Volume 6, Issue 4, April 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Study over Importance of Data Cleansing in Data Warehouse

Shivangi Rana
Research Scholar
CSE Department
Abhilashi Group of Institutions
(School of Engg.)
Chail Chowk, Mandi, India

Er. Gagan Prakesh Negi
Assistant Professor
CSE Department
Abhilashi Group of Institutions
(School of Engg.)
Chail Chowk, Mandi, India

Kapil Kapoor
Associate Professor
ECE Department
Abhilashi Group of Institutions
(School of Engg.)
Chail Chowk Mandi, India

Abstract: ***Cleansing data from impurities is an integral part of data processing and maintenance. This has lead to the development of a broad range of methods intending to enhance the accuracy and thereby the usability of existing data. In these days, many organizations tend to use a Data Warehouse to meet the requirements to develop decision-making processes and achieve their goals better and satisfy their customers. It enables Executives to access the information they need in a timely manner for making the right decision for any work. Decision Support System (DSS) is one of the means that applied in data mining. Its robust and better decision depends on an important and conclusive factor called Data Quality (DQ), to obtain a high data quality using Data Scrubbing (DS) which is one of data Extraction Transformation and Loading (ETL) tools. Data Scrubbing is very important and necessary in the Data Warehouse (DW). This paper presents a survey of sources of error in data, data quality challenges and approaches, data cleaning types and techniques and an overview of ETL Process.***

Keywords: Data Cleansing, Data warehouse, Decision Support System, Data Quality, Data Scrubbing, Extract-Transform-Load (ETL).

I. INTRODUCTION

Data cleansing is the process of detecting, diagnosing, and editing faulty data. It deals with data problems once they have occurred. The purpose of data cleansing is to detect so called dirty data (incorrect, irrelevant or incomplete parts of the data) to either modify or delete it to ensure that a given set of data is accurate and consistent with other sets in the system. Poor data quality is a well-known problem in data warehouses that arises for a variety of reasons such as data entry errors and differences in data representation among data sources. However, high quality data is essential for accurate data analysis. Data quality is very crucial for the success Data analysis. The data loaded to the data warehouse must be correct, accurate and must be of very high quality. High quality data in the data warehouse will result in the better analysis and better decision making. So this data quality issues must be addressed before the data is loaded in to the data warehouse. Data cleaning, also called data cleansing or scrubbing. Data cleaning find errors and remove errors. It also detects and deals with data redundancy and data inconsistency. Data cleansing ensures that undecipherable data does not enter the data warehouse. Undecipherable data will affect reports generated from the data warehouse via OLAP, Data Mining and KPI's. Data cleaning for data warehouse are usually performed through Extract Transform Load(ETL) operations. Either we can use our own procedures or programs to perform the ETL functions or could use specific ETL tools to perform these functions. Without a data cleansing strategy the data warehouse will be expected to suffer:

- First from lack of quality,
- Second from loss of trust,
- Third a diminishing user base, and
- Fourth loss of business sponsorship and funding.

II. SOURCES OF ERROR IN DATA

Before a data item ends up in a database, it typically passes through a number of steps involving both human interaction and computation. Data errors can creep in at every step of the process from initial data acquisition to archival storage. An understanding of the sources of data errors can be useful both in designing data collection and curation techniques that mitigate the introduction of errors, and in developing appropriate post-hoc data cleaning techniques to detect and ameliorate errors. Many of the sources of error in databases fall into one or more of
the following categories:

Data entry errors: It remains common in many settings for data entry to be done by humans, who typically extract information from speech (e.g., in telephone call centers)or by keying in data from written or printed sources. In these settings, data is often corrupted at entry time by typographic errors or misunderstanding of the data source. Another very common reason that humans enter \dirty" data into forms is to provide what we call spurious integrity: many forms require certain fields to be filled out, and when a data-entry user does not have access to values for one of those fields, they will often invent a default value that is easy to type, or that seems to them to be a typical value. This often passes the crude data integrity tests of the data entry system, while leaving no trace in the database that the data is in fact meaningless or misleading.

