International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

System and Process for Data Transformation and Migration from Libsys to Koha

Mukesh Pund[1], Parul Jain[2]

[1]Principal Scientist, IT Division, CSIR-NISCAIR ,14, Satsang Vihar Marg, New Delhi - 110067, INDIA
[2]Senior Project Fellow, IT Division, CSIR-NISCAIR ,14, Satsang Vihar Marg, New Delhi - 110067, INDIA

Abstract—The purpose for this paper is to explain the transformation and migration process from Libsys to Koha- an open source library management software. Open source is a development methodology, which offers practical accessibility to a product's source. Koha being an open source software is cost effective i.e. freely available and is customizable according to one's requirements as compared to libsys. Free/open source software Koha is an economical alternative to reliance upon commercially supplied software libsys. So to migrate from libsys to Koha, the source data is being transformed into the target format. The paper discusses various steps for accomplishment of task and the benefits of exploiting Koha over Libsys.

Keywords- open source , library management, linux, marcedit , mysql, transformation, migration, Z39.50 protocol, marc21, libsys, Koha

1. INTRODUCTION

Data migration is an emerging field nowadays because with the advancement in technology, the need grows to exploit the newer technologies instead of the older ones. The newer systems contain advanced features compared to already existing systems. Hence migration from an existing system to a new one is the need of the hour. Data Migration is a process of transferring data from one system to another and it is divided into two processes: (a) extracting data from an existing system into an extracted file and (b) loading data from extracted file into the new application. The new application usually requires data in a different format, hence transformation of data is required for successful migration. The data transformation is the process of transforming data from one format to another and is a mandatory step in data migration as the architecture of target system may be different from source system[1].In this paper, we are discussing the transformation and migration process from LIBSYS to KOHA . LIBSYS is a proprietary software product aiming most convenient and pleasing library experience through its value added features.[12] KOHA on the other hand is an open source library management software. The use of OSS i.e. open source software is becoming very popular now days in the digital libraries across the globe. According to a survey, satisfaction ratings on Koha 's performance on some aspects found "good" and value for money. The use of OSS has tremendously lower down the initial cost of setting up the libraries and improves flexibility in delivery of services to a greater extent. This is the reason for what the number of researchers and librarians are interested and continuously working on the implementation of OSS in digital libraries.[2]

2. WHAT IS KOHA?

KOHA is the **world's first free and open source library management software** that is being implemented in digital libraries. By open source software we are meant to say that the source code of software is freely available and it can be modified, customized or redistributed according to the person's requirement. As with the enhancement in technology, the need pops up for compliant replacement of existing library system and provides the user the ability to receive free software, customize and redistribute for the benefits of whole community. Also the library system should be advanced to meet the present scenario needs. So, in the year 1999, Katipo Communications proposed a new system, KOHA(the Maori word for "gift" or "donation") which was the first's open source Integrated Library Automation Package (ILAP) using open-source tools to be released under the general public licence (GPL) and installed at Horowhenua Library Trust (HLT) in New Zealand, in the year 2000.

2.1 Technical Features:

- The current version is Koha-3.22 .It runs on different platforms, including Linux, MacOSx, FreeBSD, Solaris, and Windows.[3]
- Developed on the Linux OS, Koha is written in Perl, uses the Apache web server, and has better support for multi-RDBMS like MySQL, PostgreSQL.[3]

- The Online Public Access Catalog(OPAC) interface is in CSS with XHTML. It supports all major library standards such as MARC record import/export (MARC 21), Z39.50 and SRU/Wfeature.
- Records are stored internally in an SGML-like format and can be retrieved in MARCXML, Dublin Core, OAI-DC, and Endnote; and the OPAC can be used by citation tools such as Zotero[3].

2.2 Key Features:

- **Full-featured ILS :** Koha is a true enterprise-class ILS with comprehensive functionality including basic and advanced features for customization of software according to a person's requirement. Koha will work for consortia of all sizes, multi-branch, and single-branch libraries.
- **Multilingual and translatable:** Koha has a large number of languages with enhancement and translation in various available languages.
- **Full text searching:** Koha supports powerful searching, and an enhanced catalogue display that can fetch data from Amazon , Google ,etc. It uses zebra search engine i.e. Z39.50 server and client to enhance search ability, data interchange and import data from Library of congress.
- **Web-based Interfaces:** KOHA's OPAC are all based on worldwide technologies – XHTML, CSS, javascript etc. making it a platform independent solution.
- **Attach Files to Records**: Koha's new feature to attach files to records provides the functionality to upload documents in text, pdf or image format along with metadata.
- **No Vendor Lock-in:** It is an important aspect of KOHA as libraries can freely install it if they have the in-house expertise to purchase support or development of services from best available resources or to change support company at any time if found unsatisfactory.
- **New Templates:** Koha's spine labels, barcode labels, staff and patron interfaces are developed with a template system that's easy to theme. The default templates are also provided that compose of 100% valid XHTML and CSS that can be customized.
- **Item Types:** The module is self explanatory as there are various types of items present in Koha and it gives the functionality to create the same so as to provide an attractive front end to users. It can also be used to manage inventory such as cameras, computers, etc.
- **User Management:** Koha manages users by providing integration with systems like Lightweight Directory Access Protocol **(LDAP) ,** Radius, Central Authentication Service **(CAS)** to allow single sign-on

2.3 Koha Modules:

Koha includes various modules to provide tremendous support to its users to enhance its functionalities. It includes:

- ACQUISITION: Koha's acquisition module holds suggestions, budgets, invoices, funds, currencies.
- ADMINISTRATION: It is an exclusive module of Koha that enable users to change global system preferences and other parameters in various aspects to provide better customizability.
- CIRCULATION: Koha includes a fully featured circulation module with circulation rules that are customizable to meet needs of user. It includes checking in and out of books. It also grants offline circulation feature.
- CATALOGING: Koha provides cataloguing features to its users that enable them to search migrated data both for books and serials, amend already existing records ,add a new record in any framework (default or created by user) and fetch from external sources if required.

