Chapter 5
Data Warehouse Testing

Matteo Golfarelli  
University of Bologna, Italy

Stefano Rizzi  
University of Bologna, Italy

ABSTRACT
Testing is an essential part of the design life-cycle of a software product. Although most phases of data warehouse design have received considerable attention in the literature, not much research has been conducted concerning data warehouse testing. In this paper, the authors introduce a number of data mart-specific testing activities, classify them in terms of what is tested and how it is tested, and show how they can be framed within a reference design method to devise a comprehensive and scalable approach. Finally, the authors discuss some practical evidences emerging from a real case study.

INTRODUCTION
Testing is an essential part of the design life-cycle of any software product. Needless to say, testing is especially critical to success in data warehousing projects because users need to trust the quality of the information they access. Nevertheless, while most phases of data warehouse design have received considerable attention in the literature, not much has been written about data warehouse testing.

As agreed by most authors, the difference between testing data warehouse systems and generic software systems or even transactional systems depends on several aspects (BiPM, 2009; Mookerjea & Malisetty, 2008):

• Software testing is predominantly focused on program code, while data warehouse testing is directed at data and information. The key to data warehouse testing is to know the data and what the answers to user queries are supposed to be.
• Differently from generic software systems, data warehouse testing involves a huge data volume, which significantly impacts performance and productivity.

• Data warehouse testing has a broader scope than software testing because it focuses on the correctness and usefulness of the information delivered to users. In fact, data validation is one of the main goals of data warehouse testing.

• Though a generic software system may have a large number of different use scenarios, the valid combinations of those scenarios are generally limited. Data warehouse systems are aimed at supporting any views of data, so the possible combinations are virtually unlimited and cannot be fully tested.

• While most testing activities are carried out before deployment in generic software systems, data warehouse testing activities still go on after system release.

• Typical software development projects are self-contained. Data warehousing projects never really come to an end; it is very difficult to anticipate future requirements for the decision-making process, so only a few requirements can be stated from the beginning. Besides, it is almost impossible to predict all the possible types of errors that will be encountered in real operational data. For this reason, regression testing is inherently involved.

Like for most generic software systems, different types of tests can be devised for data warehouse systems. For instance, it is very useful to distinguish between unit test, a white-box test performed on each individual component considered in isolation from the others, and integration test, a black-box test where the system is tested in its entirety. Also regression test, that checks that the system still functions correctly after a change has occurred, is considered to be very important for data warehouse systems because of their ever-evolving nature. However, the peculiar characteristics of data warehouse testing and the complexity of data warehouse projects ask for a deep revision and contextualization of these test types, aimed in particular at emphasizing the relationships between testing activities on the one side, design phases and project documentation on the other.

From the methodological point of view we mention that, while testing issues are often considered only during the very last phases of data warehouse projects, all authors agree that advancing an accurate test planning to the early projects phases is one of the keys to success. The main reason for this is that, as software engineers know very well, the earlier an error is detected in the software design cycle, the cheapest correcting that error is. Besides, planning early testing activities to be carried out during design and before implementation gives project managers an effective way to regularly measure and document the project progress state.

Since the correctness of a system can only be measured with reference to a set of requirements, a successful testing begins with the gathering and documentation of end-user requirements (Haertzen, 2009). Since most end-users requirements are about data analysis and data quality, it is inevitable that data warehouse testing primarily focuses on the ETL process on the one hand (this is sometimes called back-end testing) (BiPM, 2009), on reporting and OLAP on the other (front-end testing) (BiPM, 2009). While back-end testing aims at ensuring that data loaded into the data warehouse are consistent with the source data, front-end testing aims at verifying that data are correctly navigated and aggregated in the available reports.

From the organizational point of view, several roles are involved with testing (Haertzen, 2009). Designers draw conceptual schemata that represent the users’ requirements to be used as a reference for testing. Designers are also responsible
Related Content

A Clustering Rule Based Approach for Classification Problems
Philicity K. Williams, Caio V. Soares and Juan E. Gilbert (2012). *International Journal of Data Warehousing and Mining* (pp. 1-23).
[www.igi-global.com/article/clustering-rule-based-approach-classification/61422?camid=4v1a](www.igi-global.com/article/clustering-rule-based-approach-classification/61422?camid=4v1a)

On the Mining of Cointegrated Econometric Models
[www.igi-global.com/chapter/mining-cointegrated-econometric-models/39641?camid=4v1a](www.igi-global.com/chapter/mining-cointegrated-econometric-models/39641?camid=4v1a)

A Survey on Database Performance in Virtualized Cloud Environments
[www.igi-global.com/article/survey-database-performance-virtualized-cloud/67571?camid=4v1a](www.igi-global.com/article/survey-database-performance-virtualized-cloud/67571?camid=4v1a)

Vertical Fragmentation in Databases Using Data-Mining Technique
[www.igi-global.com/article/vertical-fragmentation-databases-using-data/1812?camid=4v1a](www.igi-global.com/article/vertical-fragmentation-databases-using-data/1812?camid=4v1a)