Measurement errors: In many cases data is intended to measure some physical process in the world: the speed of a vehicle, the size of a population, the growth of an economy, etc. In some cases these measurements are undertaken by human processes that can have errors in their design (e.g., improper surveys or sampling strategies) and execution (e.g., misuse of instruments). In the measurement of physical properties, the increasing proliferation of sensor technology has led to large volumes of data that is never manipulated via human intervention. While this avoids various human errors in data acquisition and entry, data errors are still quite common: the human design of a sensor deployment (e.g., selection and placement of sensors) often affects data quality, and many sensors are subject to errors including miscalibration and interference from unintended signals.

Distillation errors: In many settings, raw data are preprocessed and summarized before they are entered into a database. This data distillation is done for a variety of reasons: to reduce the complexity or noise in the raw data (e.g., many sensors perform smoothing in their hardware), to perform domain-specific statistical analyses not understood by the database manager, to emphasize aggregate properties of the raw data (often with some editorial bias), and in some cases simply to reduce the volume of data being stored. All these processes have the potential to produce errors in the distilled data, or in the way that the distillation technique interacts with the final analysis.

Data integration errors: It is actually quite rare for a database of significant size and age to contain data from a single source, collected and entered in the same way over time. In almost all settings, a database contains information collected from multiple sources via multiple methods over time. Moreover, in practice many databases evolve by merging in other pre-existing databases; this merging task almost always requires some attempt to resolve inconsistencies across the databases involving data representations, units, measurement periods, and so on. Any procedure that integrates data from multiple sources can lead to errors.

III. IMPORTANCE OF DATA QUALITY IN DATA WAREHOUSES

Data quality can simply be described as a fitness for use of data. To be more specific every portion of data has to be accurate to clearly represent the value of itself. This is as much important for the clarity of data as for the correlation between massive databases. Without certain standards those databases would collapse. Data quality is critical to data warehouse and business intelligence solutions. Better informed, more reliable decisions come from using the right data quality technology during the process of loading a data warehouse. It is important the data is accurate, complete, and consistent across data sources. The data quality process includes such terms as data cleansing, data validation, data manipulation, data quality tests, data refining, data filtering and tuning. It is a crucial area to maintain in order to keep the data warehouse trustworthy for the business users.

An appropriate definition of the data quality criteria will ensure that the quality of the data is measured, analyzed and subsequently improved. The data quality management in an organization requires great involvement from the business side and very often a lot of manual interventions.

ETL plays a major role in data cleansing and data quality process as it helps automate most of the tasks outlined above. The data warehouse is fed daily with an orders extract which comes from a source OLTP system. Unfortunately, the data quality in that extract is poor as the source system does not perform much consistency checks and there are no data dictionaries

The data quality problems that need to be addressed are identified using two types of **Data Quality tests**:

Syntax Tests

ReferenceTests

The syntax tests will report dirty data based on character patterns, invalid characters, incorrect lower or upper case order, etc.

The reference tests will check the integrity of the data according to the data model. So, for example a customer ID which does not exist in a data warehouse customer's dictionary table will be reported.

Also, both types of tests report using two severity levels: errors and warnings.

When an error is encountered, the record is logged and not passed through to the output.

Warnings are logged, however still loaded into the data warehouse.

Data Quality challenges

The challenges addressed by data quality platforms can be broken down into business and technical requirements. The **technical** data quality problems are usually caused by data entry errors, system field limitations, mergers and acquisitions, system migrations.

Technical DQ challenges:

- Inconsistent standards and discrepancies in data format, structure and values
- Missing data, fields filled with default values or nulls
- Spelling errors
- Data in wrong fields
- Buried information
- Data anomalies

To be able to **measure Data Quality**, data should be divided into quantifiable units (data fields and rules) that can be tested for completeness and validity.

Some business DQ challenges:

- Reports are accurate and credible
- Data driven business process work flawlessly
- Shipments go out on time
- Invoices are accurate
- Data quality should be a driver for successful ERP, CRM or DSS implementations

Approaches to Improving Data Quality

The lifetime" of data is a multi-step and sometimes iterative process involving collection, transformation, storage, auditing, cleaning and analysis. Typically this process includes people and equipment from multiple organizations within or across agencies, potentially over large spans of time and space. Each step of this process can be designed in ways that can encourage data quality. here we mention a broad range of approaches that have been suggested for maintaining or improving data quality:

Data entry interface design: For human data entry, errors in data can often be mitigated through judicious design of data entry interfaces. Traditionally, one key aspect of this was the specification and maintenance of database integrity constraints, including data type checks, bounds on numeric values, and referential integrity (the prevention of references to non-existent data). When these integrity constraints are enforced by the database, data entry interfaces prevent data-entry users from providing data that violates the constraints. An unfortunate side-effect of this enforcement approach is the spurious integrity problem mentioned above, which frustrates data-entry users and leads them to invent dirty data. An alternative approach is to provide the data-entry user with convenient affordances to understand, override and explain constraint violations, thus discouraging the silent injection of bad data, and encouraging annotation of surprising or incomplete source data.

