Chapter 1

An Introduction to Data Warehouse Testing

1.0 INTRODUCTION

Since 1970’s, organizations focus on information systems that make use of operational data to support decision-making, as a means of gaining competitive advantage. This requires support for high-level decision making through applications, which can analyze and explore current and historical data for identifying useful trends and creating ready-to-execute summaries. Pedagogically, these systems were referred to as data warehouse systems. The concept of data warehousing, although, not always under that name has been gaining-in popularity. Big organizations identify data warehouses as an important strategy to integrate heterogeneous information sources even from outside the organization, and to enable On-Line Analytic Processing (OLAP). The most successful corporate houses in India, a brief synopsis of which has been given below, are thriving on the might of Information and Communication Technologies.

a) Asian Paints has been using technology for efficient data collection, demand forecasting and for reduction in working capital and to congregate online information about material flows across factories and other locations.

b) TELCO, India’s largest commercial vehicle manufacturer is having 130-strong dealer networks online with the company’s Internet-based system. It
is also working on with a few banks to establish payment gateways between the value chain at the company-end and banks. Effectiveness of marketing campaign, product sales forecasting and capacity planning are some of the applications that gives strategic edge to the company.

c) Godrej Consumer Products Limited wanted to extract more value out of the well-formatted ERP data it had collected over the years. It implemented data warehousing and OLAP tools and applications to mine additional benefits and it also paved the path for future e-commerce initiatives.

d) The National Stock Exchange of India is the second Stock Exchange in the world that has developed a data warehouse to analyze trading and payment patterns of members of the exchange, based on well defined risk classes for risk containment and maintaining market integrity.

e) ICICI Bank is offering a host of services to its customers.” Any time any where” banking, electronic bill payment services and ATMs.

Consequently, there is an escalating trend in the use of countless data-warehousing engines (from both established database vendors as well as new players). Unfortunately, majority users are not confident with the accumulation of data to be used for analysis. The ever growing fierce competition to capture the global markets has forced many organizations to equip with information systems and informative technologies. Information systems have colossal potential to transform business processes for achieving major improvements in quality, performance and productivity through the optimal use of information resources. The ongoing
improvements in the design and increasing use of advanced information systems have helped the enterprises to improve operating efficiency and reduce costs. Earlier the multi national organizations were using information systems for automation of business processes but then small and mid sized organizations too implemented information systems to gain competitive advantage over the rival organizations. With the induction of computer based business process automation, the quick mounding up of enterprise-wide data resulted in data explosion.

The global business was facing information crises as the information systems of that era were capable of processing easy day to day transactions only and strategic information was required to devise business tactics for achieving targets and to monitor outputs on prime investments. Hence a system was needed, which can process data from numerous different resources and should produce accurate and timely information to make strategic decisions, such systems were later identified as data warehouse systems. Organizations are investing in data warehouse solutions since 1990’s. Such investments are motivated by the requirements of organizations to make better timely decisions.

Data warehouses repository can be used to support user queries, reporting, on-line analytical processing (OLAP), DSS/EIS, and data mining purposes. Data warehouses are playing a dominant role in some pivot business initiatives like enterprise resource planning, customer relationship management, electronic commerce, and supply chain management etc, are a few to specify. A short list of industries where data warehouse solutions has been used is shown table 1.1.
<table>
<thead>
<tr>
<th>Industry</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer Services</td>
<td>Forecasting, Inventory replenishment, marketing campaigns, Market analysis, Sales and marketing Research</td>
</tr>
<tr>
<td>Economic Services</td>
<td>Choosing better marketing partners, Innovative and successful new plans &amp; products, Market trend Analysis, Reduction in production costs, Spotting spending patterns and Trend forecasting etc.</td>
</tr>
<tr>
<td>Natural Resources</td>
<td>Inventory, Sales ratios, Identify quality problems, Order-book ing, Delivery performance, Identify changes in product demand, Working capital and Capacity planning.</td>
</tr>
<tr>
<td>Retailers</td>
<td>Checking fraudulent transactions, Better understanding customers and their buying patterns, Promoting brand loyalty, Managing customer contacts, Market basket analysis, Competitive Advantage analysis</td>
</tr>
<tr>
<td>Production</td>
<td>Analyzing business trends, Portfolio management, Product bookings, Contract negotiations, Demand forecasting, Detection of warning situations, Product shipments,</td>
</tr>
<tr>
<td>Cyber Operations</td>
<td>Fraud and error detection, Transaction reporting/ acknowledgement, Analyzing shopping trends, Customer Relation Management, Finding new site capability etc.</td>
</tr>
<tr>
<td>Healthcare &amp; Insurance</td>
<td>Defining characteristics of profitable business, Fraud and Abuse detection, Business and client retention, Evaluating hospital services, Offering profitable healthcare insurance plans, Detecting inappropriate tests, Analyzing medical errors and injuries etc.</td>
</tr>
</tbody>
</table>

