Identifying Prime Testing Zones to Ensure Quality ETL Routine of a Data Warehouse

Jaiteg Singh[1] *Kawaljeet Singh*[2]

Abstract

Attaining high quality accurate data for decision making is a major challenge for the data warehouse industry. The organizations practicing data warehouse are diverse in their functionality and follow varying business practices. To tackle the data quality and data management issues which are inherent in data warehouse implementation, the project team has to develop an extensive reviewing process for improving data quality. A thorough business practice review, an analysis of existing data quality and data cleansing techniques are necessary to develop a data quality assurance methodology. The quality assurance for an in house developed the extraction transformation and loading (ETL) routine using hand coded algorithms is not an easy task. This low level implementation of ETL logic may cause hidden errors at technical as well as at logical level. Hence instigation of automated testing procedures as an integrated module of ETL routine itself is essential to ensure a high quality data in a data warehouse. The identification of such crucial test cases which may be automated to enhance data quality in a data warehouse is the sole aim of this paper.

Keywords : ETL, Data Warehouse, ELT, ETL Prototype

[1]Research Scholar, University College of Engineering, Punjabi University, Patiala - 147002, India. Email : jaitegkhaira@yahoo.co.in

[2]Director, University Computer Centre, Punjabi University, Patiala - 147002, India. Email : singhkawaljeet@pbi.ac.in

Introduction

The importance of high quality data with respect to the strategic planning of any organization cannot be ignored. The Data Warehousing Institute, (TDWI), in a recent report, estimates that data quality problems currently cost U.S. businesses about $600 billion each year. Generally the benefits of high quality data are sidelined because of the expenses associated with attaining high quality data. A data quality strategy is very important specially when implementing a data warehouse. Although the effectiveness of a data warehouse is based upon the quality of its data but the data warehouse does not do a satisfactory job of cleansing data. The same data would need to be cleansed repeatedly during iterative operations. The best place to cleanse data is the ETL platform. By cleansing data in ETL instead of in the data warehouse, organizations can save time and money.

Methodology Followed : Precise data is an elementary condition for effective data warehouse development. Most data warehouses and information systems contain momentous quantity of imprecise data. There is plenty of literature available on the ETL structure, data quality and data warehousing but there is very diminutive information available on the quality assurance of ETL routines. Hence the authors followed an empirical approach to identify prime test cases essential for a quality ETL development and to understand the ETL routine structure itself. Initially an ETL prototype was developed (snapshot 1 to snapshot 5) which is capable of extracting data from a number of distinct databases following different structures and formats. After the analysis of

extracted data a number of test cases were developed to assure the quality data in the targeted data warehouse.

The organizations lack fundamental awareness of the concepts of information quality. The shortage of accurate data costs organizations dearly in correction actions along with lost customers, missed opportunities, and incorrect decisions. Most organizations are very much ignorant of the enormity of the data cleansing costs because they are unaware of the extent of inaccurate data in their database systems. Organizations are content in their belief that their data is good enough, although they have no basis for that belief [1]. In case of a data warehouse the prime responsibility for the data quality is of the ETL routine which is responsible to extract, transform and load data from different sources into the data warehouse. Hence the selection of an appropriate ETL tool is a serious concern of an organization. There are a 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 and the expenditure associated with a licensed commercial ETL tool can also be avoided. The ETL is a big term having many small independent sub systems of its own like :

