

Statistically Analyzing the Impact of Automated ETL Testing on the Data Quality of a Data Warehouse

Jaiteg Singh and Kawaljeet Singh

Abstract-- For truthful reporting and decision-making a major challenge in data warehouse industry is to ensure quality data. The Extraction, Transformation and Loading (ETL) module is crucial to attain high quality data for a data warehouse. In-house development of ETL solutions with improvised algorithms may result in unknown errors at logical or technical levels. To assure data quality one has to understand the prevailing data quality assurance practices. This paper is intended to empirically analyze the impact of automated ETL testing on the data quality of the data warehouse. The data quality was observed before and after the induction of automated ETL testing. Statistical analysis indicated a substantial escalation in data quality after the induction of automated ETL testing.

Index Terms—ETL, ELT, Data Warehouse, ETL Test Cases

I. INTRODUCTION

The Data Warehousing Institute, (TDWI), in a recent report, estimates that data quality problems currently cost U.S. businesses \$600 billion each year. Even then the benefits of high quality data are ignored because of the heavy operating expenses associated with attaining it. Data quality issue gets even more significant while implementing a data warehouse as the data warehouse itself does not do cleansing of data satisfactorily and relies only on stored procedures. These inflexible stored procedures prove to be bottlenecks as the data needs to be cleansed repeatedly. The best place to cleanse data is the ETL platform as it can save both time and money. The published work substantiates there is very diminutive information available on the quality assurance of ETL routines. Hence the authors followed an empirical approach to understand ETL structure so as to concentrate precisely on various subsystems that test incoming source data automatically so as to ensure a quality ETL routine. Initially an ETL prototype was developed which is capable of extracting data from a number of distinct databases following different structures and formats. The methodology involved loading of source data to a newly designed data sink using this ETL prototype that automatically conforms data for some common errors. Then statistically it is established that this tool is capable of making predefined transformations and data purification to the extent possible for managing the database.

Jaiteg Singh is with Punjabi University Patiala, India as research Scholar. Tel: 0091-9872029991

Dr. Kawaljeet Singh is presently with Punjabi University Patiala, India working as Director in the University Computer Center..

II. UNDERSTANDING THE ETL

In case of a data warehouse the ETL routine is primarily responsible for the data quality. Hence the selection of an appropriate ETL tool is a serious concern for an organization. There are number of ETL tools available in the market but generally small and medium sized enterprises use a hand coded ETL routine to extract and unify data. The authors perceive ETL as an assemblage of many small independent sub systems of its own like as identified below:

- **Aggregate building System:** It is for creating and maintaining physical database structures.
- **Backup system:** It is responsible for backing up data and metadata.
- **Cleansing system:** It is a dictionary driven system for information parsing. For example names and addresses of individuals and organizations etc.
- **Data Change identification system:** It is to record Source log file reads, source date and sequence number etc.
- **Error tracker and handling System:** It is for identifying and retorting to all ETL error events.
- **Fact table loading System:** It is equipped with push/pull routines for updating transaction fact tables.
- **Job scheduling System:** It is for scheduling and launching all ETL jobs.
- **Late arriving fact and Dimension Handling System:** It is for insertion of fact and dimension records that because of some reason have been delayed in arriving at the data warehouse.
- **Metadata manager.** It is for assembling, capturing and maintaining all ETL metadata and transformation logic.
- **Pipelining system:** It is required for implementing and streaming data flows.
- **Quality Checking System:** It is responsible to check the quality of incoming data flows.
- **Recovery and restart system:** It is responsible for restarting a job that has halted.

- **Security system:** It is responsible for the security of data within an ETL.
- **Source Extract system:** It includes Source data adapters along with push/pull routines for filtering and sorting at the source.
- **Surrogate key pipelining System:** It is a Pipelined, multithreaded process for replacing natural keys of incoming data with data warehouse surrogate keys.

