Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (21): 4442-4448, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.4442.4448

Research on Algorithm of Partitioning Table Set in Data Exchange Process

[1]Jin-lai Lv, [2]Qiu-lin Yang, [3]Jing Lv and [1]Ai-ping Li
[1]College of Computer Science and Technology, Taiyuan University of Technology ,Taiyuan, China
[2]National Pilot School of Software, Harbin Institute of Technology, Harbin, China
[3]College of Arts and Laws, Wuhan University of Technology, Wuhan, China

Abstract: There are some problems on the order of table data migration in XML-based data exchange system. Based on these problems, an algorithm of partitioning table set, which solves those problems on the order of table data migration effectively, is proposed. First of all, this paper analyses the problems of data among heterogeneous relational database during the data update process and demonstrates the significance of the research. After that, based on the algorithm of partitioning table set theory, which is detail discussed in this paper, an available algorithm is proposed. Finally, using an example of data exchange process based on XML, the study describes and discusses how to ensure the data order migration with the help of the algorithm of partitioning table set. The results show that the algorithm of partitioning table set improves the efficiency of resource usage and provides an excellent resolution of some problems, such as real-time quality and high efficiency of data in the system.

Key words: Relationship, reference table, algorithm of partitioning table set, XML, data exchange system

INTRODUCTION

Building a specific information technology application, developers may face the task of integrating the data from remote heterogeneous database into the data center (or integrating the data from data center into remote heterogeneous database). When fulfilling data migration and exchange among different Database applications, the heterogeneity of relational databases doesn't allow data to exchange directly among different databases (Shao-Hua and Xie, 2010; Li-Hua, 2010). There are many researches that make XML (eXtensible Markup Language) as a common data model to realize data exchange among heterogeneous relational databases (Xu *et al*., 2003; Li and Lossless, 2002; Barbosa *et al*., 2004; Shen, 2002; Hu, 2004; Tan and Li, 2011; Xiao-Heng, 2009; Rahman *et al*., 2011; Qian and Fu, 2011; Xiong-Jun, 2010). XML not only can describe relational data well but also has self-describing, extensible and platform-independent features and can be a transformation standard of data transmission and exchange among different relational database systems (Xin-Yi, 2009; Leopoldo and Loreto, 2008; Reddy *et al*., 2011). Therefore, by integrating information resources to achieve information sharing and exchange across the systems through XML, which makes using the existed information resources efficiently, has become an important trend of information technology.

In a database application, the tables are not independent of each other but correlated with each other (Rahman *et al*., 2011; Qian and Fu, 2011).

These relationships raise a question, that is, when data migrate from one database to another, the data in the referenced table must be migrated to the target database before the migrating data in the referenced table, if not, there will be data reference errors during the migration process. In complex database application system, the number of the tables in the database could be as many as hundreds or even thousands, the reference relationships among them are more complicated. So, what kind of efficient algorithms could be used to determine the order of data migration in these tables? Using related theories of set theory, this paper proposes a new algorithm on solving this question, named "algorithm of partitioning table set ", which is a good solution to this problem.

MATERIALS AND METHODS

Select Oracle 9i as the source database, MS SQL Server 2005 as the target database. Choose Kettle as the secondary development tool of data exchange application.

Kettle is ETL (Extract-Transform-Load) open source projects, mainly including four parts: Chef, Kitchen, Spoon and Pan. Chef is a job design tool. Kitchen is job executor which is used as perform operations. Spoon is a conversion design tool used for designing process of

Corresponding Author: Jin-lai Lv, College of Computer Science and Technology, Taiyuan University of Technology, 030012, Shanxi, China

data conversion. Pan is conversion executor and the transformation task is generated by Spoon.

The exchange order of each table data in the database adopt "algorithm of partitioning the table set" algorithm. More details about the algorithm (partitioning the table set algorithm) are described in chapter 3.

Using the method in the data exchange process of the extranet E-government network system not only exchanges timely, rapidly, completely and has no exception occurs but also the data of the exchange before and after is consistent. The validity of this method is better illustrated.