1. **Aggregate Building System:** It is responsible for generating and maintaining physical database structures, known as aggregates. These are used to improve query performance. It includes stand-alone aggregate tables and materialized views.
2. **Backup System:** This system is responsible for back up, metadata, recovery, restart, security, and compliance requirements.
3. **Cleansing System:** This system is usually a dictionary driven system and is generally responsible for parsing generic details of individuals and organizations etc. It can identify and remove the duplicate records and retains back references like natural keys to all participating original sources.
4. **Data Change Identification System:** It keeps an eye over Source log file readers, source date and sequence number filters etc.
5. **Data Match Up:** Its sole responsibility is data integration across multiple data sources based upon their conformed attributes and measures.
6. **Data Profiling System:** This system is responsible for the analysis of Column properties of the source tables including detection of dependent domains, and structure analysis. It further handles foreign keys and candidate keys also.
7. **Error Handler:** It is a widespread system for identifying and retorting to all ETL error events. It has special routines to handle various classes of errors, and includes real-time monitoring of ETL data quality
8. **Error Tracking System:** It is an automatic as well as a manual system for trapping and resolution of an error condition. It includes simple error log entries, operator notification, and messages for the system developer.
9. **Fact Table Loader:** It is a System for updating transaction fact tables including manipulation of indexes and partitions. Normally used to append most recent data.
10. **Job Schedule Handler:** This System is for scheduling and launching all ETL jobs. It is able to halt itself for a wide variety of system conditions including dependencies of prior jobs completing successfully. It can also post alerts.
11. **Jumble Dimension Handler:** It is responsible for the creation and maintenance of dimensions

consisting of miscellaneous low cardinality features and indicators found in most production data sources.

12. **Late Arriving Dimension Handler:** This system is responsible for the insertion and update of dimension changes that have been deferred in arriving at the data warehouse.

13. **Late Arriving Fact Handler:** This system has an insertion and update logic for fact records that have been delayed in arriving at the data warehouse.

14. **Meta Data Assembler:** It assembles the metadata context associated with each fact table and loads it to the fact table as a normal dimension.

15. **Metadata Repository Manager :** This is an indispensable system of an ETL routine for capturing and maintaining all ETL metadata and transformation logic.

16. **Multi Valued Dimension Associative Table Builder:** This module is responsible for creating an associative table used for describing many-to-many relationship between dimensions.

17. **Multidimensional Cube Builder:** It is responsible for creation and maintenance of multidimensional (OLAP) cubes, including special preparation of dimension hierarchies as dictated by the specific cube technology.

18. **Pipelining System:** This system is highly desirable to automatically invoke pipelining system for any ETL process to tackle certain conditions, such as not writing to the disk in case of transaction failure or waiting on a condition in the middle of the query under execution.

19. **Quality Checker:** This system is responsible to check the quality of all the incoming flows of data.

20. **Recovery and Restart system:** This is a common system which is responsible for restarting a job that has halted.

21. **Report and Dependency Analyzer:** It can watch the ultimate physical sources and all subsequent transformations of any selected data element, chosen either from the middle of the ETL pipeline, or chosen on a final delivered report. It can record all affected downstream data elements and final report fields which are affected by a potential change in any selected data element.

22. **Security System:** This system is responsible for the security of data within an ETL pipeline and the ETL itself.

23. **Slowly Changing Dimension Handler:** This system has transformation logic for handling of time variance for a dimension attribute which generally includes events like overwrite, create new field or create new record.

24. **Source Extract System:** This system includes Source data adapters along with push/pull routines for filtering and sorting at the source. Such system is also responsible for data format conversions, and data staging after transfer to ETL environment.

25. **Surrogate Key Pipeline:** It is a Pipelined, multithreaded process for replacing natural keys of incoming data with data warehouse surrogate keys.

26. **Version Control System:** This system is required for archiving and recovering all the metadata in the ETL pipeline. It keeps a vigil on all the check-outs and check-ins of all ETL modules and jobs. This system has source comparison capability to reveal differences between different versions.

27. **Workflow Monitor:** This system includes a control panel and reporting system to watch all job runs

initiated by the Job Scheduler. It records number of records processed, summaries of errors, and actions taken etc.

Extract Transform and Load (ETL) was believed to be the most effectual way to insert data into a data warehouse. 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 [8]. Now a days with the advancement in both hardware and data warehouse development technology, the designers now consider Extract Load and Transform (ELT) as a better approach instead of extract transform and load (ETL). Because of data explosion the organizations have to manage pentabytes of data which may require hours or even days for ETL process completion depending upon the amount of data to be extracted and the complexity of the transformation rules. In the case of ETL, before passing the data along to the data warehouse the data is moved to an intermediate platform where the transformation rules are applied. Where as 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 and the data warehouse tables instead of any intermediate staging area. The difference between both these architectures is shown in figure 1 and figure 2 respectively.

