

# ORGANIZATIONAL DATA WAREHOUSE : AN EVALUATION OF TESTING TECHNIQUES

Raj Kumar\*, Ajay Kumar Singh\*\*

---

## ABSTRACT

*This article is a position paper on the nature of the data warehouse exhaustive testing of a Data warehouse during its design on an ongoing basis for the incremental activities comprises data warehouse testing. This type of testing is getting common today. The reasons for this are manifold, prominent ones being increase in Enterprise Mergers & Acquisitions, Data Center Migrations, Increased Compliance regulations, Senior Management's increased focus on data and data driven decision makings. The paper focuses on different components in the Data warehouse architecture, its design and aligning the test strategy accordingly. The main components include the ETL process, OLAP engine and other client applications residing on the data warehouse. Data Storage has become cheaper and easier compared to the file-storage days. Hardware has become cheaper and Data Mining solutions are easily available. Data driven decision have proved to be more accurate. In this context, testing the Data warehouse implementations are also of utmost significance. Organization decisions depend entirely on the Enterprise data and the data has to be of utmost quality! Complex Business rules and transformation logic implementations mandates a diligent and thorough testing. Finally, the paper addresses some challenges for DWH Testing like voluminous data, heterogeneous sources, temporal inconsistency and estimation challenges.*

## INTRODUCTION

A Data warehouse is a composite and collaborated data model that captures the entire data of an organization. It brings together data from heterogeneous sources into one single destination. It is not just bringing together. Data is Extracted, Transformed and Loaded (ETL) into the Data warehouse. This processing of the data is usually done in what is known as a 'staging area'. The data need not be normalized, as it will be mainly for read-only purposes or basically querying and analytical purposes. Data warehouses

---

\* University department of Mathematics(IT Division), BRA Bihar University, Muzaffarpur, Bihar, rajkumar1729@yahoo.co.in

\*\* University department of Mathematics (IT Division), BRA Bihar University, Muzaffarpur, Bihar, Ajaykrsingh51@gmail.com

are mainly OLAP systems and they do not operate on OLTP data. Enterprise data comprises of multiple data residing in multiple systems. Data will be duplicated at many places; however it facilitates easy day-to-day OLAP operations. Inmon, Father of Data warehousing defines a Data warehouse as subject oriented, integrated, non-volatile & time variant collection of data in support of Management Decision.

Mergers and Acquisitions are very common in today's business arena. Every day, there are talks going on between retail giants, players in banking and insurance domains. Every quarter there are some fruitful mergers or acquisitions. What does this mean to the IT departments? Simple - Extraction, Transformation and Loading of voluminous data! There is a huge amount of Organizational data movement. Testing ensures data integrity during data movement and the data loading.

Today Senior Managements rely on Decision Support Systems for policy-making in the organization. Hence, the DSS (and the underlying data model) should have the ability to monitor historical trends, patterns and provide suggestions/conclusions. Monitoring near real-time operational data also provides the organization with the much coveted completion-advantage. It's a basic rule of thumb that the Defect detection should be as early as possible in an SDLC. This rule has utmost significance when it comes to DSS and Management Information Systems. Any defect in the data model, the warehouse implementation, the Extraction or the processing is capable of translating into disastrous decisions by the Organization.

Financial Institutions, Health Care institutions etc are made to comply with stringent IT policies in reference to the customer data and day-to-day transactions. Implementation of such regulatory applications may mandate ensuring of the compliance on the historical data as well. This calls for check on voluminous organization data again. Transformations will be applied and the same has to be tested thoroughly, as any slippage here will be a violation of compliance and will result in a huge loss to brand-image and finances for the company.

Data Mart can be described as a specialized subset of the Data warehouse. It caters to the data requirements of a specific group. E.g.: Finance, HR, and Marketing. It contains data which is assimilated, processed and is in a decipherable format for the end-user. Dependent data marts are those that derive the data from the underlying Enterprise datamarts. Independent data marts are those that get their data directly from the sources

Data warehousing is the process of building, maintaining a data warehouse, including the datamarts and any downstream client applications. In other/ words, it is the science



of re-structuring and integrating heterogeneous data residing at multiple sources. Data integrity is of utmost importance in Data warehousing. However, unlike OLTP systems, the data need not be normalized.

Some guidelines need to be considered in modeling a data warehouse. If implemented, these guidelines also serve in deciding the test strategy of the data warehouse.

- How extensive is the warehouse?
- Is it covering all the dimensions of the Enterprise data?
- Will it satisfy the data requirements of the underlying Decision Support Systems (DSS)?
- Is the warehouse Scalable & Robust. Will it meet the future needs of the organization?

