TESTING THE VALIDATION OF DATA FOR QUALITY ASSURANCE USING AUTOMATION FOR DATA MIGRATION

N.Sudheer1*, S.Hrushi Kesava Raju2, Nesar Ahmad3

1Department of Computer Science and Engineering, Siddharth Institute of Engineering &Technology, Puttur, Andhra Pradesh, India.
2Department of Computer Science and Engineering, Siddharth Institute of Engineering &Technology, Puttur, Andhra Pradesh, India.
3Department of Computer Engineering, Research Committee, Aligarh Muslim University, Aligarh, Uttar Pradesh, India.

Email: nidamanuri.sudheer@gmail.com

Received on: 18.10.2016 Accepted on: 11.11.2016

Abstract

Data Migration has become one of the most demanding proposals for IT company managers. Even though these projects earn high business benefits, such as reduced costs, improved productivity, and data manageability, they likely to involve a high level of risk due to the huge volume and criticalness of moved data. In order to reduce risk and guarantee that the data has been migrated and transformed successfully, it is essential to employ a thorough Quality Assurance (QA) strategy in migration projects. During the process of the Data Migration (DM), too much data is extracted, transformed, structured, and loaded (ETL) from legacy/source database into a newer structure that is the target database. This process leads to various types of changes in data, known as data corruption. Therefore, it is essential to perform data validation testing after data migration process over. The validation of transformed data for QA process is substantially more demanding to accomplish.

Not just there is a high risk that the structure has been modified, furthermore it might happened that the data itself has been liable to considerable updates. A number of attempts were made to validate data manually or automating the process; but there are limitations in measuring dimensions like time, cost and resource utilized. Here, authors emphasis on automation of data validation approach for quality assurance (QA) in the project of data migration. The paper describes existing data validation methods with their limitation. Afterwards, it elaborates in detailed architecture and process flow of implemented automated tool.

Keywords: Automation, Data Validation, ETL, Migration, QA, Testing.
1. Introduction

Many established businesses have legacy databases that are costly to maintain, risky to modify; affecting on business's competitiveness, reputation, and outcomes. Solution to tackle with problem is 'Data Migration'. Data migration is the process of moving a data from existing/legacy source database to targeted newer database. Though data migration term sounds simple and easy to understand; it is the most complicated, risky, time and budget overflowing but unavoidable process. Data Migration Projects Are Risky: 84% of data migration projects fail to meet expectations, 37% experience budget overruns, and 67% are not delivered on time [7]. Data validation is the key operation need to be perform after migration process over, to ensure that data migration is properly processed and that it will not corrupt the target system or loose important information. The goal is to produce a target system's applications free of errors and inconsistencies. There are various solutions to validate data. Sampling is most commonly and cheap solution to validate data, where a random records are selected from target dataset and matched with source dataset [5]. Sampling works on principle that errors are distributed in dataset uniformly. Another solution is writing MINUS queries, where first SELECT queries are executed on source and target dataset separately and then MINUS operation is performed to find the differences between them [4]. Some migration tools do provide data validation as an integrated phase of migration process.

Validating data indicates the comparison of original data set with the resulting target data set by applying a manual or automated comparison, or combination of both. Sampling is the easier way of validating data, but it covers far less than 10% of data under test. Thus, remaining 90% of data is untested which is a big data volume to find defects. MINUS operation on data just show mismatch results, but fail to detect other validation defects like data type mismatch, null values, data corruption, duplicate values etc. Even though some tools do perform data validation as an integrated phase of migration process, still fails to detect some special types of defects. Thus business decisions taken based on migrated data may fail due to incorrect, corrupted data, resulting in big loss for the firm [8]. Goal of validation phase in migration process is to get the exactly same copy of source data and must retain same functional correspondence with the source one. A single migration process may involve thousands of records, making it suspicious for defects. This paper proposed an automated data validation approach. The proposed model works at record level, where record to record check is perform on source-and-target datasets, testing data thoroughly to guaranty its efficiency and to prove that data quality is preserved or even improved by removing data inconsistencies.
Background

Data migration is mostly a part of bigger projects like CRM, ERP, BI, and many more. Data migration means simply transferring data from an application to another application. Process may involve merging of two or more databases, normalizing databases or designing new database from scratch. There are many reasons to undergo data migration process like application transfer to newer system, database upgradation, technological progress and upgrades or platform change, merger and acquisition (M&A) activities, for implementing novel business models and processes etc.