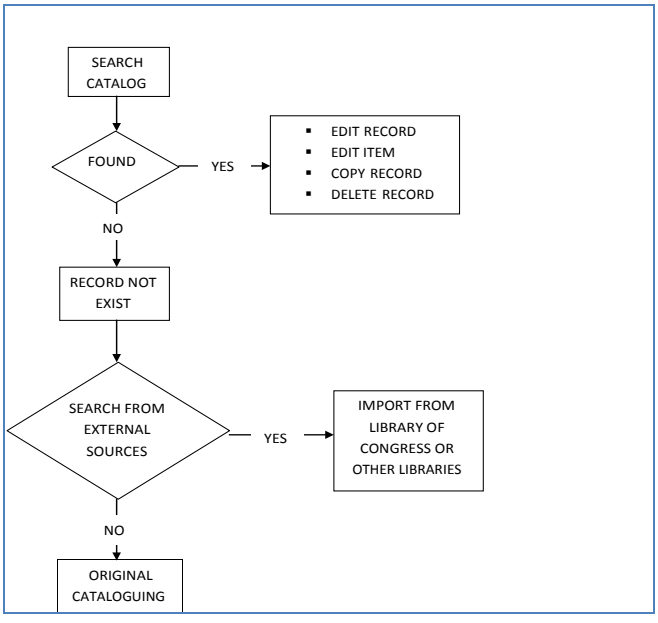

Fig -1: Cataloguing Flowchart

- PATRONS: It enables us to create patrons who exploits circulation module.

- REPORTS: This module provides users the ability to query the data stored in database and generate various reports accordingly.
- SERIALS: The Serials module in Koha is used for keeping track of journals, newspapers and other items that come on a regular schedule.[11]
- TOOLS: Tools in Koha perform some sort of action like notices, slips, patron cards, Batch item management, Records import and management, Calendar, Task Scheduler, etc.

3. SYSTEM ARCHITECTURE WE FOLLOWED:

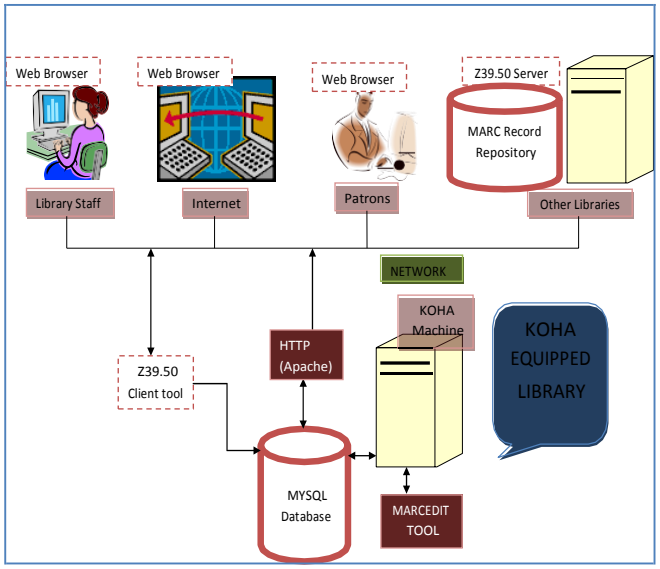

Fig -2: Koha System Architecture

4. WHY KOHA?

S.NO.	CHARACTERISTICS	LIBSYS	KOHA
1.	Nature of developing organization	Commercial	Open source i.e. FREE of cost
2.	Ownership	Libsys	Katipo communications
3.	License	Commercial	Under GPL General Public License
4.	Price	In Lacs	Freely available and free support
5.	Customization	Libsys charge users to provide customized solutions [12]	source code is freely available for innovation to provide new features at users end. New versions are added freely.
6.	Training manual	No system manual is provided to users except user manual to get AMC[4]	YES, manual includes everything for user convenience[4]
7.	Database	Software can be used either with with SQL Server, ORACLE or MYSQL as a backend RDBMS with ODBC compatibility[4]	MYSQL dual database design (Text based and RDBMS). Scalable enough to meet the transaction load of library. [4]
8.	Support	Costly on the basis of AMC(annual maintenance contract) usually 10 to 20% of total costs[4]	Online support and discussion forums free of cost. No human ware for this purpose. Open and constant dialogue with developers.[4]
9.	Vendor Lock –in	Restrictions – can ask for support only from particular vendor	No restrictions , no set term contracts on changing support
10.	Addition of new features/new version	Charge extra cost to upgrade to new version or add new features [4]	Very frequently new versions are coming and added for free[4]
11.	Web Server	Only Apache and IIS	Apache, IIS and others[4]

5. DATA TRANSFORMATION

The transformation of data is a necessary step in data migration as the target format may have a different system architecture which is differentiable from the previous one. It includes data collection, combination, filtration, reformat and so on. It is necessary to find an efficient and effective method for the same so as to improve quality of data. One of the solutions we have undergone for transformation of data is as follows:

5.1 Data In Source Format

The data we have in Libsys is in the format of text file. We have multiple files with accession number as a mandatory field along with other fields:

File 1:

a. Accession number(barcode) – used to uniquely identify a book
b. Title of the book
c. Publisher name and place of publication separated by any delimiter for identification

File 2:

a. Accession number
b. Volume of book and year of publication separated by another delimiter for identification
c. Author of book

File 3:

a. Accession number
b. Classification number
c. Pages, Edition of book separated by some another delimiter and so on.