Early data warehouses ETL systems were not proficient of managing the extensive processing required to perform the complex transformations involved in the warehouse load process. So third-party tools like IBM's WebSphere DataStage and Informatica were used to organize data movement between source systems and the data warehouse. Now days with the advancement in hardware and data warehouse development technology, the designers consider Extract Load and Transform (ELT) as an alternative to ETL. ETL logic states that before loading the data to the data warehouse, the data should be moved to an intermediate platform where the transformation rules should be applied. On the other hand the ELT follows a standard data transfer mechanism such as File Transfer Protocol (FTP) to transfer the bulk data directly to the data warehouse. The transformation rules are then applied to the data warehouse tables with the help of preloaded procedures instead of any intermediate staging area. A comparative analysis of the two architectures is summarised as below:

TABLE I. ETL VS ELT

	ETL	ELT
1	A dedicated external system is applied to take care of transformation logic for data standardization and business rules thus reducing the unnecessary burden from the data warehouse.	There is no dedicated external system responsible to tackle transformation logic and business rules. The transformations are done on the data already loaded into the data warehouse.
2	The whole data has to travel first from source to staging area and then from staging area to the data warehouse through the network thus causing excess network traffic when there is no dedicated link between the ETL server and the data warehouse.	The files are loaded from the source systems to the data warehouse via FTP or other secure file transfer methods, hence the network traffic is least affected.
3	The ETL server requires high performance CPU and huge disk capacity to sustain the transformation process. This can lead to the need for expensive and highly sophisticated hardware.	Transformation logic to be applied on the stored data of a data warehouse will utilize additional data warehouse resources for execution.
4	ETL tools have the capability to interact with other external engines for data validation before the data is loaded on the data warehouse. such as Geographic Information Systems (GIS)	Complex transformations which may require external sources of data are not easy to implement with the stored procedures of the data warehouse.
5	Errors if any that occur during the transformation process can be located and corrected before data is loaded in the data warehouse	Database roll-backs are inevitable in case an error occurs during the transformation process. Generally these rollbacks are taken on temporary tables.

	table thus reducing the need for time consuming database roll-backs.	
6	Depending on the number of sources feeding the data warehouse, the ETL licensing may become a costly affair.	The cost for loading the data warehouse is quite lower than the ETL architecture as there is no additional software licence is required.
7	Time spent on bringing in the data to the data warehouse is higher.	Time for getting data to the data warehouse is reduced as there is no staging process required.

Before selecting the loading procedure one must weigh up the data transformation requirements along with the desired data quality of the targeted database. If the transformation rules are intricate and cannot be carried out using stored procedures of the database than ELT architecture should be avoided. ELT is best suited for environments where text parsing routines are required to implement data standardization and cleansing. For hefty environments where numerous sources with terabytes of transactional data are involved, ETL is the best suited architecture. On the other hand ELT is best suited for loading small data sets where relatively simple transformation logic is applied. ELT is also best suited for manipulating business data for populating data marts that has a physical infrastructure similar to that of the data warehouse.

III. BUILD OR BUY?

According to the data unification needs the business organizations need to decide whether to build or buy an ETL tool. Although the ETL tools offered by various vendors are very much proficient in their functionality but still there are many organizations that believe to hand code ETL programs than to use readymade ETL software. These companies advocate their decision by aiming at the high cost of many ETL tools and the profusion of programmers on their staff. Table II summarises Build Vs. Buy options.

Table II. BUILD VS BUY

	BUILD	BUY
1	It is cheaper and quicker to code ETL programs than use a vendor ETL tool.	ETL tools are expensive to purchase and requires renewal of license.
2	It is cheaper to maintain as it is geared with a specific business.	They are developed for generalized use.
3	Code written is based upon custom specifications and meta data model.	Industry specifications are considered instead of custom specifications.
4	No need to pay unnecessary training or maintenance fee to any vendor.	One has to bear paid training sessions and heavy maintenance costs to introduce vendor ETL tools.
5	Easily available object oriented technology is best suited for ETL development.	Licensed vendor ETL tools are not desirable for small businesses
6	Challenging factors like migrating source data into a data warehouse along with data cleansing jobs can easily be performed with the hand coded ETL routine.	Vendor ETL tools follow a generalized approach to migrate source data into a data warehouse. They are quiet about how to identify and clean dirty data or to build interfaces to legacy systems.
7	Metadata is rarely maintained	Meta data is highly maintained.
8	Complex mappings can be	Only predefined mapping procedures can be carried out