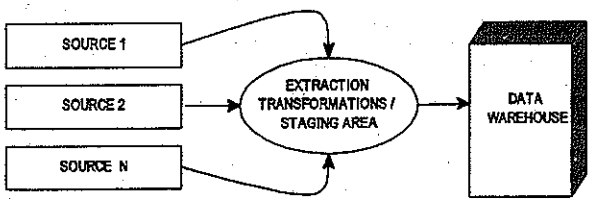


Figure 1 : The ETL Architecture

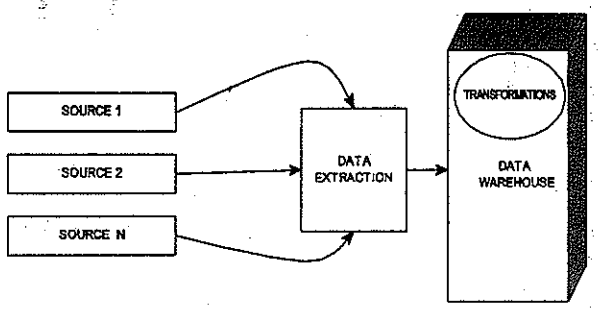


Figure 2 : The ELT Architecture

Both these architectures are responsible for fetching and merging data from different sources into the data warehouse. The only factor that differentiates the duo is the staging area. In ETL architecture the extracted data is first placed onto a staging area where the transformation logics are applied to consolidate as well as filter data to enhance its quality before loading this purified data into the data warehouse [13]. In case of ELT there is no staging area the data is loaded directly into the data warehouse. The transformations are applied within the data warehouse server itself. However after the rigorous analysis of both these architectures the following characteristics have been identified:

Table 1 : 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	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

	traffic when there is no dedicated link between the ETL server and the data warehouse.	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 table thus reducing the need for time consuming database roll-backs.	Database roll-backs are inevitable in case an error occurs during the transformation process. Generally these rollbacks are taken on temporary tables.
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 and in case where routines require text parsing ETL architecture are excellent for data standardization and cleansing [12]. For hefty environments where more than ten source systems are feeding the data warehouse and one terabyte or greater of transactional data is involved, the 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.

As business organizations investigate their data unification needs, the first decision they need to make is 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 it is better to hand write ETL programs than use off-the-shelf 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. Although it is very crucial decision but to facilitate this build or buy decision the following aspects can be considered.

Table 2 : Build Vs Buy Analysis

	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	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.
7	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 generally don't address the challenging facet of migrating source data into a data warehouse, typically they don't answer how to identify and clean dirty data, build interfaces to legacy systems, and deliver real time data.
8	Meta data is rarely maintained.	Meta data is highly maintained.
9	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.
10	Complex mappings can be handled easily with custom built ETL code.	Only predefined mapping procedures can be carried out which generally elude complex mappings.
11	It is difficult to ensure adequate stability, reliability and performance.	Adequate stability, reliability and performance is well ensured in advance.
12	Team of expert programmers is required to develop a customized ETL tool.	Highly salaried and experienced programmers are already employed by ETL vendors for developing, training and maintenance purposes.
13	Creating and integrating user defined functions is a cumbersome and difficult job.	The well designed proficient modules are integrated in advance and one has to study the user manual only to make it work.
14	Rigorous testing and debugging effort is required.	No need of testing and debugging only maintenance is required.
15	Source data is well understood in advance to tackle a particular database.	They follow a generalized approach in understanding source databases.