### DATA WAREHOUSE TESTING

A good understanding of data modeling and the need for the data warehouse will equip the test analyst with guidelines for coming up with an apt testing strategy. Hence, it is very important that during the Requirement Analysis phase, importance must be given in understanding the Data warehouse implementation to the maximum possible extent. Data warehouse testing strategies will, in most cases be a consortium of several smaller strategies. This is due to the nature of Data warehouse implementation. Different stages of the Data warehouse implementation require testing team's participation. Unlike traditional testing, the test execution does not start at the end of implementation. In short, test execution itself has multiple phases and is staggered throughout the life cycle of the Data warehouse implementation.

Functional testing needs to be carried out to validate the Transformation logic.

During the setup of Data Load functionality, specific testing on the load module is carried out. The Parameters and Settings for data load are tested here. Initial Data Load is when the underlying databases are loaded for the first time. Performance testing is of significance here. Data Quality, once tested and signed off during the ETL testing phase is re-tested here. Once the initial data load is done, the Data warehouse is ready for an end-to-end functional validation. UI testing and Interface testing are carried out during this phase. Data from the operational Database should be input into the Data warehouse periodically. During such periodic updates, regression testing should be

executed. This ensures the new data updates have not broken any existing functionality. Periodic updates are required to ensure temporal consistency.

## 2. DATA WAREHOUSE TESTING: FOCUS POINTS

At a high-level, any strategy should focus on the 2 main aspects mentioned below.

- Underlying Data
- Data warehouse Components

### Data Coverage

The primary test for data completeness is to ensure all the data is covered or loaded into the target. Special case data (records) exist in every application. The coverage of all such special cases and boundary cases must be ensured. This includes, but is not limited to, validating that all records, all fields and the full contents of each field are loaded. Special checks for truncation, null records etc. should be performed. The full content of each field can be populated to ensure that no truncation occurs at any step in the process.

Checksum or Record counts must be compared between the source data and the data loaded. Data rejected should also be analyzed.

This helps in pointing out a variety of possible data errors without doing a full validation on all fields. The range and the value boundaries of fields in the tables should be measured using profiling tools. Defects are hidden in the boundaries. Boundary cases should be given importance during the test case generation.

Maximum possible data scenarios should be included to ensure success in testing data quality. Typically, data quality rules are defined during design, for example:

- Reject the record if a certain decimal field has nonnumeric data.
- Non numeric values in a decimal field - this should be cleaned up, by updating the field either as a NULL or just initializing (placing zeros).
- Validate and correct the state field if necessary based on the ZIP code.

The Operational database tables will not have validations for every field in a database. E.g.: The Gender column will not have any constraints, or the UI may not mandate a user in entering Male or Female. Such data, if used for purposes like taking the % of Males, will give an incorrect count, since value 'male' may not be present for all records.



### **3. DATA COMPLYING WITH THE TRANSFORMATION LOGIC IN ACCORDANCE WITH THE BUSINESS RULES.**

Business Requirements get translated into Transformation logic. Hence, the Requirements Elicitation phase is of utmost importance. During the implementation of the transformation logic, care must be taken to consider all special cases of data. Once the data is transformed, thorough testing has to be executed to confirm that the underlying data complies with the expected transformation logic. Multiple techniques can be adopted for the same. Sampling is one technique, where in sample records are picked up and compared to validate data transformations. A combination of automated data profiling and data feed is another better long-term strategy. This ensures more test coverage.

#### **Performance and scalability**

Performance and Scalability testing ensures that loading of the initial data and subsequent queries on the same does not kill the system and are within acceptable performance limits. This also ensures that the system is scalable and can sustain further growth. As the volume of data in a data warehouse grows, ETL load times can be expected to increase and performance of queries can become a concern. This can be avoided by having in place a scalable architecture and good ETL design. The aim of the performance testing is to point out any potential weaknesses in the ETL design, such as reading a file multiple times or creating unnecessary intermediate files.

#### **The following strategies will help discover performance issues:**

Load the database with peak expected production volumes to ensure that this Volume of data can be loaded by the ETL process within the agreed timeframe.

Scalability of the system can be ensured by executing ETL loads at different volumes and comparing the time taken for different loads. A study of this comparison will help in measuring the scalability of the system. Compare the ETL processing times component by component to point out any areas of weakness. E.g.: For a DWH system implemented using abinitio, test each of the 'graphs' rather than treating the entire set of graphs as a 'black box'.

Perform simple and multiple join queries to validate query performance on large database volumes. Work with business users to develop sample queries and acceptable performance criteria for each query.

### Component Orchestration Testing

This covers the testing of multiple Data warehouse components together, in an integrated fashion. Ensures that the ETL process functions well with other upstream and downstream processes. A Data warehouse implementation is a collection of many components, and hence integration testing is a must.

Typically, system testing only includes testing within the ETL application. Integration testing shows how the application fits into the overall architecture. Individual components behaving correctly is no guarantee for the entire system to behave as expected. Integration brings with it many new issues like 'resource conflict', 'deadlocks' etc.