	handled easily with custom built ETL code.	which generally elude complex mappings.
9	It is difficult to ensure adequate stability, reliability and performance.	Adequate stability, reliability and performance are well ensured in advance.
10	Team of expert programmers is required to develop a customized ETL tool.	Highly salaried and experienced programmers are hired by ETL vendors for developing, training and maintenance purposes.
11	To keep the costs down and for better turnaround times one can outsource the ETL development code to Asian countries	Generally they are built by highly skilled, salaried and in house trained coders, outsourcing is avoided to avert any kind of data breach.
12	These tools are flexible and can be adjusted in accordance with the changing business dimensions	These are not much flexible and one has to look in for readymade plug ins to cope with changing business dimensions.
13	Creating and integrating user defined functions is a cumbersome and difficult job.	The well designed proficient modules are integrated in advance.
14	Rigorous testing and debugging effort is required.	No need of testing and debugging only maintenance is required.
15	Source data is well understood.	They follow a generalized approach in understanding source databases.

The authors perceive a hand coded ETL routine as the easiest way to migrate data but this low level implementation makes the maintenance of such migration solutions a complicated job and may cause hidden ETL errors at logical or technical level. Accurate data does not come free. It requires careful attention to the design of ETL systems [8], constant monitoring of data collection activities and assertive actions to correct problems that generate or propagate inaccurate data. Any hand coded ETL module can not be used on real life data management until and unless its performance has been assured by testing it rigorously [14]. To identify prime testing zones for a hand coded ETL tool the authors coded an ETL tool which is capable of extracting data from databases like Oracle, SQL, MS Access, MS Excel and flat files like those of MS word [12] [13]. This tool is capable of making predefined transformations and data purification to the extent possible for managing the database.

IV. THE MAJOR DATA QUALITY PROBLEMS

While performing data synthesis without inducting automated ETL testing the authors identified the following categories of data anomalies.

- 1) **Data type Mismatch:** The type mismatch imposes a major problem during data extraction and loading process. For example in source one the personal records have the serial no. attribute declared as text where in source two there serial number attribute declared as an integer. Both these fields are acting as a primary key for source concerned. The fusion of these records is hard to manage as it is feasible to insert integer values into a text field but integer field cannot hold a text value.
- 2) **Specification Mismatch:** This is a problem of multiple abbreviations. For example in two different source tables the sex specification field is having same data type but expressed differently. In source one personal record

database sex attribute is specified as M/F where as in source two it is specified as male/female.

- 3) **Missing Values:** values of some fields were missing in either the source or the target database. As a null record may have mappings/dependencies with some another fields in some another tables which may further aggravate data quality problems.
- 4) **Typing Errors:** This type of errors is almost inevitable because the data is being fed into the database by humans. If some one has spelled male as amle in the sex specification field then it is will cause a complete but vague record.
- 5) **DBMS Constraints:** Every DBMS is having its own pros and cons along with their compatibility issues. For example if the source data is in ms access and the target table is in oracle then some major transformations has to be made to merge data from source to target.
- 6) **Transformation Problems:** To eliminate data consistency and integrity problems if one tries to purify data through data transformations it is again not an easy job. For example one cannot insert null values into an integer field because by default its value is set to zero.
- 7) **Normalization Breakup:** Merging data from different sources may denormalize the target database. A poorly designed database may provide erroneous information and may become difficult to use, or may even fail to work properly. Most of these problems are the result of two bad design features called: redundant data and anomalies. Redundant data is unnecessary reoccurring data; Anomalies are any occurrence that weakens the integrity of the data due to irregular or inconsistent storage [1] [2].

Basically, normalisation is the process of efficiently organising data in a database. There are two main objectives of the normalization process: eliminate redundant data (storing the same data in more than one table) and ensure data dependencies make sense (only storing related data in a table) [4] [5]. Both of these are valuable goals as they reduce the amount of space a database consumes and ensure that data is logically stored. Normalized data bases can provide better query response times.