The use of different software systems from different vendors in order to provide full coverage of the business functions and the integration between these systems often requires real time interaction among them [5]. The easiest way to data migration is with the help of in-house built solutions with hand-coded algorithms for Extraction, Transformation and Load (ETL). This low level implementation makes the maintenance of such migration solutions a complicated job and can be a cause for hidden errors with the ETL processes on logical or technical level and once it has been decided to build an ETL project there begins a new chore of assuring its quality. Accurate data does not come free. It requires careful attention to the design of ETL systems, 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 [6,10]. A successful test case is one which makes the system halt at the occurrence of a particular event. 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. This tool is capable of making predefined transformations and data purification to the extent possible for managing the personal records database. 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. Hence in this paper only those test cases have been discussed in table 3. which are essential to enhance the quality an ETL routine of a data warehouse.

Table 3 : Prime Test Cases for ETL Quality Assurance

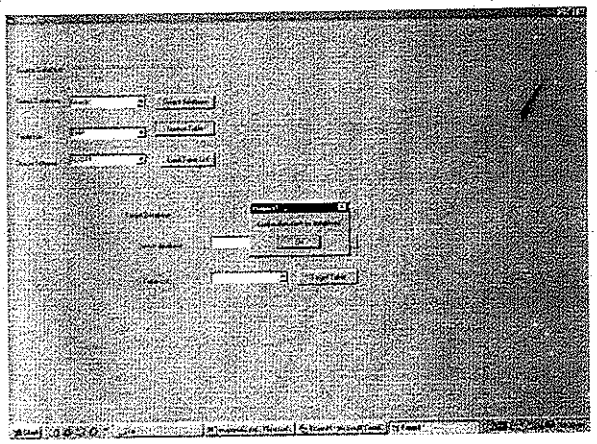
S.No	Test Case Desc.	Input	Expected Outcome	Actual Result	Assigned to	Defect Severity *	Result (P/F)	Remarks
1	Connecting to the Source Database	ETL routine tries to connect the targeted Source database	Source Database is connected	There is no connectivity of the targeted source database	Development team	Major	Fail	Provide the source and destination address carefully if in a networked environment provide the source and destination IP address along with the port no to be opened, also check if a firewall is interrupting the connection
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	Parley for a time slot with the source DBA and break the extraction into smaller chunks
3	Availability of the Source Database	ETL routine tries to extract source data	Source database is swiftly available through ODBC OLEDB connections	Target source database requires a particular driver because we are unable to use ODBC OR OLEDB connections	Development team	Major	Fail	Acquire or get hold of database drivers and install on ETL server
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	Major	Fail	Check for the specified name of the source database, some databases like teradata requires a suffix to the name and if on a networked environment try it with the help of IP address else ask the network administrator to add the name on system DNS
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	The source DBA should manage two accounts, an admin account for the development purpose and a functional account for ETL routines.

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	Upgrade the driver.
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	specify the source system administrator about the desired structure of the file system.
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	Decide an archive period keeping in view the backup procedure, reprogram the ETL routine for the shifting of extracted files to archive directory and to delete archived files older than the archive period.
9	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 columns is also disturbed	Development team	Minor	Fail	Avoid the usage of semicolons, colons, commas and tabs etc. file delimiters. use some uncommon characters as delimiters like ~(tilde) and pipes
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 columns is also disturbed	Development team	Minor	Fail	Prefer column to column mapping, Try to preserve the order of columns and agree with the source DBA on the number of columns which are going to be extracted.
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	Minor	Fail	Arrange a meeting with the source DBA and decide a mutual agreed upon schema.