Table 1.1 Brief Summary of Industry wise utilization of Data Warehouse Applications
1.1 DISSECTION OF A DATA WAREHOUSE

Dissection of data warehouse architecture can reveal vital information about its components. In reference to figure 1.1 the data warehouse architecture can be stated as an integration of four independent components/systems. These autonomous systems exhibit virtual layers of the data warehouse in which data from one layer are derived from data of the previous layer.

Figure 1.1 Dissection of a Data Warehouse

As shown in figure 1.1 a warehouse environment has five major subsets/systems. The four virtual layers are required to define the boundaries for these subsets. In between these boundaries the sub systems are liable to perform following activities [104].
1.1.1 Source Data Acquisition

Every computer based system follows the fundamental rule of garbage in, garbage out hence source data acquisition is indispensable for quality data warehouse solutions. This subsystem digs out data from numerous sources with varying platforms and schemas/structures. Following are the issues associated with successful acquisition of source data.

a) Source credentials: categorize source applications and understand source structures.

b) Mode of data extraction: as per the structure and type of source data one has to decide whether the extraction process should be manual or tool-based.

c) Time constraints: online databases can not go offline beyond a certain time length; hence it is required to denote a time window for the data extraction process.

d) Job scheduling: in order to acquire data from numerous different sources, strict time constraints are to be followed hence scheduling the execution of data extraction processes is necessary.

e) Exception handling: exception denotes an event which is not expected in application environment. During source acquisition the data extraction routines may also encounter such events. A remedial measure for such events is necessary for successful data acquisition.

1.1.2 Data Scrubbing

Data staging component acts as a host for data scrubbing and transformation activities. Data scrubbing/cleansing is vital for quality data warehouse management. Data captured from traditional OLTP systems (online analytical processing systems) and data marts may contain incorrect, incomplete, inconsistent and unwanted
duplicate records; they need to be transformed and cleansed before they are loaded into the data warehouse. Further varying formats and representations used in various sources make the cleansing process essential in order to get correct and qualitative data into the data warehouse. The data scrubbing component of data warehouse is accountable for performing following activities:

a) **Transformations:** this process transforms a variety of data representations within incoming data streams as per data warehouse standards.

b) **De-duplication:** it is responsible to capture and eliminate duplicate information.

c) **Sanitization:** it is responsible to enrich data with correct and valid values. This component reconciles differences among multiple sources due to the use of synonyms and homonyms or different units of measurement.

1.1.3 **Data Loading**

Inordinate time is required to migrate data from staging area to the data warehouse. This data migration is managed by dedicated loading routines. These loading routines are of great concern because during the loading procedure data warehouse has to go offline. Therefore the whole loading process is divided into smaller chunks and a few files are populated at a time. The whole loading procedure can be classified into following categories:

a) **Initial Load:** it is concerned with populating the DW tables for the very first time.

b) **Incremental Load:** it is repetitive in nature and is responsible for applying ongoing changes as required in a periodic manner.

c) **Complete data reset:** these loads are rare in nature and are responsible to refresh the complete data within a table during recovery procedures.
1.1.4 Data Storage and Placement:

The loading routines are responsible for placing the cleansed data from staging area into the data warehouse in accordance to any fundamental data design technique. The data design literature identifies two basic data models known as star schema and snowflake schema. In accordance to these schemas the incoming data has to be placed in two types of tables known as fact and dimension tables. A fact table represents those attributes, which represents the quantitative measure of any subject of interest. On the other side a dimension table stores the textual description of any identified fact of business interest [82]. The relationship among fact and dimension tables using star and snowflake schemas is shown in figure 1.2 and 1.3 respectively.

