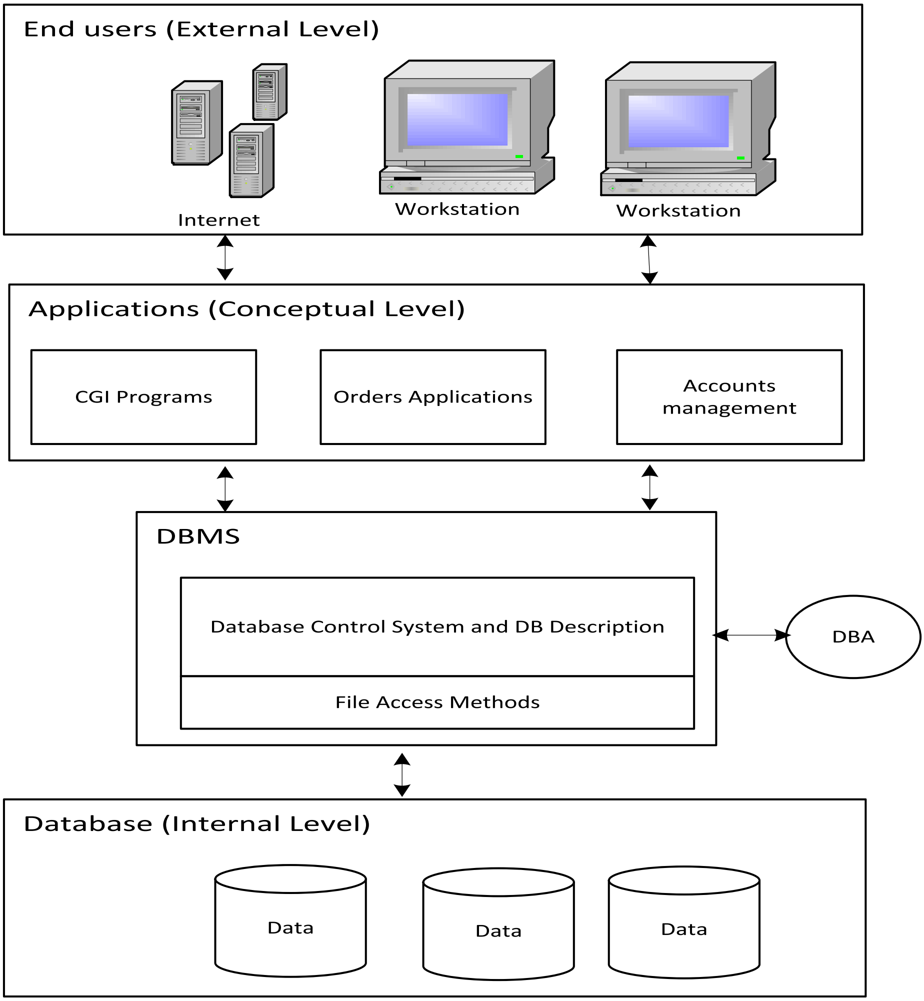

End users (External Level)
Internet
Workstation
Workstation
Applications (Conceptual Level)
CGI Programs
Orders Applications
Accounts management
DBMS
Database Control System and DB Description
File Access Methods
DBA
Database (Internal Level)
Data
Data
Data

Figure 1. RDBMS

4. Database schema and metadata

A database schema is a set of formulas called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real world entities are modeled in the database.

Metadata are considered a key success factor in data warehouse (DW) projects. They capture all kinds of information necessary to analyze, design, build, use and interpret the contents of the DW (Vetterli *et al.*, 2000). With the enormous increases in the storage capacity of rapid-retrieval data storage, the number now includes thousands, even tens of thousands of files. Clearly, additional information is needed simply to intelligently track, identify and use these files (Bedoll and Kimball, 1990).

Metadata are often called data about data or information about information. Specifically, metadata discover and know all that is necessary to know about the structures of a source data. At a minimum, the following are known (Aspin, 2012):

- Tables used;
- Columns available;
- Data types;
- Domain of data in columns; and
- Column nullability.

Certain sources also must be known:

- Primary and unique keys constraints;
- Indexes; and
- Triggers.

There are three main types of metadata:

(1) *Descriptive metadata* are used for discovery and identification. They can include elements such as title, abstract, author and keywords.

(2) *Structural metadata* indicate how objects are put together, for example, how pages are ordered to form chapters.

(3) *Administrative metadata* provide information to help manage a resource, such as when and how it was created, the file type and other technical information and who can access it.

With the advent of computers and the incessant need for data, new techniques have made it possible to store data permanently in secondary storage. These data can be retrieved and used by application programs. File managers are used to store and retrieve data from the secondary storage. To accomplish their job, file managers use such metadata as field names and filenames. This use of metadata, along with the actual data, is now ingrained in database management technology (Lee *et al.*, 2006).

5. Data quality assessment in literature

DQ is a key concern in any applications system area, for example, business support systems (transactions systems, enterprise resource planning (ERP), decision support systems, etc.), air traffic systems, transportation systems, military systems, energy management system and maintenance management systems. For such systems, DQ has a huge impact on decisions and their consequences; some research has focused on the impact of poor or insufficient DQ in applications (Sains and Teknologi, 2012). The literature provides a wide range of techniques to assess and improve the quality of data, including record linkage, business rules and similarity measures. Over time, these techniques have evolved to cope with the increasing complexity of DQ in networked information systems (Batini *et al.*, 2009).

DQ assessment methods are generally based on measurement theory. Each dimension of DQ consists of a set of attributes. Each attribute characterizes a specific

DQ requirement, thereby defining the standard for DQ assessment. There is flexibility in the methods used to measure DQ, as each attribute can be measured by several different methods (Chen *et al.*, 2014). Early DQ research focused on developing techniques to query multiple data sources and build large DWs. According to International Organization for Standardization/International Electrotechnical Commission 15,939, measurements are "a set of operations having the object of determining a value of a quantitative or categorical representation of one or more attributes". In addition, "measurements should have a clearly defined purpose". The purpose for measuring the DQ of a given scenario is to satisfy an "information need" to manage objectives, goals, risks and problems (Caballero *et al.*, 2007).

Wang and Madnick (1989) used a systematic approach to study related DQ concerns. They identified and addressed entity resolution issues that arose when integrating information from multiple sources with overlapping records. They explored ways to determine whether separate records actually correspond to the same entity. This is now known as record linkage, record matching and, more broadly, data integration and information integration (Wang and Madnick, 1989).