Data migration is a three step process: Plan, Migrate, and Validate. Planning works at structure level of source and target databases. Migrate is actual data movement phase using Extract, Transform and Load (ETL) process/tools. While Validate is the testing phase for confirmation that moved data is same copy of source database. Major hurdle in the migration process is its huge volume of data. It is assumed that there is a high chance of data/structure has been altered. Thus data validation after data migration process is key step to follow. Mapping document is one of the outcomes of data migration process. It consists of table-to-table, column-to-column level mapping data between source and target database.

Table showing below is a sample of mapping document. The column fields of mapping document can change based on database and its inner configuration. Before data validation actual starts, the correct fields need to be match between source and target databases.

**Table 1: Mapping Document.**

<table>
<thead>
<tr>
<th>Mapping ID</th>
<th>Mapping Name</th>
<th>Source Table</th>
<th>Target Table</th>
<th>Source Column</th>
<th>Source Column Data Type</th>
<th>Target Column</th>
<th>Target Column Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD01</td>
<td>Emp_Cust_id</td>
<td>Employee</td>
<td>Customer</td>
<td>Emp_id</td>
<td>Integer</td>
<td>Cust_id</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>Mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD02</td>
<td>Emp_Cust_Name</td>
<td>Employee</td>
<td>Customer</td>
<td>Emp_name</td>
<td>Varchar</td>
<td>Cust_na</td>
<td>char</td>
</tr>
<tr>
<td></td>
<td>Mapping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>me</td>
<td></td>
</tr>
</tbody>
</table>

Related Work

Methodology of data migration and its validation process is described in several papers. In paper [1], author proposed an automated data validation testing approach and concludes with table of data inconsistencies that have to be checked as a part of data validation process. A detailed migration process is thoroughly discussed in [2]. Paper also focused on various
risks in data migration projects along with types of testing that needs to perform for confirmation of successful migration process. Data quality is one of the major dimensions in QA. If there are data quality issues at source database side, how these leads to an errors at ETL process and also data modeling process in described in [3]. There are a lot of works about data validation testing by manual approach or using automated tool. Sampling approach for data validation is suggested in [4]. Authors has developed a tool data will output sample data size that needs to test, based on data population of data source. In the article [5], authors emphasis on automation of data validation testing for its quality and security. There proposed methodology uses SQL queries with blending of MINUS operator; used to find out missing and extra data from source and target database. A thorough study on DW testing with its various aspects is described in [6]. For perfect solution to test DW, paper discuss general framework of DW. Data Migration Projects Are Risky: 84% of data migration projects fail to meet expectations, 37% experience budget overruns, and 67% are not delivered on time [7].

**Existing System with Limitations**

This section will light on various existing methodologies of data validation testing. Some approaches are manually while some are automated. Ultimately data validation is a process to measure the data migration quality and will assure that the application will have the same functional behaviour after migration.

**Lines of Code:** This is simplest and easily performed data validation process. Simple SELECT query with blending of COUNT operator is executed on individual datasets will work for it. This validation by row counting is like primary testing. If row counts of source and target databases are same, then it assume that migration is successfully done.

```sql
SELECT count(emp_id) from emp_source;
SELECT count(cust_no) from cust_target;
```

This method is beneficial in terms of high speed, especially when COUNT operation is performed on tables which have primary or unique keys. Though method is simple, but doesn't verify data values, especially in migration which is accompanied by the data type transformation.

**Sampling Techniques**

This technique assumes that error is uniformly distributed. Thus, randomly any record is peaked from source DB and checked against in target DB. Since method do not test all records, fails to all types of errors. Even these comparisons are time consuming with limited coverage.
MINUS Queries

This technique uses `SELECT` query along with `MINUS` operation. The query is executed on both source and target separately, to detect what source/target has that target/source lacks. Thus `MINUS` queries needs to be executed twice (Source-to-Target and Target-to-Source).

SELECT cust_id FROM cust_source;

MINUS

SELECT cust_no FROM cust_target;

Along with consumption of time and resources, it does no other validation like data type mismatch, null values, data corruption etc. The methodologies described above does not guarantee 100% data coverage, does not detect various types of data inconsistencies; even process is time consuming. In the next section authors has proposed an automated approach of data validation with using simple SQL queries. The Proposed system is implemented and got better results in comparison with existing methodologies with respect to time and cost factors.

Our Analysis: Typically, data migration is a part of various big projects, where migrating or upgrading database is a key requirement; and hardly perform as a single project. Even, regularly data is not moved as it is; rather many business rules, normalization concepts are applied. Here, authors proposed an automated approach for data validation testing in migration project. The tool automates the comparison of all data from source and target dataset very quickly, thus assuring the quality (QA) of data after migration. Tool will compare all data across the databases and log the errors such as data truncation, extra records in source/target, data type mismatch, missing data, duplicate records, transformation logic errors, NULL translation etc. Using proposed solution, one can effectively save time, cost, and manual efforts; along with data quality assurance (QA).