***Organizational management*:** In the business community, there is a wide-ranging set of principles regarding organizational structures for improving data quality, sometimes referred to as Total Data Quality Management. This work tends to include the use of technological solutions, but also focuses on organizational structures and incentives to help improve data quality. These include streamlining processes for data collection, archiving and analysis to minimize opportunities for error; automating data capture; capturing metadata and using it to improve data interpretation; and incentives for multiple parties to participate in the process of maintaining data quality

Automated data auditing and cleaning: There are a host of computational techniques from both research and industry for trying to identify and in some cases rectify errors in data.

Exploratory data analysis and cleaning: In many if not most instances, data can only be cleaned effectively with some human involvement. Therefore there is typically an interaction between data cleaning tools and data visualization systems. Exploratory Data Analysis [Tukey, 1977] (sometimes called Exploratory Data Mining in more recent literature [Dasu and Johnson, 2003]) typically involves a human in the process of understanding properties of a dataset, including the identification and possible rectification of errors. Data profiling is often used to give a big picture of the contents of a dataset, alongside metadata that describes the possible structures and values in the database. Data visualizations are often used to make statistical properties of the data (distributions, correlations, etc.) accessible to data analysts.

In general, there is value to be gained from all these approaches to maintaining data quality. The prioritization of these tasks depends upon organizational dynamics: typically the business management techniques are dictated from organizational leadership, the technical analyses must be chosen and deployed by data processing experts within an Information Technology (IT) division, and the design and rollout of better interfaces depends on the way that user tools are deployed in an organization (e.g., via packaged software, downloads, web-based services etc.) The techniques surveyed in this report focus largely on technical approaches that can be achieved within an IT organization, though we do discuss interface designs that would involve user adoption and training.

IV. DATA CLEANING: TYPES AND TECHNIQUES

Focusing more specifically on data cleaning, there are many techniques in the research literature, and many products in the marketplace. (The KDD Nuggets website [Piatetsky-Shapiro, 2008] lists a number of current commercial data cleaning tools.) The space of techniques and products can be categorized fairly neatly by the types of data that they target. Here we provide a brief overview of data cleaning techniques, broken down by data type.

Quantitative data are integers or floating point numbers that measure quantities of interest. Quantitative data may consist of simple sets of numbers or complex arrays of data in multiple dimensions, sometimes captured over time in time series. Quantitative data is typically based in some unit of measure, which needs to be uniform across the data for analyses to be meaningful; unit conversion (especially for volatile units like currencies) can often be a challenge. Statistical methods for outlier detection are the foundation of data cleaning techniques in this domain: they try to identify readings that are in some sense \far" from what one would expect based on the rest of the data. In recent years, this area has expanded into the more recent field of data mining, which emerged in part to develop statistical methods that are efficient on very large data sets.

Categorical data are names or codes that are used to assign data into categories or groups. Unlike quantitative attributes, categorical attributes typically have no natural ordering or distance between values that fit quantitative definitions of outliers. One key data cleaning problem with categorical data is the mapping of different category names to a uniform namespace: e.g., a \razor" in one data set may be called a \shaver" in another, and simply a \hygiene product" (a broader category) in a third. Another problem is identifying the miscategorization of data, possibly by the association of values

with \lexicons" of known categories, and the identification of values outside those lexicons [Raman and Hellerstein, 2001]. Yet another problem is managing data entry errors (e.g. misspellings and typos) that often arise with textual codes. There are a variety of techniques available for handling misspellings, which often adapt themselves nicely to specialized domains, languages and lexicons [Gravano et al., 2003].