Later, Wang and Madnick (1990) developed a polygen (poly for multiple, gen for source) model to consider the processing of data source tags in the query processor to answer DQ-related questions such as "Where are these data from?" and "Which intermediary data sources were used to arrive at these data?" (Wang and Madnick, 1990). Follow-up research included the development of a modeling method (quality entity relationship model) to systematically capture comprehensive DQ criteria as metadata in the conceptual database design phase (Wang *et al.*, 1993), using an extended relational algebra to allow the query processor to process hierarchical DQ metadata (Wang *et al.*, 1995a, 1995b).

This stream of research has had impacts on modern database research and design, such as data provenance and data lineage (Buneman *et al.*, 2001) and on extensions to relational algebra for data security and data privacy management. More importantly, early research efforts motivated researchers to embark on systematic inquiry into the whole spectrum of DQ issues, which, in turn, led to the inauguration of the MIT Total Data Quality Management (TDQM) program in the early 1990s and the later creation of the MIT Information Quality Program (Zhu *et al.*, 2014).

TQDM has been introduced as a guideline for DQ analysis in information systems, with four main categories of DQ dimensions: intrinsic, accessible, contextual and representational. Each dimension contains several DQ matrices. The intrinsic dimension refers to the fact that information has qualities in its own right. Contextual means the IQ must be considered within the context of the task at hand. Accessible and representational dimensions emphasize the important roles of information systems. Figure 2 shows TQDM for information products (Sains and Teknologi, 2012).

Scannapieco *et al.* (2004) presented an architecture for managing DQ in cooperative information systems by focusing on two specific modules, the DQ Broker and the Quality Notification Service. The former allows querying and improving DQ values. The latter is specifically targeted at the dissemination of changes in DQ values. The investigation of DQ by the Cooperative Information System (DaQuinCIS) project started in 2001. The project involves three universities in Italy, Universita Di Roma, Polictechnico Di Milano and Universita Di Milano, with more than 20 professors,

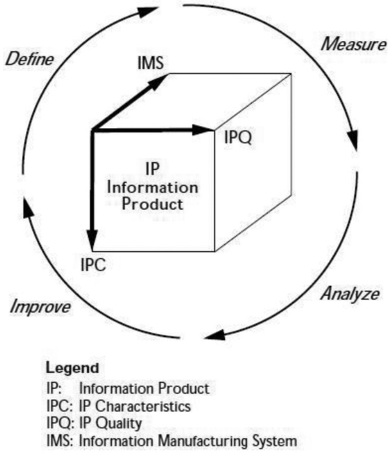


Figure 2.
Schematic of the TDQM methodology

Source: Sains and Teknologi (2012)

doctoral students, technicians and researchers working together on it. Figure 3 shows the DaQuinCIS framework (Scannapieco *et al.*, 2004).

Jeusfeld *et al.* (1998) suggested an approach to assess the DQ of a DW via a semantically rich model of quantity management in DW. A DW relies on meta-databases to control its operation and aid its evolution because of the dynamic changes in DW requirements and environment. The model allows stakeholders to design abstract quality goals that are translated to executable analytic queries of quality measurements in the DW's meta-database. Figure 4 shows the quality meta model in the DW architecture (Jeusfeld *et al.*, 1998).

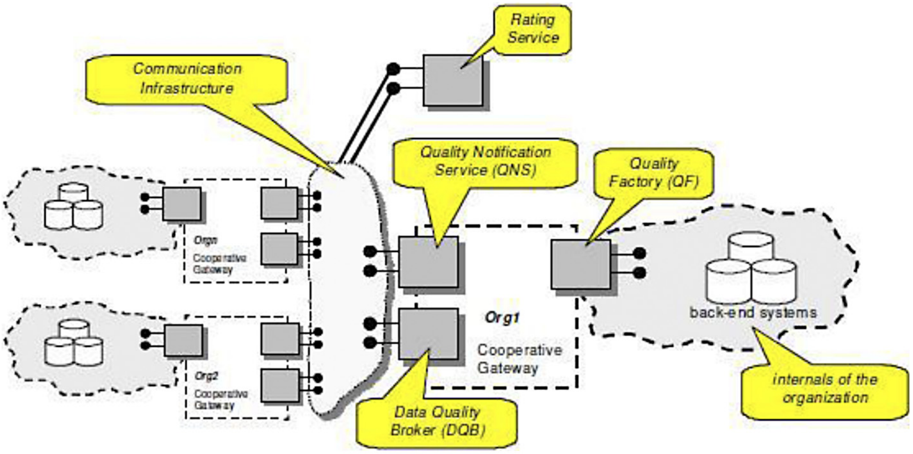


Figure 3.
DaQuinCIS framework

Source: Scannapieco *et al.* (2004)

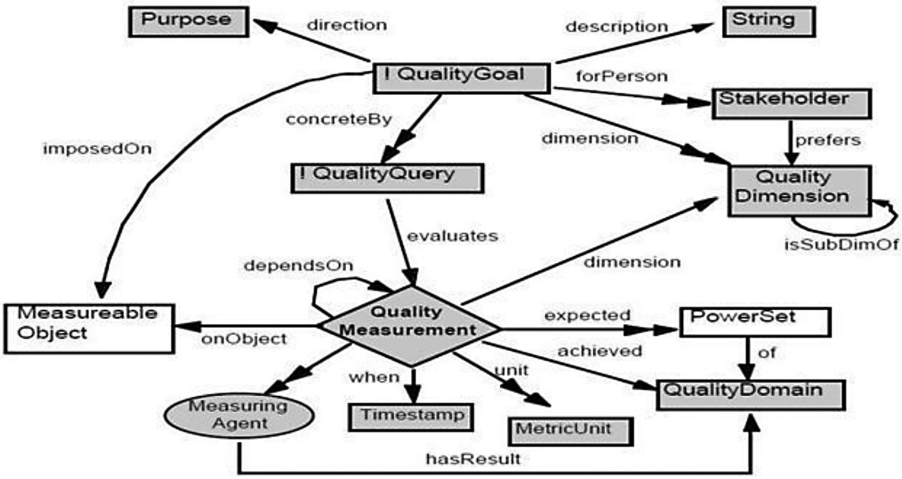


Figure 4.
Quality meta model

Lee *et al.* (2002) suggested a model for basic IQ assessment and benchmarking known as AIM Quality. It encompasses a model for IQ, a questionnaire to measure IQ and an analytic technique to interpret IQ. Other research in computer science has considered specific areas, such as database technical solutions, DW and data integration, enterprise architecture, networks and performance. As an example, Davidson and Tayi combined data mining techniques with DQ matrices. The basic idea was that most data mining research focuses on discovering patterns in organizational databases without considering the DQ knowledge of the databases. Their research developed a general purpose method of incorporating DQ matrices into the data mining classification task by looking for accuracy, contextual, semantic interpretability and database quality matrix (Ballou and Tayi, 1999). Zhang *et al.* (2009) conducted research to improve the quality of DW data by emphasizing data structure correctness, data consistency, integrity and data atomicity. Madnick *et al.* produced a paper on semantic issues in DQ. Many DQ problems are a result of data misinterpretation, that is, problems caused by heterogeneous data semantics. The authors suggested context interchange technology can be used to capture data semantics and reconcile semantic heterogeneities, thereby improving DQ (Madnick and Zhu, 2006).