ALGORITHM OF PARTITIONING TABLE SET

The purpose of data exchange as described in Fig. 1, is to migrate the table structure and (or) the corresponding data in the database A, database B, etc., to the data center database. A particular point is that during the data migration process, for the existing relationships among database tables, there are certain sequence requirements to the table data migration and the referenced table (main table) data must be migrated before the corresponding referring table (detailed table) data, otherwise, there will be reference exceptions.

When the number of tables in the database are not many and the relationships among the tables are not very complex, it is comparatively easy to solve the problem of table data migration order in database but when the number of tables are huge and the relationships among the tables is complex, it becomes a great headache to deal with the problem of table data migration order. While very few domestic and foreign scholars have made researches in this regard, this paper demonstrates the corresponding processing algorithm based on set partitioning theory of set theory.

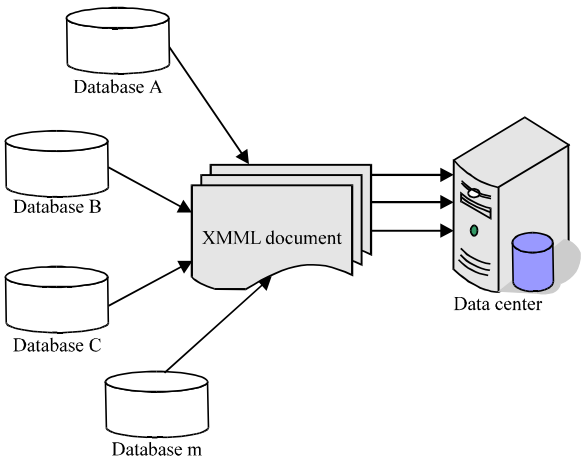

Fig. 1: Data exchange process modelc

Principle of table sequential algorithm: Divide collection of database tables by reference relationship to determine the order of table data migration/update. Principle 1 shows formal description about how to partition the table set in accordance with the reference relationship.

Principle 1 Given a non-empty set A, family of sets, π = {A1, A2,···, Am-1, Am}, constituted by non-empty subsets of A, is a partition of A, if π has the properties (Yi-He, 2007; Duo-Qian; Guo-Dao, 2008):

$$\forall A_i, A_j \in p, i, j = 1, 2, \ldots, m, , \text{if } A_i \neq A_j, \text{ then } A_i \cap A_j = \Phi$$

And:

$$\bigcup_{A_i \in p} A_i = A$$

Get partition of A, π, π = {A1, A2, ¡–, Am-1, Am} by dividing non-empty table set A. The division of the set is as follows:

- Put all the tables without foreign keys in table set A into set A1, same as tables that refer to the primary key of themselves
- Put all the tables in table set A that refer to the primary key of tables in A1 into set A2
- Put all the tables in table set A that refer to the primary key of tables in A2 into set A3 and these tables may also refer to the primary key of tables in A1
- Put all the tables in table set A that refer to the primary key of tables in Ak (k>0) into set Ak+1 (k+1 ≤m) and these tables may also refer to the primary key of tables in Aj (0<j<k)
- Repeat step (4) until π, division of table set A is got

Realization of table sequential algorithm: The specific realization process corresponding to principle of the algorithm is as follows:

- The formation of the initial matrix of table relationship
- Assumption is that there are n tables, ta1, ta2··· tan in A, a non-empty table set. The relationship matrix of all tables in the entire database is shown in Table 1

In Table 1:

Table 1: Relationship matrix

RM	ta_1	ta_2	…	ta_n
ta_1	a_{11}	a_{12}	…	a_{1n}
ta_2	a_{21}	a_{22}	…	a_{2n}
…	…	…	…	…
ta_n	a_{n1}	a_{n2}	…	a_{nn}

$$a_{ij} = \begin{cases} 0 & \text{when } ta_j \text{ is not referenced by } ta_i \\ 1 & \text{when } ta_j \text{ is referenced by } ta_i \end{cases}$$

Formation of a table subset Ai: For the existing matrix of table relationship, travel each row of the matrix, when there are lines whose value of each element is 0, put the corresponding table into table subset Ai, that is how table subset Ai is got (The value of i here starts from 1, whenever there is a new table subset generated, i increases the value by 1).