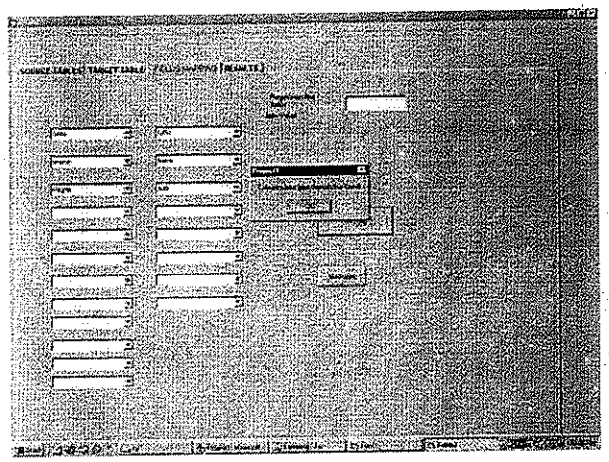
12	Extracting Relational Databases	ETL approaches the relational database	Relational database is extracted and primary key attribute is maintained	Violation of the Primary key	Development team	Minor	Fail	The most probable reason may be that the specified primary key in the source table is not declared as a primary key in the target database
13	Extracting Relational Databases	ETL approaches the relational database	Iterative extraction of only modified rows is possible	There is no well defined criteria to download only changed rows from the source system	Source DBA	Minor	Fail	The source table must have time stamp columns, identity columns or transaction dates etc. so that one should keep track of last updated records, last successfully read record etc.
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	Minor	Fail	Make sure that time stamp should be updated every time the row in the table changes.
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	Minor	Fail	Provide read only access to the ETL routines so that the ETL should not spoil the source data.
16	Extracting Relational Databases	ETL approaches the relational database	ETL can fetch the information about every record from related tables / every related table is reachable from the main entity source table	Related tables are not reachable from the main source table	Development team and Source DBA	Minor	Fail	Make sure that the related tables are reachable from the central source table
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	Minor	Fail	Count of source values= count of values appended + count of values in the error log

18	Extracting Relational Databases	ETL approaches the relational database	Data Extraction is completed within the specified time window	some error has occurred and the ETL routine is waiting endlessly to complete its job	Development team	Minor	Fail	When the ETL routine should extract data from a particular source system it needs to be completed within a certain time slot.
19	Extracting Relational Databases	ETL approaches the relational database	The ETL log is updated with every ETL transaction failure keeping the older entries.	With every new incremental access of the ETL the older log files get deleted	Development team	Minor	Fail	Either decide the time for which a log file has to be archived or develop a procedure to automatically backup log files before starting next incremental ETL request.
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 tables	Development team and Source DBA	Minor	Fail	If a record in the source table is updated then this updation should also be represented in the target table, otherwise the data in a data warehouse will not be reliable
21	Extracting Relational Databases	ETL approaches the relational database	Data is extracted without Lexical anomalies	Lexical or Syntactical anomalies are present for example a name is represented by a numeric value	Development team	Minor	Fail	Implement a strong mapping procedure to enforce strict data type and instance value checking
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	This category specifies those errors where the value of a given attribute does not conform to the anticipated domain. Generally they specify a correct record in a wrong format like last name shown in lieu of first name and vice versa. Efficient mapping procedure of ETL routine can handle such errors smartly.

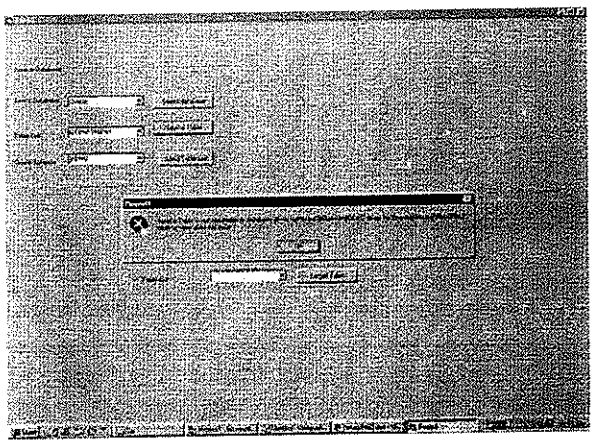
23	Extracting Relational Databases	ETL approaches the relational database	The ETL routine is capable of handling irregularities in data representations	Irregular representations are there	Development team	Minor	Fail	The ETL routine should consolidate the data representations in the target database.
24	Verifying ETL Functionality	User tries to access ETL staging area	The Staging area is not accessible by non Administrator users	Any User can access Staging area	Development team	Minor	Fail	Any unauthorized access to the data staging area may cause critical data quality issues
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	Minor	Fail	Improper transformations may lead to bogus records
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	The target database DBA might have granted read only permissions because of security reasons
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	Minor	Fail	Backup is required to successfully retrieving data in case of transaction failure
28	Verifying ETL Functionality	The backed up data has to be recovered	The Backed up data is in stable form and ca be retrieved easily.	The retrieval algorithm is not defined	Development team	Minor	Fail	A powerful Retrieval algorithm is desirable for reviving the backed up staging area data