6. Data quality attributes

The DQ literature provides a thorough classification of DQ dimensions; the most basic set of dimensions includes *accuracy*, *completeness*, *consistency* and *timeliness* (Batini *et al.*, 2009). DQ dimensions can be measured using both qualitative (subjective) survey evaluations and quantitative (objective) metrics. In either case, the result of the measurement is the data value.

According to the DQ literature, typical DQ measurement methods to determine data values implement a formula like the following (Lee *et al.*, 2006) (Caballero *et al.*, 2007):

$$Ratio = 1 - [Number\ of\ undesirable\ outcomes/Total\ outcomes] \qquad (1)$$

The measure is composed of two base measures, *number of undesirable outcomes* and *total outcomes*, with 1 representing the most desirable and 0 the least desirable score. In this case, the measurement method is objective: it simply consists of counting the number of data units accomplishing the criterion (Caballero *et al.*, 2007).

The overall score of DQ is found using a weighted summation. The values of n single variable metrics are aggregated as follows (Lee *et al.*, 2006):

$$DQ = \sum ni = 1\,(ai.Mi) \qquad (2)$$

Where ai is a weighting factor, $0 \leq ai \leq 1$, a1 + a2 + […] + an =1, and Mi is a normalized value of the assessments of the i-th attribute.

In this study, DQ attributes metrics are based on the ratio in equation (1). These attributes metrics are discussed in the next sections.

7. Data quality assessment model

Measurement is a key activity in DQ management. As noted above, the purpose for measuring DQ is to satisfy an "information need" to manage objectives, goals, risks and problems. Knowing the information needs and context, a plan can be drawn to determine the following:

(a) the measure to be made;

(b) where the objects to measure are;

(c) how to measure these objects;

(d) how many objects must be inspected to have statistically significant evidence;

(e) whose objects these are;

(f) to whom results must be delivered; and

(g) when measures can be done, so as to not interfere in any other process (Caballero *et al.*, 2007).

Although DQ literature contains numerous measurement proposals, many research challenges remain. This study proposes a model for overall DQ assessment. The model merges subjective assessment with objective assessment; see Figure 5. Defining DQ metrics is crucial for the objective assessment process, and the following sections discuss these metrics. The subjective assessment of DQ includes user surveys, focusing on such attributes as usability and believability. The proposed subjective model is discussed in a separate study.

This paper limits itself to the objective assessment model. As noted, the quantitative assessment merges metadata and content. The flowchart of the proposed model in Figure 6 shows the assessment process.

7.1 Metadata analysis to measure data consistency and accuracy

Information technology applications enable firms to have a simple selection and internalization process of their knowledge. KMS can be used not to manage all the existing knowledge inside the organization, but to manage that knowledge needed by people within the organization, which could help them in achieving their expected benefits (Cricelli *et al.*, 2014). Metadata are part of KMS as structured information that

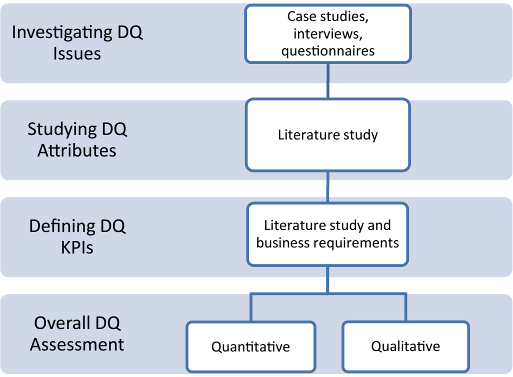


Figure 5. Overall DQ assessment model

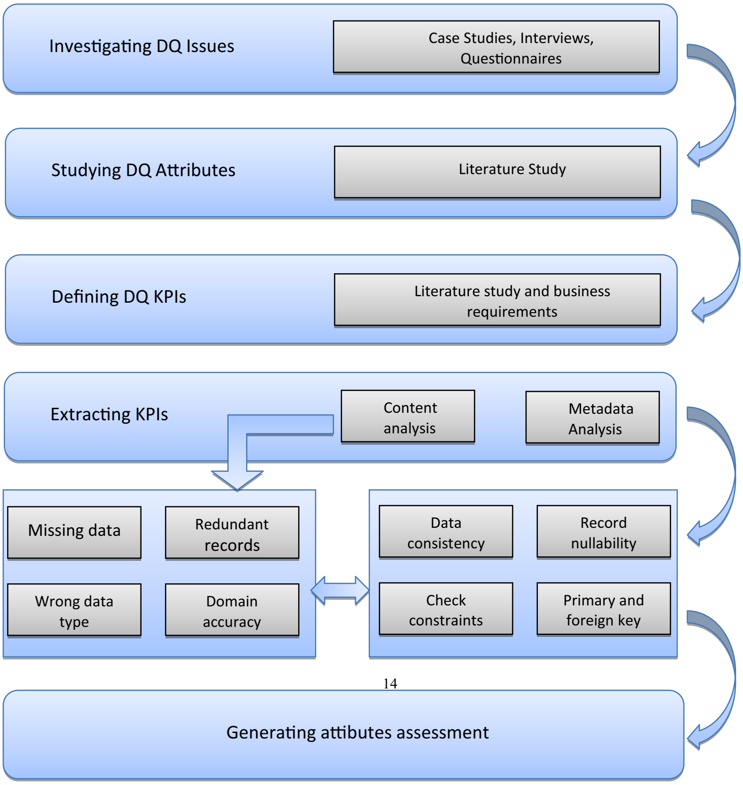


Figure 6. Quantitative DQ assessment model

describe, explain, locate or otherwise make it easier to retrieve, use or manage an information resource. Metadata are often called data about data or information about information (Brand *et al.*, 2003). In a database, metadata record the names of basic entities in the system (users, schemas, tables, columns, indexes, etc.) and their relationships and is stored as a set of tables in the database. By keeping the metadata in the same format as the data, the system is made more compact and simpler to use: users can employ the same language and tools to investigate the metadata that they use for other purposes (Hellerstein *et al.*, 2007).