Correction of the value of table relationship matrix: For each table element tak (k = 1, 2,···, m) in table subset Ai, set the value of each element in k-column of table relationship matrix 0.

Row reduction of table relationship matrix: In the matrix of table relationship, delete the rows corresponding to each table element tak (k = 1, 2,···, m) in table subset Ai to form a new matrix of table relationship;

Continue to generate a subset of the table: Check for the new generated matrix of table relationship, if the number of rows in the matrix of table relationship is not 0, return to step (2) and generate a new subset of this tables; if the number of rows in the matrix of table relationship is 0, then it is the end of the table set partitioning.

Data migration process: Suppose R is a binary relation on the table set partition π and R={<Ai, Aj>, i<j}. Binary relation R-1 is an inverse relationship on binary relation R. When updating the latest inserted data (Insert data) from the source database to the target database, for any Ai and Aj ($0 < i < j \leq m$), the Insert data in tables of set Ai must be first updated to target database, then the Insert data in tables of set Aj. The update of the Insert data in set Ai must be completed before Aj, otherwise it will lead to data reference exceptions. Therefore, the binary relation R is anti-reflexive, anti-symmetric and transitive. By the definition of quasi-order (Duo-Qian and Guo-Dao, 2008) the binary relation R is a quasi-order on π.

When updating the latest deleted data (Delete data) from the source database to the target database, for any Ai and Aj ($0 < i < j \leq m$), the Delete data in tables of set Aj must be first deleted from target database, then the Delete data in tables of set Ai. The delete of the Delete data in set Aj must be completed before Ai, otherwise it will lead to data reference exceptions. Therefore, the binary relation R-1 is anti-reflexive, anti-symmetric and transitive. Similarly, by the definition of quasi-order (Duo-Qian and Guo-Dao, 2008), the binary relation R-1 is a quasi-order on π.

When updating the latest updated data, that is the Update data, from the source database to the target database, it will not cause data reference exceptions. Therefore, there is no need to update the Update data to target database by a fixed order.

CASE STUDY

Selection of test case: Select Oracle 9i as the source database, MS SQL Server 2005 as the target database. Shown in Fig. 1, assume that tables stored in the database A are shown in Fig. 2, where the arrows pointing to are the referenced table, tables without arrow lines connected to have no reference relationship.

Implementation of data exchange: The entire process of data migration from database A to dada center is shown in Fig. 3.

Establish auxiliary table of main table: The purpose of establishing auxiliary table is to facilitate the data update on the target database when there are insert, update and delete operations on tables in the source database. The data types of auxiliary table's primary key are the same as the main table's primary key, the value of auxiliary table's operatype column is limited to 1, 2, or 3 (1, 2, 3 represents the types of operation on records-insert, update, delete). In order to prevent the trigger and Kettle (a data exchange tool) from operating on one auxiliary table at the same time, two auxiliary tables need to be created, Auxiliary Table 1 and Auxiliary Table 2. When Kettle operates on Auxiliary Table 1, if the trigger would also operate on Auxiliary Table 1, let the trigger operates on Auxiliary Table 2. When Kettle completes its operation on Auxiliary Table 1, move data in Auxiliary Table 2 to the Auxiliary Table 1. That is, Auxiliary Table 2 is created to make Auxiliary table 1 complete its work.

Establishment of the trigger of the main table: The function of trigger is to record the type of each operation on main tables to auxiliary tables.