V. THE ETL QUALITY CHECKS

After identifying the aforesaid data quality issues it was the time to introduce automated ETL testing [6] [7]. The fact, queries that perform satisfactorily on small datasets may fail miserably in the real life environment. This necessitates establishing a system that runs queries on fully scaled data. Legal implications and business ethics do not allow performing testing with real business data. Hence the authors put efforts to generate test data ensuring correct balance and skewness making sure that the ratio of fact to dimension is

TABLE III TEST CASES FOR QUALITY ETL ROUTIN

S.No	Test Case Desc.	Input	Expected Outcome	Actual Result	Assigned to	Defect Severity *	Result (P/F)
1	Connecting to the Source Database	ETL routine tries to connect the targeted Source database	Source Database is connected	There is no connectivity of the targeted source database	Development team	Major	Fail
2	Availability of the Source Database	ETL routine tries to extract source data	Source data is available every time the ETL tries to extract data	Source data is available at particular times	Development team	Major	Fail
3	Availability of the Source Database	ETL routine tries to extract source data	Source database is swiftly available through ODBC OLEDB connections	In absence of ODBC OR OLEDB connections, specific drivers are needed.	Development team	Minor	Fail
4	Availability of the Source Database	ETL routine tries to extract source data	Source data server name is accepted	The target source server name is not accepted	Development team	Minor	Fail
5	Availability of the Source Database	ETL routine tries to extract source data	Source data is available and the ETL is capable of querying the database	Targeted source database is available but ETL is unable to query the database	Development team	Major	Fail
6	Availability of the Source Database	Source OLTP database is called in	Source OLTP database version is in accordance with the drivers installed on the ETL server.	The database version of the OLTP is newer than the driver on the ETL server	Development team	Minor	Fail
7	Data Extraction from a flat file	File is imported	Structure of the file is justified and bulk data is imported from the file	Structure of file is not justified	source DBA	Minor	Fail
8	Data Extraction from a flat file	File is imported and file is cleared from archive directory	Bulk data is imported from the file while cleaning the source archive directory.	Data has been extracted but replica is still existing on archive directory	Development team	Minor	Fail
9	Data Extraction from a flat file	Data is to be extracted from the file	Data file format is clearly specified to ensure the quality of data.	There is a confusion over the file format and extracted data is not meaningful	Source DBA	Major	Fail
10	Data Extraction from a flat file	Data is to be extracted from the file	Order and number of columns extracted are same as anticipated.	Number of columns are more than expected, the order of is also disturbed	Development team	Minor	Fail
11	Extracting Relational Databases	ETL approaches the relational database	Source database schema is in accordance with warehouse schema	There is a mismatch in the count of attributes defined for a single entity in the source schema and in the target schema.	Development team and Source DBA	Major	Fail
12	Extracting Relational Databases	ETL approaches the relational database	Relational database is extracted and primary key attribute is maintained	Violation of the Primary key	Development team	Major	Fail
13	Extracting Relational Databases	ETL approaches the relational database	Iterative extraction of only modified rows is possible	No criteria available to download only changed rows from the source system	Source DBA	Major	Fail

