Automated ETL Testing on the Data Quality of a Data Warehouse

Sara B. Dakrory
Computer Science Department, Faculty of Science, Minia University, El-Minia, Egypt

Tarek M. Mahmoud
Computer Science Department, Faculty of Science, Minia University, El-Minia, Egypt

Abdelmgeid A. Ali
Computer Science Department, Faculty of Science, Minia University, El-Minia, Egypt

ABSTRACT
Testing ETL (Extract, Transform, and Load) procedures is an important and vital phase during testing Data warehouse (DW); it's almost the most complex phase, because it directly affects the quality of data. It has been proved that automated testing is valuable tool to improve the quality of DW systems while the manual testing process is time consuming and not accurate so automating tests improves Data Quality (DQ) in less time, cost and attaining good data quality. In this paper the author's propose testing framework to automate testing data quality at the stage of ETL process. Different datasets with different volumes (stared from 10,000 records till 50,000 records) are used to evaluate the effectiveness of the proposed automated ETL testing. The conducted experimental results showed that the proposed testing framework is effective in detecting errors with the different data volumes.

General Terms
Data warehouse, data quality, ETL testing.

Keywords
Automated ETL Testing, Data Quality, Data Warehouse, Data Quality checking Routines.

1. INTRODUCTION
The ETL importance came from the definition of its functionality and the development effort. ETL is responsible for fetching the data from the heterogeneous sources systems into the DW so each failure in the ETL functionality leads to loading incorrect data in DW, which in turn leads to provide managers with incorrect data that leading to inaccurate decisions. Rainardi V [1] gives the ETL about 50 % of the development effort. "Data warehouse projects fail for many reasons, all of which can be traced to a single cause: nonquality" [2]. This raises the need to ensure that the data in the source is consistent with the data that reached the DW. Singh and Singh [6], Wayne [9], and Kimball [10] consider ETL stage as the most crucial stage in DW process since the maximum responsibility of data quality efforts resides in this stage. This leads to considering ETL stage as plenty area of DQ problems. Needing to automate ETL testing on DQ came from the importance to test the overall data that come from the data sources and ensure that it's loaded correctly into the destination DW. The aim in testing the data quality in the ETL is to ensure the correctness of ETL procedures and whether or not it need to be re-designed to mitigate the issues. The aim of this paper is to automate test routines that check the data quality parameters (completeness, consistency, uniqueness, validity, timeliness, and accuracy).

1.1 ETL Automation Challenges and Limitations:
It is commonly accepted that not all the DW tests can be automated, but some critical and repetitive tests can be achieved using automating tools. Here are some of the challenges and limitations as discussed by Vucevic and Yaddow [14]:

2. DW consists of many tables and records, which raise the testing complexity.
3. DW rely on extracting data from multiple source systems, and during test process it's mandatory to check the data between those come from sources and loaded in the DW.
4. Automated tests can not completely replace the manual tests. Manual tests are still needed to handle complex cases where automation may not catch everything.
5. Business objects reports testing are still a challenge to be automated.
6. The cost of automated tools leads to keep away from using it and depend on manual tests.

1.2 ETL Automation Benefits:
1. Decrease the consumed time in the testing phases, as automation tools speed up the test cases implementation.
2. Reusability of tests, since the stored tests can be used many times later.
3. Save the human consumed effort and time spent in manual tests.
4. Take the advantage of the automated tools to generate reports and records test results.
5. Reduce the effort spent in the regression tests by using the test cases generated by the automation tools to confirm that after each change in data or business rules other parts of the system does not affect.

The rest of this paper is organized as follows: Section 2 gives a review of the studies of data quality problem in the DW and the few works proposed to automate DQ testing in the ETL stage. Section 3 presents the framework architecture. Section 4 presents the implementation description of the proposed framework. Section 5 presents a case study to test the ability of the implemented system to detect the errors through the generated tests. Finally, the conclusions and future works are shown in section 6.