In relational DBMS, metadata are extensively used to define data. These metadata include relation names, attribute names, key and domain information (Lee *et al.*, 2006). Therefore, metadata contain constraints that control data integrity, consistency, accuracy and completeness. Common kinds of constraints are as follows:

- *not null* – value in a column must not be null;
- *unique* – value(s) in specified column(s) must be unique for each row in a table;
- *primary key* – value(s) in specified column(s) must be unique for each row in a table and not be null; normally, each table in a database should have a primary key used to identify individual records;
- *foreign key* – value(s) in specified column(s) must reference an existing record in another table (may be primary key or some other unique constraint); and
- *check* – an expression is specified which be true for a constraint to be satisfied.

7.1.1 Primary key check. A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A primary key's main features are the following: it must contain a unique value for each row of data; it cannot contain null values; and it is either an existing table column or a column specifically generated by the database according to a defined sequence. Having a primary key will ensure uniqueness and prevent data redundancy. Hence, any table without a primary key will have problems when selecting, joining, linking, etc.

7.1.2 Foreign key check. A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them; see Figure 7.

A foreign key constraint does not have to be linked to a primary key constraint in another table; it can also be defined to reference the columns of a unique constraint in another table. A foreign key constraint can contain null values; however, if any column of a composite foreign key constraint contains null values, verification of the foreign key constraint is skipped.

7.1.3 Check constraint. This constraint controls data accuracy and consistency. The relational theory distinguishes two fundamental categories of integrity constraints: *intra-relation constraints* and *inter-relation constraints*. Interrelation constraints define the range of admissible values for an attribute's domain. Examples are "Age must range

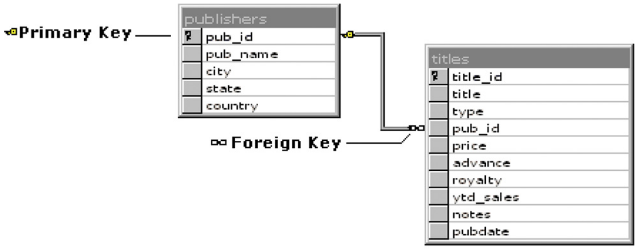


Figure 7.
Foreign key constraint

between 0 and 120” or “If Working Years is lower than 3, then Salary cannot be higher than 25.000 Euros per year” (Batini *et al.*, 2009). Check constraints enforce domain integrity by limiting the values accepted by one or more columns. A check constraint can be created with any logical (Boolean) expression that returns true or false. For example, the range of values for a salary column can be limited by creating a check constraint that allows only data that range from $15,000 to $100,000. This prevents salaries from being entered beyond the regular salary range. The logical expression would be the following: salary $\geq$ 15,000 AND salary $\leq$ 100,000.

We can apply multiple check constraints to a single column or apply a single check constraint to multiple columns by creating it at the table level. For example, a multiple-column check constraint can be used to confirm that any row with a country/region column value of USA also has a two-character value in the state column. This allows multiple conditions to be checked in one location.

Check constraints are similar to foreign key constraints in that they control the values put in a column. The difference is in how they determine which values are valid: foreign key constraints obtain the list of valid values from another table, and check constraints determine the valid values from a logical expression.

7.1.4 Nullability. This metadata constraint relates to the data completeness attribute. In the research area of relational databases, completeness is often related to the meaning of null values. A null value has the general meaning of a missing value, a value that exists in the real world but is not available in a data collection. Null is frequently used to represent a missing value or invalid value, for example, from a function that failed to return or a missing field in a database, as in null in SQL. To characterize completeness, it is important to understand why the value is missing. A value can be missing because it exists but is not known, or because it does not exist, or because it is not known whether it exists (Atzeni and De Antonellis, 1993). In database systems, this constraint is considered during the design of the database. For example, RDBMS contains this constraint in the metadata to determine if table columns are allowed to have null values or not (Batini *et al.*, 2009).

7.2 Content analysis to measure completeness, consistency and accuracy

This study conducts several types of database analysis. These are described below.

7.2.1 Data type checking. The data type check is used to ensure all data values follow the datatype property defined during the database design. Datatype property is registered in the metadata records. In this check, the number of fields violating the datatype property is counted:

$$Data\ \ Type = 1 - (no.\,of\,items\,violating\,Data\,Type/total\,no\,of\,items)$$

This attribute check is used to assess data accuracy.

7.2.2 Domain range checking. This can be considered an extension to the data accuracy attribute defined in Section 7.2.5. As an extension of simple validation, range checking ensures a value is within allowable minimums and maximums. Values should be specified during the database design via check constraints. This attribute check is also used in the data accuracy assessment.

7.2.3 Completeness checking. Completeness is defined as the degree to which a given data collection includes data describing a corresponding set of real-world objects (Batini *et al.*, 2009). It is the degree to which all data relevant to an application domain have been recorded in an information system. It can be also considered the degree to which expected values are present in a data collection. When an incomplete value represents an unknown or missing value in the real world, or it represents a value yet to be entered into a database, a value of null is used (Lee *et al.*, 2006):

$$\textit{Completeness} = 1 - (\textit{No. of incomplete items/total no of items})$$

7.2.4 Redundancy checking. This check relates to the data consistency attribute. Duplicate records are a common problem in information systems. They are created by mistake, simply because the user is not aware that the record exists already, or because of system limitations. Common problems include systems that cannot store different information for the same vendor; consequently, the vendor is created multiple times as a workaround. Duplicate records for the same customer or vendor will result in incorrect reporting and directly affect a company's business:

$$\textit{Redundancy} = 1 - (\textit{No. of redundant items/total no. of items})$$

This attribute check is used in the data consistency assessment.

7.2.5 Accuracy. Accuracy denotes the extent to which data are correct, reliable and certified (Wang and Strong, 1996). Ballou and Pazer (1985) specify that data are accurate when the data values stored in the database correspond to real-world values (Ballou and Pazer, 1985). The dimension of accuracy itself, however, can consist of one or more variables, only one of which is whether the data are correct (Lee *et al.*, 2006). To count the number of data units in error, the metric is as follows:

$$\textit{Accuracy} = 1 - (\textit{no. of items in error/total no of item})$$

An item could be a file or a record. In this study, the accuracy assessment depends on two evaluations: data-type accuracy and domain accuracy values.