.No	Test Case Desc.	Input	Expected Outcome	Actual Result	Assigned to	Defect Severity *	Result (P/F)
14	Extracting Relational Databases	Data is extracted iteratively	The timestamp columns are well maintained at the source database.	The timestamp columns are not reliable	Development team and Source DBA	Major	Fail
15	Extracting Relational Databases	ETL approaches the relational database	The ETL routine has read only access to the source database	The ETL is capable of modifying the source database	Source DBA	Major	Fail
16	Extracting Relational Databases	ETL approaches the relational database	ETL can fetch the information about every related table as it is reachable from the main entity source table	Related tables are not reachable from the main source table	Development team	Minor	Fail
17	Extracting Relational Databases	ETL approaches the relational database	Count of source records matches with the count of updated records in the target database.	The count of source records does not match with the count of updated records in the target table.	Development team	Major	Fail
18	Extracting Relational Databases	ETL approaches the relational database	Data Extraction is completed within the specified time window	ETL routine is waiting endlessly to complete its job	Development team	Minor	Fail
19	Extracting Relational Databases	ETL approaches the relational database	The ETL log is updated with every ETL transaction failure keeping the older entries.	New incremental access of the ETL is deleting older log files.	Development team	Major	Fail
20	Extracting Relational Databases	ETL approaches the relational database	There are no data leaks, means the rows in target table are also present in the source table	some rows which were present in the target tables were missing in source	Development team and Source DBA	Major	Fail
21	Extracting Relational Databases	ETL approaches the relational database	Data is extracted without Lexical anomalies	Lexical or Syntactical anomalies are present	Development team	Major	Fail
22	Extracting Relational Databases	ETL approaches the relational database	The ETL routine is capable of handling format errors	Format Errors are Present	Development team	Minor	Fail
23	Extracting Relational Databases	ETL approaches the relational database	The ETL routine is capable of handling irregularities in data representations	Irregular representations are there	Development team	Major	Fail
24	Verifying ETL Functionality	User tries to access ETL staging area	Non administrator users cant access staging area	Any User can access Staging area	Development Team	Major	Fail
25	Verifying ETL Functionality	The transformation logics are applied to the staging area	There are well defined transformation logics for every attribute	The transformations made are not reasonable	Development team and Project Management	Major	Fail
26	Verifying ETL Functionality	ETL tries to update Target Database	ETL has full access to the target database and the Target table is Updated	Update Fails	Development Team	Minor	Fail
27	Verifying ETL Functionality	ETL tries to back up data in case of update failure	ETL provides a backup facility in case of a failure	There is no such facility available	Development Team	Major	Fail
28	Verifying ETL Functionality	The backed up data has to be recovered	The Backed up data is stable and can be retrieved easily.	The retrieval algorithm is not defined	Development Team	Major	Fail

TABLE IV ANOMALIES OBSERVED BE

Sample No.	Lexical Errors	Format Errors	Irregularities	Integrity Constraints	Contradictions	Duplicates	Semantic Errors	Sample Total	Sample Mean (\bar{x})	Sample Range (R)
I	65	40	50	80	40	35	75	385	55	45
II	75	25	35	60	45	40	80	360	51.42	55
III	50	35	45	75	50	60	60	375	53.57	40
IV	40	45	60	40	65	50	65	365	52.14	25
V	65	230	70	45	45	270	45	770	110	225
VI	45	55	40	45	55	75	50	365	52.14	35
VII	120	10	65	60	62	45	60	424	60.57	110
									$\sum \bar{x} = 434.84$	$\sum R = 535$

TABLE V ANOMALIES OBSERVED AFTER AUTOMATED TESTING

Sample No.	Lexical Errors	Format Errors	Irregularities	Integrity Constraints	Contradictions	Duplicates	Semantic Errors	Sample Total	Sample Mean (\bar{x})	Sample Range (R)
I	4	4	2	4	2	0	6	22	3.142	6
II	6	0	1	2	0	2	3	14	2	6
III	2	3	0	1	3	1	0	10	1.428	3
IV	0	6	4	0	0	0	0	10	1.428	6
V	6	0	6	3	1	3	4	23	3.285	6
VI	2	4	0	2	2	2	1	13	1.857	4
VII	0	0	2	0	0	0	0	2	0.285	2
									$\sum \bar{x} = 13.425$	$\sum R = 33$

correct and so on. As result a test data generator was developed to generate synthetic test data enough to test the Extraction, Transformation and Loading routine performance [15].

How ever one can write test cases for the click of a button but it will merely be a test case which will not prove anything fruitful. A successful test case is one which makes the system halt at the occurrence of an erroneous event. Keeping in view the various ETL sub systems test scripts were written and embedded into the hand coded ETL routine to impose quality checks as and where required [9]-[11]. Test cases of prime importance for quality data were first identified and then inculcated to the ETL module for performing automated ETL testing. These test cases have been discussed in TABLE III.