2. LITERATURE REVIEW
Golfarelli and Rizzi [3] introduced a classification of testing activities in two coordinates: what is tested and how it is tested. The "what" coordinate, concerns with the data quality testing that require an accurate check on the data loaded by ETL procedures and accessed by the frontend tools. The second coordinate "how" is achieved by defining seven types

of tests: functional test, usability test, performance test, stress test, recovery test, security test, and maintainability test.

Manjunath et al [11] analyzed data quality factors in each stage of data warehouse system i.e., data sources, staging area and target system (multidimensional schema).

From other perspective Singh and Singh [6] gave a descriptive classification of causes of data quality problems at all the phases of data warehousing: data sources, data integration & data profiling, data staging and ETL, data warehouse modeling and schema design.

While, Jarke et al [4] divided the data quality types into 3 main categories: data quality, schema quality, and operational quality.

1- Data Quality: it's main concern is quality of data in each stage of DW stages (i.e., data in data sources, data in the data warehouse, data in the data marts, or processed data presented to the end user).

2- Schema Quality: concerns the design quality of each component in the data warehouse system

3- Operational Quality: concerns the data warehouse system quality when it launched when put into operation and the quality of the process of developing the data warehouse system.

Also in [7], Singh and Singh established that there is a lack in the information available on the quality assurance of ETL routines, and suggested that by automating very basic quality checks for data quality management have given satisfactory results.

And in [8], Gill and Singh concluded to many researchers have considered the following data quality issues in data warehouse environment: naming conflicts, structural conflicts, date formats, missing values, changing dimensions. These data quality issues have been implemented through various tools, but no single framework has provided a solution to all the above problems at a single place. Moreover, the frameworks implemented which covers all the issues are implemented through paid tools.

Moreover in [12], Rodic and Baranovic proposed a DQ rules generator that generates data quality rules and integrate the generated rules into ETL process. They also split the rules into three categories: Database integrity rules, Match and merge rules, and Business rules.

Vucevic and Yaddow [14] defined the DW automation test according to the functionality of the used tools, That can be one of four categories:1)Test Execution Automation tools, 2) Data Compare Automation tools, 3)Test Preconditions set up tools, 4)other test control and test reporting functions.

3. THE PROPOSED AUTOMATING DQ TESTING FRAMEWORK ARCHITECTURE

The proposed data quality framework main goal is to automate tests that check data quality in ETL process by automating the creation and execution of these tests. The testing process of ETL procedures carried out after the multidimensional schema is tested and ensured that it fulfillment the user requirements. By analyzing the existing standards and guidelines related to software testing, Figure (1) represents the sequence diagram introduced by Tanuška et al [13] that summarized the steps followed to test data warehouse projects. By considering the previous testing process and the corresponding activities needed to be achieved, Figure (2) summarizes the proposed framework that used to automate test routines for checking the data quality parameters defined in Table (1) and consider the dependencies between the DQ dimensions. Table (1) displays the Quality dimensions definition, checking routines needed to achieve the most coverage, and the related dimensions that reflect the dependency between the quality parameters. The proposed framework consists of: three repositories, mapping document, two processes and the results reports. The three repositories are: DW Metadata Repository, which stores the DW metadata, Database Model Repository, which used to store the needed information from the mapping document and the metadata repository, and Test Routine Repository which stores quality parameters definitions and their related test cases. Two processes will be used to generate and execute the test cases based on the data stored in the previous repositories, and finally generate result report about the passed test cases.

