International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

A COMMODITY DATA CLEANING SYSTEM

Ajay C. Gohel[1], Avishar V. Patil[2], Pratik P. Vadhwana[3] , Harsh S. Patel[4]

[1 234]*Department of Computer Engineering,YTCEM, India*

Abstract-- *Large number of data is recorded into company's database server. Presently to maintain & cleaning data is very important, time consuming & costly process. For big organization Accurate is one of the important aspects. For example, Database of bank should be accurate & consistent. Any error into database can damage bank reputation. Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis. The data mining is used to transform, clean, compress & store data. The data collected can be dirty. Dirty in terms of inconsistent & inaccurate. In current system the faculties have to take care while making the entries. They have to be careful while making entries as well as they have to check for ensuring the data is correct or not. Even after checking if the errors are still present bank have to hire an expert who will perform cleaning operation which is costly & time consuming. By using our system, we import bank database to our system. After importing dashboard will be appear which will show the data of database & at the bottom there are buttons given to perform operation. This operation solves issues like inconsistency, redundancy & inaccuracy.*

1. INTRODUCTION

Data has become an important asset in today's economy. Extracting values from large amounts of data to provide services and to guide decision making processes has become a central task in all data management stacks. Ensuring the quality of the data with respect to business and integrity constraints has become more important. There is no end-to-end solution to error detection and correction. In particular, there is no commodity platform similar to general purpose DBMSs that can be easily customized and deployed to solve application-specific data quality problems. So, This project is design for organizations or companies where large number of data is recorded into company's database server. In current system the faculties have to take care while making the entries. They have to be careful while making entries as well as they have to check for ensuring the data is correct or not. Even after checking if the errors are still present bank have to hire an expert who will perform cleaning operation which is costly & time consuming. By using our system, we import bank database to our system. After importing dashboard will be appear which will show the data of database & at the bottom there are buttons given to perform operation. This operation solves issues like inconsistency, redundancy & inaccuracy.

2. LITERATURE SURVEY

Michele Dallachiesa, Amr Ebaid [1] published a paper *"NADEEF: A COMMODITY DATA CLEANING SYSTEM"* in 2013 in New York city. There is no commodity platforms that can solve data quality problems. In this paper they represent, NADEEF architecture, an extensible, generalized and easy-to-deploy data cleaning platform. NADEEF distinguishes between a programming interface and a core to achieve generality and extensibility. The programming interface allows the users to specify multiple types of data quality rules, which uniformly define what is wrong with the data and how to repair it through writing code that implements predefined classes. The core provides algorithms to detect error and to clean data. Treating user implemented interfaces as black-boxes, the core provides algorithms to detect errors and to clean data. The core is designed in a way to allow cleaning algorithms to cope with multiple rules holistically, i.e., detecting and repairing data errors without differentiating between various types of rules. Using real-life data, we experimentally verify the generality, extensibility, and effectiveness of our system.

Akshata dagade, manish mail, nerandra pathak [2] published a paper *"Survey of data duplication detection and elimination in domain dependent and domain independent database."* on 05 may 2016. In this paper author explains about data duplication problem. In data duplication problem, record matching is done because of that we can find out duplicate records from the system. To eliminate duplicate records from the database, data cleaning is needed which the important phase. Duplication detection is done in two ways either detecting duplicate record in the single the database or detecting duplicate record in multiple other the databases. It improves the data quality to provide better decisions support system. This paper provides the survey on data duplication detection and elimination using several methods which reduced the record linkage problem, record comparison and elimination time in single and multiple the databases.

Ihab F. Ilyas, Sanjay Krishnan [3] published a paper "*Data Cleaning: Overview and Emerging Challenges*" on 26 June 2016. Detecting and repairing dirty data is one of the perennial challenges in data analytics, and failure to do so can result in inaccurate analytics and unreliable decisions. There has been a surge of interest from both industry and academia on data cleaning problems including new abstractions, interfaces, approaches for scalability, and statistical techniques. Using Machine Learning to improve

the efficiency and accuracy of data cleaning and considering the effects of data cleaning on statistical analysis.

Dileep Kumar Koshley, Raju Halder [4] published a paper"*Data Cleaning: An Abstraction-based Approach" in 2015.* The presence of bad data in databases may affect badly the quality of query answers in information processing systems. For instance, in case of decision making processes, the influence of incorrect or inconsistent data on the result of data-analysis may lead to a false decisive stand for an organization. Bad data may occur due to various reasons and falls into various categories.