VI. RECORDING THE OBSERVATIONS

Similar synthetic data of thirty five thousand records from similar sources were synthesized again but this time the ETL was equipped with automatic testing procedures. The resultant records were divided into seven samples of five thousand records each. Among these five thousand records, five hundred tuples were selected randomly. The manually observed value of each data quality issue before and after the induction of automatic testing has been recorded in table IV and table V respectively. The results produced by this prototype were analyzed statistically using t- test. The t- test was applied on the identified types of errors within the database of thirty five thousand records assuming null hypothesis. The hypothesis was rejected afterwards as major enhancement was observed in the quality of database.

VII. MEAN CHARTS OF THE OBSERVATIONS

The mean chart of the observations before automated testing is shown as under. From table IV it can be derived that:

$$x = \frac{\sum \bar{x}}{7} = \frac{434.84}{7} = 62.12 \text{----- (1)}$$

$$\bar{R} = \frac{\sum R}{7} = \frac{535}{7} = 76.42 \text{----- (2)}$$

Upper and Lower control lines are given by:

$$UCL = \bar{x} + A_2 \bar{R} \text{----- (3)}$$

$$= 62.12 + (0.419) * 76.42$$

$$[\because \text{for } n=7 \text{ and } A_2 = 0.419]$$

$$= 62.12 + 32.01 = 94.13$$

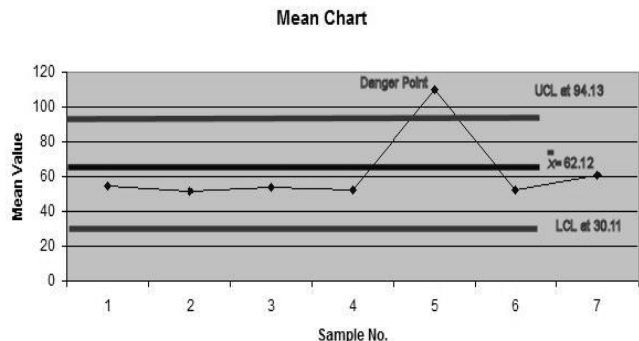
$$LCL = \bar{x} - A_2 \bar{R} \text{----- (4)}$$

$$= 62.12 - (0.419) * 76.42$$

$$[\because \text{for } n=7 \text{ and } A_2 = 0.419]$$

$$= 62.12 - 32.01$$

$$= 30.11$$



(a) Mean Chart of observations before introducing automatic testing
The control chart for mean is constructed by taking the sample number along the horizontal scale, (x axis) and the statistic mean along the vertical scale. Sample points are than plotted as points against the corresponding sample number.

The central line, UCL and LCL are plotted as horizontal lines at the computed values given in table IV.

The mean charts of the observations after the induction of automated testing is shown below.

From table V it can be derived that:

$$\bar{x} = \frac{\sum \bar{x}}{7} = \frac{13.425}{7} = 1.9178 \text{ ----- (5)}$$

$$\bar{R} = \frac{\sum R}{7} = \frac{33}{7} = 4.7142 = 4.71 \text{ ----- (6)}$$

Upper and Lower control lines are given by:

$$UCL = \bar{x} + A_2 \bar{R} \text{ ----- (7)}$$

$$= 1.9178 + (0.419) * 4.71$$

$$[\because \text{for } n=7 \text{ and } A_2 = 0.419]$$

$$= 1.9178 + 1.97349$$

$$= 3.89129$$

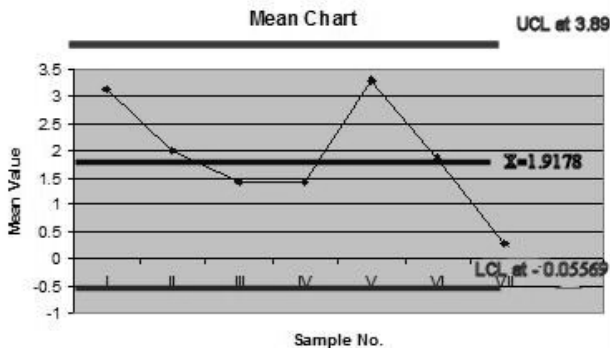
$$LCL = \bar{x} - A_2 \bar{R} \text{ ----- (8)}$$

