Data Integration of Separated Databases Using Schema Based Approach

Swe Zin Win
University of Computer Studies, Yangon
swe1288@googlemail.com

Abstract

Today, many business applications continue the ongoing creation of separated data stores; data mining often requires data integration for those data. This process becomes significant in a variety of situations both commercial and scientific. The main contribution of data integration system is that user can focus on specifying what data they want rather than on describing how to obtain it. This paper proposes a system for integrated access to data that are related, but exist in separated databases. In this system, a data integration tool is created for different data that are stored in different databases. This system will make integrated access based on participant databases' export schemas by using data dictionary and type dictionary. Each export schema contains information about attributes containing in each database. This system also defines attribute names, attribute types, and attribute length for tables that user wants to integrate and gives integrated results to user.

1. Introduction

To efficiently retrieve information from separated data sources, these source need to be integrated into one single system such that the user can retrieve the desired information through the integrated system by a single query. First solution is finding similarities between entities within data sources. Second solution is finding the data elements that are most highly associated to each column and then match columns that share these important data elements. In this system, second solution is used and data integration is based on participant databases' export schemas. The schema contains information about data elements. This paper is organized as follows: Section-2 presents the theory, objectives and benefits of the data integration, Section-3 presents the proposed system architecture, Section-4 presents the implementation of the system, Section-5 presents the evaluation of the system, Section-6 presents the conclusion.

2. Background Theory

2.1 Data Integration

Data integration is the process of the standardization of data definitions and data structures by using common conceptual schema across a collection of data sources. One important step in integrating separated data sources is matching equivalence attributes. Another important step is determining which field in two databases refers to the same data [6]. Integrating data sources involves combining the concepts and knowledge in the individual data sources into an integrated view of the data which isolates users from the individual system details.

The main contribution of data integration system is that user can focus on specifying what data they want rather than on describing how to obtain it. Data integration system involves combining data residing in different sources and providing users with a unified view of this data. Data integration is the process in which:

1. Takes as input two separated databases (export schemas), and
2. Produces as output a single unified description of the input schemas (the integrated schema) and associated mapping information supporting integrated access to existing data through the integrated schema.

2.2 Schema Based Integration

A schema is a model of data sets which can be used for both understanding and querying data. Schema integration is the process of combining local schemas into a global, integrated view by resolving conflicts present between the schemas [13]. Schema gives the best way to understand the semantics of the data. In schema based integration, system creates export schemas for participant databases. Each schema contains attribute names, attribute types, and attribute lengths for each database. There are main steps for schema based integration as follows:

1. System performed object matching stage by using information from export schema. Object matching is

important step for this system. Object matching is based on object names of original databases. Object Matching is intended to determine which object in one database corresponds to which object in another. It takes two database schemas and produces a mapping between elements of the two schemas that correspond semantically to each other.

2. System performed attribute matching stage depends on objects and export schemas. Attribute matching is important step for this system. Matching attributes is based on attribute name, attribute type and attribute length of original databases. Attribute Matching is intended to determine which attribute in one table schema corresponds to which attribute in another. It takes two table schemas and produces a mapping between attributes of the two schemas that correspond semantically to each other.

The proposed system resolves naming conflicts and type conflicts by using data dictionary and type dictionary.

2.3 Objectives of the Proposed System

The proposed system used schema based approach to resolve type conflicts and naming conflicts using data dictionary and type dictionary. By integrating separated databases, user can reduce data duplicating and costing time for accessing data. This system can also resolve foreign key problem and primary key problem.

2.4 Related Work

As related work, significant research progress has been made towards Data Integration. First work is interoperability of heterogeneous databases using the WWW by Gustavo Zanini Kantorski [6]. This work presents a tool for integrated access to heterogeneous databases through the Internet. The development of such tool was based on two assumptions:(a) the use of WWW resources for database access,(b) integrated access to available information without any conceptual schema, database or local applications modifications. As a second work, there has been an explosion of work [13]) in integrating heterogeneous databases using neural networks. This work involves extracting semantics, expressing them as metadata, and matching semantically equivalent data elements that present a procedure using a classifier to categorize attributes and train a neural network to recognize similar attributes.

