

Data Warehouse Testing: Why QA Projects Need Automation

Eric Ceres
RTTS
Senior Test Engineer

To successfully deploy a data warehouse, an organization must be confident that the reports generated display the proper data. The accuracy of these reports is dependent on two primary factors: the integrity of the data in the data warehouse and the proper functioning of the reporting tool. In the Pharmaceutical industry, where the correctness of Safety Data is paramount for reasons of both public health and compliance with Federal regulation, the issue of data quality takes on even greater proportions. The functionality of the reports is tested using methods similar to any front-end application, such as use of an automation tool and test scripts. The focus of this whitepaper is to explain how to ensure that the data in the data warehouse is properly loaded from the source system(s).

Executive Summary

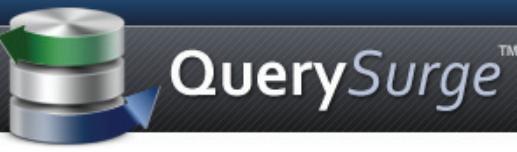
The number one concern of a Safety Data data warehouse is integrity of the data. Without reliable data, the investment in building the warehouse will provide little return for the organization. This becomes a pressing issue when testing large amounts of data that are derived from multiple sources, which can vary substantially over time. The solution presented in this paper reveals the process of automating the testing of each data element based on business requirements. This is accomplished with the use of an automation tool that extracts correlated data from a source system and the data warehouse. The tool compares the data from both systems and identifies any discrepancies. The testing team can then research any failures in the Extraction, Transformation and Load (ETL) process.

Standard Testing Approaches: Manual Sampling

Testing the integrity of the data is a crucial process in the development of a data warehouse. Though common methods for testing provide businesses with confidence in their data, a data warehouse is always at high risk for defects that may not be readily visible. This section covers several conventional techniques for testing a data warehouse to demonstrate where uncertainty in the data may appear.

End-to-end testing is the practice of entering data into the various systems from which the data warehouse extracts and then verifying that this data is correctly displayed on the reports. Specifically chosen test cases allow testing of scenarios which may or may not currently exist in the data warehouse. The limitation of this technique, however, is both in the amount and variety of data created for the purpose of testing – to fully test a complex data warehouse by this approach may require the entry of very large volumes of data. This large volume of data may need to be assembled by subject-matter experts (SMEs). In addition, warehouses may have data that the test team cannot replicate for a number of reasons; for example, due to retirement of front-end systems, records may exist in a data warehouse that can't be re-entered from the current front-end.

Another approach, which focuses on verifying metadata, is the use of row counting. To ensure that no data is lost in the ETL process, row counts on the data warehouse and source systems are performed after ETL. This works well when all records from the source are brought into the data warehouse. In this case, if the number of rows for a table in the data warehouse does not match the number of rows in the corresponding table(s) in the source systems, data is clearly being lost.



This verifies the amount of data being loaded is proper, but, as a metadata check, it does not in any way check the validity of the data. Furthermore, this approach becomes challenging to implement when the ETL process has complex business logic to filter the data that is admitted to the warehouse. In this latter case, one source record may result in writing a number of records to the data warehouse that is dependent on the data content of the source record.

Aside from rows being lost, another metadata-related problem that may occur during ETL is that the data may be truncated if the data warehouse field is shorter than the data being loaded into the field. By comparing the field size in the source systems with that of the data warehouse, this problem can be reduced. Still more complications arise in instances where fields are concatenated together, parsed, or decoded to lists of values. This approach is used to ensure that the resulting data will fit into the table in the data warehouse. These actions do not always ensure that the data are not truncated during the ETL process.

Sampling the data for data verification purposes is an extraordinarily important approach, but its success depends on the same criteria as any sampling approach: the sample size used, and the degree that the sample used is representative of the whole. Clearly, the sample set must be a good representation of all the data in the system in order to be successful. Identifying data that are unique (and therefore must be included in the sample), however, can be a difficult and time-consuming task. Once testing is complete, the question of lost or mangled data in the data warehouse from data outside the sample still remains. This is an issue with the sampling approach that can only be addressed by increasing the sample size.

Standard Testing Approaches: The Reporting Tool

The primary goal of functional testing of reports created in the data warehouse reporting tool is to ensure that the reports correctly present the data in the warehouse. Testing that the reports display the desired information is an important step in the test process, but the amount of data that is able to be tested with this approach is often significantly limited even with automated methods, because automated run execution times per datum verified are typically long compared with the time required to verify each datum run with a back-end approach. While data verification at the reporting interface is an important complement to data integrity testing, for practical reasons it usually cannot supply a complete solution.

