Developing a Real Time Method for the Arabic Heterogonous DBMS Transformation

S. M. Hadi , S. Murtatha
Department of Information & Comm. Eng. College of Engineering Al-Khawarizmi ,University of Baghdad
Department of Computer Science, College of Science, University of Baghdad
Received in : 12, June, 2011
Accepted in : 13, July, 2011

Abstract

A common problem facing many Application models is to extract and combine information from multiple, heterogeneous sources and to derive information of a new quality or abstraction level. New approaches for managing consistency, uncertainty or quality of Arabic data and enabling e-client analysis of distributed, heterogeneous sources are still required. This paper presents a new method by combining two algorithms (the partitioning and Grouping) that will be used to transform information in a real time heterogeneous Arabic database environment.

Key word: Heterogeneous DB, Mapping DB, cleaning data and real time transforming

Introduction

One of the first and most important steps in any data processing task is to verify that the data values are correct or, at the very least, conform a set of rules. Data warehouses require and provide extensive support for data cleaning. They load and continuously refresh huge amounts of data from a variety of sources so the probability that some of the sources contain"dirty data" is high. Furthermore, data warehouses are used for decision making, so that the correctness of their data is vital to avoid wrong conclusions [1].

Many applications today need information from diverse data sources, in which related data may be represented quite differently, in one common scenario, if a DBA wants to add data from a new source to an existing warehouse DB, the data in the new source may not match the existing warehouse schema and this will cause a problem [2].

In any scenarios, one or more data sets must be mapped into a single target representation. Needed transformations may include two approaches:

- Schema transformations (changing the structure of the data)
- And data transformation with cleansing (changing the format and vocabulary of the data and eliminating or at least reducing duplicates and errors).

In each area, there is a broad range of possible transformations, from simple to complex. Schema and data transformation has typically been studied separately. We believe they need to be handled together via a uniform mechanism [3].

In addition to a consistent and uniform access to heterogeneous sources there is a further value of integration. The integrated data can contain information, which are not integrated data explicitly but in the form of dependencies, relationships or patrons over the various sources [4].

The Data Cleaning

Data cleaning, also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data. Data quality problems are present in single data collections, such as files and databases, e.g., due to misspellings during data entry, missing information or other invalid data. The data warehouses require and provide extensive support for the data cleaning [5].

They load and continuously refresh huge amounts of data from a variety of sources so the probability that some of the sources contain"dirty data" is high. Furthermore, data warehouses are used for decision making, so that the correctness of their data is vital to avoid wrong conclusions. The need for data cleaning increases significantly. This is because the sources often contain redundant data in different representations. Provide access to an accurate and consistent data, consolidation of different data representations and to elimination the duplicate information, in this future age the cleaning process become necessary [6].

During the ETL process (extraction, transformation, loading), illustrated in Figure (1), further data transformations deal with schema/data translation and integration, and with filtering and aggregating data to be stored in the warehouse. In this Figure, all data cleaning is typically performed in a separate data staging area before loading the transformed data into the warehouse. A large number of tools of varying functionality are available to support these tasks, but often a significant portion of the cleaning and transformation work has to be done manually or by low-level programs that are difficult to write and maintain.

While a huge body of research deals with schema translation and schema integration, data cleaning has received only little attention in the research community. A number of authors focused on the problem of duplicate identification and elimination, while some research groups concentrate on general problems not limited but relevant to data cleaning, such as special data mining approaches and data transformations based on schema matching. More recently, several research efforts propose and investigate a more comprehensive and uniform treatment of data cleaning covering several transformation phases, specific operators and their implementation. [7].

In this paper we focus on the new problem that faces transformation and conversion in ETL process related to our DB environment, which is the Arabic data. Several standard programs and methods (working under windows operating system) are available for transformation of English data in any database. But the same transformation will be so difficult when we deal with an Arabic data in a real time from any special database or warehouses.