1.1.4.1 Star Schema: The star schema is the simplest data warehouse schema. It is called so because of its resemblance with a star. It is simple to implement because it requires only one join to query the database.

![Figure 1.2 The Star Schema](image-url)
1.1.4.2 **The snowflake schema:** it is a variation of the star schema used to store data in a data warehouse. It is comparatively complex because the tables, which describe the dimensions, are normalized and require multiple joins to query data from a data warehouse.

![Figure 1.3 The Snowflake Schema](image)

1.1.5 **Information Delivery:** The information delivery component facilitates users to access the information within the Data Warehouse. The Data Warehouse has to satisfy users from all levels of management. It is responsible for providing information to the users as and when required by means of reports, Excel pivot tables, OLAP and data mining technologies.
Keeping in view the involvement of the divergent source systems, evaluation of prevailing technology, lack of training and expertise and rigorous need of requirement analysis, one can consider data warehouses as a challenging technical undertaking. The availability of complicated and advance data warehousing options it has been a burdensome task for many companies to evaluate and select data warehouse system that can fit into their budget and time constraints. The complexity of data warehouse architecture makes it even harder to test the data warehouse. Generally the tools used in data warehouse development belong to different vendors and they may introduce severe compatibility issues if not tested properly. Data warehousing applications are dynamic in nature and that is why they keep on changing with shifting business requirements. Data warehouse in real sense is not a product it is an architecture which can be customized according to needs of the business implementing the data warehouse. The absence of real data for testing and analysis, the tight time lines and the scarce literature about testing methods deployed in data warehousing has further enhanced the complexity of data warehouse testing. Testing is crucial to the success of a data warehousing project because the business organizations need to trust the quality of the information they access. Although detailed literature on each phase of data warehouse design is available but much less has been discussed about data warehouse testing. This might be because of the fact that traditionally data warehouse testing is considered equivalent to that of application testing. Unlike application testing which is dedicated to the concerned application only, the data warehouse testing has to comply with much higher standards of quality assurance for data as well as applications involved. On following grounds one can justify the demand of a separate testing model for data warehouse applications.
a) Software testing is primarily focused on testing the program code, while data warehouse testing is concentrated on data and information.

b) The prime focus of data warehouse testing is to know the data and the query requirements of the user, they are supposed to answer.

c) Unlike generic software systems testing, data warehouse testing involves a huge data volume, which ultimately impacts performance and productivity of data warehouse.

d) Data warehouse testing has a broader scope than software testing because it is focused on the validation of information delivered to users. In fact, data validation is one of the main goals of data warehouse testing.

e) Although general software systems have large number of different use case scenarios, but the valid combinations of those scenarios are limited. Contrarily the data warehouse systems are aimed at supporting any views of data, so the possible combinations are virtually unlimited and cannot be fully tested.

f) Most testing activities in generic software systems are carried out in house before deployment, where as data warehouse testing activities may go on even after system release.

g) Software development projects have a well defined boundary but data warehousing projects are virtually everlasting because it is almost impossible to anticipate future requirements for the decision-making process, so only a few requirements can be stated from the beginning.
h) It is almost impossible to predict all the possible types of errors that a data
warehouse may encounter in real operational data.

1.2 DATA WAREHOUSE TESTING INITIATIVE

Testing is a process of locating errors within a system and keeping in view the
aforesaid issues one may find it hard to define the testing sequence for a data
warehouse. Although a stepwise procedure, which includes Unit Testing, Integration
Testing and System testing can be followed but still exhaustive testing for a data
warehouse is yet not feasible [4]. Aforesaid testing categories are discussed as
under:

1.2.1 Unit Testing

Unit testing a data warehouse includes validation of mappings performed during
Extraction, Transformation and loading process. It further verifies the authenticity of
front-end reports generated by the data warehouse system. Following tests can be
carried out during unit testing:

a) Whether extraction methods are having approach to the required source data.

b) To check that all the business rules are followed properly and the quality of
data entering data warehouse after transformations is acceptable.

c) Testing for slowly changing dimension like the marital status of a customer.

d) Checking of format specifications.

e) To check whether the data from data warehouse tables/schema is reflected
correctly in reports.