7.3 Developed analysis tool

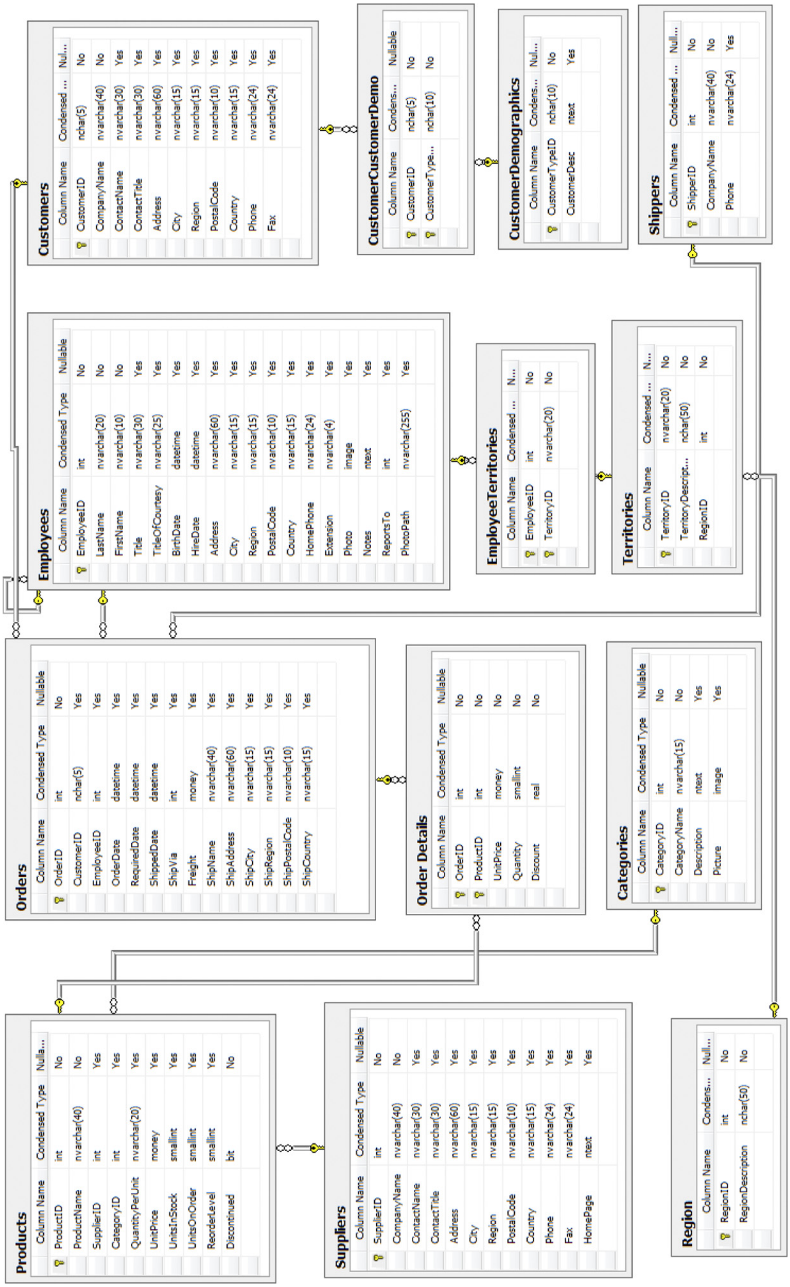
To validate the proposed measures, we use the Northwind database, a sample database that comes with Microsoft Access. Basically, the database is for a fictitious company named "Northwind Traders". A diagram of the database is shown in Figure 8.

The database captures all sales transactions between the company and its customers as well as the purchase transactions between Northwind and its suppliers. The developed tool investigates the DQ attributes discussed in the previous sections. Again, the study has two main analytic categories: metadata and content analysis.

The metadata analysis investigates general data representation to determine whether there are data constraints, and if so, how they were designed. It checks for constraints such as primary keys, foreign keys and check keys. DQ dimensions of integrity and accuracy are included in this investigation as well.

Database content analysis considers other DQ dimensions, including completeness, value accuracy, value data type (if they follow the column data constraint listed in the metadata), data domain values (if they follow the check constraint listed in the metadata).

The interface of the developed tool is shown in Figure 9.



Customers

Column Name	Condensed ...	Nul...
CustomerID	nchar(5)	No
CompanyName	nvarchar(40)	No
ContactName	nvarchar(30)	Yes
ContactTitle	nvarchar(30)	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
Phone	nvarchar(24)	Yes
Fax	nvarchar(24)	Yes

CustomerCustomerDemo

Column Name	Condens...	Nullable
CustomerID	nchar(5)	No
CustomerType...	nchar(10)	No

CustomerDemographics

Column Name	Condens...	Nul...
CustomerTypeID	nchar(10)	No
CustomerDesc	ntext	Yes

Shippers

Column Name	Condensed ...	Nul...
ShipperID	int	No
CompanyName	nvarchar(40)	No
Phone	nvarchar(24)	Yes

Employees

Column Name	Condensed Type	Nullable
EmployeeID	int	No
LastName	nvarchar(20)	No
FirstName	nvarchar(10)	No
Title	nvarchar(30)	Yes
TitleOfCourtesy	nvarchar(25)	Yes
BirthDate	datetime	Yes
HireDate	datetime	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
HomePhone	nvarchar(24)	Yes
Extension	nvarchar(4)	Yes
Photo	image	Yes
Notes	ntext	Yes
ReportsTo	int	Yes
PhotoPath	nvarchar(255)	Yes

EmployeeTerritories

Column Name	Condensed ...	N...
EmployeeID	int	No
TerritoryID	nvarchar(20)	No

Territories

Column Name	Condensed ...	N...
TerritoryID	nvarchar(20)	No
TerritoryDescript...	nchar(50)	No
RegionID	int	No

Orders

Column Name	Condensed Type	Nullable
OrderID	int	No
CustomerID	nchar(5)	Yes
EmployeeID	int	Yes
OrderDate	datetime	Yes
RequiredDate	datetime	Yes
ShippedDate	datetime	Yes
ShipVia	int	Yes
Freight	money	Yes
ShipName	nvarchar(40)	Yes
ShipAddress	nvarchar(60)	Yes
ShipCity	nvarchar(15)	Yes
ShipRegion	nvarchar(15)	Yes
ShipPostalCode	nvarchar(10)	Yes
ShipCountry	nvarchar(15)	Yes