4. IMPLEMENTATION DESCRIPTION

4.1 Test Environment Set Up

In this step first of all, the references will be defined to the involved databases: source database, staging database (if exists) and DW. Secondly, the ETL logical mapping document and the metadata repository involved in this step. The Logic data mapping document contains the journey for each extracted filed from the source system till the final destination. The responsibility of creating this document is for the business analyst. This is usually an excel sheet, also called a crosswalk or Interface Design Document (IDD). These are a layout that analyzes the source, or legacy, systems and also describes the business rules and transformations. It contains the following fields: target table name, target column name table type, Slowly changing Dimension (SCD) type, source database, source table name, source column name, and transformation [10]. Finally, a connection to the metadata repository is established. There are many tools exist in market that enables the exchange of metadata between DW phases for example: Teradata Meta Data Services (MDS) [15], Meta Integration Model Bridge (MIMB) [16], Pragmatic Works [17].

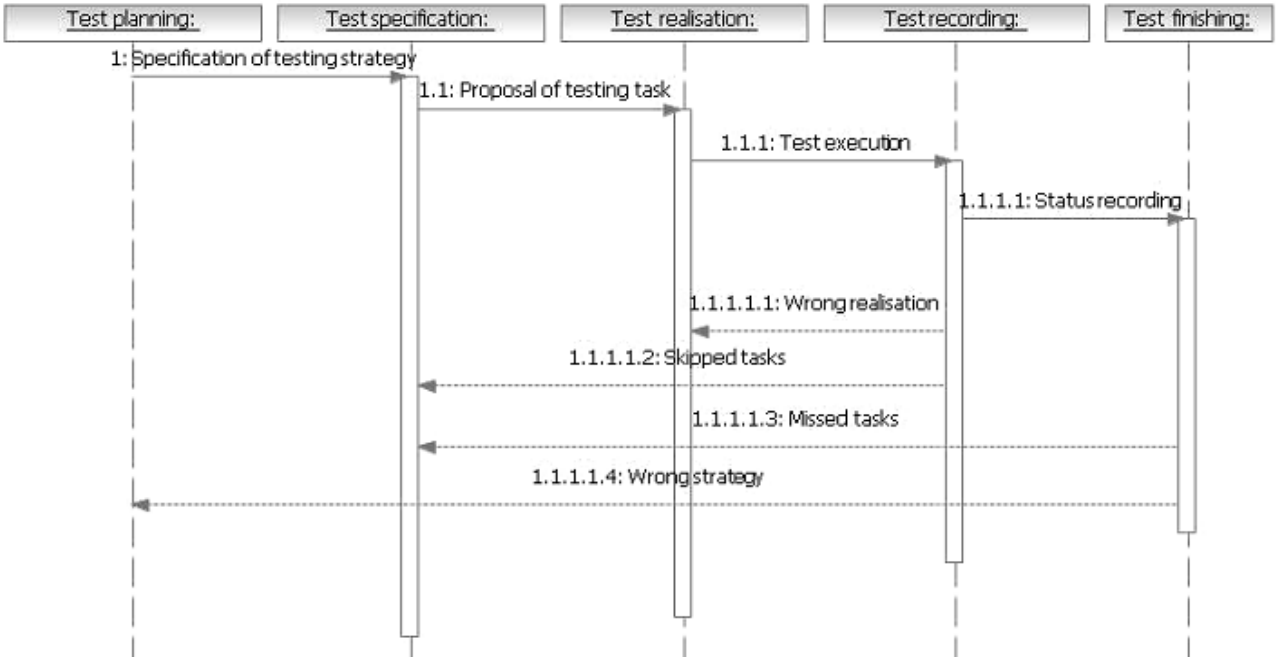

Fig 1: Testing Process Sequence Diagram.