To determine the migration order of the table:

- To Fig. 2, the table set δ = {user1, vip, incomeshare, relationship_7, role, ordersheet, relationship_5,

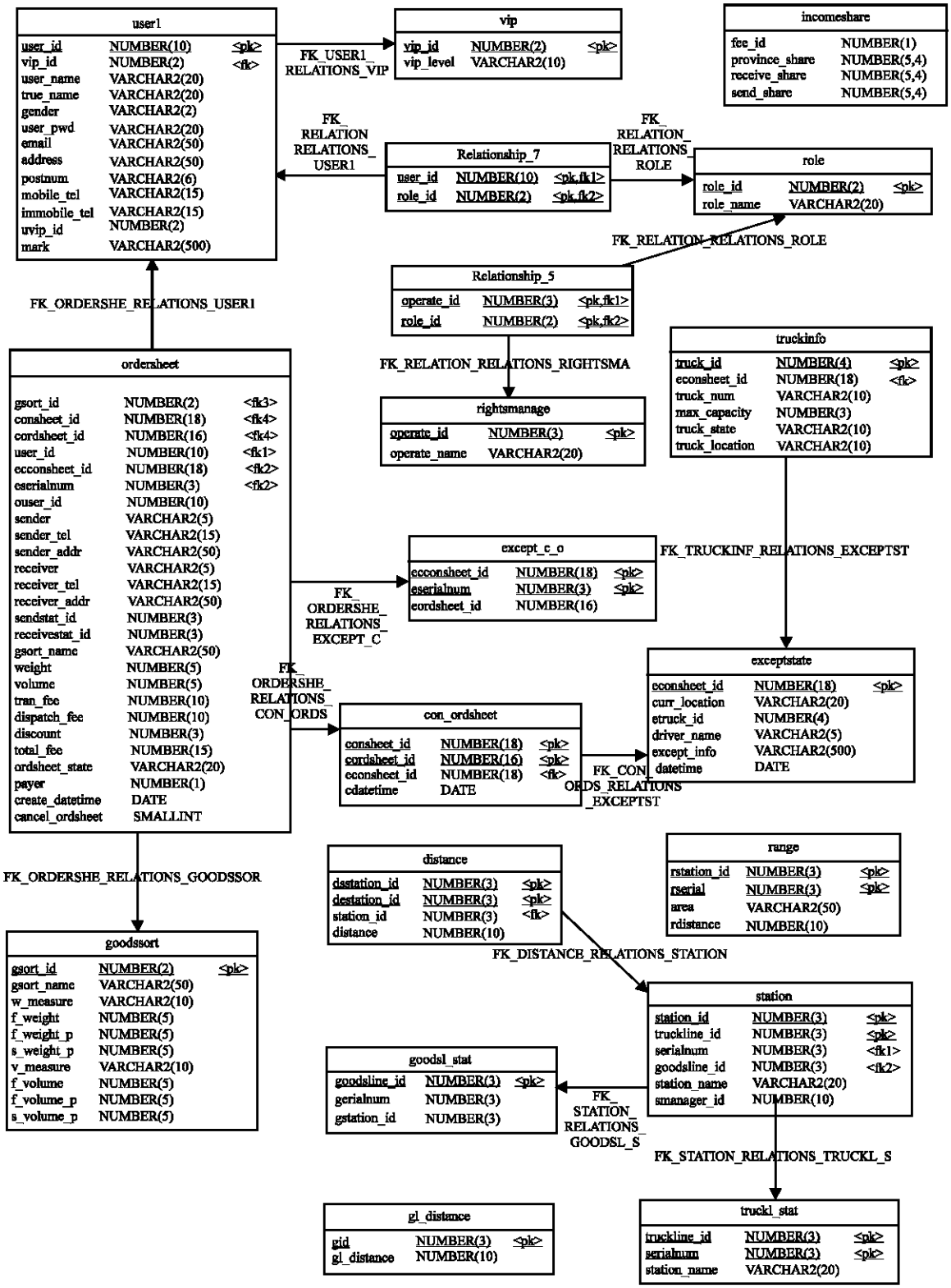

Fig. 2: Physical relationship model of database table

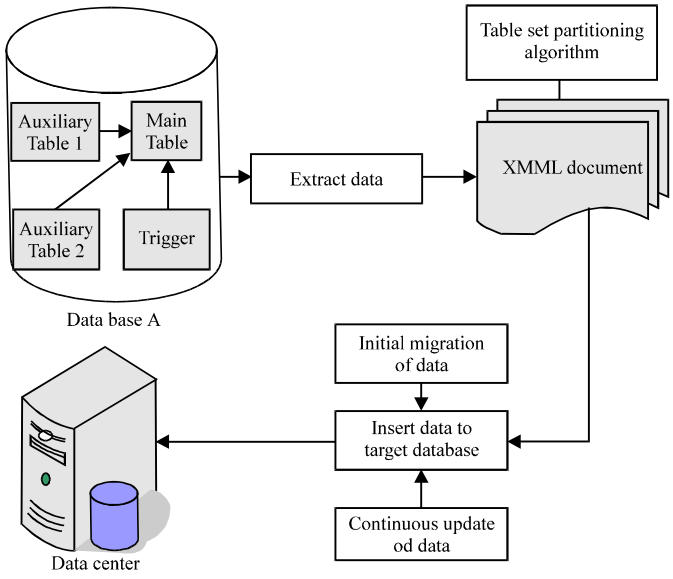

Fig. 3: Model of data migration process from database A to dada cent

Table 2: Reference matrix

RM	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T
A	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
G	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

rightsmanage, truckinfo, except_c_o, exceptstate, con_ordsheet, goodssort, distance, station, goodsl_stat, truck_stat, range, gl_distance}, through the analysis of the relationship of each table, get the reference matrix of table relationship as shown in Table 2 (A--user1, B--vip, C--incomeshare, D--relationship_7, E--role, F--ordersheet, G--relationship_5, H--rightsmanage, I--truckinfo, J--except_c_o, K--exceptstate, L--con_ordsheet, M--goodssort, N--distance, P--station, Q--goodsl_stat, R--truck_stat, S--range, T--gl_distance).

- Comparing to the relationship matrix of Table 2, travel each row of the matrix and add table in which the value of all elements in a row are 0 to table subset A1, then gets A1 = {vip, incomeshare, role, rightsmanage, except_c_o, exceptstate, goodssort, goodsl_stat, truck_stat, range, gl_distance}
- Modifying the relationship matrix of Table 2, according to each table element tak (k = 1, 2,···, 11) in relationship matrix A1 and set the value of all elements in the k queue of this relationship matrix to 0
- Based on each table element tak (k = 1, 2,···, 11) in table subset A1, delete their corresponding rows in the matrix of Table 2 and then a new table relationship matrix like Table 3 formed
- Repeat the operation from (b) to (d), then gets A2 = {user1, relationship_5, truckinfo, con_ordsheet, station}, A3 = {relationship_7, ordersheet, distance}. As a result, Partitioning table π = (A1, A2, A3) gets and the migration order of

Table 3: New reference matrix

RM	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

he table data we get is as the following: Vip, incomeshare, role, rightsmanage, except_c_o, exceptstate, goodssort, goodsl_stat, truck_stat, range, gl_distance, user1, relationship_5, truckinfo, con_ordsheet, station, relationship_7, ordersheet, distance