Order Details

Column Name	Condensed Type	Nullable
OrderID	int	No
ProductID	int	No
UnitPrice	money	No
Quantity	smallint	No
Discount	real	No

Categories

Column Name	Condensed Type	Nullable
CategoryID	int	No
CategoryName	nvarchar(15)	No
Description	ntext	Yes
Picture	image	Yes

Products

Column Name	Condensed Type	Nulla...
ProductID	int	No
ProductName	nvarchar(40)	No
SupplierID	int	Yes
CategoryID	int	Yes
QuantityPerUnit	nvarchar(20)	Yes
UnitPrice	money	Yes
UnitsInStock	smallint	Yes
UnitsOnOrder	smallint	Yes
ReorderLevel	smallint	Yes
Discontinued	bit	No

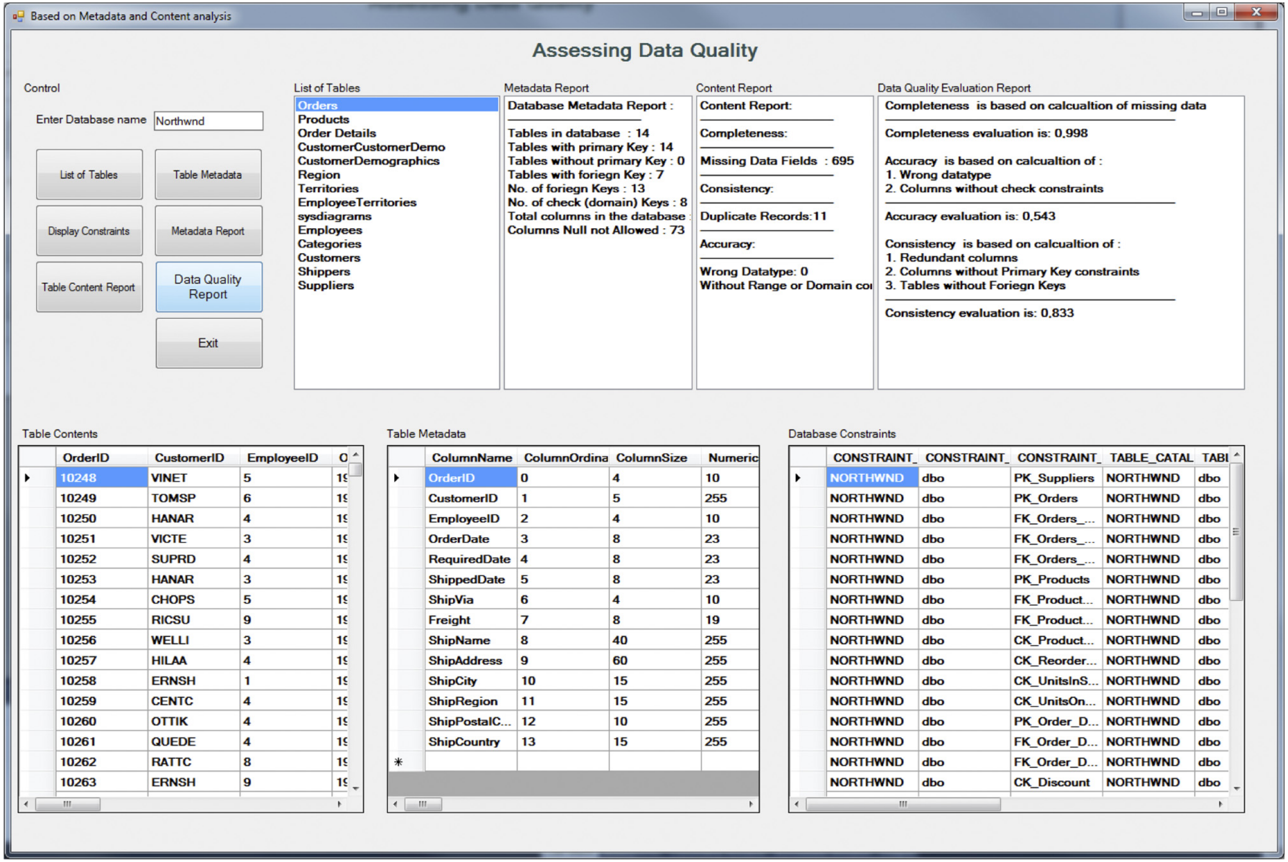
Suppliers

Column Name	Condensed Type	Nullable
SupplierID	int	No
CompanyName	nvarchar(40)	No
ContactName	nvarchar(30)	Yes
ContactTitle	nvarchar(30)	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
Phone	nvarchar(24)	Yes
Fax	nvarchar(24)	Yes
HomePage	ntext	Yes

Region

Column Name	Condens...	Nul...
RegionID	int	No
RegionDescription	nchar(50)	No

Figure 8.
Northwind database structure



Assessing Data Quality

Enter Database name: Northwnd

List of Tables

Orders
Products
Order Details
CustomerCustomerDemo
CustomerDemographics
Region
Territories
EmployeeTerritories
sysdiagrams
Employees
Categories
Customers
Shippers
Suppliers

Metadata Report

Database Metadata Report :

Tables in database : 14
Tables with primary Key : 14
Tables without primary Key : 0
Tables with foriegn Key : 7
No. of foriegn Keys : 13
No. of check (domain) Keys : 8
Total columns in the database
Columns Null not Allowed : 73

Content Report

Content Report:

Completeness:

Missing Data Fields : 695

Consistency:

Duplicate Records:11

Accuracy:

Wrong Datatype: 0
Without Range or Domain co

Data Quality Evaluation Report

Completeness is based on calcualtion of missing data

Completeness evaluation is: 0,998

Accuracy is based on calcualtion of :
1. Wrong datatype
2. Columns without check constraints

Accuracy evaluation is: 0,543

Consistency is based on calcualtion of :
1. Redundant columns
2. Columns without Primary Key constraints
3. Tables without Foriegn Keys