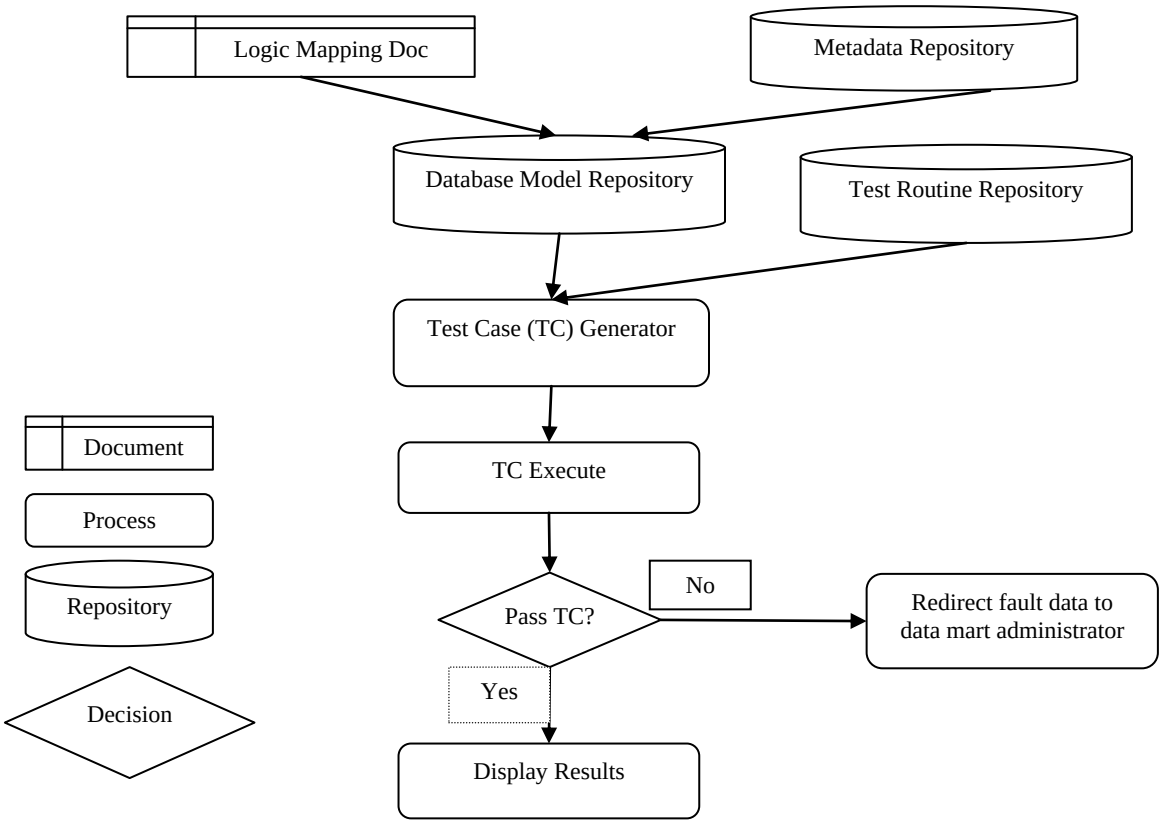

Fig 2: The proposed Automating DQ Testing Framework

Table 1: Quality Dimensions Definition.

	Completeness	Consistency	Uniqueness	Validity	Timeliness	Accuracy
Definition	Deals with to ensure whether all relevant data stored [5].	Deals with to ensure whether all values consistent across data sets.	Deals with to ensure there is no duplications in stored data.	Deals with to ensure that data are conforms to the syntax (format, type, range) of its definition.	Deals with to ensure whether all data is stored with the required time frame.	Deals with to ensure whether all data correctly describes the "real world" object or event being described.
Checking Routines	1-Record Count Validation 2-Data Duplicate Check. 3-Integrity Constraints Check. 4-Data Boundaries.	1-Field Mapping. 2-Integrity Constraints. 3-Measure Aggregation. 4-Hierarchy Level Integrity.	1- Data Duplicate Check. 2-Integrity Constraints Check.	1- Integrity Constraints Check. 2-Field Data Type Check. 3-Field Length Check.	1-Data Access. 2-Data Freshness.	1-Field-to-Field Comparison. 2-Data Boundaries. 3- Integrity Constraints.
Related Dimension	Validity and Accuracy	Validity, Accuracy and Uniqueness	Consistency	Accuracy, Completeness, Consistency and Uniqueness	Accuracy	Validity

4.2 Loading the Extracted Data into the Database Model

After extracting required data from the Logic mapping document and DW metadata, this data is loaded into the database model. But in some cases may be the metadata repository is not ready in this point of DW phases so, instead of building the whole metadata repository, the Schema information for both data source database and staging database (assuming that it have the same structure of DW) will be queried to retrieve the metadata. Each database management system manages its own schema information. The database model schema that used to store the extracted data will be shown in Figure (3).