Once several testing methods have been applied, the quality of the testing effort must be evaluated. If all the records being extracted into the data warehouse are very similar, this can be a simple task that readily builds confidence in the data.

However, if the data warehouse is transforming unclean data that is inconsistent in structure (e.g. a particular text field in the database is used for one purpose for several releases, and then is used for a different purpose in succeeding releases), the end users and upper management may need a higher level of confidence in the accuracy of the data warehouse. Attaining that higher confidence level can be a significant challenge.

Automating Data Integrity Testing

The solution to ensuring data integrity is simple in concept - compare all of the data in the source system with all of the data loaded into the data warehouse. This solution, however, is often impossible to achieve due to the sheer volume of data. As with any other type of testing, use of an automation tool can increase the scope of the testing to approach this level.

Any of the industry-leading automation tools can be used to implement this type of approach, however, none of the currently-available tools are optimized for this type of project (and some of them have severe shortcomings in SQL and database handling).

To address this problem, RTTS built QuerySurge™ for a client with high data-quality standards. The tool is built around a central repository that stores SQL queries, allowing for testing of thousands, even millions of rows of data during a run. Once the repository is populated with SQL queries that are customized for each source system and each business rule (no small task), volume data testing can begin. The tool framework's job is fairly simple – it pulls queries from the repository and executes them against the source systems and the data warehouse.

(Refer to Figure 1). The queries are written to return the data in a common format, regardless of differences in database structure. The tool then compares the results of these queries to find discrepancies in the data within the different systems.

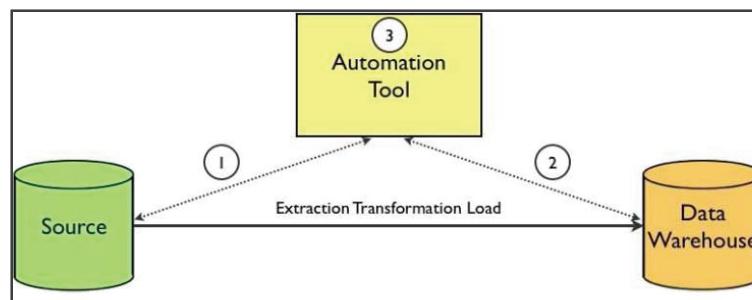


Figure 1. Tool Interactions

1. The tool queries the source database and captures results formatted to meet a business requirement.
2. Next, the tool queries the data warehouse and captures results for the same business requirement as in Step 1.
3. The tool compares the results and reports any discrepancies.

The SQL queries are written based on business requirements for the extraction and transformation of data from each of the source systems to the data warehouse. Each set of queries creates a test case for a particular business requirement. The test team can also use the tool to create skeleton queries, which may then be extended for specific business requirements. This allows for a lower maintenance cost, as a subset of the query library can be modified by slightly changing the fundamental skeleton query.

The tool utilizes a server that connects to agents on remote clients. The software quality engineer connects through a client to modify, execute and view results of the tests. This permits distribution of large query libraries over several clients, as tests can be run concurrently on several clients (24 x 7) in order to reduce the total required execution time (Figure 2). The decrease in testing cycle time, in turn, can reduce the time between builds in the development cycle, enabling more cycles and builds prior to release.