Generate the mapping model: Mapping model mainly deals with three kinds of data update behaviors, which are insert, modify and delete on tables in the target database. When inserting new data to the target database and modifying existing data in the target database, the mapping model processes XML data documents according to the determined migration order of table data. If deleting data in the target database is wanted, the mapping model processes XML data documents according to the inverse migration order of table data.

The initial migration of data

The purpose of the initial migration of data is to migrate data in the source database to the target database. Generic process of initial data migration:

- Get XML data files from the FTP server, which are extracted from the source database, put XML data files on the local directory to facilitate local data operations
- Insert data in the XML files corresponding to tables in set A1 into the data center
- Insert data in the XML files corresponding to tables in set A2 into the data center; and then repeat the same operations on the set A3, ..., Am-1, Am until all data are inserted into the target database (data center)

Continuous update of data: The purpose of continuous data update process is to update the recently updated data in the source database to the target database in time, to ensure the accuracy of the data in the data center. The process of continuous update of data is:

- In accordance with a certain time interval, check the FTP server directory, if there are XML data files, get XML data files and put them on the local directory, then empty FTP server directory to delete XML data files, if there are no XML data files, then this process ends, re-execute step 1
- Update all the data in Update_XML, Insert_XML and Delete_XML files to the target database (data center)
- Go to step 1 remark. Up to now, the authors in the study don't see any thoughts or views similar to the table sequential algorithm proposed here, so when solving data migration problems which resemble the process displayed here, experimenters can try this method boldly and bravely.