4.3 Export the Data Quality Parameters from the Test Routine Repository

For each quality parameter in the DQ model a number of routines are assigned. Assigning each quality parameter to test routines were done manually by studying all quality issues that affect each quality parameter and finding test routines that detects these quality issues. The assigned tests routines will be executed on each dimension/fact table exist

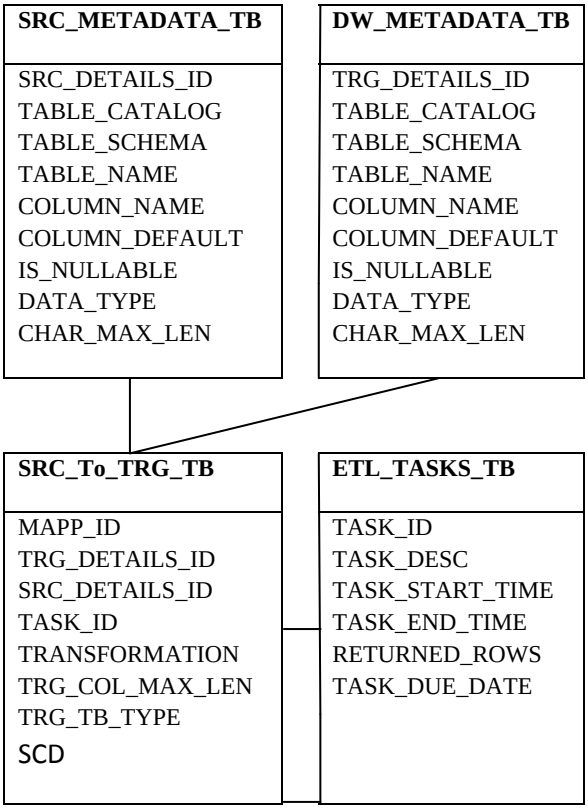

Fig 3: Database Model.

4.4 Executing the Test Routines

In this step each filed in the extracted test case will be assigned to a value from the database model (table name, column name etc.). The algorithm used for the execution is:

```
--For each dimension/fact table in the logic data
mapping document, do this:

Begin
For each quality parameter in the test routine
repository:
Begin
Fetch the corresponding test cases

        For each test case:
            Map the fields of the test case to its
        corresponding values in the database model.
            Execute test case.
        If Test Case failed, then redirect test case
        details to data mart administrator.

        Else Show result.
        End if;
        End for each;

End for each;
End;
End;
```

4.5 Test Action Step

After executing the tests on the selected item, successful test cases will be passed directly to the next step. In the case of failed test case(s) many strategies can be used for dealing with ETL errors determined during requirement analysis phase such as: automatically clean faulty data, reject faulty data, hand faulty data to data mart administrator, etc. [3]. The last option has been considered.

4.6 Displaying the Test Result

TC results can be export into reports and can be stored in the results table for future use.

5. CASE STUDY

In this section, the proposed framework and the implemented system will be demonstrated through a case study. An ETL process has been considered which populates DimEmployee table. The Employee dimension table has been populated from another multiple tables in an operational database. Figure (4) represents the tables used to populate the Employee dimension table.