Also the data contains various blank lines, unwanted content after every record, one record may be separated in different lines and one record may be repeated twice. So it is required to remove all flaws like duplicacy and consolidate the different data into the desired format. The following snapshot will provide you a clearer version of the source data:

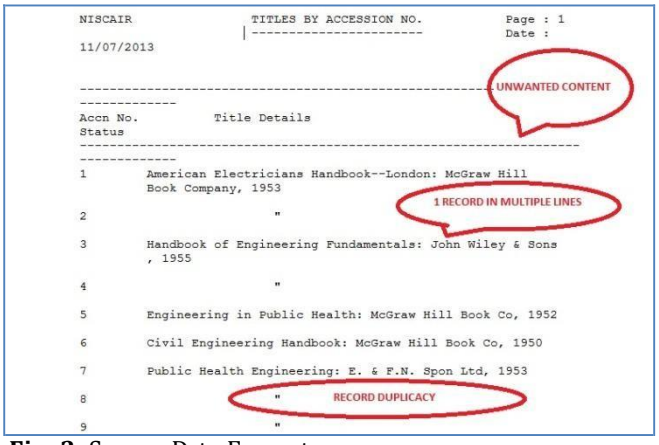

Fig -3: Source Data Format

Here we have multiple files in the same format as above. The first target here is to bring the data in such a format that would be legible and easy to understand. Also the data in various files must be accumulated in a single file for migration.

So question arises is **HOW TO TRANSFORM?**

One of the solutions we have worked to accomplish the task:

5.2 Transformation

It is required that the transformation process should be simple and effective. Each received file is sorted separately and then they are merged afterwards. The Steps followed to sort out the data:

1. Bring the source data in MS –Excel for further processing. Here we have chosen **Microsoft Visual Basic (VBA in MS-EXCEL**) to process data.
   a. Go To the received file and open with→ Microsoft Office excel
   Now here we have different fields in different columns like accession number in first column, title in another and so on. Also

we may have multiple fields in same column. So we can use MS-EXCEL functionalities as well as code in VBA for processing our data.

b.

At first we will use **"Text to Columns in Data Tab"** functionality of excel. We have two options here: fixed width and delimited.
Fixed width: It is used when we have two fields in one column and they are separated by a fixed width. So here we will select that column, use this fixed width option and separate the two fields by a certain width and then Click on "OK". This will separate the two fields in two columns as desired.
Delimited: It is used when we have two or more fields in one column separated by some delimiter say by comma or semicolon. So here also we will select that column, use delimited option and specify that delimiter. It will give a preview of fields in different columns as desired. Clicking on "OK" will give the required output in excel.

c.

But there are also some flaws, like if the source data contains fields say title of book, publisher name separated by delimiter say comma .So there may be a possibility that title may also have comma in it. If we apply delimited method here, it will separate from every position wherever it will witness a comma. So part of title will also be separated in multiple columns along with publisher name .In that case, delimited is not the efficient way to separate. Here programming in VBA in MS-Excel will help to have the desired output.

d.

Press Alt + F11 to go to window where programming needs to be done. Here the programs created for solving problems are called as **MACROS.**

The following figure explains the procedure:

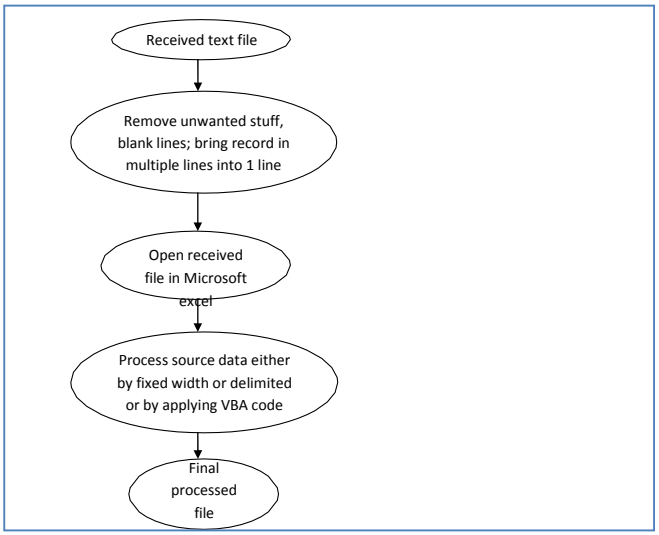

Fig -4: Data Transformation Flowchart

2. Now data in snapshot is assumed to explain the procedure for carrying out the task:

a.

Open the files with Microsoft office excel.

b.

Remove the unwanted content by following algorithm:

Algorithm for removing errors

Step 1: Start
Step 2: Declare variables iRow, LastRow .
Step 3: Initialize variables
iRow = 1
LastRow = ActiveSheet.UsedRange.Rows.Count
Step 4: Repeats the steps until iRow = LastRow
4.1 If data in cell of iRowth row and 1st column contains text "NISCAIR" or "Date :" or "Accn" or "---" , then Delete that iRow
4.2 iRow ← iRow + 1
Step 5: Stop

c.

Row all errors are removed. Then select first column and by fixed width option in text to columns functionality, separate the accession number and title in different fields. It is required to remove all the blank

lines so we created another macro for this task:

Algorithm for removing blanks

Step 1, Step 2 ,Step 3 and step 5 are same as above algorithm

Step 4: Repeats the steps until iRow = LastRow

4.1 If data in cells of iRowth row and 1st 2nd ,3rd ,4th ,5th columns are blank then Delete that iRow

4.2 If data in cells of iRowth row and 1st column is not blank but iRowth row and 2nd, 3rd, 4th, 5th columns are blanks ,then Delete that iRow

4.3 iRow ← iRow + 1

d.

