Data Warehouse Testing 101

As organizations develop, migrate, or consolidate data warehouses, they must employ best practices for data warehouse testing. The success of any on-premise or cloud data warehouse solution depends on the execution of valid test cases that identify issues related to data quality. Extract, Transform, and Load (ETL) is the common process used to load data from source systems to the data warehouse. Data is extracted from the source, transformed to match the target schema, and loaded into the data warehouse.

ETL testing ensures that the transformation of data from source to warehouse is accurate. It also involves verifying data at each point between the source and destination. This article will focus on the traditional ETL testing process.

The Importance of Data Warehouse Testing

With data driving critical business decisions, testing the data warehouse data integration process is essential. Data comes from numerous sources. The data source affects data quality, so data profiling and data cleaning must be ongoing. Source data history, business rules, or audit information may no longer be available.

Additionally, in the ETL process, data flows through a pipeline before reaching the data warehouse. You must test the entire ETL pipeline to ensure each type of data is transformed or copied as expected. Most importantly, the data warehouse is a strategic enterprise resource. Testing is required.

Understanding the ETL Testing Process

A solid understanding of data modeling provides testing teams with information to develop the right testing strategy. During the analysis phase, the testing team must learn and understand the different stages of the data warehouse implementation including but not limited to:

- Source data profiling
- Data warehouse design
- ETL development
- Data loading and transformations

ETL testing includes multiple phases, and testing should be executed throughout the lifecycle of the data warehouse implementation, not just at the end.

Preparing for ETL Testing

A data warehouse implementation must include end-to-end testing. The QA team must test initial and incremental loads for the entire ETL process beginning with identifying source data to report and portal functions. They must also test each point between extract and load, including data load from the source extract to staging, staging to transformation and once the data reaches the data warehouse, test data extraction for display and reporting.

With traditional ETL test planning, there are six phases:

1. Understanding business requirements/analysis
2. Creating test plans and estimating time to completion
3. Designing test cases and selecting test data
4. Executing tests with bug reporting and closure

5. Report summary and analysis

6. Test completion

ETL testing is performed in five stages:

1. Identifying data sources and requirements

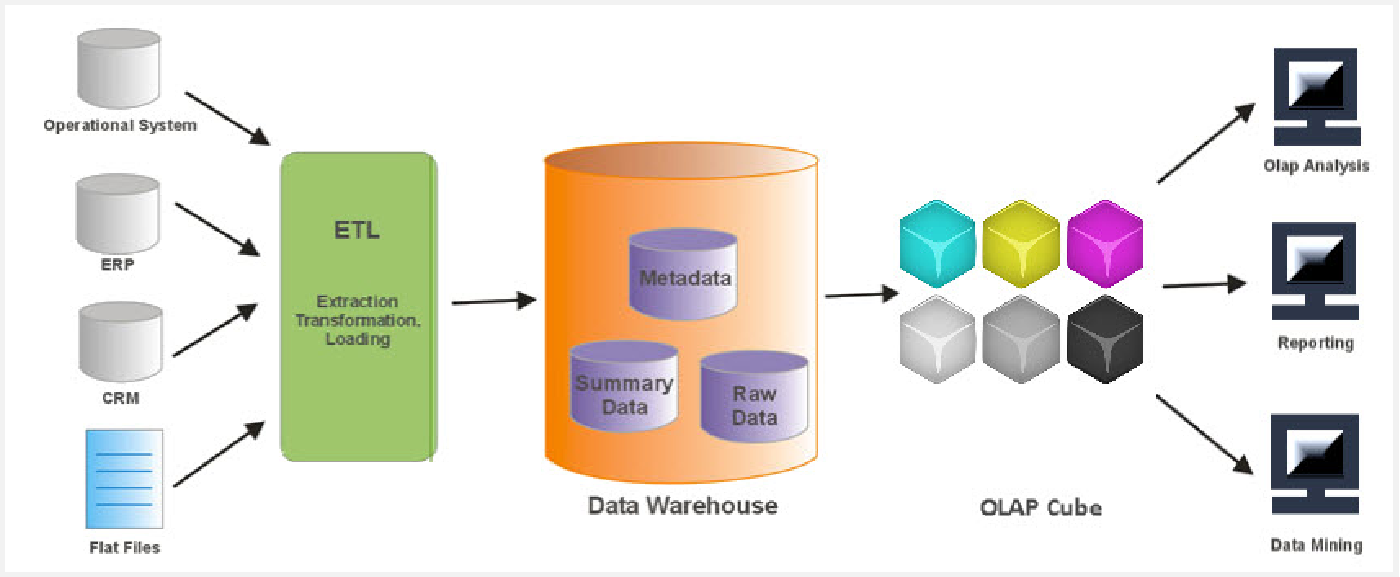
2. Data acquisition

3. Implement business logic and Dimensional Modeling

4. Build and populate data

5. Build Reports

Test Planning Process



Identify Data Sources	Data Acquisition	Implement business logic and Dimensional	Build and populate	Build reports

		Modeling	analytical cube	
Requirement/Analysis	Design and Coding	Design and Coding	Design and Coding	QA and Deployment

Test Execution Process

Validating required data /data sources:	Review metadata/data dictionary	Review archival/purge strategy	Test data prep
Data profiling:	Validate source to target mapping	Error logging, exception handling, recoverability	ETL testing (end to end)
Data quality and performance acceptance criteria:	Validate ETL/Data Warehouse architecture	Parallel Execution and Precedence	OLAP and cube testing
Data transformation rules:	Validate data model – Dimensional modeling and normalized approach	ETL pull logic – full/incremental	Report testing – drill down/drill through

Common ETL Testing Types

Organizations use data warehouse testing to ensure that required business functions are implemented correctly. This phase includes data verification, which tests the quality of data populated into target tables. The table below describes the most common testing types.

Test	Description
Metadata Testing	Metadata testing confirms that the table definitions conform to the data model and application design specifications. This test should include data type check, data length check, and index/constraint check.
Data Completeness Testing	Data Completeness testing validates that all the expected source data has been successfully loaded to the target. Tests include: Compare and Validate counts, aggregates (min, max, sum, avg), and actual data between the source and destination.
Data Quality Testing	Data Quality tests validate the accuracy of the data. Data profiling is used to identify data quality issues, and the ETL is designed to fix or handle these issue. Automating the data quality checks between the source and target system can help to mitigate problems post-implementation.
Data Transformation Testing	Data Transformation comes in two flavors: white box testing and black box testing. White box data transformation testing examines the program structure and develops test data from the program logic/code. Testers review the transformation logic from the mapping design document and the ETL code to create test cases. Black-box testing examines the functionality of an application without looking at internal structures for transformation testing; this involves reviewing the transformation logic from the mapping design document creating the appropriate test data.
ETL Regression Testing	ETL Regression testing validates that the ETL produces the same output for a specific input before and after the change.
Incremental ETL Testing	Incremental ETL testing verifies that updates on the sources are getting loaded into the target system correctly.
ETL Integration Testing	ETL integration testing is end-to-end testing of the data in the ETL process and the target application.
ETL Performance Testing	ETL performance testing is end-to-end testing to ensure that the all steps in the ETL process are working with expected data volumes. One pitfall of this testing method is the lack of actual data to emulate appropriate volumes.

Testing mission-critical data warehouse infrastructure is required. Testing is an essential part of building a new data warehouse (or consolidating several), and it must be part of the development pipeline when the ETL process is modified or extended.

Testing data and systems systematically for inconsistencies before moving into production is necessary if the data warehouse is to be the central source of business information. Whether it is a newly built data warehouse or the consolidation of several, you must develop a thorough data warehouse testing process to help you test for, resolve, and prevent unnecessary exposure.

Learn More About Data Warehouses

- The Difference Between a Database and a Data Warehouse
- Data Warehouse Architecture: Traditional vs. Cloud
- Data Warehouse Concepts: Traditional vs.Cloud

Built by Panoply