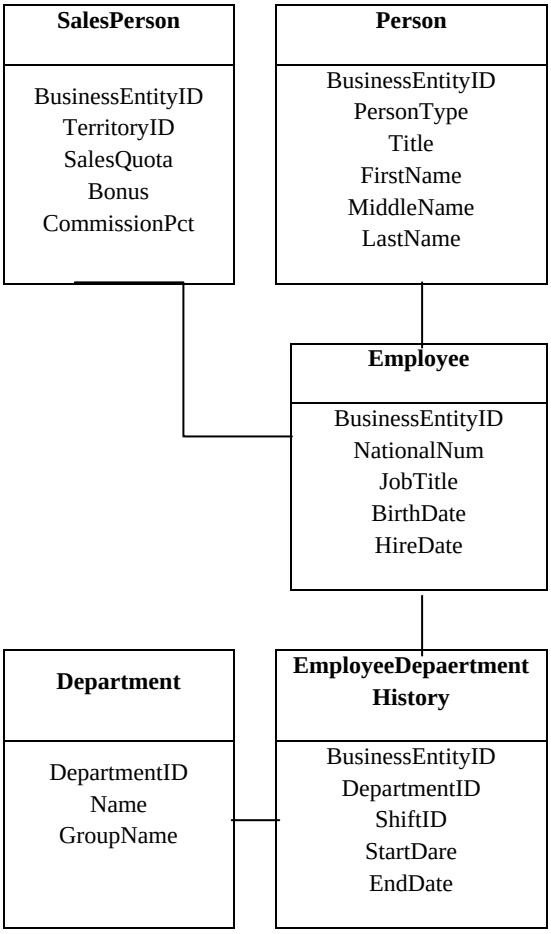

Fig 4: ERD for source database.

The data flow diagram for the ETL process is shown that the specific columns will be extracted from the source database and the extracted data will be compare with those in the DW table, only the new records will be added and the others will be rejected. Figure (5) displays the data flow for employee dimension table.

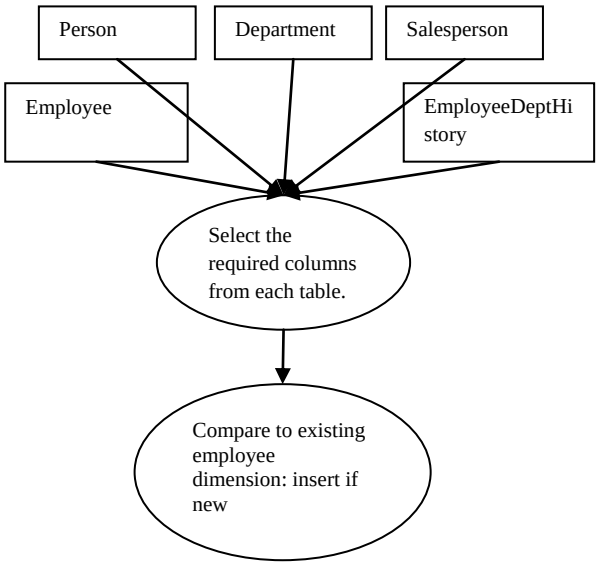

Fig 5: Data flow for Employee Dimension Table.

The proposed tool has been tested by carried out two different databases, including respectively 1) real data that loaded in a

dimension table; (2) data simulating the presence of faulty data of different kinds represent non-effective ETL process. Table (2) shows the generated test cases and their results after seeding errors in the tested dataset. The TC result can be one of two options Pass/Fail but after seeding the errors all the TCs are failed.

Table 2: The Generated Test Cases Results.