$$= 1.9178 - (0.419) * 4.71$$

$$[\because \text{for } n=7 \text{ and } A_2 = 0.419]$$

$$= 1.9178 - 1.97349$$

$$= -0.05569$$



(b) Mean Chart of observations after introducing automatic testing

The control chart for mean is constructed by taking the sample number along the horizontal scale, (x axis) and the statistic mean along the vertical scale. Sample points are then plotted as points against the corresponding sample number. The central line, UCL and LCL are plotted as horizontal lines at the computed values given in table V.

VIII. INTERPRETATION OF MEAN CHART

A process is termed to be under statistical control if the mean charts exhibit control, i.e. all the sample points lie within the control limits in both the charts. If one or more of the points in any or both of the charts go out of control limits than one can say that the process is out of control or it is not under statistical control. Such a situation indicates the presence of some assignable causes of erratic fluctuations which must be traced, identified and eliminated so that the process may return to operation under stable statistical conditions. On the other hand a process under control implies that there are no apparent erratic causes of variation.

IX. WORKING OF THE ETL PROTOTYPE

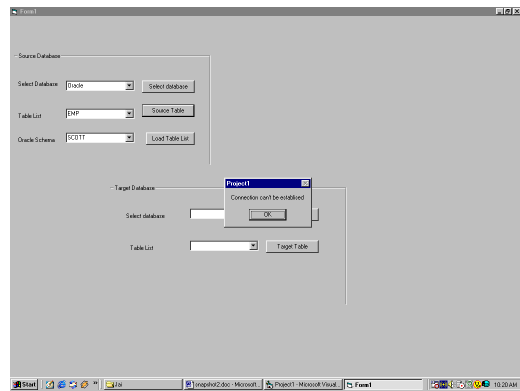


Fig.1 Specification of source and target databases

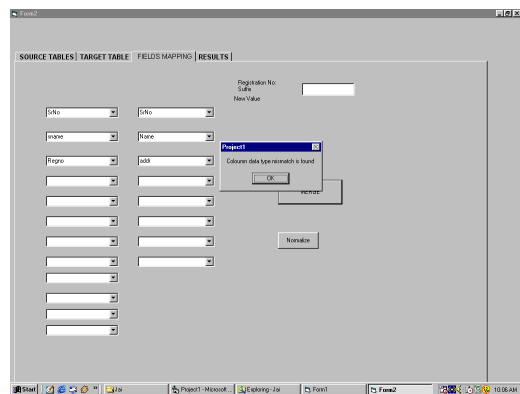


Fig.2 Field Mapping in ETL Prototype

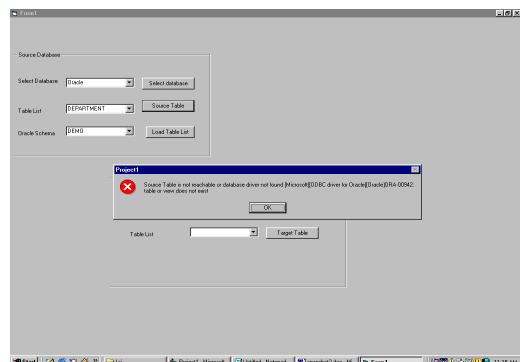


Fig. 3 Automated Testing Procedure

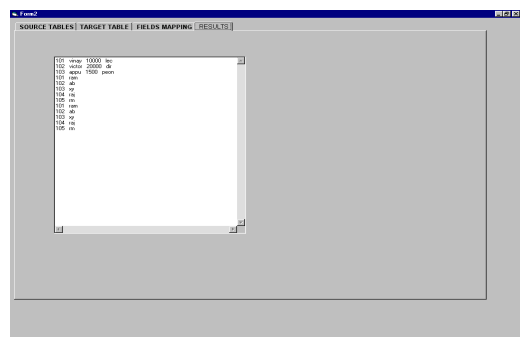


Fig. 4 Rejected Records

X. CONCLUSION AND FUTURE WORK

The primary reason for the convolution of the data extraction and transformation functions are the diversity of the source systems. This diversity includes bewildering combination of computing platforms, operating systems,