All blank lines are now removed. Some titles are divided in multiple lines so it is required to bring them into a single line. For this we have created another macro.

Algorithm for merging multi row records

Step 1, Step 2 , Step 3 and step 5 are same as above algorithm

Step 4: Repeats the steps until iRow = LastRow

4.1 If 1st column corresponding to (iRow + 1)th row is blank, then Data in iRowth row and 2nd column and (iRow+1)th row and 2nd column gets merged into iRowth row and 2nd column and so on for 3rd column

4.2 If 1st column corresponding to (iRow + 1)th row is blank , then Delete that iRow

4.3 iRow ← iRow + 1

Now we have all the titles in one line. Data also contains same records like if one title is repeated again in the next row then instead of writing the title again, " is written in the next row to signify that the title repeats itself. So for solving this, we created another

macro:

Algorithm for same records

Step 1, Step 2 , Step 3 and step 5 are same as above algorithm

Step 4: Repeats the steps until iRow = LastRow

4.1 If data in cell of iRowth row and 2nd column contain double quotes as a symbol of repetition, then
Data in cell of (iRow-1)th row and 2nd column come in place of iRowth row and 2nd column

4.2 iRow ← iRow + 1

f.

We have some data sorted now but a column with multiple fields separated by delimiter is not yet sorted. Here in the data, we have 'year' separated by comma(,) ; 'publisher' by (:) ; and place by (--). So it is reqAuired to create macro for separating them.

Algorithm for using delimiter to separate using macro

Step 1: Start

Step 2: Declare variables iRow, LastRow , pos, str, le.

Step 3: Initialize variables
iRow = 1
LastRow = ActiveSheet.UsedRange.Rows.Count
str = data in cell of iRowth row and 2nd column
le = length of str
pos = 1st position of comma from right to left in str

Step 4: Repeats the steps until iRow = LastRow

4.1 If pos = 0 , then
Data in cell of iRowth row and 3rd column is blank
Data in cell of iRowth row and 4th column is string str
Else
Data in cell of iRowth row and 3rd column is right part after comma
Data in cell of iRowth row and 4th column is left part before comma

4.2 iRow ← iRow + 1

Step 5: Stop

g.

Similarly publisher and place are also

separated by changing the delimiter sign and column number where the separated field needs to be placed.

h.

We are using MARC 21 library standard which includes various marc tags to which the fields are mapped to migrate data into Koha.

One among the tag is **tag 008.** This tag is used to reflect in Koha whether the title is a handbook or dictionary or encyclopedia and so on. For creating this tag, macro is written as follows:

Algorithm for marc tag 008

Step 1: Start
Step 2: Declare variables iRow, LastRow , pos, str, le.
Step 3: Initialize variables
iRow = 1
LastRow = ActiveSheet.UsedRange.Rows.Count
str = data in cell of iRowth row and 2nd column
Data in cell of iRowth row and 3rd column=

"131209s2013\\\\xx\\\\\\\\\\\\\000\0\eng\d"
Step 4: Repeats the steps until iRow = LastRow
4.1 If str contains text "handbook"
Change bit position 24 to "f" in above string of column 3
4.2 If str contains text "encyclopedia" or "encyclopaedia"
Change bit position 25 to "e" in above string of column 3
4.3 If str contains text "BS"
Change bit position 26 to "e" in above string of column 3
4.4 If str contains text "Proceedings"
Change bit position 29 to "1" in above string of column 3
4.5 iRow ← iRow + 1
Step 5: Stop

5.3 Merging Files

All files are processed separately by following above procedure respectively. It is required to merge all files into a single one. To carry out this task, **VLOOK UP FORMULA** is applied in Excel sheet.

FORMULA:
VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)

Lookup_value : The value to search in the first column of the table array. Lookup_value can be a value or a reference. If lookup_value is smaller than the smallest value in the first column of table_array, VLOOKUP returns the #N/A error value.
Table_array: Two or more columns of data. Use a reference to a range or a range name. The values in the first column of table_array are the values searched by lookup_value. These values can be text, numbers, or logical values.
Col_index_num: The column number in table_array from which the matching value must be returned. A col_index_num of 1 returns the value in the first column in table_array; a col_index_num of 2 returns the value in the second column in table_array, and so on. If col_index_num is:

- Less than 1, VLOOKUP returns the #VALUE! error value.
- Greater than the number of columns in table_array, VLOOKUP returns the #REF! error value.

Range_lookup: A logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match:

- If TRUE or omitted, an exact or approximate match is returned. If an exact match is not found, the next largest value that is less than lookup_value is returned. Values in table_array must be sorted.[13]
- If FALSE, VLOOKUP will only find an exact match. In this case, the values in the first column of table_array do not need to be sorted. If an exact match is not found, the error value #N/A is returned.

The following procedure we have used:

a. Open processed excel sheets and move one by one to a single sheet by right click on sheet name → select move or copy → select the sheet where you want to move

b. Select table_array in each sheet.

c. Apply VLOOKUP in single sheet where all need to merged.