Most issues found during integration testing are either data related or resulting from false assumptions about the design of another application. Production data is ideal for any testing; however Regulatory requirements might prevent use of production data. Next best option is to use data closest to the LIVE data. An Effective Requirements Elicitations phase will ensure that no gaps are left out. To help bridge this communication gap, ensure the involvement of every team (all components, all up/downstream applications) from the initial stages of the project itself.

### Regression Testing

Data warehouse implementation is not a one time job. This requires continual up gradation as the Operation Data stores change, and also periodic data loads are necessary. A well-defined, baselined Regression suite ensures existing functionality remains intact each time such a change happens.

Regression test will be executed multiple times and hence it's a good idea to automate the regression testing. Multiple test packages can be created based on the priority business scenarios. Smaller changes may not require execution of the entire regression suite. Test cases can also be prioritized as per business risk and importance. This helps in deciding which test cases to be run for each of the new releases or changes to the up/down stream applications.

A simple but effective and efficient strategy to retest basic functionality is to store source data sets and results from successful runs of the code and compare new test results with previous runs. When doing a regression test, it is much quicker to compare results to a previous execution than to do an entire data validation again.



#### 4. DATABASE TESTING VS. DATA WAREHOUSE TESTING

The difference between a Database and a Data warehouse is not just in the data volume as we saw earlier. ETL is the building block for a Data warehouse. Data warehouse testing thus should be aligned with the Data modeling underlying a data warehouse. Specific test strategies should be designed for the Extraction, Transformation and for the Loading modules.

##### Challenges in Data warehouse Testing

- Voluminous data, from heterogeneous sources.
- Data Quality not assured at source.
- Difficult to estimate. Only volume might be available. No accurate picture of the quality of the underlying data.
- Business Knowledge. Organization-wide Enterprise data knowledge may not be feasible.
- 100% Data verification will not be feasible. In such cases, the extraction , transformation and loading components will be thoroughly tested to ensure all types of data behaves as expected, within each of these modules.
- Very High Cost of Quality. This is because any defect slippage will translate into significantly high costs for the organization.
- The Heterogeneous sources of data will be updated asynchronously. Temporal Inconsistency is part and parcel of a Data warehouse implementation.
- Transaction-level traceability will be difficult to attain in a Data warehouse.

#### CONCLUSION

Focus on Data Quality: If the data quality is ascertained, then the testing of ETL logic is pretty much Quality straight forward. One need not even test 100% of the data. The ETL logic alone

##### Identify Critical Business Scenarios

The Sampling technique will have to be adopted many times in Data ware house testing. However, what constitutes a sample? 10 % or 20 % or 60 %? Identification of Critical Business Scenarios and including more test data here is a route to success.

### Automation

Automate as much as possible! The Data warehouse Test Suite will be used time and again as the database will be periodically updated. Hence a regression test suite should be built and be available for use at any time.

This will save much of a time.

### Tools: Data warehouse testing

There are no standard guidelines on the tools that can be used for Data warehouse testing. Majority of the testing teams go with the tool that has been used for the data ware house implementation. A drawback of this approach is redundancy. The same transformation logic will have to be developed for DWH implementation and also for its testing. Try selecting an independent tool for testing of the DWH. E.g.: Transformation logic implemented using a tool 'X' can be tested by reproducing the same logic in yet another tool - say 'Y'.

Tool selection also depends on the test strategy viz exhaustive verification, Sampling, Aggregation etc. Reusability & Scalability of the Test Suite being developed is a very important factor to be considered.

Tools with built-in test strategies help in deskillling.

### REFERENCES

- [1] [ASII91] American Supplier Institute, Inc.. Taguchi Methods: Introduction to Quality Engineering. Allen Park, Mich. 1991.
- [2] [BaTa89] D.P. Ballou and K.G. Tayi. Methodology for allocating resources for data quality enhancement. Communications of the ACM (CACM), vol. 32, no. 3, 1989.
- [3] [BBBB95] D. H. Besterfield, C. Besterfield-Michna, G. Besterfield and M. Besterfield-Sacre. Total Quality Management. Prentice Hall, 1995
- [4] [Boed87] R.F. Boedecker. Eleven Conditions for Excellence. The IBM Total Quality Improvement [OiBa92] M. Oivo, V. Basili. Representing software engineering models the TAME goal-oriented approach. IEEE Transactions on Software Engineering, 18(10), 886-898, (1992). Process. Quincy, Mass. American Institute of Management, 1987.
- [5] [Wang98] R.Y. Wang. A product perspective on total data quality management. Communications of the ACM (CACM), vol. 41, no. 2, February 1998.
- [6] [www.wikipedia](http://www.wikipedia)