Consistency evaluation is: 0,833

Table Contents

OrderID	CustomerID	EmployeeID	O
10248	VINET	5	19
10249	TOMSP	6	19
10250	HANAR	4	19
10251	VICTE	3	19
10252	SUPRD	4	19
10253	HANAR	3	19
10254	CHOPS	5	19
10255	RICSU	9	19
10256	WELLI	3	19
10257	HILAA	4	19
10258	ERNSH	1	19
10259	CENTC	4	19
10260	OTTIK	4	19
10261	QUEDE	4	19
10262	RATTC	8	19
10263	ERNSH	9	19

Table Metadata

ColumnName	ColumnOrdina	ColumnSize	Numeric
OrderID	0	4	10
CustomerID	1	5	255
EmployeeID	2	4	10
OrderDate	3	8	23
RequiredDate	4	8	23
ShippedDate	5	8	23
ShipVia	6	4	10
Freight	7	8	19
ShipName	8	40	255
ShipAddress	9	60	255
ShipCity	10	15	255
ShipRegion	11	15	255
ShipPostalC...	12	10	255
ShipCountry	13	15	255

Database Constraints

CONSTRAINT_	CONSTRAINT_	CONSTRAINT_	TABLE_CATAL	TABI
NORTHWND	dbo	PK_Suppliers	NORTHWND	dbo
NORTHWND	dbo	PK_Orders	NORTHWND	dbo
NORTHWND	dbo	FK_Orders_...	NORTHWND	dbo
NORTHWND	dbo	FK_Orders_...	NORTHWND	dbo
NORTHWND	dbo	FK_Orders_...	NORTHWND	dbo
NORTHWND	dbo	PK_Products	NORTHWND	dbo
NORTHWND	dbo	FK_Product...	NORTHWND	dbo
NORTHWND	dbo	FK_Product...	NORTHWND	dbo
NORTHWND	dbo	CK_Product...	NORTHWND	dbo
NORTHWND	dbo	CK_Reorder...	NORTHWND	dbo
NORTHWND	dbo	CK_UnitsInS...	NORTHWND	dbo
NORTHWND	dbo	CK_UnitsOn...	NORTHWND	dbo
NORTHWND	dbo	PK_Order_D...	NORTHWND	dbo
NORTHWND	dbo	FK_Order_D...	NORTHWND	dbo
NORTHWND	dbo	FK_Order_D...	NORTHWND	dbo
NORTHWND	dbo	CK_Discount	NORTHWND	dbo

Figure 9.
Developed DQ analysis tool

This tool is developed using Visual C# and can be applied to any SQL server database connected to visual studio. The user needs to enter the database name; after that, the desired functions are included, as shown in Figure 9, displaying the table's metadata, content, database constraints, reports about database metadata and content investigations.

After generating DQ KPIs using the developed tool which deploy the proposed model, DQ attributes assessment can be calculated. Figure 10 shows an example of assessment attributes such as completeness, consistency and accuracy of Northwind database.

8. Results and discussion

In this study, we suggest that metadata can be used to help in IQ assessment, but this assessment needs to be combined with database content assessment. We test and empirically validate the hypothesized model using data called Northwind.

Our study makes two key contributions to DQ research. First, it provides a link between metadata quality and IQ because metadata is the registry for the quality information of any database system. Second, it contributes to the DQ assessment model by presenting metadata quality as a key aspect of information system quality because the assessment models in literature consider content analysis and expert evaluations in general. The proposed model improves the assessment process by analyzing metadata and generating quality KPIs.

Figure 6 shows the proposed model with all steps that describes the applied scenario and the relationship between the metadata and overall IQ. Assessing the quality of metadata helps assess DQ attributes such as integrity and accuracy. Overall, the findings show a significant direct impact of metadata quality on IQ and on the quality of decisions made based on these data.

As most of the decisions are based on data, decisions based on dirty data may lead to negative impacts to any organization. These negative impacts can be related to financial, safety, satisfaction productivity and decision-making process as well as by providing delayed and wrong decisions. Having an estimation of the quality of database can help the decision makers to know the current states of their data, and hence, the quality of the decision they may take.

The developed software tool provides an overview of metadata quality. For optimum DQ assessment, however, the metadata and content analysis explained here should be combined with user assessment (explained in the previous study).

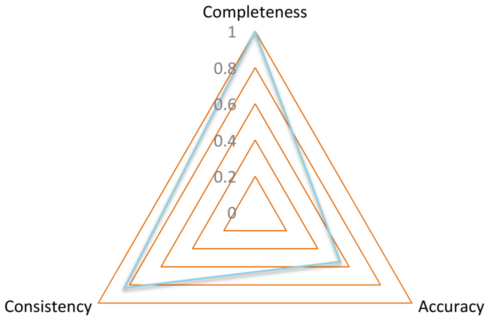


Figure 10.
DQ assessment based on proposed model

9. Conclusions

The modified DQ assessment model presented here extends previous models, resolving ambiguities in terminology and relationships of quality attributes. It merges the use of metadata with content analysis to objectively assess DQ. Previous DQ models are based on user evaluation of DQ, with some DQ attributes such as missing data, data accuracy; others are calculated based on the content only. This model uses both content and metadata to assess DQ. However, this assessment should be combined with subjective assessment to find the overall DQ.

The key contribution of the research is to integrate metadata, content and user satisfaction. The proposed methodology provides a practical IQ tool for organizations to identify IQ problems, prioritize areas for IQ improvement and monitor IQ improvements over time.

References

Aspin, A. (2012), “SQL server sources”, *SQL Server 2012 Data Integration Recipes*, Springer, New York City, pp. 241-283.

Atzeni, P. and De Antonellis, V. (1993), *Relational Database Theory*, Benjamin-Cummings Publishing, Redwood City, CA.

Ballou, D.P. and Pazer, H.L. (1985), “Modeling data and process quality in multi-input, multi-output information systems”, *Management Science*, Vol. 31 No. 2, pp. 150-162.

Ballou, D.P. and Pazer, H.L. (2003), “Modeling completeness versus consistency tradeoffs in information decision contexts”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15 No. 1, pp. 240-243.

Ballou, D.P. and Tayi, G.K. (1999), “Enhancing data quality in data warehouse environments”, *Communications of the ACM*, Vol. 42 No. 1, pp. 73-78.

Batini, C., Cappiello, C., Francalanci, C. and Maurino, A. (2009), “Methodologies for data quality assessment and improvement”, *ACM Computing Surveys (CSUR)*, Vol. 41 No. 3, p. 16.

Bedoll, R. and Kimball, C. (1990), “The importance of metadata in mass-storage systems”, *Digest of Papers: Tenth IEEE Symposium on Mass Storage Systems, 1990: crisis in Mass Storage*, Monterey, CA, pp. 111-116.