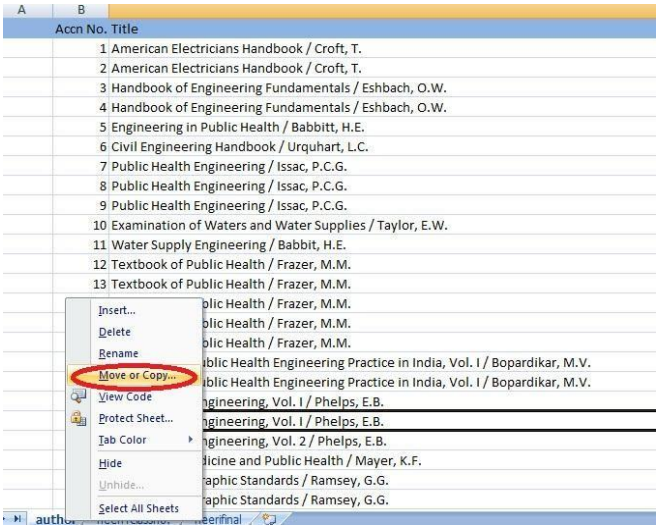

Fig -5: Move or copy one sheet to another

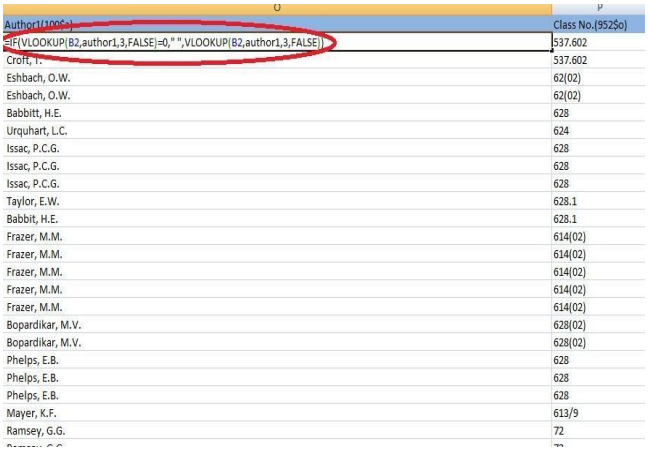

Fig -6: VLookUp Formula

Now we have complete data sorted in multiple fields merged in a single file. The following snapshot will do so:

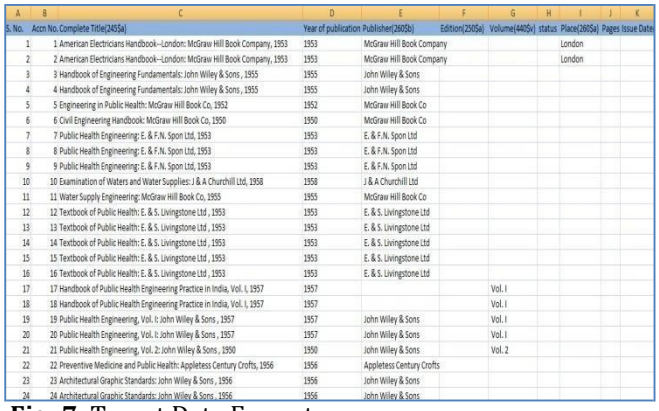

Fig -7: Target Data Format

6. DATA MAPPING

The fields in final excel sheet obtained are mapped with MARC tags. Before moving ahead, let me explain about **WHAT is MARC and WHY it is required?**

Machine – Readable Cataloguing (MARC) was conceived in 1966 as a method of converting the data on Library of congress cards to machine readable form in order to print bibliographic products. At the turn of new millennium it has become an international standard communication format and newest version has appropriately been renamed MARC 21. [5]

Now Question arises WHY there is need for MARC 21?

There is a tendency to transfer towards the MARC 21 because of need for exchange of bibliographic data within the framework of world library network that is based on the MARC 21 format. Reasons are:

- **Standardization:** Standardisation in the exchange formats and structure of a database is essential to facilitate exchange of data in efficient and effective way between the libraries. The adoption of different standard creates incompatibility in exchanging data which act as a major barrier in the use of bibliographic and related information. Format compatibilities are necessary for computerized cataloguing data and these are being standardized by the ISO. The MARC 21 format is one of the popular standard exchange format which adhere to ISO 2709 standard and are using majority of the countries in the world for exchanging data in machine readable form. [6]
- **Other standards under development:** Other standards for encoding digital information in machine readable form such as Dublin core,

extensible mark-up language(XML) are still under development.[5]

➔

Carries information: It carries lot of information in a standard, easy-to-process, clearly designated sequence of bytes.[5]

Now fields in final sheet are mapped with MARC 21 tags. The tags are followed by the name they represent. Examples include:

0XX Control information, numbers, codes
1XX Main entry
2XX Titles, edition, imprint
3XX Physical description etc.
4XX Series statements
5XX Notes
6XX Subject added entries
7XX Added entries other than subjects
8XX Series added entries
9XX Items table information like barcode, etc.[9]

In MARC 21 tags, the notation XX is often used to refer to a group of related tags. For example : 1XX refers to all the tags in the 100s; 100, 110, 130 & so on.
We have mapped fields with corresponding MARC 21 tags .For carrying out this task, we have used **MARCEDIT TOOL** which is a simplified metadata processing tool that provides simplest way to convert excel sheets to marc files – marc text files(.mrk) and machine readable cataloguing file(.mrc) which is required to migrate data into Koha .