Some of them are:-

- Data Ambiguity
- Measurement errors
- Data Integration errors

3.DESIGN

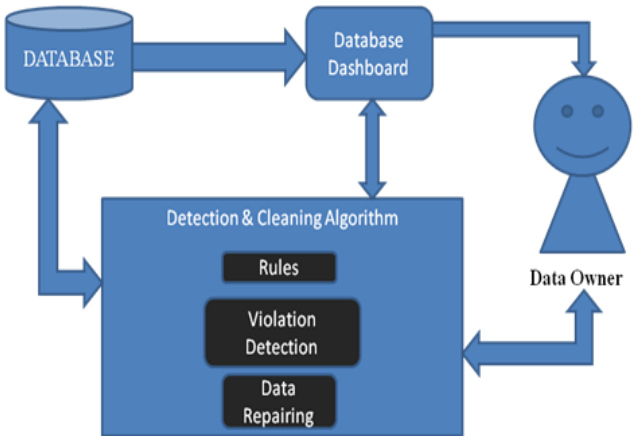

Fig. 1 System Architecture

3.1.Methodology

In fig 1 we have shown architecture of our system. Initially it will take two database. First database is Master DB and second is Transaction. As the data contain various violation & errors so it is called dirty data. This dirty data is given to our system. Our system shows this database on our system's dashboard. On system dashboard user will perform various cleaning operation. The core of our system consists of rules, violation detection, and data repairing.

Rule block consist of many algorithm which resolve the data errors present in dirty data. Our system provide rules for resolving inconsistency, redundancy & inaccuracy. Based on requirement user select this rule. This rules are given to violation detection block.

Violation detection detects error values into the database. This violation is highlighted shown on the dashboard. So user can see what the errors are. These detected errors are given to data repairing block.

Data repairing block takes rules & the data which is detected in previous block is taken as input. This block insert, delete & update the value of databases. For inconsistency it will find error value & replace it with correct values. For redundancy, duplicate rows will be shown to user & it will be merged & stored into database. Once the data repairing is performed it will be updated into database as well as it will display it on dashboard so it is available to our system.

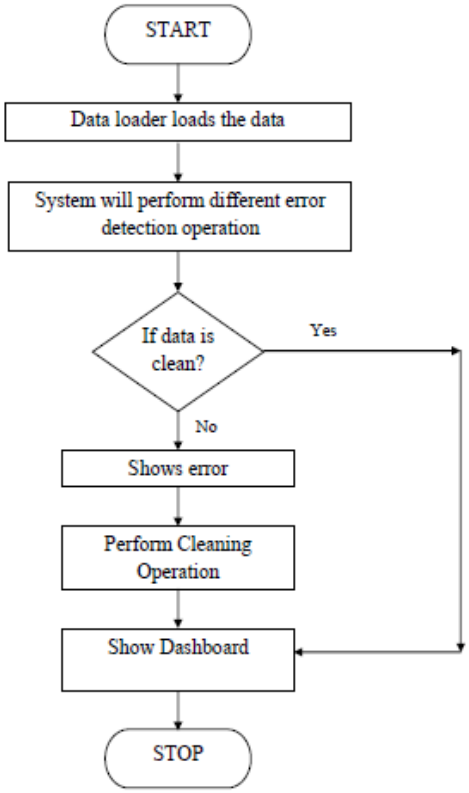

Fig. 2 Flow Chart

4. Example

Consider two databases D1 and D2 from a bank:D1 maintains customer information collected when credit cards are issued and D2 records credit card transactions. The databases are specified by the following schemas:
bank (FN; LN; St; city; CC; country; tel; gd),
tran (FN; LN; str; city; CC; country; phn; when; where).

Here, a bank record specifies a credit card holder identified by first name (FN), last name (LN), street (St), city, country code (CC), country, phone number (tel) and gender (gd). A tran tuple is a record of a purchase paid by a credit card at time when and place where, by a customer identified by first name (FN), last name (LN), street (str), city, country

code (CC), country and phone (phn). Example instances of bank and tran tables are shown in Figs. 3(a) & 3(b) respectively.