System Architecture: The proposed automated tool is robust, agile and platform independent, used to configure, design and execute test cases.

Figure 1: Automated Data Validation Tool Architecture.
The tool support data validation testing between the database data sources like MySQL, Oracle, SQL Server, MS Access; also file data sources such as delimited file (CSV), fix length file.

A test case is designed and configure for source and target databases individually. Applying primary data validation filter before data comparator is the optimized solution to lessen TC execution time. The objective of data validation testing is to compare target database against source; and retain the exactly same copy of source data into target database, along with applied business rules. Source and target database records are compared based on Key (Record identifier). The automated tool support N * N data sources comparison with extendable reader for supporting new data sources. The proposed tool here support simple to complex business rules design.

Finally, test case execution reports are generated with flexible graphical representation for test case success/failure percentage, summery and detailed record level data comparison report. Report is also customized to show complete detailed report, only fail records and only pass and fail records identifier.

System Implementation

This section will present a thorough inner implementation detailing of tool. Author has implemented the model using Java technology.

Software Requirements: JRE 1.6, My SQL database server

Development and Testing Platform: Microsoft Windows XP

Read only copy of source and target database and a proper database connection are the prerequisites for tool. Test Suite is collection of Test Cases (TC) where every TC corresponds to a test scenario. The tool works around TC designing, configuration, execution and reporting.

![Figure 2: System Implementation Flow.](image)

**TC_Design**: A test case is designed for a specific scenario. The fields associated to test case (TC) like project details, along with TC description are filled.

**TC_Configure**: Configuring TC is nothing but setting up connection with source and target database separately, and testing it. Here, key (record identifier) is chosen for source and target database individually. Proposed automation tool compares data at record to record level using key identifier.

![Image of Data Comparator](data_comparator.png)

*Figure 3: Test Case Configuration.*

**TC_RULE_DESIGN**: Tool is provided with various options of rule design like Equity rule, Comparison rule and Checking rule to apply on column level.

![Image of Rule Design](rule_design.png)

*Figure 4: Test Case Rule Design.*

**TC_Execute**: Finally, properly designed TC is executed. Tool shows a process status bar for user information. Tool provides a function to execute more than one test case simultaneously under one Test Suite.
TC_Report: The outcome of test execution is three different types/levels of reports are generated.

![Test Execution Report]

Summary Report: A summary report of test suite execution shows the no. of pass and fails test cases with percentage of pass/fail in one Test Suite. Report consists of fields like total execution time, total no of records, result status with graphical representation using pie chart.

Detail Report: A detailed report contains individual test case level result details. It contains various fields of information such as total records, total events, total passed/failed events, duplicate records in source and target database, start and end time of test case execution.

Record Level Comparison: This is a record level data comparison result showing expected and actual values from source and target database with highlighted fail records.

Experimental Result

Following table shows Data Volume (No. of records) verses Total Time of Execution (in Sec.) to validate data. From result, easily we can say automation of data validation makes it very simple and lessen the time efficiently.

![Automation Tool Result]
Table: Automation Tool Result.

<table>
<thead>
<tr>
<th>Data Volume (No. of Records)</th>
<th>Total Execution Time (in Sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>2.949</td>
</tr>
<tr>
<td>10000</td>
<td>3.854</td>
</tr>
<tr>
<td>15000</td>
<td>5.226</td>
</tr>
<tr>
<td>20000</td>
<td>6.365</td>
</tr>
<tr>
<td>25000</td>
<td>7.456</td>
</tr>
</tbody>
</table>

Conclusions

This paper presents a practice of the data validation in a project Database migration for QA process. Authors have effectively automated the data validation testing after data migration; the proposed model is tested for Comma Separated Files (CSF), fixed length files, MySQL, Oracle database and cross combinations of these databases. The implemented system is tested on databases having more than thousands of records and proven its efficiency. The proposed skeleton will fit for any sort of source data, spot of customization is needed in terms of complexity and configuration. This process is designed for data validation testing where massive volume of data is involved between the different databases in less time more accurate and reducing effort by 70 to 80 %. This best practice and methodology is ideally suits for any data migration QA.

References


**Corresponding Author:**

**Dr. N.Sudheer***

Email: nidamanuri.sudheer@gmail.com