6.1 Excel → .mrk

a. Use delimited and click on "NEXT" to accept excel sheet as an input file and .mrk as an output file. The following snapshots will do so:

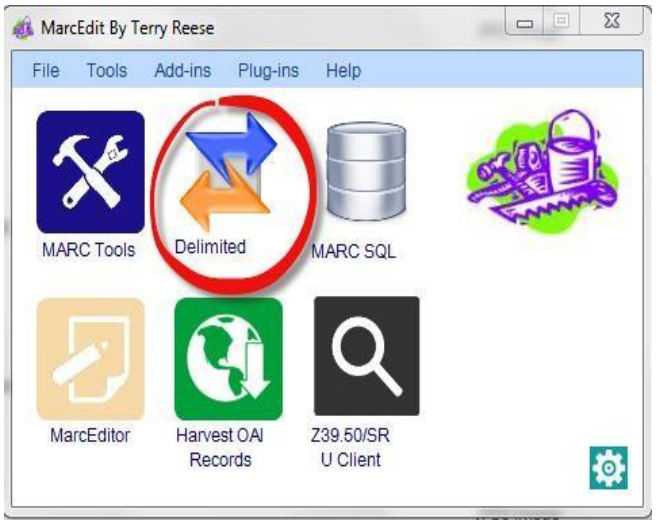

Fig -8: MarcEdit Tool

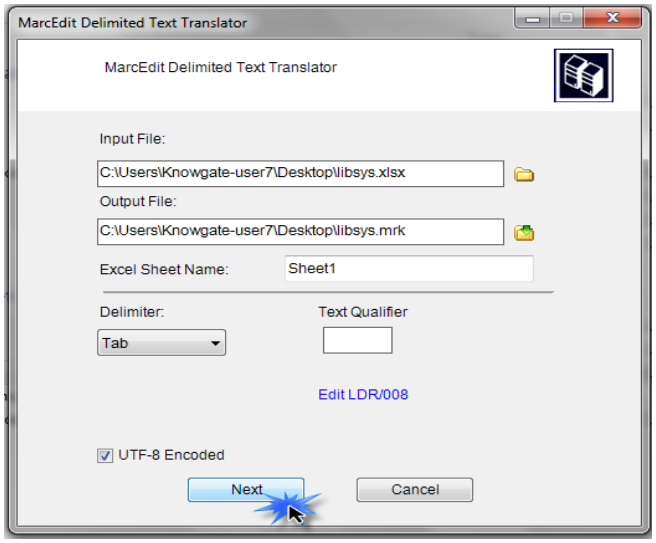

Fig -9: Convert Excel Sheet into .mrk file

a. Map with Marc 21 tags , Join similar items, and click on Finish

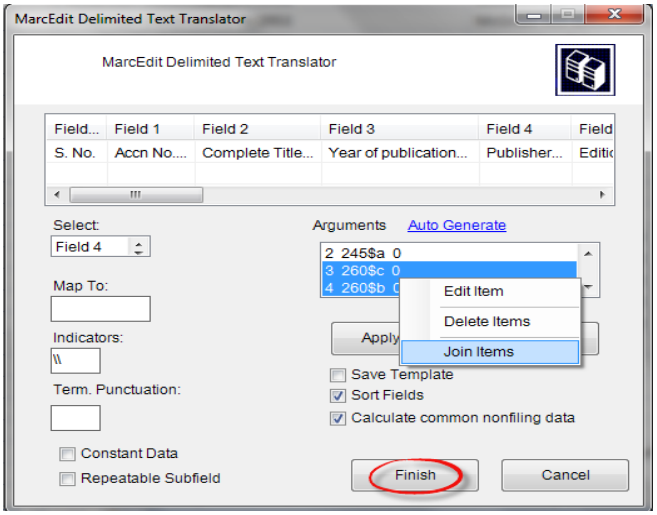

Fig-10: Mapping with marc tags

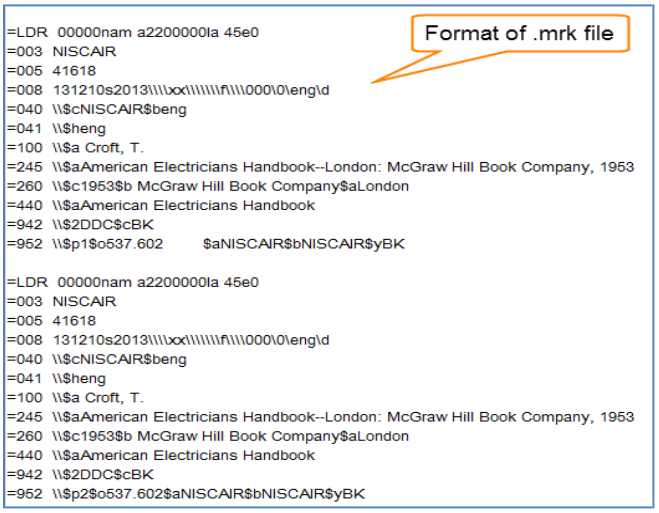

Fig -11: Format of .mrk File

6.2 mrk→ .mrc

a. Select MARC Tools

b. Make .mrk file as an input and .mrc as an output and Use MarcMaker and Execute

c. Format of .mrc file (Fig.14)