Postal Addresses represent a special case of categorical data that is sufficiently important to merit its own software packages and heuristics. While postal addresses are often free text, they typically have both structure and intrinsic redundancy. One challenge in handling postal address text is to make sure any redundant or ambiguous aspects are consistent and complete {e.g. to ensure that street addresses and postal codes are consistent, and that the street name is distinctive (e.g., \100 Pine, SanFrancisco" vs. \100 Pine Street, San Francisco".). Another challenge is that of deduplication: identifying duplicate entries in a mailing list that differ in spelling but not in the actual recipient. This involves not only canonicalizing the postal address, but also deciding whether two distinct addressees (e.g. \J. Lee" and \John Li") at the same address are actually the same person. This is a fairly mature area, with commercial offerings including Trillium, QAS, and others [Piatetsky-Shapiro, 2008], and a number of approaches in the research literature (e.g., [Singla and Domingos, 2006], [Bhattacharya and Getoor, 2007], [Dong et al., 2005]). The U.S. Bureau of the Census provides a survey article on the topic [Winkler, 2006].

Identifiers or keys are another special case of categorical data, which are used to uniquely name objects or properties. In some cases identifiers are completely arbitrary and have no semantics beyond being uniquely assigned to a single object. However, in many cases identifiers have domain-specific structure that provides some information: this is true of telephone numbers, UPC product codes, United States Social Security numbers, Internet Protocol addresses, and so on. One challenge in data cleaning is to detect the reuse of an identifier across distinct objects; this is a violation of the definition of an identifier

that requires resolution. Although identifiers should by definition uniquely identify an object in some set, they may be repeatedly stored within other data items as a form of reference to the object being identified. For example, a table of taxpayers may have a unique tax ID per object, but a table of tax payment records may have many entries per taxpayer, each of which contains the tax ID of the payer to facilitate linking the payment with information about the payer. Referential integrity is the property of ensuring that all references of this form contain values that actually appear in the set of objects to which they refer. Identifying referential integrity failures is an example of finding inconsistencies across data items in a data set. More general integrity failures can be defined using the relational database theory of functional dependencies. Even when such dependencies (which include referential integrity as a subclass) are not enforced, they can be \mined" from data, even in the presence of failures [Huhtala et al., 1999].

This list of data types and cleaning tasks is not exhaustive, but it does cover many of the problems that have been a focus in the research literature and product offerings for data cleaning. Note that typical database data contains a mixture of attributes from all of these data types, often within a single database table. Common practice today is to treat these different attributes separately using separate techniques. There are certainly settings where the techniques can complement each other, although this is relatively unusual in current practice.

V. DATA CLEANSING IN THE DATA WAREHOUSE

Data cleansing is a valuable process that can help companies save time and increase their efficiency. Data cleansing software tools are used by various organisations to remove duplicate data, fix and amend badly-formatted, incorrect and amend incomplete data from marketing lists, databases and CRM's. They can achieve in a short period of time what could take days or weeks for an administrator working manually to fix. This means that companies can save not only time but money by acquiring data cleaning tools.

Data cleansing is of particular value to organisations that have vast swathes of data to deal with. These organisations can include banks or government organisations but small to medium enterprises can also find a good use for the programmers. In fact, it's suggested by many sources that any firm that works with and hold data should invest in cleansing tools. The tools should also be used on a regular basis as inaccurate data levels can grow quickly, compromising database and decreasing business efficiency. Data cleansing is often the most time intensive, and contentious, process for data warehousing projects. Data cleansing ensures that undecipherable data does not enter the data warehouse. Undecipherable data will affect reports generated from the data warehouse via OLAP, Data Mining and KPI's.

Data cleaning for data warehouse are usually performed through Extract Transform Load (ETL) operations. Either we can use our own procedures or programs to perform the ETL functions or could use specific ETL tools to perform these functions. ETL plays a major role in data cleansing and data quality process.

Why Extract, Transform and Load (ETL)?

Extract, Transform and Load (ETL) refers to a category of tools that can assist in ensuring that data is cleansed, i.e. conforms to a standard, before being entered into the data warehouse. Vendor supplied ETL tools are considerably easier to utilize for managing data cleansing on an ongoing basis. ETL sits in front of the data warehouse, listening for incoming data. If it comes across data that it has been programmed to ***transform***, it will make the change before ***loading*** the data into the data warehouse.ss

ETL tools can also be utilized to ***extract*** data from remote databases either through automatically scheduled events or via manual intervention. There are alternatives to purchasing ETL tools and that will depend on the complexity and budget for your project. Database Administrators (DBAs) can write scripts to perform ETL functionality which can usually

suffice for smaller projects. Microsoft's SQL Server comes with a free ETL tool called Data Transforming Service (DTS). DTS is pretty good for a free tool but it does have limitations especially in the ongoing administration of data cleansing.