database management systems, network protocols, and source legacy systems etc. Hence there is a need to pay special attention to the various sources and there is a need of generating a complete record of the source systems too. With this record as a starting point one should work out all the details of data extraction. The difficulties encountered in the data transformation function should also be related to the heterogeneity of the source systems. The loading procedure might seem to be the simplest one but it is solely responsible for consolidation and integration of targeted database. Although by automating very basic quality checks for data quality management have given satisfactory results but still there is a need to study the scope of automated testing in extraction, transformation and loading routines independently.

Dr. Kawaljeet Singh is Ph.D in Computer Science from TIET Patiala, now he is working as Director University Computer Center, Punjabi University Patiala. His areas of interest are Data Warehousing and System Simulation.

REFERENCES

- [1] Larry P. English, "Improving Data Warehouse and Business Information Quality", New York: John Wiley & Sons, 1999.
- [2] Michael H. Brackett, "Data Resource Quality: Turning Bad Habits into Good Practices", New York: Addison-Wesley, 2000.
- [3] Richard J. Orli, "Data Quality Methods," based on a public document prepared for the United States government, 1996. [<http://www.kismet.com/cleand1.html>].
- [4] Man-Yee Chan and Shing-Chi Cheung, "Applying white box testing to database applications", Technical Report HKUST-CS9901, Hong Kong University of Science and Technology, Department of Computer Science, February 1999.
- [5] Man-Yee Chan and Shing-Chi Cheung, "Testing database applications with SQL semantics", In Proceedings of the 2nd International Symposium on Cooperative database Systems for Advanced Applications, pages 363-374, March 1999.
- [6] David Chays, Saikat Dan, Phyllis G. Frankl, Filippos I. Vokolos, and Elaine J. Weyuker, "A framework for testing database applications", In Proceedings of the 7th International Symposium on Software Testing and Analysis, pages 147-157, August 2000.
- [7] H. Galhardas, D. Florescu, D. Shasha and E. Simon, "Ajax: An Extensible Data Cleaning Tool", SIGMOD'00, pp.590, Texas, 2000.
- [8] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, "A Framework for the Design of ETL Scenarios", CAiSE'03, Klagenfurt, Austria, 2003.
- [9] E. Rahm, H. Do, "Data Cleaning: Problems and Current Approaches", Bulletin of the Technical Committee on Data Engineering, 23(4), 2000.
- [10] V. Raman, J. Hellerstein, "Potter's Wheel: An Interactive Data Cleaning System", VLDB'01, pp. 381-390, Roma, Italy, 2001.
- [11] Nikolay Iliev, Senior Application Consultant "Best Practice ETL for Validata using a staging area", validata organization white paper June 2007.
- [12] P. Vassiliadis, A. Simitsis, S. S.oulos, "Conceptual modeling for ETL processes", Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP, 2002.
- [13] A. Simitsis, "Mapping conceptual to logical models for ETL processes", Proceedings of the 8th ACM international workshop on Data warehousing and OLAP, pp.67-76, 2005.
- [14] D. Loshin, "Rule based data quality", Proceedings of the eleventh international conference on Information and knowledge management, pp.614- 616, 2002.
- [15] Singh, J.; Singh, K. "Designing a Customized Test Data Generator for Effective Testing of a Large Database", Advanced Computer Theory and Engineering, 2008. ICACTE apos;08. International Conference on Volume , Issue , 20-22 Dec. 2008 Page(s):84 – 88 Digital Object Identifier 10.1109/ICACTE.2008.39

Jaiteg Singh is graduated in Economics (Hons.) and Masters in Computer Applications from Punjabi University Patiala (India). He did M.Phil (Comp.Sci.) from Madurai Kamaraj University (India). Now he is pursuing Ph.D under the Faculty of Engineering and Technology, Punjabi University Patiala. He is senior lecturer at RIMT, Mandi Gobindgarh (India). His areas of interest are Data Warehousing, Data Quality Assurance and Software Engineering.