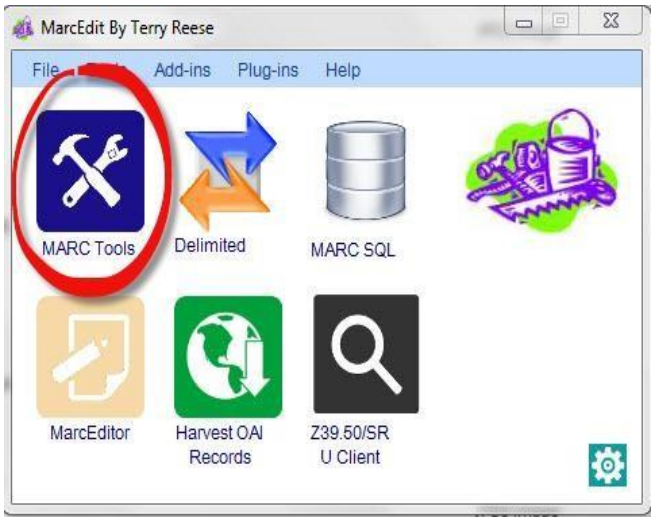

Fig -12: Select Marc Tools

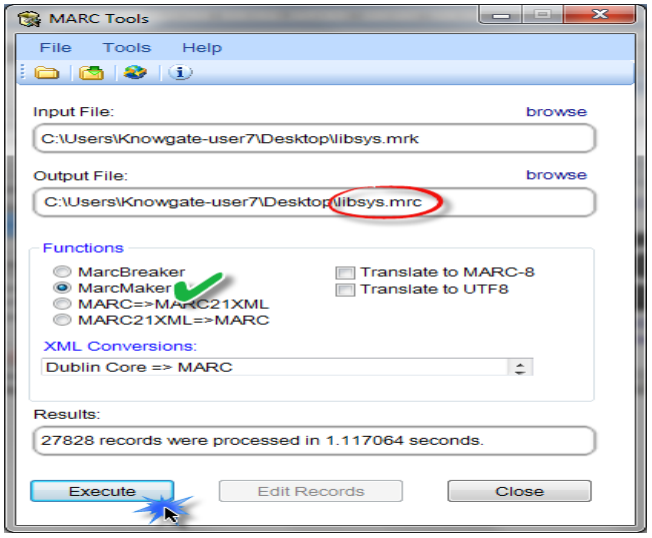

Fig -13: Use Marcmaker

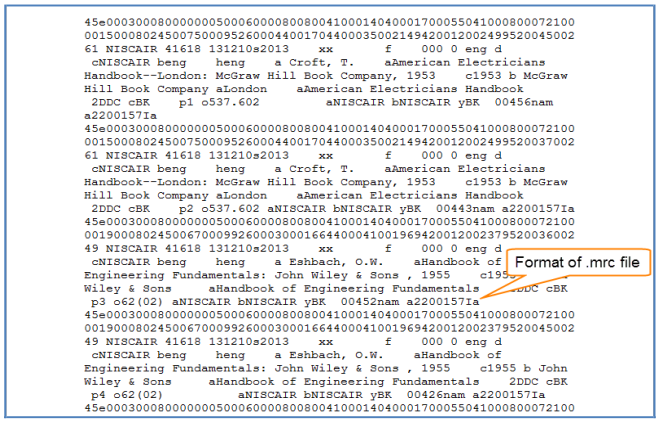

Fig -14: Format of .mrc file

7. DATA MIGRATION

Data migration is the process of transferring data from one system to another. It is an important step and is a

critical process that directly influences the quality of data management. Data migration had affected on the quality of the data, such as, accuracy, data elements, and data accessibility, and all data performances. So it is important that data migration should not hamper quality of data. The steps we have followed to accomplish data migration process includes:

7.1 Upload .mrc File

Upload .mrc file created by MarcEdit tool:

- Go to KOHA→ Home → Tools → Stage Marc Records for Import
- Browse and upload .mrc file created

7.2 Import Batch Into Catalog

- Go To KOHA→ Home →Tools → Stage Marc record management
- Manage Staged Records
- Select framework
- Import batch into catalog

7.3 Rebuild Zebra

One of the frequently used search engines is Zebra. Zebra is used for indexing structured documents (such as e-mail, XML, MARC records) and for the retrieval of documents using the Z39.50 protocol and SRW/U.[7] Records can also be imported from library of congress through Z39.50.
Command used in Linux to rebuild zebra so that all the records get updated in MYSQL database.
[koha @localhost]# perl -I /usr/share/koha /lib/ /usr/share/koha /bin/migration_tools/rebuild_zebra.pl -r -b -v -a

The following Data Flow Diagram explains the complete procedure from Data Transformation to Data Migration

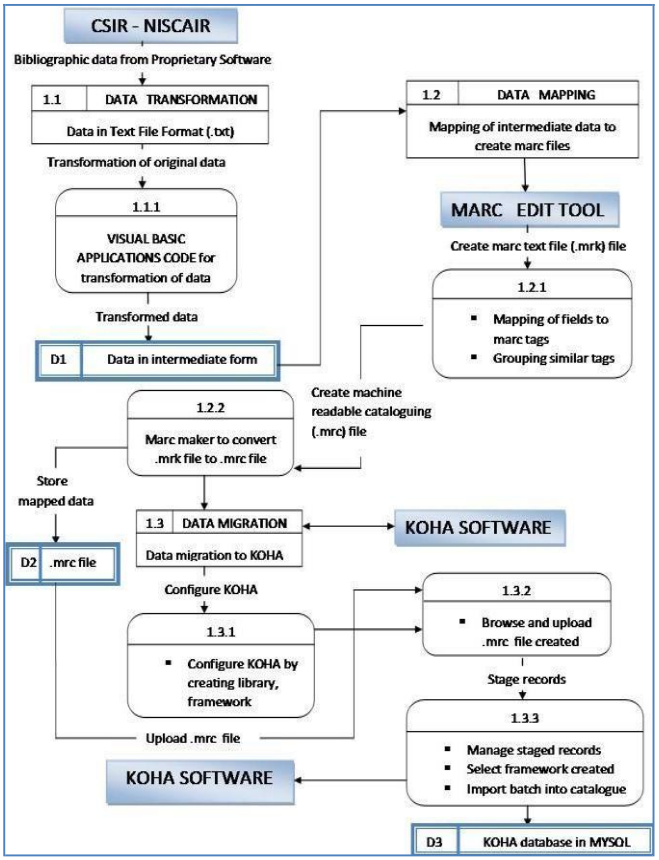

Fig -15: Data Flow Diagram

8. DATA VALIDATION

Data validation is the process of ensuring data quality. Data migration is a critical process that directly influenced the quality of data management. The accuracy of data is fundamental dimension in order to ensure the higher quality of data if the data were wrong, the other dimensions matter little[8].
The following figure defines the common problems faced in data quality during data migration:

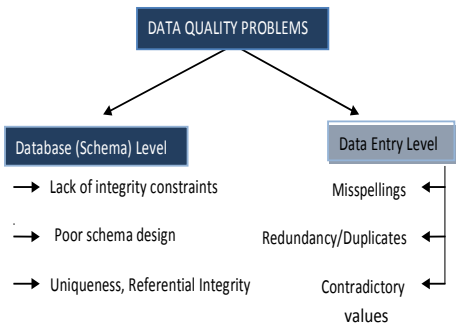

Fig -16: Data Validation

At the database level, MYSQL is used as database, following commands are run to ensure data quality:

[root@localhost]# mysql –u root –p koha
{Koha is the name of database}

Mysql > select * from biblio;
{This will display all biblionumbers and their related information in biblio table}

Mysql > select * from biblioitems;
{This table includes marc information done in data mapping step}

Mysql > select * from items;
{This table holds all information of items migrated to Koha }

Referential Integrity is maintained in the way:
We use various tables in Koha database which are connected to each other via primary key- foreign key hence fulfilling referential integrity. The following figure will show referential integrity among 3 tables: Biblio , Biblioitems and Items [10].

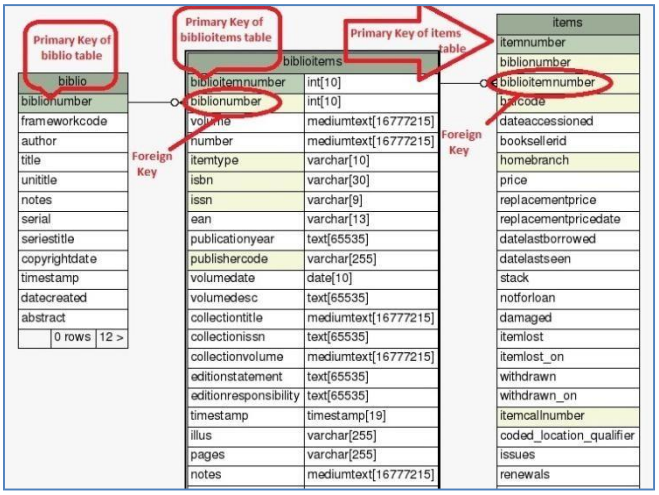

Fig -17: Referential Integrity

At data entry level, problem of misspellings, redundancy and contradictory values are resolved in data transformation process itself (Refer Fig. 3 and Fig.7)

Hence the correctness and effectiveness of transformation and migration process has been validated and thereby data quality is ensured in Koha .

9. CONCLUSION

With the advent of new technology and growth of information technology, it becomes necessary to migrate the data from their legacy system to a new one. The migration cannot be overlooked as a simple step. It is a complex process that holds various phases which makes it prone to failures. Thorough understanding of purpose of migration , proper migration design and predicting the migration output can bring down the possible chances of failure drastically. Therefore, being aware of modern software and technology and current issues in data migration, execution of steps becomes easier and can prove to be critical in successfully accomplishing a migration project.

10. ACKNOWLEDGEMENT

Special thanks and appreciation goes to Sanjay Burde, Senior Principal Scientist, Charu Verma, Principal Scientist and Salim Ansari, Senior Technical officer for their tremendous support.

11. REFERENCES

[1] Cheong Youn and Cyril **S.** Ku Bell Communications Research, “Data Migration”, Piscataway, NJ 08855-1379,p.1255,1992.
[2] Dr. Sanjay Kataria, Mohit Sharma and Anshul Pachouri, “Integrating Open Source Knowledge Management Tools into Library Management for Automation: A case study of Jaypee Institute of Information Technology University”, Noida, India, p.317, 2010.
[3] K.T. Anuradha, R. Sivakaminathan and P. Arun Kumar, “Open-source tools for enhancing full-text searching of OPACs-Use of Koha, Greenstone and Fedora”,Bangalore,India,p.233,2011.
[4] Shivpal Singh Kushwah, J. N. Gautam and Ritu Singh, “Library Automation and Open Source Solutions Major Shifts & Practices: A Comparative Case Study of Library Automation Systems in India”, India, p. 148, 2008.
[5] Zahiruddin Khurshid, “From MARC to MARC 21 and beyond: some reflections on MARC and the Arabic language”,Dhahran, Saudi Arabia,p.370, 2002.
[6] Dhrubajit Das, “MARC 21 : The Standard Exchange Format for the 21st Century”, Ahmedabad, India, p.154, 2004.
[7] Branko Milosavljevic, Danijela Boberic´ and Dusˇan Surla, “Retrieval of bibliographic records using Apache Lucene”, Novi Sad,Serbia,p.526,2009.
[8] Ikhlas Fuad Zamzami, Hanan Abdullah A. Fatani and Nuha Abdullah H. Zammarah, “Data Migration Challenges: The Impact of Data Quality”, Kuala Lumpur,Malaysia,p.1.
[9] http://www.loc.gov/marc/bibliographic/
[10] http://schema.koha-community.org
[11] http://manual.koha-community.org/
[12] http://www.libsys.co.in/
[13] https://support.office.com/

12. BIOGRAPHIES

Principal Scientist & Principal Investigator, CSIR Knowledge Gateway Project at CSIR-National Institute of Science Communication and Information Resources, New Delhi
E-mail:
mukeshpund@niscair.res.in

Senior Project Fellow, CSIR Knowledge Gateway Project at CSIR-National Institute of science communication & Information Resources, New Delhi
E-mail:
er.paruljain91@gmail.com