

MASTER DATA MANAGEMENT TEST ENABLER

Sagar Porov¹, Arupratan Santra², Sundaresvaran J³

Infosys, (India)

ABSTRACT

All Organization needs to handle important data (customer, employee, product, stores, financial data and etc.) and its correctness. Most of the organizations start adopting Master Data Management solution to keep the data as source of truth for rest of the downstream applications. Any enterprise applications coupled with master data management (MDM) tools require a system for cleansing, standardizing, de-duplicating and enriching data. Presently testing of these critical processes is happening by traditional approach to remove redundancy, incorrect data, format, etc. This article is about accelerating Master Data Management testing with automated solution. It's not specific to any MDM tool, here manual checks of these implementation process are replaced with automated process.

Keyword: MDM, Schema, Query builder, Gold copy

I. INTRODUCTION

1.1 What is Master Data Management

Master Data Management is the technology to create and maintain sanitized huge volume of specific entity data of any organization [1]. MDM tools generates a golden record for the entity (Master Data) from the information gathered from different source systems or different organization databases of various regions (providers). MDM also shares golden record information with downstream systems (consumer) for consumption. Master Data may include data about customers, employees, inventory, suppliers, territory and etc. [2].

1.2 Why Master Data Management

All organizations would need to master the entity information (it could be customer, employee, product, etc.) to derive valuable information for business reasons.

One of the reasons for MDM is, information about the entity might come from more than one sources in different formats and any of that could be the latest and correct. We need MDM system to process the incoming data based on its correctness, uniqueness, trust worthy score and integrity.

Another reason for MDM is, frequent changes in data and data flow [3]. Suppose an employee of a company moved from address A to address B. After moving the customer applied to change his address and got confirmation. After few days employee resigned and joined another company. After few years same employee wants to withdraw PF amount but there is no information as it is more than 5 years. Employee lost his PF amount and company lost his credentials. MDM will help to both employee and organizations to connect current and old employee without duplicating the employee entity record.

1.3 Traditional Testing Approach

In MDM testing[1,2,3] generally, we test the backend data. While testing a MDM implementation we follow below traditional testing approaches:

- Data migration testing
- Data cleansing testing&standardization testing
- Data consolidation testing

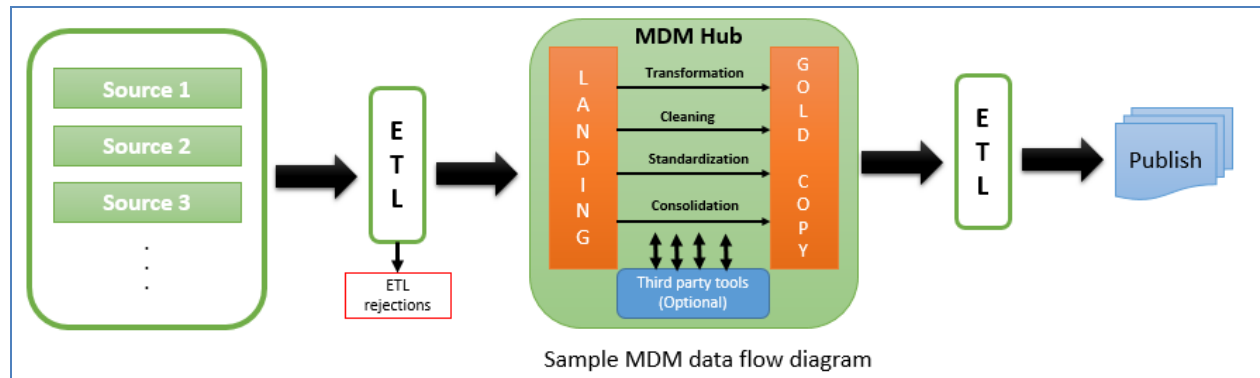


Figure: 1

When data comes from source application/database to MDM hub (Fig. 1) then it may come via landing table with the help of ETL process. Testing team needs to check whether data has been loaded properly and completely [1,3] by running SQLs queries to compare the row count and sampling data validation.

If the migrated source data in MDM have some special character[4] formatting conditions or spelling mistake. MDM tool needs remove these special characters and loads the fresh and clean data with corrected format and spelling [5]. This process is called data cleansing and standardization. To perform standardization and cleansing, we prefer SQL queries than third party tool.