DQ Dimensions and the Generated Test Cases Names		Expected Outcome	Actual Result	TC Result
Completeness	1) Duplicate Values Checking	Values in columns are unique	Data in the destination table are redundant	Fail
	2) Integrity Constrains Checking	Foreign key/ primary key is maintained	Violation of the foreign key / primary key constrain	Fail
	3) Out of Boundaries Value Checking	Data in the destination table are in specific rang	Violation of rang specification in the destination table.	Fail
	4) SCD Checking	Data in the SCD column are compatible with the specified type (1, 2, or 3)	Values in the SCD attribute are not compatible with the defined type.	Fail
	5) Record count validation Checking	The count of records in the destination table is the same number in the source table	Mismatch in the count of attributes between the source and destination	Fail
Consistency	1) Field Mapping Checking	Fields are mapped depend on logic mapping document specification	Mismatch in fields mapping	Fail
	2) Measure Values Checking	Measure values are correctly calculated	The results of the measure function are erroneous	Fail
Uniqueness	1) Duplicate Values Checking	Values in columns are unique	Redundant data in the destination table.	Fail
	2) Integrity Constrains Checking	Foreign key/ primary key is maintained	Violation of the foreign key constrain/ primary key	Fail
Validity	1) Data Type Checking	The data type of the source filed is the same data type in the destination	Data type mismatch	Fail
	2) Field length Checking	Field length in the source table is the same length in the destination	Field length mismatch	Fail
Timeliness	1) Data Freshness Checking	Data represent reality from the required timestamp	The time stamp columns are not reliable	Fail
Accuracy	1) Out of Boundaries Values Checking	Data in the destination table are in specific rang	Values in the destination tables are out of boundaries	Fail
	2) Truncated Values Checking	Data in the destination table are the same in the source	There is a mismatch between the data in the source and the data loaded in the destination	Fail

After getting the TCs results from the previous table an evaluation of the system effectiveness is made by compare the detected defeats with the number of seeded defects .Figure (6) shows the seeded errors' types and there frequency in the tested data set.

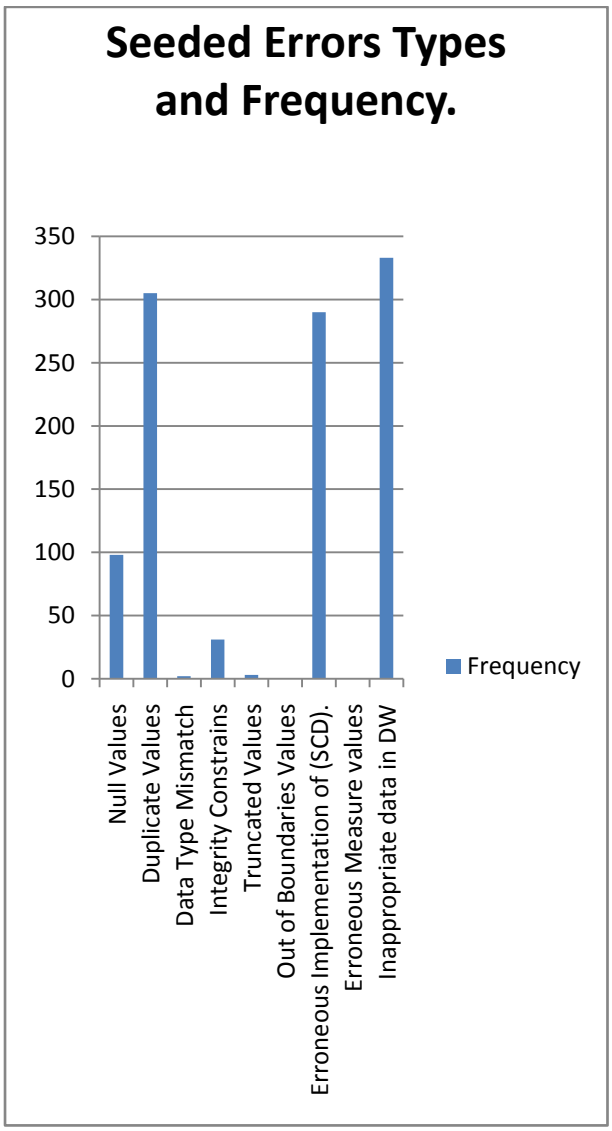

Fig 6: Seeded Errors Types and Frequency.