The Proposed System

In the present work, four methods have been proposed to solve the problem in a sequential approach to assess the advantages and disadvantages of each method. All are based on finding new relationships between the original Arabic data and the converted un-clear (rubbish) one (using the standard methods). To check the validity of these proposed methods, a case study is conducted to retrieve an Arabic database file from FoxPro DB under Dos to Oracle 10g under window. The proposed system is based on a server to servers' architecture connection; or what we call it as a three tire architecture that will consist of the following:

A- The Server (destination server): This server will represent the database server that collect data from multi-source and can be act as a warehouse server, oracle DBMS will be the core that hold the warehouse data which will be treated and cleaned in the middleware**.**

B- The Middleware: is the software that do ETL process, this part can be built in the destination server or could be build in a separated server to increase the performance of the database and to avoid the load that may be happened on the destination server as shown in Figure (2) that contain the proposed system architecture.

C- The Clients (sources servers): the second server will be the multi source local database servers that contain all the desired data ,each source can be use different DBMS (MS-Access, FoxPro under-DOS, Excel, Mysql…etc).

Most of the Arabic information has a problem when transforming it from one database to another due to many reasons. The reason why the data appears to be different (rubbish) is not related to the difference in the DBMS only. This problem might appear also when transferring the Arabic data to the same kind of DMBS with the same version. The real fact and the main reason to this problem is the difference in the Character Set used in the Operating System combined with the DBMS.

For example, the Operating System (DOS) uses a special character set for the Arabic language and the Operating System WINDOWS uses the (MS-WIN 1256) Character Set for the Arabic language and (MS-WIN 1252) Character Set for the English language. From other hand the Operating System (LINUX), use the different set character for the Arabic language and for the English language. So if we tried to transfer the data that have been written in Arabic language from Oracle10g DBMS using the (MS-WIN 1256 Character Set) hosted by Windows Server 2003 Operating system to another Oracle10g DBMS hosted by another computer that has the same operating system but using the (MS-WIN 1252 Character Set) that will not support the Arabic language, then the transferred data will defiantly appear to be different in shape and meaning (rubbish).

There are many software programs used to solve this problem that will transfer the data to its original form regardless of what the data is, but the real problem is that all these programs works using the Single Patch form, which means it transfers all the data directly and at once for the same DBMS. Because of that the needs to design a new method become required that will enable us to transfer huge amounts of data from any language in Real Time's environment between different DBMS(s) and operating systems.

The Problem Solutions

In this proposed system we implement four solutions to solve the problem that mansion before each of these methods have been implemented on a real case study that contain a kind of an Arabic data. The four methods will be illustrated sequentially as the following:

- **First Method**

It has been thought of many ways to solve this problem, one solution used to be a new method was designed to insert all the Arabic letters and their movements (The movement in the Arabic language used to be treated as a separated letter in the computer that will has a separated ASCII code) into the system that will transfer data from it, the result of translating these data to the related ASCII Code before the operation of transforming and after, can be seen in the Look up Table shown in table (1). This method requires separating the words (data) letter by letter, and each letter is to be worked on and swapped with the corresponding letter in the Look up Table. This method can be explained as the following steps which have been written as a program with the (Procedural Language/Structured Query Language) known as (PL/SQL) in Oracle10g,

1. The data will be converted for each table column by column,
2. And each column will be read raw by raw,
3. And each raw will be read cell by cell,
4. Then each cell will be read word by word
5. Finally each word will be read letter by letter (of course the movement will be treated as a separated letter).

When we transform the Arabic data and getting the rubbish view as mentioned before these rubbish letters will be managed and after reading each one, it will be compared with the Look up Table to obtain the original letter, this is where the swapping procedure will take place. Each rubbish letter will be swapped with the original letter. This method has been highly

efficient when reading the transferred data, however, it is not suitable for huge amounts of transferred data (just like a warehouse database), and due to the fact it takes too much time for the transformation process. Therefore, this method can be suitable only for the small Database that has small amount of data and this will guide us to the second method.