MDM tool searches common records and merge them. Data match and merging process is complex and varies with organization. Types of match process: Exact match and Fuzzy match. Testing team prefers SQL queries to test this merging process [6]

II. RESEARCH BACKGROUND

Above traditional testing approach are human dependent and time consuming. Every process has to be designed properly with respect to SQL query and test scenario. If there is any miss of test assumption then SQL query will be wrong and final test will fail. Any MDM system will have millions of record so probability of failing is very high and as well to make successful testing, we need always experienced MDM skilled resources. To overcome this test complexity and human dependency, a MDM test enabler has been designed to automate various steps. Detailed descriptions of research outcome are explained below.

III. PROPOSED AUTOMATED TESTING APPROACH:

MDM Test Enabler (MTE) will connect to any database e.g. Oracle, SQL Server etc. through connection wizard as shown below and fetch all metadata from database which can be used further for data validation and verification. This tool will have various features as below and can be tested automatically to check various

features. Immediately this tool cannot replace SQL query completely but our future target is to minimize manual intervention as much as possible.

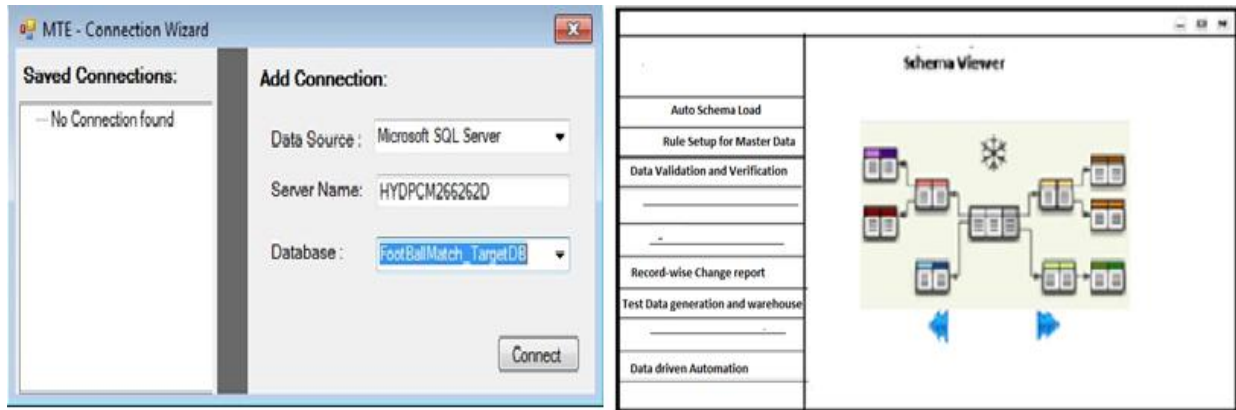


Figure: 2

MTE (Fig.2) will have automatic Schema load feature, Query builder, Report generation, Data validation features and etc. Further MTE can be enhanced to larger extend, by which complete MDM traditional testing approach will be fully automated.

3.1 Auto Schema Load: The inbuilt add-ins for connecting to any kind of database will be provided to perform this activity. All tables and schema available in the MDM data base will be loaded automatically. User can view all the tables and schema once they are loaded into the tool using Schema Viewer as below. User will be able to browse through landing, staging and base objects table schemas using the arrows available on the schema viewer. This schema metadata will be used for query building, data validations and other checks. Schema is displayed using entity relationship diagram as below (Fig.3):