1.2.2 Integration Testing:

Integration testing is associated with the second phase of data warehouse quality
assurance. Testing of individual units may rectify faults local to any sub system, but
it does not guarantee a flawless data warehouse. A data warehouse system comprises
independent subsystems/modules, which needs to interact with each other for
answering user queries. Integration testing is the activity of exercising such
interactions by hauling together the different modules composing a data warehouse
system. It is characterized by involving different interacting units which have been
in general developed by different programmers. In case of Extraction,
Transformation and Loading individual jobs are made to run as a batch. This is
usually done by using advanced batch facilities in the ETL tool. Finally a series of
batches are generated to run in a single shot. Integration testing is performed to test
the following conditions:

a) The sequence of batches during the single shot runs.
b) Generation of error logs.
c) Impact of failure of a job on the subsequent batch jobs.
d) Processing of altered records which are updated at a later stage after being
   initially processed.
e) Automatic or manual restarting in case of any failure.
f) Processing of rejected records.
g) Generation of logs created at the end of every process.

1.2.3 System Testing

System and acceptance testing are basically different from each other. But it is
advantageous to combine the two phases in case of tight time lines and budget
constraints. Here the system is tested for its full functionality keeping in view the
requirements of the user. At the end of this phase the system should be acceptable to
the client in terms of ETL process integrity and business functionality. It is
necessary to understand the working of the business process for verifying the
functionality of the data warehouse application. The major challenge here is preparation of test data. An intelligently designed input dataset can bring out the flaws in the application more quickly. Wherever possible use near to real data. Test data generators are a good option to create synthetic test data but it should ensure all possible combinations of input to specifically check out the errors and exceptions. An unbiased approach is required to ensure maximum efficiency.

1.2.4 Scope of Automated Testing in Data Warehouses

Automation is generally a blend of Errands automation and Tests automation. Under Errands automation in data warehouse, there is a need to do the pre-requisite steps before starting the ETL procedure. For example if one has to extract data from flat files then there is a need to bring the flat files on to a landing platform by using copy operation. It is followed by the remodelling of all the files in a format acceptable to the ETL module. The test automation process can be considered as a three phase procedure. After initial processing automated ETL can be launched to read the data from flat files and to load it on to a staging area where data scrubbing can be performed. Here good automation logic is required to validate the loading process.

Test automation is not much different from errand automation as both proceeds in parallel. Test automation is concerned with the implementation of business logics. It further ensures that the automated errands are behaving as expected. Automation of data warehouse testing has immense potential to enhance the quality and performance of the data warehouse systems.

Data warehouse testing is substantial as it is oriented towards the correctness and validation of data/information supplied for decision making. Keeping in view the
idiosyncratic characteristics of data warehouse testing and the complexity of data
warehouse projects, there might be a scope to revise and contextualize the traditional
testing strategies. The data warehouse realization involves interaction among a
number of autonomous sub systems. As these systems are operational within the
boundary of data warehouse hence it is extremely hard to test such virtually
independent but conceptually unified systems. The ETL (Extraction, Transformation
and Loading) system is probably the most complex and critical system from testing
perspective. ETL is responsible for consolidation and synthesizing data into a data
warehouse according to prescribed business rules. Involvement of complex business
rules and interaction with numerous data sources makes the ETL testing both
important as well as difficult. ETL unit testing can either be performed
diachronically or synchronically. In diachronic standards for ETL testing it is
essential to write test cases for every identified fact with correlated dimensions of
the concerned entity domain. Whereas synchronic standards lay emphasis on
separate test cases for extraction, cleansing, transformation and loading procedures.
The ETL integration testing verifies the correctness of data flows.