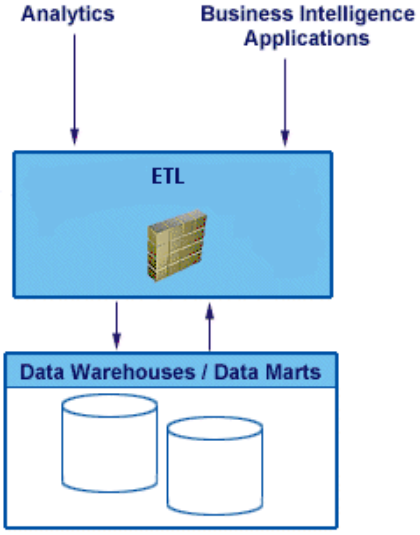


Figure 1. ETL sits in front of Data Warehouses

ETL (Extract-Transform-Load)

ETL comes from Data Warehousing and stands for Extract-Transform-Load. ETL covers a process of how the data are loaded from the source system to the data warehouse. Currently, the ETL encompasses a cleaning step as a separate step. The sequence is then Extract-Clean-Transform-Load. Let us briefly describe each step of the ETL process.

ETL Process

Extract

The Extract step covers the data extraction from the source system and makes it accessible for further processing. The main objective of the extract step is to retrieve all the required data from the source system with as little resources as possible. The extract step should be designed in a way that it does not negatively affect the source system in terms or performance, response time or any kind of locking.

There are several ways to perform the extract:

- Update notification - if the source system is able to provide a notification that a record has been changed and describe the change, this is the easiest way to get the data.
- Incremental extract - some systems may not be able to provide notification that an update has occurred, but they are able to identify which records have been modified and provide an extract of such records. During further ETL steps, the system needs to identify changes and propagate it down. Note, that by using daily extract, we may not be able to handle deleted records properly.
- Full extract - some systems are not able to identify which data has been changed at all, so a full extract is the only way one can get the data out of the system. The full extract requires keeping a copy of the last extract in the same format in order to be able to identify changes. Full extract handles deletions as well.

When using Incremental or Full extracts, the extract frequency is extremely important. Particularly for full extracts; the data volumes can be in tens of gigabytes.

Clean

The cleaning step is one of the most important as it ensures the quality of the data in the data warehouse. Cleaning should perform basic data unification rules, such as:

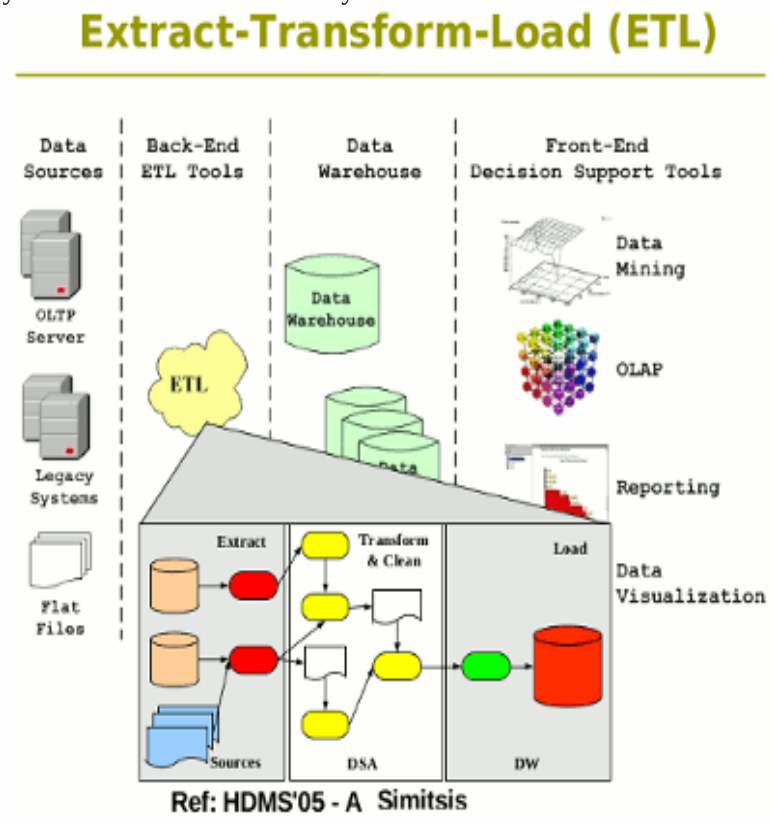
- Making identifiers unique (sex categories Male/Female/Unknown, M/F/null, Man/Woman/Not Available are translated to standard Male/Female/Unknown)
- Convert null values into standardized Not Available/Not Provided value
- Convert phone numbers, ZIP codes to a standardized form
- Validate address fields, convert them into proper naming, e.g. Street/St/St./Str./Str
- Validate address fields against each other (State/Country, City/State, City/ZIP code, City/Street).