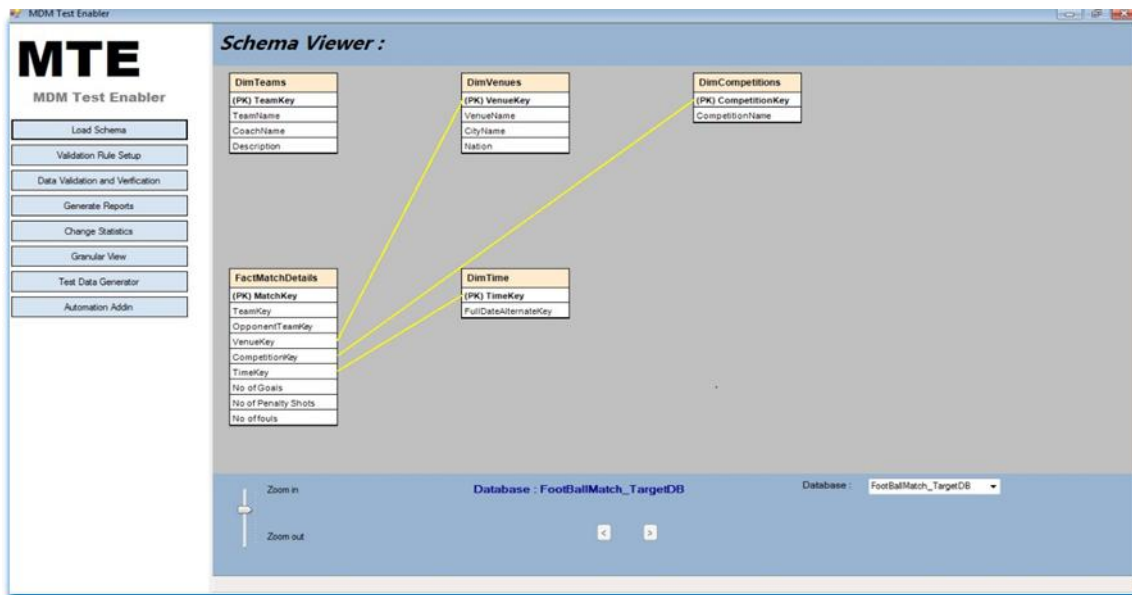


Figure: 3

This tool will show all referential integrity between tables at each stage (landing, Stage and Base Objects) by reading all Primary Key (PK) and Foreign Key (FK) information available in the database. This wizard or tool will also show the missing link between the tables if there is no PK or FK relationship as above. All this schema information will be used for validation during data migration testing. This process will remove the manual

intervention of traditional approach for data migration testing. Tableschema/structure validation can be also be done using schema viewer functionality.

3.2 Rule Setup for Master Data: Tool will provide configurable interface(Rule Setup Wizard) to build different sets of master data rules or validation checks on tables like mandatory check, duplicate check, match records check, and spell check, special character check ,client specific master data rules etc. These rules will be used for determining data health and generating health check report in MDM system. Data validation and verification tab available in the tool will use the rules and checks defined for a table during data validation.

3.2.1 Alert Mechanism: Tool will generate alerts for any Master data rule violation or data issue e.g. duplicate records, mandatory attribute missing, potential match records found etc. Below is the high-level model for alert notifications (Fig.4):

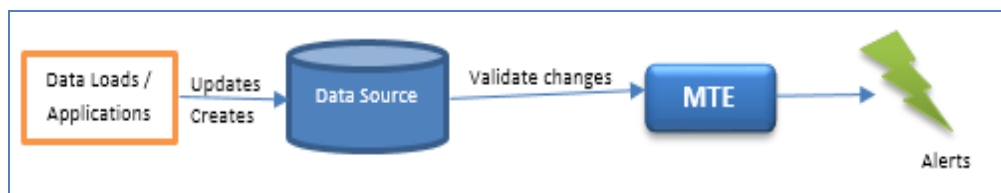


Figure: 4

MTE will monitor every change in the database and validate table data against rules defined for it and in case of violation will trigger alerts.

3.2.2 Reporting Functionality: Tool will provide the customized reports for Master data health and Data change statistics report.

3.3 Data Validation and Verification: Schema loader will display the exiting table details and its dependencies. Tool will provide interface to build queries for data mapping between two schemas within the database.