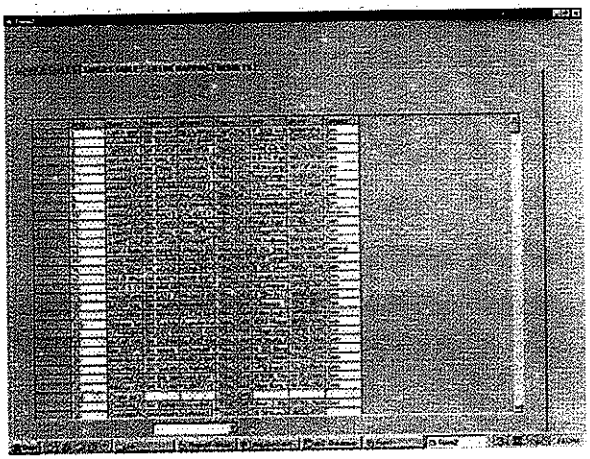
Snapshot 1 Working of the ETL Prototype (Specification of source and target databases)



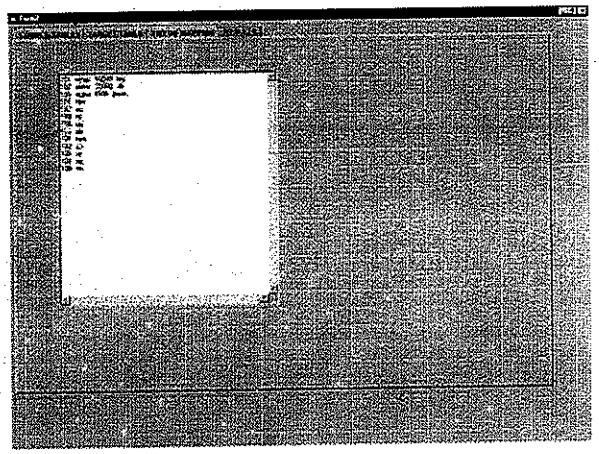
Snapshot 2 Working of the ETL Prototype (Field Mapping)



Snapshot 3 Working of the ETL Prototype (Automated testing procedure)



Snapshot 4 Working of the ETL Prototype (The source and target table views)



Snapshot 5 Working of the ETL Prototype (Results showing rejected records

In the above said prototype the authors have automated all these prime test cases along with many others. The results produced by this prototype were satisfactory as the quality of resulting database improved considerably. It has further been observed that if quality checks are imposed during the data extraction stage than the effort required to refine and transform data can be reduced considerably resulting in the saving of time and money. Business houses are investing heavily on new generation databases so as to gain competitive edge. As with any development project, a data warehouse also needs testing. The complexity and size of data warehouse systems make

comprehensive testing both "more difficult and more necessary". 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. The scarcity of this fully scaled test data is again a crucial problem hence one may use a test data generator tool for generating synthetic test data.

Conclusion & 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 begin with generating a complete record of the source systems. 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 the authors have observed the enhancement in the data quality of the target database but still the statistical analysis of the results is under process and the authors are hopeful to statistically prove the impact of ETL on data quality of a data warehouse.

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. kismeta.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, PP. 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, PP. 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.

Author's Biography

Jaiteg Singh is a research scholar from Punjabi University Patiala, Punjab, India. His area of research includes Data Warehousing and Testing.

Dr. Kawaljeet Singh is Director, UCC, Punjabi University Patiala, Punjab, India. His area of research includes system simulation and Data Warehousing.