All the result reports, produced by the system, showed that the seeded errors have been detected, which demonstrates its effectiveness in testing DQ in ETL stage

The implemented system tested on different datasets with different volumes (stared by 10,000 records To 50,000 records) and the system proved an efficient in detecting errors with the different data volumes.

By analyzing the previous results, the causes of these faults can summarized in the following points:

1) The look up process didn't work correctly since there are many duplicate values. This indicates an erroneous of the load strategy.

2) Data integrity constrains in data staging tables are disabled which permit loading of wrong data and relationships.

3) The Filed length in the source database must be greater than or the same of the field length in the DW to save data from truncation.

4) SCD process is not implemented right and didn't satisfy the rules of the specified type (1,2, 3, 4, 5,or 6)

5) Inappropriate data results in some times from wrong data mapping.

6. CONCLUSION

In this paper a framework for automating ETL testing for data quality has been proposed. The proposed framework delivers a wide coverage for data quality testing by framing testing activities within a modular methodology that can be customized according to ETL specificities, business rules, and constraints. The proposed test routines associated to quality parameter does not satisfy the fulfillment of data quality completely, but it raises the level of confidence in the data warehouse with respect to this quality parameter. More test routines will be added to the proposed matrix in the future to increase the test coverage.

7. REFERENCES

[1] Rainardi, V. Testing your Data Warehouse. in *Building a Data Warehouse with Examples in SQL Server*, Apress, 2008.

[2] English, L. P. (1999). Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits, John Wiley and Sons, Inc.Data Quality Issues.

[3] Golfarelli, M. and Rizzi, S. Data Warehouse Testing:A prototype-based methodology. Information and Software Technology, 53 (11). 1183-1198.

[4] Jarke, M., Jeusfeld, M. A., Quix, C., and Vassiliadis, P. (1999). "Architecture and Quality in Data Warehouses: An Extended Repository Approach." Information Systems,24(3), 229-253.

[5] Askham, N., and Cook, D. (2013). "Defining Data Quality Dimensions: The six Primary Dimensions for Data Quality Assessment." Enterprise Data and BI Conference,London, UK.

[6] Singh R, and Singh K. (2010). A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. International Journal of Computer Science Issues (IJCSI). 7(4).

[7] Singh R, and Singh K. (2009). Statistically analyzing the Impact of automated ETl Testing on the Data Quality of a Data Warehouse, International Journal of Computer and Electrical Engineering, Vol. 1, No. 4.

[8] Gill R., Singh J., 2014, A Review of Contemporary Data Quality Issues in Data Warehouse ETL Environment, Available at www.chitkara.edu.in/publications.

[9] Wayne W. E., 2004, "Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data ",The Data warehouse Institute (TDWI) report ,available at www.dw-institute.com .

[10] Kimball R. and Caserta J., 2004, The Data Warehouse ETL Toolkit. John Wiley & Sons.

[11] Manjunath T.N, Ravindra S Hegadi, Ravikumar G K. "Analysis of Data Quality Aspects in Datawarehouse Systems", (IJCSIT)-Jan-2011.

[12] Rodic, J. and Baranovic, M., Generating Data Quality Rules and Integration into ETL Process, Proceeding of DO- LAP'09, Hong Kong, 2009, pp. 65-72.

[13] Tanuška, P., Pavel, V. and Peter, S., The Partial Proposal of Data Warehouse Testing Task. 2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)

[14] Vucevic, D., and Yaddow, W. (2012). Testing the Data Warehouse Practicum- Assuring Data Content, Data Structures and Quality, Trafford

[15] Available at: http://www.teradata.com/tools-and-utilities/meta-data-services. Last access, Oct, 2015

[16] Available at: http://www.metaintegration.net/Solution. Last access, Oct, 2015.

[17] Available at: http://pragmaticworks.com. Last access, Oct, 2015.