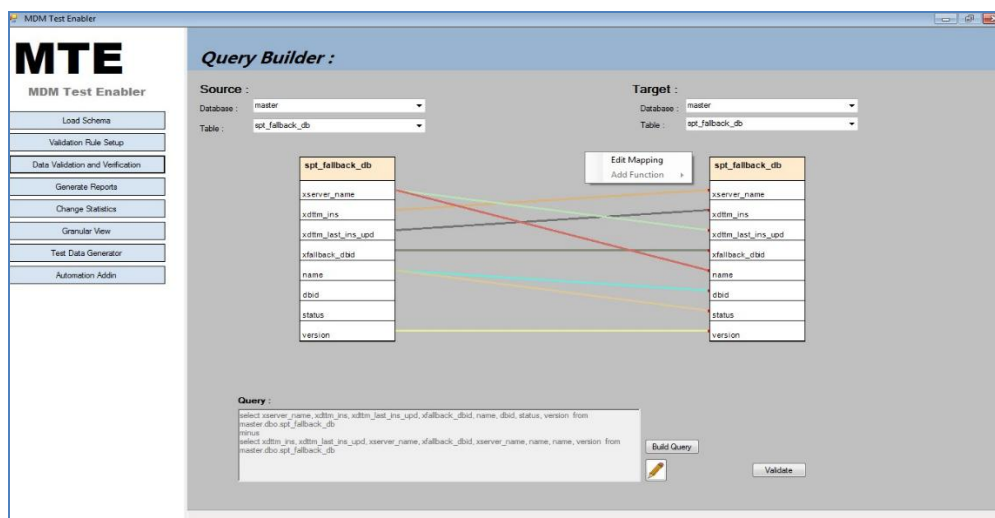


Figure: 5

With the help of query builder or mapping creator, manual SQL query creation effort has been minimized by drag and drop. Different sets of queries like count verification, data verification, duplicate check, mandatory check etc. will be created by the tools based on user mapping and validation rules setup for the target table and

same will be executed in the database by the tool to ensure data correctness (Fig.5). It will also generate detailed test report. It also allows user to check all potential matches based on match rule in Rule Setup functionality and verify that all matched records are merged or not.

3.4 Test Data Generation and Warehouse

Production data is considered to represent the ‘truth’ for testing. Migrating full copies of production databases for testing is expensive. The most common solution is to extract subsets of data to satisfy the tests. However, production data is sanitized, and it covers 10-30% of the test scenarios.

Test Data Generation is a complex problem in MDM implementation. It is harder to anticipate the program flow which makes nearly impossible for the Test Data Generators to generate exhaustive Test Data. MTE uses Intelligent Test Data Generators, which do sophisticated analysis of the master data rules and generate the test data accordingly. This approach generates the test data quicker and ensure 90% test coverage. Below are few of the benefits of MTE test generators (Fig.6):

- Synthetic test data creation: Data which contains all of the characteristics of production without sensitive content.
- Quickly create test data that is referentially intact and consistent
- Assemble more complex test scenarios from initial subsets within the Test Data Warehouse
- Rapidly create large volumes of data for performance testing
- Clone interesting defect scenarios and rare data attributes to ensure they are always available



Figure: 6

3.4.1 Test Data Export Functionality: MTE provides flexibility of exporting data in any format e.g. Comma separated csv, a delimited text file or an excel sheet. MTE has its own test data warehouse. Complete snapshot or a subset can be pulled into user specified format using the export functionality.

3.4.2 Data driven Automation: In MDM projects, a single test script needs to be checked against number of inputs and it may behave differently for each input parameter. Manual searching of multiple complex back-end systems or manual data creation is time-consuming and prone to error. This will affect the stability and effectiveness of test automation.

Using the concept of Test data matching, MTE is able to identify, mine and link data to automated test cases, from multiple sources. This reduce the time required for data creation or searching by 90%. It also eliminates usage of “out of date” or “bad” data.

IV. CHALLENGES AND BENEFITS

Challenges in MDM testing has wide variety space. Most challenges are prompting across MDM hub and its operations. Major challenges are:

- Preparation of test data
 - Access to relevant database objects and its availability
 - When more than one source systems, different data ownership for few attributes
 - Handling the error, rejections and how to correct these rejections
 - Understanding of cleansing and standardization as per organization's requirements
 - Concurrent testing for various process and frequent requirement changes
- There are various benefits after implementation of MDM test enabler.
- Implement automation solution for validating the Notification/alert Services to achieve better scope coverage and effort savings and increased accuracy.
 - Ensure better test coverage during load process when lakhs of notification XML's are generated.

REFERENCES

- [1] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi. Designing data marts for data warehouses. *ACM Transactions on Software Engineering Methodologies*, 10(4):452-483, 2001.
- [2] R. Bruckner, B. List, and J. Schiefer. Developing requirements for data warehouse systems with use cases. In *Proc. Americas Conf. on Information Systems*, pages 329-335, 2001.
- [3] R. Cooper and S. Arbuckle. How to thoroughly test a DWH. In *Proc. STAREAST*, Orlando, 2002.
- [4] P. Giorgini, S. Rizzi, and M. Garzetti. GRANd: A goal-oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, 5(1):4-21, 2008.
- [5] M. Golfarelli, D. Maio, and S. Rizzi. The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7(2-3):215-247, 1998.
- [6] M. Golfarelli and S. Rizzi. *Data warehouse design: Modern principles and methodologies*. McGraw-Hill, 2009.