Bergdahl, M., Ehling, M., Elvers, E., Földesi, E., Körner, T., Kron, A. and Nimmergut, A. (2007), “Handbook on data quality assessment methods and tools”, *Handbook on Data Quality Assessment Methods and Tools*, Wiesbaden, pp. 9-10.

Brand, A., Daly, F. and Meyers, B. (2003), *Metadata Demystified: A Guide for Publishers*, Sheridan Press and Niso Press, Baltimore.

Buneman, P., Khanna, S. and Wang-Chiew, T. (2001), “Why and where: a characterization of data provenance”, *Database Theory: ICDT 2001*, Springer, Heidelberg, pp. 316-330.

Caballero, I., Verbo, E., Calero, C. and Piattini, M. (2007), “A data quality measurement information model based on ISO/IEC 15939”, *ICIQ*, Cambridge, MA, pp. 393-408.

Chen, H., Hailey, D., Wang, N. and Yu, P. (2014), “A review of data quality assessment methods for public health information systems”, *International Journal of Environmental Research and Public Health*, Vol. 11 No. 5, pp. 5170-5207.

Cricelli, L., Grimaldi, M. and Hanandi, M. (2014), “Decision making in choosing information systems: an empirical study in Jordan”, *VINE: The Journal of Information and Knowledge Management Systems*, Vol. 44 No. 2, pp. 162-184.

Date, C.J. (2003), “Edgar F. codd”, *Sigmod Record*, Vol. 32 No. 4, p. 4.

DeLone, W.H. and McLean, E.R. (1992), "Information systems success: the quest for the dependent variable", *Information Systems Research*, Vol. 3 No. 1, pp. 60-95.

Even, A. and Shankaranarayanan, G. (2009), "Dual assessment of data quality in customer databases", *Journal of Data and Information Quality (JDIQ)*, Vol. 1 No. 3, p. 15.

Haskin, R.L. and Lorie, R.A. (1982), "On extending the functions of a relational database system", *Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data, New York, NY*, pp. 207-212.

Hellerstein, J.M., Stonebraker, M. and Hamilton, J. (2007), *Architecture of a Database System*, Now Publishers, Boston.

Jeusfeld, M.A., Quix, C. and Jarke, M. (1998), "Design and analysis of quality information for data warehouses", *Conceptual Modeling–ER'98*, Springer, pp. 349-362.

Lee, Y.W., Pipino, L.L., Funk, J.D. and Wang, R.Y. (2006), *Journey to Data Quality*, MIT Press, Cambridge, MA.

Lee, Y.W., Strong, D.M., Kahn, B.K. and Wang, R.Y. (2002), "AIMQ: a methodology for information quality assessment", *Information & Management*, Vol. 40 No. 2, pp. 133-146.

Madnick, S. and Zhu, H. (2006), "Improving data quality through effective use of data semantics", *Data & Knowledge Engineering*, Vol. 59 No. 2, pp. 460-475.

Pigott, D.J. and Hobbs, V.J. (2011), "Complex knowledge modelling with functional entity relationship diagrams", *Vine*, Vol. 41 No. 2, pp. 192-211.

Pipino, L.L., Lee, Y.W. and Wang, R.Y. (2002), "Data quality assessment", *Communications of the ACM*, Vol. 45 No. 4, pp. 211-218.

Redman, T.C. and Blanton, A. (1997), *Data Quality for the Information Age*, Artech House, Norwood.

Rogova, G.L. and Bosse, E. (2010), "Information quality in information fusion", paper presented at the *13th Conference on Information Fusion (FUSION)*, Edinburgh, pp. 1-8.

Sains, F. and Teknologi, U.M.T. (2012), "Data investigation: issues of data quality and implementing base analysis technique to evaluate quality of data in heterogeneous databases", *Journal of Theoretical and Applied Information Technology*, Vol. 45 No. 1.

Scannapieco, M., Virgillito, A., Marchetti, C., Mecella, M. and Baldoni, R. (2004), "The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems", *Information Systems*, Vol. 29 No. 7, pp. 551-582.

Sint, R., Schaffert, S., Stroka, S. and Ferstl, R. (2009), "Combining unstructured, fully structured and semi-structured information in semantic wikis", *Fourth Workshop on Semantic Wikis–The Semantic Wiki Web 6 The European Semantic Web Conference Hersonissos, Crete, Greece, June 2009*, p. 73.

Vetterli, T., Vaduva, A. and Staudt, M. (2000), "Metadata standards for data warehousing: open information model vs common warehouse metadata", *ACM Sigmod Record*, Vol. 29 No. 3, pp. 68-75.

Wang, R.Y., Kon, H.B. and Madnick, S.E. (1993), "Data quality requirements analysis and modeling", paper presented at the Proceedings of Ninth International Conference on Data Engineering, pp. 670-677.

Wang, R.Y., Storey, V.C. and Firth, C.P. (1995a), "A framework for analysis of data quality research", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7 No. 4, pp. 623-640.

Wang, R.Y., Reddy, M.P. and Kon, H.B. (1995b), "Toward quality data: an attribute-based approach", *Decision Support Systems*, Vol. 13 No. 3, pp. 349-372.

Wang, R.Y. and Strong, D.M. (1996), "Beyond accuracy: what data quality means to data consumers", *Journal of Management Information Systems*, Vol. 12 No. 4, pp. 5-33.

Wang, Y.R. and Madnick, S.E. (1989), "The inter-database instance identification problem in integrating autonomous systems", *Proceedings of Fifth International Conference on Data Engineering, 1989*, *Los Angeles, CA*, pp. 46-55.

Wang, Y.R. and Madnick, S.E. (1990), "A polygen model for heterogeneous database systems: the source tagging perspective", *Proceedings of the Sixteenth International Conference on Very Large Database*, *Brisbane, Queensland*, pp. 519-538.

Zhang, J., Wen, Q. and Zhang, H. (2009), "The research in improving the quality of DW data: the job-scheduling and checking based program in upgrading DW performance", *WiCom'09 Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, *Beijing*, pp. 1-4.

Zhu, H., Madnick, S.E., Lee, Y.W. and Wang, R.Y. (2014), "Data and information quality research: its evolution and future", *Data and Information Quality Research*, 1st ed., Taylor & Francis Group, LLC, Abingdon.

Corresponding author
Mustafa Aljumaili can be contacted at: mustafa.aljumaili@ltu.se