	FN	LN	St	city	CC	country	tel	gd
t_1:	David	Jordan	12 Holywell Street	Oxford	44	UK	66700543	Male
t_2:	Paul	Simon	5 Ratcliffe Terrace	Oxford	44	UK	44944631	Male

(a) D_1: An instance of schema bank

	FN	LN	str	city	CC	country	phn	when	where
r_1:	David	Jordan	12 Holywell Street	Oxford	44	UK	66700543	1pm 6/05/2012	Netherlands
r_2:	Paul	Simon	5 Ratcliffe Terrace	Oxford	44	UK	44944631	11am 2/12/2011	Netherlands
r_3:	David	Jordan	12 Holywell Street	Oxford	44	Netherlands	66700541	6am 6/05/2012	US
r_4:	Peter	Austin	7 Market Street	Amsterdam	31	UK	55384922	9am 6/02/2012	Netherlands

(b) Database D_2: An instance of schema tran

Fig. 3 Bank table & Transaction Table

Based on the application business logic, users may impose the following rules:

1. (on table tran) if a customer's CC is 31, but his/her country is neither Netherlands nor Holland, update the country to Netherlands.
2. (on tables bank and tran) if the same person from different tables has different phones, the phone number from table bank is more reliable.
3. (on table tran) a country code (CC) uniquely determines a country.

4.1. Algorithms:

Rule 1: Find & Replace.

1. Start.
2. Select column CC from master & transaction table.
3. If CC from master & transaction is same & country name is same than go to step 7.
4. Else check CC column of both Table.
5. Ask User to insert Correct CC into transaction table.
6. Update Correct Value In transaction table.
7. Stop.

Rule 2: Match & Merge.

1. Start.
2. In transaction table, compare each row with other rows.
3. If value of all rows are different go to step 7.
4. Else Select matching row from table & show it to user.
5. Merge this row in single row.
6. Update this value into table.
7. Stop.

Rule 3: Compare Phone numbers.

1. Select tel column from both the table.
2. If tel is same & all the other details are same than go to step 6.
3. If all the details are same & phone number is different show it to user.
4. Ask user to insert correct phone no.
5. Update the values into database.
6. Stop.

5. CONCLUSION

This System is very user friendly & efficient to use. Core part of our system implements algorithms to detect & repair dirty data by treating multiple types of quality rules holistically. A commodity data cleaning system resolves errors like inconsistency, accuracy& redundancy. Also it stores the clean data back into original database so it is accurate & available to users.

Future Scope:

Currently the system is focusing on bank scenario. We will try to improvise our system so it can be used for other organizations. It detect violation like inconsistency & redundancy so we will introduce some new functions which can resolve other database constraints.

Several extensions underway-

1. To handle large volume of data.

2. New programming interface which allows users to flexibly define multiple type of data quality rules.

3. We will design more user friendly interface to help users define their rules easier.

REFERENCES

[1] Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F. Ilyas, Mourad Ouzzani: Nan Tang. SIGMOD '13 Proceedings of the 2013 ACM SIGMODI nternational Conference on Management of Data. New York, NY, USA ©2013.

[2] Akshata dagade, manish mail, nerandra pathak : Survey of data duplication detection and elimination in domain dependent and domain independent database, 05 may 2016.

[3] Ihab F. Ilyas, Sanjay Krishnan: Data Cleaning: Overview and Emerging Challenges, 26 June 2016.

[4] Dileep Kumar Koshley, Raju Halder **:** Data Cleaning: An Abstraction-based Approach, 2015.

[5] Maksims Volkovs, Fei Chiang, Jaroslaw Szlichta : Continuous Data Cleaning.

[6] Peng Taoxin presented : A FRAMEWORK FOR DATA CLEANINGS IN DATA WAREHOUSES.

[7] Mong Li Lee et al : IntelliClean: A knowledge-based intelligent data cleaner.

[8] Chris Mayfield, Jennifer Neville, Sunil Prabhakar: A statistical method for integrated data cleaning and imputation, 2015.

[9] Sapna devi, dr. arvind kalia:Study of data cleaning and comparison of data cleaning tools, 2015.

[10] Akshata anil dagade, manish mail,nerandra pathak: Survey of data duplication detection and elimination in domain dependent and domain independent, 2016.

[11] Dileep kumar ,raju halder : The Abstract Interpretation framework to clean dirty database,2013.

[12] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. TPLP, 3(4-5), 2003.

[13] C. Batini and M. Scannapieco. Data Quality: Concepts, Methodologies and Techniques. Springer, 2006.

[14] https://github.com/daqcri/NADEEF

[15]http://ccom.unh.edu/vislab/VTDP_web_pages/VTDP_Data_Cleaning.html

[16]https://asu.pure.elsevier.com/en/publications/katara-a-data-cleaning-system-powered-by-knowledge-bases-and-crow