However data warehouse testing is an everlasting process still one can consider
the following propositions for performing data warehouse testing:

a) Test Input before Output: Traditionally the emphasis of testing is on
analyzing output, not input. The entire digital world follows the theme of
“garbage in, garbage out”. Verifying the input as well before analyzing the
output may save a lot of effort and money involved in data warehouse
testing.

b) Evaluate Source and Target Data: The business users may not believe in
the newly developed data warehouse until they are sure of its information
quality. Users generally cling to their comfort zone and may continue to rely upon information from legacy systems until they are convinced that the new data warehouse provides the better quality information in an easier way.

c) **The User’s Concern:** The business users are the sole reason for the data warehouse existence. If the users are not involved in testing the system it may finally fail at some level. The user concern is essential because the user recommendations and preferences are necessary for the customization of any data warehouse application. As the acceptance testing will ultimately be done by the users hence it becomes necessary to involve the users from designing till testing.

d) **Choice of testers:** In the final expanse of the data warehouse development, there is a strong impulse for shedding the development costs. This may shift the testing responsibilities to resources with limited technology and business experience. This practice should be avoided as to develop a quality data warehouse it is essential to test it thoroughly. The people involved in the system design including the user representatives should continue to guide the testing methodologies. A wrong choice of testers may guarantee system failure.

e) **The ETL Effort Estimate:** Although because of ready to use ETL routines, a minimal amount of code may be written for the testing purposes, but this can be very misleading. A good ETL effort estimate can foresee possible data quality issues and can further rectify such problems in advance because addressing such issues at a later stage may prove to be much more time consuming.
f) **The Response Time:** The response time is very crucial for any data warehouse project. The development team has to ensure that the ETL load and processing time should fit within the existing batch window. Judging the response time may require careful thinking and simulation of the test environment.

The Extract-Transform-Load (ETL) system is the foundation of the data warehouse. A properly designed ETL system is capable of extracting data from numerous source systems, it can enforce data quality and consistency standards, can refurbish data to use separate data sources jointly, and finally delivers data in a presentation-ready format so that application developers can build applications and end users can make better decisions.

1.3 **NEED OF THE STUDY**

It has been observed that many companies who failed to completely retire their outdated systems may now be maintaining duplicate data. Data redundancy dilemma is further exacerbated as companies developed independent systems which were limited to different core functional business subjects. This means every strategist in the organisation is maintaining its own database. Making use of different application software as per the particular demand of every strategist in the organisation has also been a part of the cause to the proliferation of redundant data. To shift operations to robust open-system architecture from the prevailing legacy system, ETL has an important role to play in moving the data from relational tables to dimension tables (altogether supported with different data structures). This problem can further be elaborated with the help of figure 1.4.
The figure above highlights the flow of data within a data warehouse. Data redundancy initiates from Product system and follows through the warranty system, sales system, production system, marketing system and eventually the data warehouse. Accordingly, superfluous data is to be extracted and loaded. Also, this data need transformation because the data structures of participating systems are completely diverse. This ultimately results in complicated and unnecessary ETL development and maintenance of redundant data [5]. To further aggravate the problem, one minor structural change in the source can create an expensive maintenance nightmare in the ETL maps and the target database with no value addition. Data transformations are needed to support any changes in the structure, representation or content of data so as to deal with integrity constraints, or to migrate a legacy system to a new information system, or when multiple data sources are to be integrated. These data problems can be largely divided into two parts namely single source problems and multi source problems.
1.3.1 Single data site Issues: The data quality of a source largely depends on the degree to which it is governed by schema and integrity constraints for controlling acceptable data values. For sources without schema, such as flat files, there are few restrictions on what data can be entered and stored, giving rise to a high probability of errors and inconsistencies. Database systems, on the other hand, enforce restrictions of a specific data model for example the relational approach requires simple attribute values, referential integrity, etc. as well as application-specific integrity constraints. Schema-related data quality problems thus occur because of the lack of appropriate model-specific or application-specific integrity constraints, e.g., due to data model limitations or poor schema design. Instance-specific problems relate to errors and inconsistencies that cannot be prevented at the schema level e.g. misspelling. These problems can be understood by analyzing the Table 1.2.