- **Second Method**

The second method has been programmed using SQL Server 2008 as a package in order to get benefit of using the Integrated Services characteristics. This method is quite similar to the first method; the only diference is that it works as a Parallel for the tables and the columns. In fact by using this method it will be possible to transfer more than one table in the same time and even transferring all the columns in one table in the same time, but the process will remain to be Serial for any columns' data (Serial concerning the rows). Obviously this method gives us more speed in the transformation process comparing with the first method.

- **Third Method**

The third method has been built on the second method with some development. With This method we use the Table Partitioning Technique, through this technique the table will be divided in to dummy parts according to the stored data, in a way that each part will contain some portion of the data from all columns (that's mean the dividing will be horizontally not vertically). Here we will have parallel processing for all the Table's parts at the same time, and this will ensure to make this method faster than the second and the first method, but we find that the coming method will be the optimum solution for the problem represented here.

- **Forth Method**

This method is a development of the third method. After studying and analyzing the data in the Look up Table, we discover that an implementation of a minus procedure performed between the ASCII code of the original data and the rubbish data will give us a new fact. This operation of having the difference between the Two ASCII cod will gain a special number that will be related to a certain amount for each group of letters (each group of letters beholds one difference as shown in Table No. (1). by using some analysis we will reduce the possibilities of searching from (35) possibilities to (10) possibilities only. This will lead the program not to search in all the Look up Table In another word by eliminating the amount of possibilities we reduce the I/O operations.

In this method we will divide all the Arabic letters to only (10) ASCII groups, then during the transformation process we will check each letter in which group it will be related, then we will make the swap operation on the original ASCII code according to its group number. This method will become the fastest comparing with the other 3 methods mentioned above; it is indeed the method in which we recommend when transferring huge amounts of data in the Real Time environment as shown in Figure (3).

Result and Conclusions

Through applying an analysis methods concerning the 4 methods and the time required to perform the transportation operation in a real time environment we have the result of this mathematical analysis shown in Figure (4) which will appear that the forth method will be the new fasts method that will solve the problem of transferring a huge Arabic data from one DBMS to another without any rubbish data and the most important in a real time form.

As a conclusion for implementing this method on a real data taken from the database of the Ministry of Internal Affairs we can list the following points:

1. Through the implementation process we concluded that the forth developed method will work very fast in an accurate manner concerning the transformation operation for any DBMS hosted by any Operating System placed in any Network architecture.

2. It has been found that this method can be implemented successfully not only on the Arabic data, in fact it will be powerful also with any language data which have the
3. same characteristics of the Arabic language just like the Kurdish, Chinese, Korean, …..ext.
4. By using the data mining with this method we could obtain different data and information from the DBMS of the different source destination servers and in different languages not only in the Arabic.

References

1. Niswonger, B. ; Haas, L. M. , Miller, R. J.(2009), Transforming Heterogeneous Data with Database Middleware beyond Integration,www.10.1.1.56.1853.edu.
2. Heiner Stuckenschmidt, Ubbo Visser, (2008), Using environmental Information Efficiently*:* Sharing Data and Knowledge From Heterogeneous Sources, The Computing Technologies Center.
3. Kai_Wie sattler and Gunter Sakee, (2006),supporting Information Fusion with Federated database Technology, IEEE computer transaction.
4. Susan B. Davidson and Kosky , S. (2006)**,** A Language for Database Transformations and Constraints**,** www.ICDE97.com.
5. Sundus Norry Shukry, (2007), Decision support system of JAVA infrastructure Analysis and control, Research Journal of Applied Science,USA.
6. Erhard Rahm Hong Hai Do, (2009), Data Cleaning: Problems and Current Approaches**,** www.ghh.com.
7. Ronald Cody, Ed.D., Robert Wood Johnson, (2009), Data Cleaning 101, www.ats.ucla.edu.