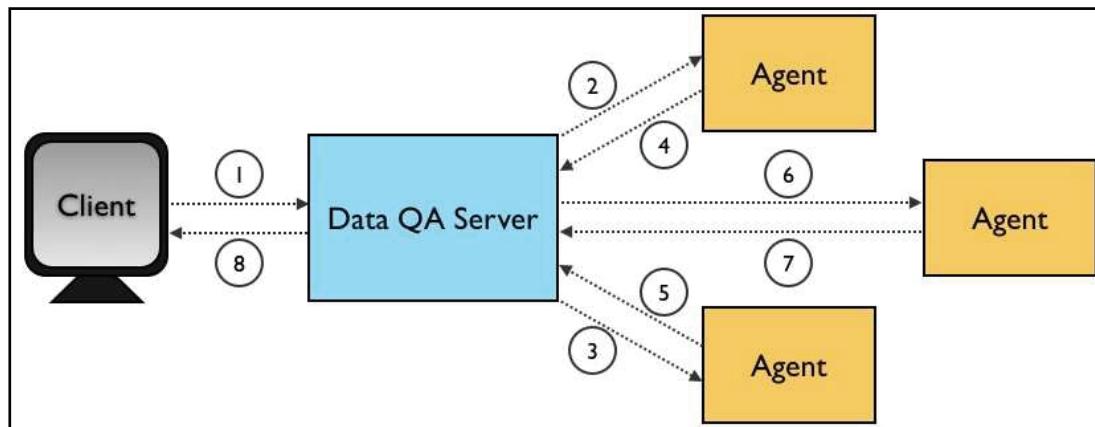


Figure 2. Functionality of an Automation Tool

1. Client sends request to server to execute test.
2. Server assigns an agent to retrieve results from the source system for the test
3. Server assigns another agent to retrieve the corresponding results from the target system.
4. The initial agent notifies server that the source information is retrieved.
5. The other agent notifies the server that the destination result is retrieved (may complete before #4).
6. Server assigns any available agent to compare the results and report any errors.
7. Agent returns to the server the outcome of the test case.
8. Server notifies all clients of the results of the test.

Benefits of Automating the Process

This level of testing can help ensure that data meets the business requirements specified, building confidence in the accuracy of the data contained in the data warehouse. The measured level of reliability provided by data integrity testing in the data warehouse underscores the value in the investment.

The customized automation tool described here to measure the quality of the warehouse offers the same benefits as most automation tools:

- » The ability to run complete regression tests
- » Increased test cycle speed
- » Improved accuracy in testing
- » Enhanced maintainability of the test suite for future releases of the data warehouse

While this approach has many advantages, its implementation does have an associated cost. The approach requires resources skilled in understanding the business requirements and in writing SQL to test each requirement. Because resource costs are not insignificant, a successful implementation requires an organization-level commitment to the software quality effort. Once that investment is made, the per-cycle costs for executing a data integrity measurement are relatively low. The return on the investment comes from the ability to measure and track data warehouse data integrity over time with relatively low per-cycle overhead. The organization knows the measured quality of its data at any time, and can proceed with confidence in its data warehousing operations based on the measured data quality.

Related Case Study

As a result of mergers and acquisitions, a Fortune 100 pharmaceutical firm needed to incorporate three legacy source systems into a data warehouse. In order to ensure the data being extracted from these systems was correctly processed, the testing approach described above was implemented. Executing this testing on these systems identified several hundred defects in the ETL process for all three sources. Most impressive was the ability of the tool not only to test volume data, but to identify within that volume a single record that was not loading properly into a field in the data warehouse, and to allow the development team to understand what issue led to the specific defect.

The quality of this testing effort was so apparent that the client expanded the project to include source systems already in production. The results of this additional work led to discovery of defects in just over 15% of the business requirements tested for each of the production source systems.

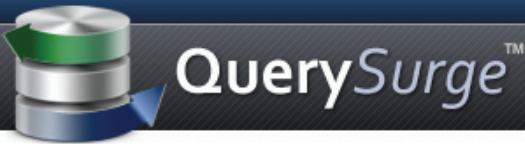
Please visit our website for a more detailed version of this case study:
http://www.rtsweb.com/research/studies/study_pharma-data-verification.jsp

Summary

By automating the process of data validation, the scope of the testing can increase dramatically. Once validation of a business requirement is complete the automation tool may be of further use in regression testing. This facilitates complete regression testing of data for every build released.

About the Author

Eric Ceres is a Senior Test Engineer at RTTS. He has worked with large-scale, automated quality assurance testing implementations for software development and pharmaceutical companies. Eric is a graduate of Rutgers University with a BS in Computer Science.



About RTTS

RTTS offers the most comprehensive suite of quality assurance services to help organizations drive positive results from their critical software projects. Headquartered in New York, NY, our expert team has worked closely with over 400 clients to improve their testing processes, tool knowledge, and application deployment outcomes. RTTS was founded in 1996, and has forged partnerships with the world's leading test tool vendors. Our satellite locations are in Philadelphia, Atlanta, and Phoenix, and many of our consulting and education services are offered through the cloud. No matter where you are, RTTS will ensure application functionality, performance, scalability, and security for your organization.



About QuerySurge™

RTTS' team of test experts developed QuerySurge™ to address the unique testing needs in the data warehousing space. It has been implemented on projects ranging from data warehousing and ETL processes to data migrations and database upgrades. QuerySurge™ can verify as much as 100% of all data from source systems, through the ETL process, to the target data warehouse. The tool has increased test coverage and reduced test cycle time for several organizations, helping them to mitigate risk and meet business requirements.



QuerySurge™ is offered exclusively by RTTS, as are any accompanying Data Warehouse Testing services. If you are interested in learning more, or to schedule a private demonstration, please contact us by e-mail here: info@rttsweb.com. If you would like to speak with our Sales team, please call (212) 240-9050.