It has been observed that different problem scopes always tell the difference between schema and instance level problems as shown in Tables 1.2; one may find it interesting to observe that uniqueness constraints specified at the schema level do not prevent duplicated instances, e.g., if information on the same real world entity is entered twice with different attribute values then it is not easily identifiable. Additionally, it is almost impossible to clean the source data but preventing the way-in of dirty data is obviously an important step to reduce the cleaning problem. This requires an appropriate design of the database schema and integrity constraints along with a robust design of the data entry applications. Also, the discovery of data cleaning rules during warehouse design can suggest improvements to the constraints enforced by existing schemas.
1.3.2 Multiple data site Issues: The problems present in single sources are aggravated when multiple sources need to be integrated. Each source may contain dirty data and the data in the sources may be represented differently, overlap or contradict. This is because the sources are typically developed, deployed and maintained independently to serve specific needs. This results in a large degree of heterogeneity with respect to data management systems, data models, schema designs and the actual data itself. The main problems with respect to schema design are naming and structural conflicts. Naming conflicts arise when the same name is used for different objects (homonyms) or different names are used for the same object (synonyms). Structural/Schema-level conflicts occur in many variations and may refer to different representations of the same object in different sources, e.g., attribute vs. table representation, different component structure, different data types, different integrity constraints, etc. In addition to schema-level conflicts, many conflicts appear only at the instance level and are known as data conflicts. All problems from the single data site case may occur with different representations in different sources. Furthermore, even when there are the same attribute names and data types, there may be different value representations or different interpretation of the values across sources. A main problem for cleaning data from multiple sources is to match records referring to the same real-world entity. This problem is also referred to as the object identity problem. From the above specified facts it has been observed on our part that the promulgation of contemporary testing techniques can be of great use for effective engineering of data warehouses.
<table>
<thead>
<tr>
<th><strong>Problem</strong></th>
<th><strong>Dirty Data</strong></th>
<th><strong>Reason</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Illegal values</td>
<td>bdate=30-2-80</td>
<td>Value of date field is not acceptable</td>
</tr>
<tr>
<td>Violated attribute dependencies</td>
<td>age=22, bdate=12.02.80</td>
<td>age = (current date (Date of recording the transaction)– birth date) should be true</td>
</tr>
<tr>
<td>Uniqueness violation</td>
<td>emp1=(name=&quot;jai Singh&quot;, pan no.=&quot;123456&quot;) emp2=(name=&quot; Dr. Kawaljeet Singh&quot;, pan no.=&quot;123456&quot;)</td>
<td>uniqueness for PAN( Permanent Account Number) violated</td>
</tr>
<tr>
<td>Referential Integrity violation</td>
<td>emp=(name=&quot;JTS&quot;, deptno=Mgmt)</td>
<td>referenced department Mgmt may not exist</td>
</tr>
<tr>
<td>Missing values</td>
<td>phone=9999-999999</td>
<td>Data recorded is representing dummy values.</td>
</tr>
<tr>
<td>Misspellings</td>
<td>city=&quot;Putialag&quot;</td>
<td>Such entries usually result from typographic, phonetic errors</td>
</tr>
<tr>
<td>Cryptic values,</td>
<td>experience=&quot;B&quot;;</td>
<td>‘B’ may not have a toning definition across multiple sources.</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>occupation=&quot;Tchr.&quot;</td>
<td>Tchr. may not have a global system definition.</td>
</tr>
<tr>
<td>Embedded &amp; misfiled values</td>
<td>name=&quot;Jaiteg singh 12.02.70 patiala&quot; city= Delhi</td>
<td>multiple values entered in one attribute, Patiala is not in Delhi</td>
</tr>
<tr>
<td>Violated dependencies</td>
<td>city=&quot;patiala&quot;, pin=77777</td>
<td>city and pin code should correspond</td>
</tr>
<tr>
<td>Substitutions</td>
<td>name1 = “j singh” substituted with name2 =&quot;Jaiteg singh.&quot;</td>
<td>Such entries usually result from a free-form field</td>
</tr>
<tr>
<td>Duplicated records</td>
<td>emp1=(name=&quot;Jaiteg Singh&quot;, eno: 123456); emp2=(name=&quot;J Singh&quot;, eno:123456)</td>
<td>same employee represented twice due to some data entry errors</td>
</tr>
</tbody>
</table>
Table 1.2 Possible Data Quality Issues