Transform

The transform step applies a set of rules to transform the data from the source to the target. This includes converting any measured data to the same dimension (i.e. conformed dimension) using the same units so that they can later be joined. The transformation step also requires joining data from several sources, generating aggregates, generating surrogate keys, sorting, deriving new calculated values, and applying advanced validation rules.

Load

During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible. The target of the Load process is often a database. In order to make the load process efficient, it is helpful to disable any constraints and indexes before the load and enable them back only after the load completes. The referential integrity needs to be maintained by ETL tool to ensure consistency.



Extract Transform Load (ETL) Tools

- Ab-initio
- Informatica
- IBM InfoSphere DataStage
- Oracle Data Integrator

Commonly used ETL tools are Ab-initio, Informatica, Data Stage etc. These graphical tools are specially designed to perform the ETL function without much worry about coding. These tools come up with various built in functions. In some tools these built-in functions are called components.. We can use these functions (or components) according to our business requirements. But sometimes these built-in functions are not sufficient to perform our business requirements. In such situations the tools provide the facility to create custom components or functions to meet the specific requirement. The ETL developer has to follow the specifications given by the tool to create custom functions/components. Some of these tools also provide facilities to write unix scripts/shell scripts to handle the data to be fit for use with the built-in functions/components in the ETL tools.

VI. CONCLUSION

Data cleaning will be considered one of the most important frontiers and one of the most promising interdisciplinary developments in Information technology. In this paper we try to briefly review the data cleaning process, sources of error in data. We further outlined the importance of data quality its challenges and provide an overview of the main solution approaches. Furthermore we provide an overview of data cleansing in data warehouses using ETL Tool. This study would help the researchers to focus on the various issues of data cleansing. Data Cleansing is useful for both public and private sectors for finding patterns, forecasting, discovering knowledge in different domains such as finance, marketing, banking, insurance, health care and retailing. Data cleansing is commonly used in these domains to increase the sales, to reduce the cost and enhance research to reduce costs, enhance research.

REFERENCES

[1] Li Lee Mong , *Cleansing Data for Mining and Datawarehousing*, school of computing National University of Singapore, 1999 .

[2] Rahm E. & Hai Do Hong, *Data Cleaning: Problems and current approaches*, IEEE Bulletin of the Technical Committee on Data Engineering, 2000

[3] S. Abiteboul, S. Cluet, T. Milo, P. Mogilevsky, J, Siméon, S. Zohar **Tools for Data Translation and Integration.** Bulletin of the Technical Committee on Data Engineering, March 1999,Vol. 22, No. 1, 3-8

[4] Tamraparni Dasu and Theodore Johnson. Exploratory Data Mining and Data Cleaning. Wiley,2003.

[5] Doan, A.H.; Domingos, P.; Levy, A.Y.: *Learning Source Description for Data Integration*. Proc. 3rd Intl. Workshop The Web and Databases (WebDB), 2000.

[6] Hernandez, M.A.; Stolfo, S.J.: *Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem*. Data Mining and Knowledge Discovery 2(1):9-37, 1998.

[7] Lee, M.L.; Lu, H.; Ling, T.W.; Ko, Y.T.: *Cleansing Data for Mining and Warehousing*. Proc. 10th Intl. Conf.Database and Expert Systems Applications (DEXA), 1999.

[8] Munawar, Naomie Salim and Roliana Ibrahim, "Towards Data Quality into the Data Warehouse Development", IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 2011

[9] R.R Nemoni and R Konda, "A Framework for Data Quality in Datawarehouse", In J. Yang et. Al (Eds): UNISCON 2009, Springer-Verlag Berlin Heidelberg, 2009.

