Overview

The characteristics of Data Warehouse applications pose a challenge for any organization planning a Data Warehouse initiative:

How can we be sure our data is accurate and reliable when there are enormous amounts of it, and when it comes from multiple systems with different data structures?

Like any other software application involving different technologies (such as the mainframe, Java or Microsoft), testing is a very crucial aspect in the Software Development Lifecycle for Data Warehouse(DW) projects. Testing for DW carries unique challenges and requires specialized approaches. However, the testing function for this highly dynamic technology area is at a very nascent stage of maturity. This paper discusses the various aspects associated with testing for DW.

About DW

Why and how is testing for DW different from testing for other technologies? Part of the answer lies in definition of what constitutes DW

"A data warehouse is a subject oriented, integrated, time variant, non volatile collection of data in support of management's decision making process".

Data warehouse includes huge set of data, and complex aggregations logics. There are different pieces of a DW like reporting, ETL (extract, transform, load), data mining, data cleansing, staging servers, Cubes, ad hoc-reporting and all them require a different testing strategies. Data Quality and freshness are the two key factors for success of any data warehouse.
There are three types of data warehouses:

1. **Enterprise Data Warehouse** - An enterprise data warehouse provides a central database for decision support throughout the enterprise.

2. **ODS (Operational Data Store)** - This has a broad enterprise wide scope, but unlike the real enterprise data warehouse, data is refreshed in near real time and used for routine business activity.

3. **Data Mart** - Datamart is a subset of data warehouse and it supports a particular region, business unit or business function.

**Testing Activities with DW**

When building Data Warehouse, organizations employ either a top-down or bottom-up development approach. In the top-down approach, an enterprise data warehouse (EDW) is built and information processes are created each business area wise with underlying dependant data marts pulled out from the EDW contents. In the bottom-up approach, independent data marts with information processes are created with the view to integrating them into an EDW in the future. Either way, a major blind spot in the data warehousing project delivery quality is testing. Data warehouse testing may have the same principles/fundamentals of a general testing project, but testing DW & BI projects involves significant programming work as there are limited front end screens but mostly backend processes that work on the data sets.

Data warehouse testing involves validation of following areas:

- **Data completeness.** Ensures that all expected data is loaded.

- **Data transformation.** Ensures that all data is transformed correctly according to business rules and/or design specifications.

- **Data quality.** Ensures that the ETL application correctly rejects, substitutes default values, corrects or ignores and reports invalid data.
• **Performance and scalability.** Ensures that data loads and queries perform within expected time frames and that the technical architecture is scalable.

• **Integration testing.** Ensures that the ETL process functions well with other upstream and downstream processes.

• **User-acceptance testing.** Ensures the solution meets users' current expectations and anticipates their future expectations.

• **Regression testing.** Ensures existing functionality remains intact each time a new release of code is completed.

**Testing strategy**
Performance Testing of DW

With increasing volume of data, stability and scalability become critical test parameters. Under stress from large transactional data volumes, data warehouses will typically not scale, and eventually fail, unless they are tested and issues are fixed. To avoid such problems, it is essential that the test team design and execute series of tests that validate the performance and scalability of the system under different loads. As part of this activity, the following tests can be executed:

- Shutdown the server during batch process and validate the result
- Perform ETL with load that is twice or thrice the maximum possible imagined data (for which the capacity is planned)
- Run huge volumes of ad-hoc queries mimicked from multiple users simultaneously
- Run large number of scheduled reports
- Monitor the timing of the reject processes and check system behavior when handling large volumes of rejected data

Tester Skill set

Test resources are a debatable topic which often Project managers and test managers end up in meetings. Data warehouse projects require more test resources than usual web applications or traditional projects.

Example: - There is a small change in a table, so for development its change at one place, but from testing perspective, that table is used in 10-15 different calculations differently, and 10 reports, and testers have to validate each and every place.
Organisation users do not give enough time to do a complete testing job as their operations job take precedence over testing work. They also lack skills and perspective to carry out testing, more so in Data warehouse testing. The attitude and methodology required for DW testing are not same as those required for normal testing.

The skill sets required include resources with a strong aptitude and with technology specialization and programming knowledge.

**Conclusion**

Data warehouse solutions are becoming almost ubiquitous as a supporting technology for the operational and strategic functions at most companies. Data warehouses play an integral role in business functions as diverse as enterprise process management and monitoring, and production of financial statements. The approach described above combines an understanding of the business rules applied to the data with the ability to develop and use testing procedures that check the accuracy of entire data sets. This level of testing rigor requires additional effort and more skilled resources. By employing this methodology, the team can be more confident, from day one of the implementation of the DW, in the quality of the data. This will build the confidence of the end-user community, and it will ultimately lead to a more effective implementation.