<table>
<thead>
<tr>
<th>Problem</th>
<th>Dirty Data</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contradicting records</td>
<td>emp1=(name=&quot;Jaiteg Singh&quot;, bdate=12.02.70); emp2=(name=&quot;Jaiteg Singh&quot;, bdate=12.12.70)</td>
<td>The same real world entity processes different prime values</td>
</tr>
<tr>
<td>Wrong references</td>
<td>emp=(name=&quot;Jaiteg Singh&quot;, deptno=9)</td>
<td>Deptno 9 may exist but emp may not be working there.</td>
</tr>
</tbody>
</table>

1.4 SCOPE OF THE STUDY UNDERTAKEN:

The ETL system can make or break the data warehouse. Although building the ETL system is considered as a back room activity and is not very visible to end users but according to experts ETL consumes nearly 70 percent of the resources needed for implementation and maintenance of a typical data warehouse. The main objective of this study was to understand the existing data warehouse architecture which may further help in refining the existing architecture for effective engineering of data warehouses; the research is also intended to highlight data quality issues identified during the extraction, transformation and loading (ETL) procedure while populating data; so as to ensure data quality and integrity. This research follows an incremental model to mimic data warehouse environment. The findings from literature review were used to fabricate blueprints for data set generator for generating synthetic test data, to hand code an ETL routine and for proposing web technology as a way-out for managing data distributed globally. Further the fabricated blueprints were tested empirically in this research. Keeping in view the aforesaid need for study the objectives of this research can be postulated as:

a) Understanding Data Warehouse Architecture.

b) Understanding the ETL framework through hand coded ETL routine.
c) Identifying prime ETL testing zones.

d) Statistically analyzing the impact of automated ETL testing on data quality.

e) Proposing a refined architecture for effective management of distributed data.

Akin to other research studies, this research has some limitations too. This study has been carried out under de rigueur constraints of time and resources. Through extensive literature review, an effort to integrate all available literature was made yet understanding and findings may have been constrained by the vision of the researcher.

As the data warehouse development requires expensive data servers and hardware that can store terabytes or even pentabytes of data with extremely fast processors to ensure fast information retrieval. Being unable to afford such expensive hardware and patented platforms needed to develop a data warehouse, this study is restricting itself to simulating a prototype of data warehouse. As the ETL tools and services offered by big brands do not allow to study or analyze their internal structure and functionality hence it was decided to hand code an ETL routine for extracting data from divergent logical sources. The hand coded ETL routines are always domain specific and are constrained to the business rules for an organization. This research has developed an ETL routine to manage personal records data. The prototype is built on a standalone personal computer and the windows IIS (Internet Information Server) utilities were also used to masquerade networked environment. Although the prototype is flexible in selecting the source database type among the predefined list of database management systems but MS access is opted as the only available data sink. Microsoft access was selected as targeted database because of its unmatched compatibility with Visual Basic 6.0, used
to develop the front end of the prototype. After the induction of automatic quality assurance in hand coded ETL, the effectiveness of ETL testing procedures has been adjudged in terms of the number of dirty records populated in the target data warehouse.

Every human creation is structured and patterned and the data warehouse is no exception to that. No single term/object carry any significance. The data warehouse too comprises multifarious terms and objects. A data warehouse signification pre-supposes the existence of a relation. So the conjunction and disjunction are very important to decode the terms and objects prevalent in the data warehouse. The two terms/objects can be distinguished uniquely only if they have a simultaneous correlation and opposition. Hence the terms/objects in the data warehouse can be studied on the basis of a research methodology designed synchronically and diachronically. Every structure has a synchronic and diachronic pattern. The synchronic pattern describes the state of a variable studied at any point of time and are designed to answer what? clause. On the other hand a diachronic pattern demonstrates the evolution/innovation of technology across a period of time. From the aforesaid discussion it can be inferred that the synchronic-Diachronic research method may best suite to examine and evaluate the possible contemporary testing techniques and further typological and metric analysis may be useful to reconstruct comparative syntagm and paradigms. The inferences thus derived can be used for the precision test to enhance quality, authenticity, effectiveness and trouble-free operation of data warehouses.