Table :(1) Contains the Arabic letters and its corresponding ASCII Code

group	ASCII Difference	Original ASCII	Original Character	rubbish ASCII	rubbish Character	Seq.
Group 1	-11	213	ض	224	À	1
Group 2	-9	216	ط	225	ل	2
	-9	217	ظ	226	Â	3
	-9	218	ع	227	م	4
	-9	219	غ	228	ن	5
	-9	222	ق	231	Ç	6
	-9	223	ك	232	È	7
Group 3	-8	221	ف	229	ه	8
	-8	225	ل	233	É	9
Group 4	-7	227	م	234	Ê	10
	-7	228	ن	235	Ë	11
	-7	229	ه	236	ى	12
	-7	230	و	237	ي	13
Group 5						14
	-2	236	ى	238	Î	rubbish
	-2	237	ي	239	Ï	15
Group 6	24	211	س	187	<<and >>	16
Group 7	39	212	ص	173	-	17
Group 8	40	199	ا	159	ں	18
	40	200	ب	160	space	19
	40	201	ة	161	،	20
	40	202	ت	162	¢	21
	40	203	ث	163	£	22
	40	204	ج	164	¤	23
	40	205	ح	165	¥	24
	40	205	ح	165	لا	25
	40	206	خ	166	¦	26
	40	207	د	167	§	27
	40	208	ذ	168	¨	28
	40	209	ر	169	©	29
	40	210	ز	170	ھ ||||؛	30
	40	211	س	171	«	31
	40	212	ش	172	¬	32
Group 9	41	193	ء	152	ک	33
	41	196	ؤ	155	›	34
Group 10	57	196	ؤ	139	‹	35

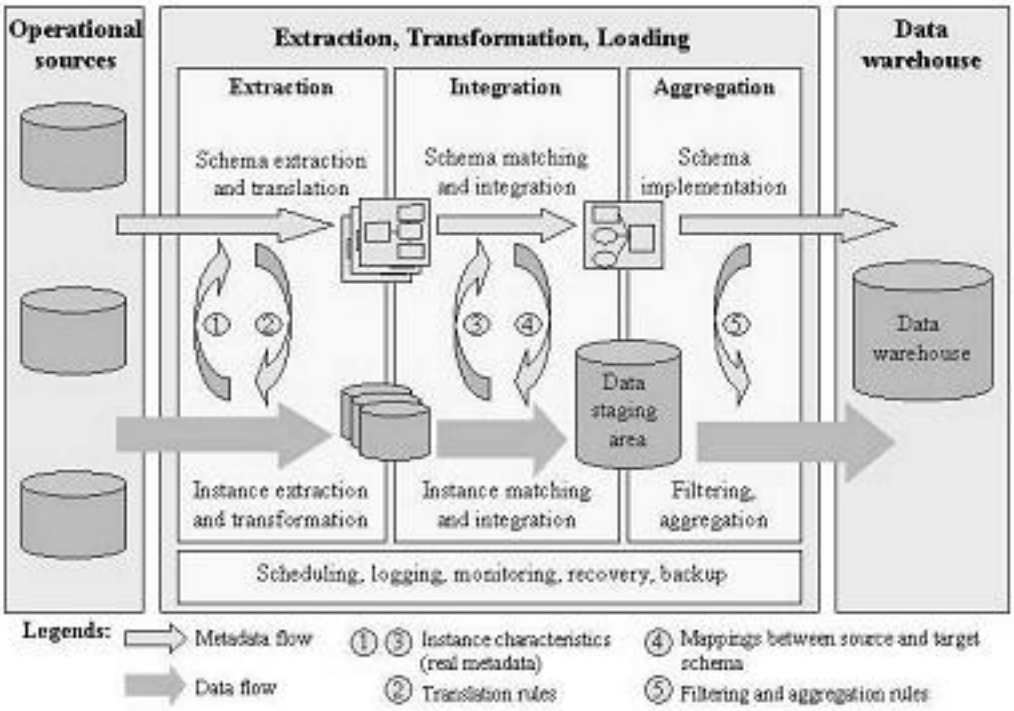

Fig. (1): The ETL process in a data warehouse.[6]

