Data Warehouse / MIS Testing: Corporate Information Factory

Introduction

Data warehouse commonly known as "**DWH**" is a central repository of data that is created from several diverse sources. Businesses need the data to review their historical trends and monitor operational records for its decision support systems. This system aids them in carrying out informed decision that decides its competitive edge in modern business environment.

MIS and Data Warehouse testing is a special type of software testing where testing principles are applied to MIS Applications /Data warehouse components under test.

Problem Statement

A leading UK Mobile service provider wanted to move from existing "one-stop-team" model to more sophisticated team structures to transfer the ownership of Operational Support, Validation, Release, and development to the respective teams.

The project consists of variety of billing, operational support, networking, inventory, trouble ticketing, insurance, Interactive VOICE Response (IVR), marketing and campaign management, Reporting, Telesales and contracts management systems which were developed in variety of technical platforms.

Solution

TestQ has facilitated this change from testing point of view by aligning MIS/Data warehouse testing process with existing validation and then project delivery process.

How?

Typically, data is extracted from diverse sources of operational data such as different databases, feeds, application files, or flat files. This derived data is transformed to a common format and loaded into the data warehouse. Once data warehouse is populated, front end applications need to be built in to facilitate querying, analysis and reporting.

The former part of **E**xtracting, **T**ransforming, and **L**oading data is typically known as data warehouse ETL process whereas the later part which deals with front end applications is called as MIS applications. The below given figure illustrates the process of ETL:

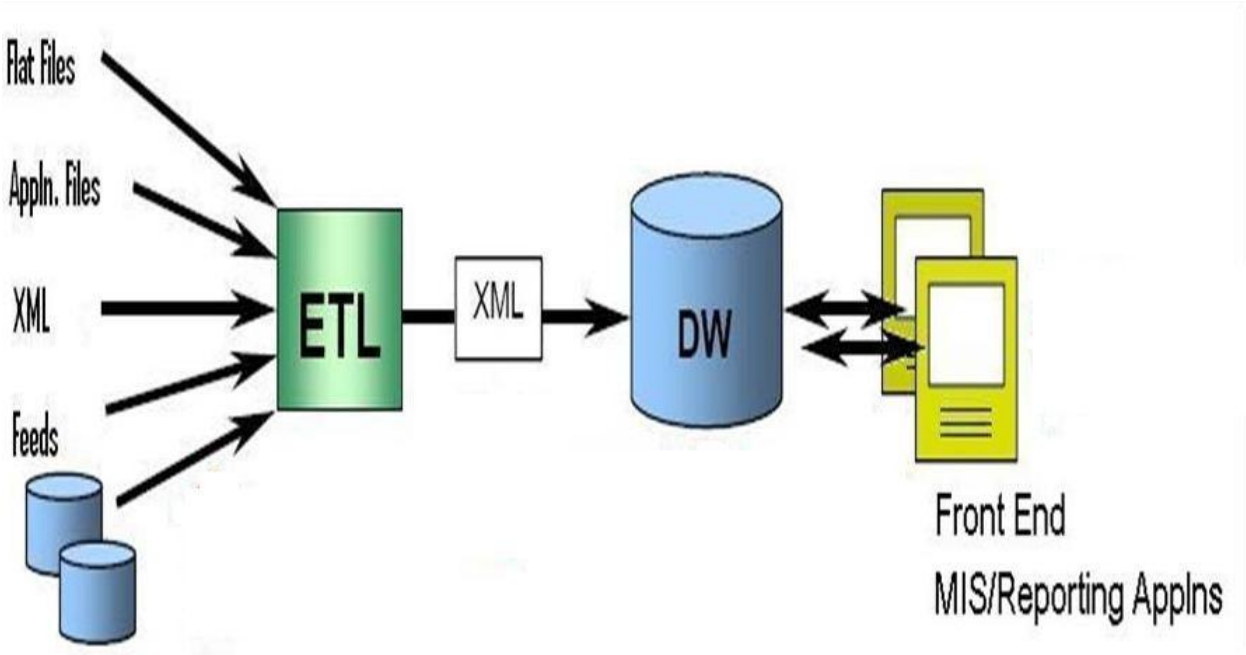

Figure 1: ETL and Data warehouse process

While Extracting and Loading data from different sources, there are quite a few chances to loss data. Some of the other major challenges faced in ETL process are:

1. Loss of data during ETL process
2. Inaccurate and incomplete source data
3. Duplicate source data
4. Data Quality issues

Strategies for Testing MIS / Data warehouse Applications:

An effective MIS/Data Warehouse testing strategy focuses on the main structures within the data warehouse architecture:

1. The ETL layer
2. The data warehouse itself
3. The front-end data warehouse applications

In order to overcome the above given challenges, these structures must be tested individually and their various combinations with respect to:

1. **Data completeness:** Ensures that all expected data is loaded. e.g. Record counts between source and loaded data
2. **Data transformation:** Ensures that all data is transformed correctly as per business rules and/or design specifications.
3. **Data quality:** Ensures that the ETL application correctly rejects, substitutes default values, corrects or ignores and reports invalid data.
4. **Performance and scalability:** Ensures that data loads and queries perform within expected time frames and that the technical architecture is scalable.
5. **Integration testing:** Ensures that the ETL process functions well with other upstream and downstream processes.

6. **UAT:** Ensures the solution meets users' current expectations and anticipates their future expectations.
7. **Regression testing:** Ensures existing functionality remains intact each time a new release of code is completed.
8. **Front End testing:** Validate user queries based on a data warehouse of known data composition

Figure 2 shows the different test entry points in the architecture

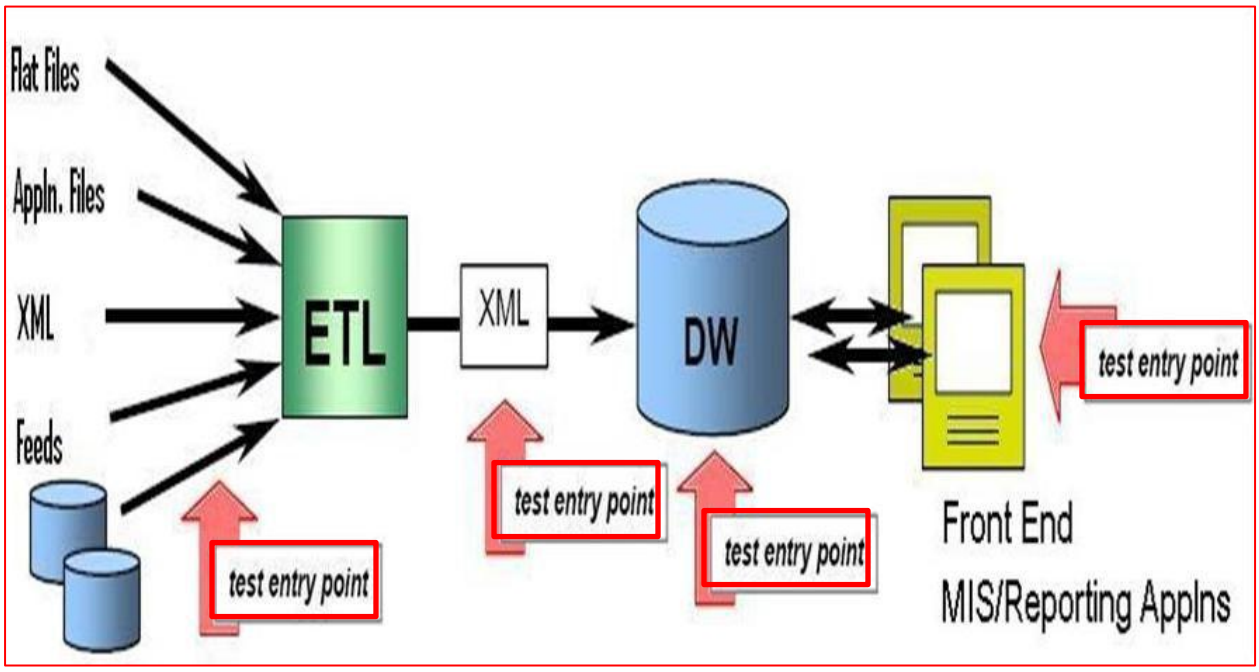

Figure 2 shows the different test entry points in the architecture

Tools required undertaking Data warehouse / MIS testing

1. Tool to schedule and run ETL batches
2. Database Access tools to manipulate the operational databases and data marts
3. A secure client used to access various environments using UNIX
4. Automated / Manual test scenarios documents/spreadsheets to populate and verify the intermediate/final values
5. Test Management tools to execute the scenarios and log issue