CONCLUSION

Algorithm partitioning table set provides a good solution for the order problem of data exchange/update in data migration process, which lays the foundation of establishing efficient data exchange system. Efficient data exchange system not only can avoid data inconsistence and untruthfulness caused by duplicated construction of various departments but also can expand the range of date usage and improve the efficiency of data usage.

ACKNOWLEDGMENTS

This study was financially supported by the Science Research Project in Shanxi Province (No. 20080322011, No. 20110311039-2, No. 20120321030) and Youth Foundation of Taiyuan University of Technology (No. 2012L067).

REFERENCE

Barbosa, D., J. Freire and A.O. Mendelzon, 2004. Information preservation in XML-to-relational mappings. Proceedings of the International XML Database Symposium, August 29-30, 2004, Toronto, Canada, pp: 66-81.

Duo-Qian, M. and L. Guo-Dao, 2008. Rough Sets Theory Algorithms and Applications. Tsinghua University Press, China.

Hu, X., 2004. Developer's Guide of Oracle9i XML Network Database. Beijing Hope Electronic Press, China.

Leopoldo, B. and B. Loreto, 2008. Information sharing agents in a peer data exchange system. Proceedings 1st International Conference on Data Management in Grid and Peer-to-Peer Systems, September 3, 2008, Turin, Italy, pp: 70-81.

Li, W. and Lossless, 2002. Mapping from semi-structured data to structured data. J. Southeast Univ., 18: 46-53.

Li-Hua, Z., 2010. Research on heterogeneous exchange database technology based on XML. Sch. Newspaper Suzhou Coll. Technol., 23: 77-80.

Qian, S. and Z. Fu, 2011. Design and implementation of the data share and data exchange system based on SOA. Proceedings of the 3rd International Conference on Communication Software and Networks, May 27-29, 2011, Xi'an, China, pp: 697-699.

Rahman, M.A., S.M. Masud Karim and I. Kiringa, 2011. Tabular representation of schema mappings of a data exchange system. Proceedings of the 14th International Conference on Computer and Information Technology, December 22-24, 2011, Bangladesh, Dhaka, pp: 423-427.

Reddy, K.S.K., G. Narayana and M. Padmavathamma, 2011. Learner integrated data exchange system architecture. Proceedings of the 2nd International Conference on Software Engineering and Service Science, July 15-17, 2011, Beijing, pp: 63-65.

Shao-Hua, H. and H. Xie, 2010. Integrated model of heterogeneous relational database based on XML technology. Comput. Eng. Design, 31: 5285-5288.

Shen, Z., 2002. Integrate Application of Java and XML Database. Tsinghua University Press, China.

Tan, M. and Y. Li, 2011. Design and implementation of general distributed heterogeneous data exchange system. Proceedings of the IEEE 3rd International Conference on Communication Software and Networks, May 27-29, 2011, Xi'an, pp: 416-420.

Xiao-Heng, J., 2009. XML word store in the research of relational database. Comput. Progr. Skills Maintenance, 24: 56-57.

Xin-Yi, Z., 2009. XML Simple Tutorial. Tsinghua University Press, China.

Xiong-Jun, Z., 2010. Development and Application of Network Database. Tsinghua University Press, China.

Xu, Z., Q. Liu and Y.H. Dong, 2003. Review of XML storage technology based on relational database. Comput. Eng. Appl., 39: 197-200.

Yi-He, W., 2007. Introduction to Discrete Mathematics. 3rd Edn., Harbin Institute of Technology Press, USA.