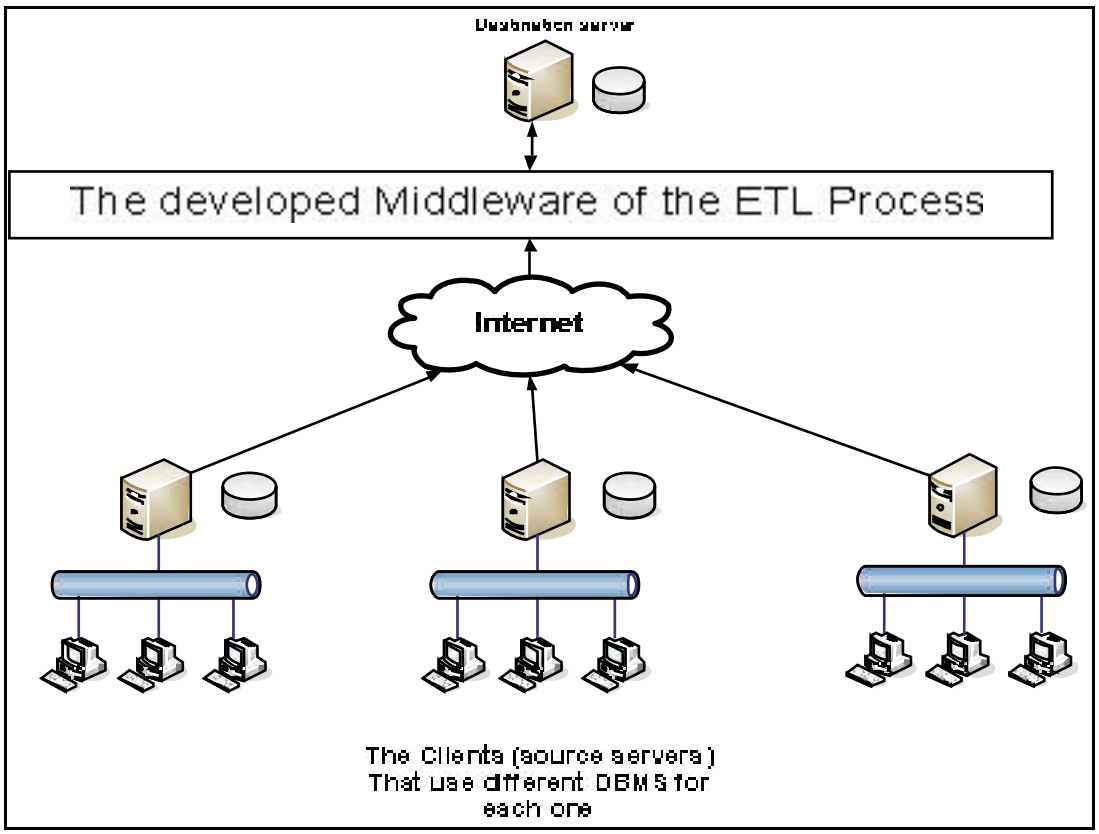

Fig. (2): Represent the Middleware for the proposed System

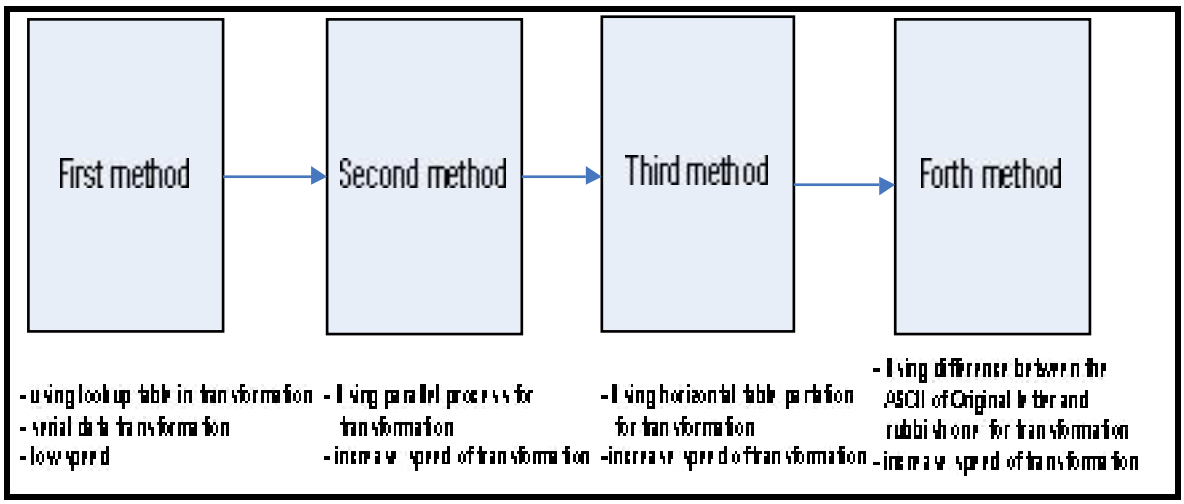

Fig. (3): The Block Diagram of the 4 methods

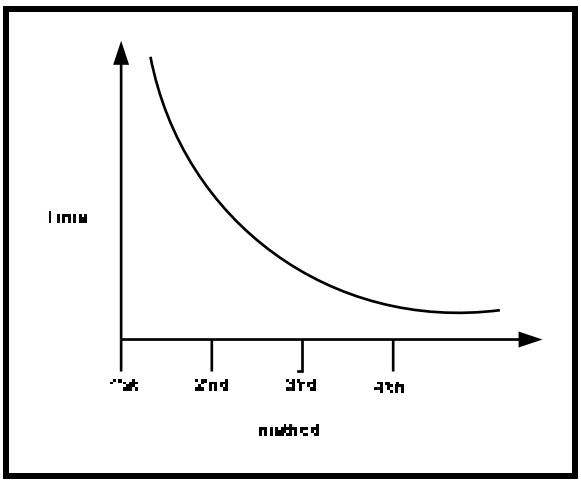

Fig. (4): The Time Analysis for the 4 Methods

تطويرطريقة ضمن الزمن الحقيقي لتحويل البيانات العربية في قواعد البيانات الموزعة

سها محمد هادي ، صفاء مرتضى
قسم هندسة المعلومات والاتصالات ، كلية الهندسة الخوارزمي ، جامعة بغداد
قسم علوم الحاسبات،كلية العلوم ، جامعة بغداد
استلم البحث في : 12،حزيران، 2011
قبل البحث في: 13،تموز، 2011

الخلاصة

ان واحدة من اهم المشاكل التي تواجه نماذج تطبيقات قواعد البيانات يمكن تلخيصها بكيفية التعامل وادارة المعلومات ودمجها من المصادر المختلفة لقواعد البيانات التوزيعية ليتم تخزينها في قاعدة بيانات موحدة . ان الحاجة الى توافر طريقة جديدة للتعامل بثبات من اجل الحصول على نوع المعلومات المخزنة باللغة العربية ضمن قواعد البيانات التوزيعية اصبح ضروريا في الوقت الحاضر .ولهذا سيتضمن هذا البحث تقديم طريقة جديدة تتالف من دمج أثنين من الخوارزمانات المستعملة لنقل البيانات باللغة العربية دون حدوث اي معوقات اوتشويه للبيانات و ضمن الوقت الحقيقي من خلال بيئة قواعد البيانات التوزيعية.

الكلمات المفتاحية: قواعد البيانات التوزيعية، تنظيف البيانات، نقل البيانات ضمن الزمن الحقيقي