3. Proposed System Architecture

3.1 System Flow Diagram

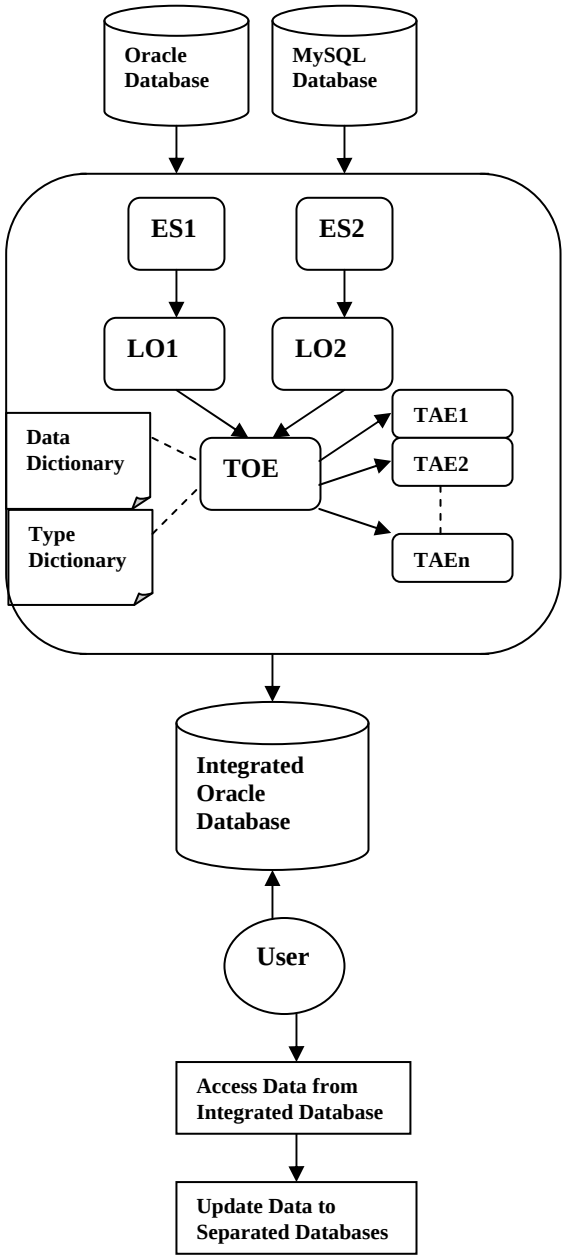

Figure.3.1 System Flow Diagram

In Figure (3.1), the required abbreviations are defined as follows:

ES1 is Export Schema 1 for Oracle Database.
ES2 is Export Schema 2 for MySQL Database.
LO1 is Local Object1 for Oracle Database.
LO2 is Local Object2 for MySQL Database.
TOE is Table of Object Equivalence.
TAE is Table of Attribute Equivalence.

Figure.3.1 shows the system flow of this paper. Input data is got from Oracle and MySQL data stores.

When system received data, system performed four steps as follow:

Step 1: System defined ES1 and ES2 for each databases. ES1 is Export Schema of the Oracle Database and ES2 is Export Schema of MySQL Database. Each ES contains text descriptions for *attributes* in each database such as attribute names, attribute types, and table names for each attribute.

Step 2: System created LO1 and LO2 for each databases based on ES1 and ES2. LO1 is Local Object for Oracle Database and LO2 is Local Object for MySQL Database. Each LO contains *available objects* for each database. Each object depends on information of attributes contained in ES.

Step 3: After the creation of ES and LO, the system creates TOE by matching LO1 and LO2. TOE is Table of Object Equivalence. It contains objects that can be access by client and attribute names for each object. By matching objects from LO1 and LO2, the system can resolve foreign key problem.

Step 4: In this step, system creates TAE1…TAEn. TAE is Table of Attribute Equivalence. TAE contains information about attributes corresponding with each objects containing in TOE. Number of TAE is number of objects from TOE. When TAEs were created, the system uses Data Dictionary and Type Dictionary for deciding what attributes are same and integrate. Data Dictionary contains synonyms and default words for attributes. Type dictionary contains type names and default type. By creating Data Dictionary, the system finds out that what two attributes are semantically matched. By creating Type Dictionary, the system finds out that what two attributes are typically matched. In this step, all matching for attributes are based on Data Dictionary and Type Dictionary.

After system performed above four steps, the system set data to integrated databases. User can access data from resulting integrated database. If user wants to update attributes to tables of separated databases, user can update and can integrate after update.

4. System Implementation

4.1 Database Design for Oracle Database

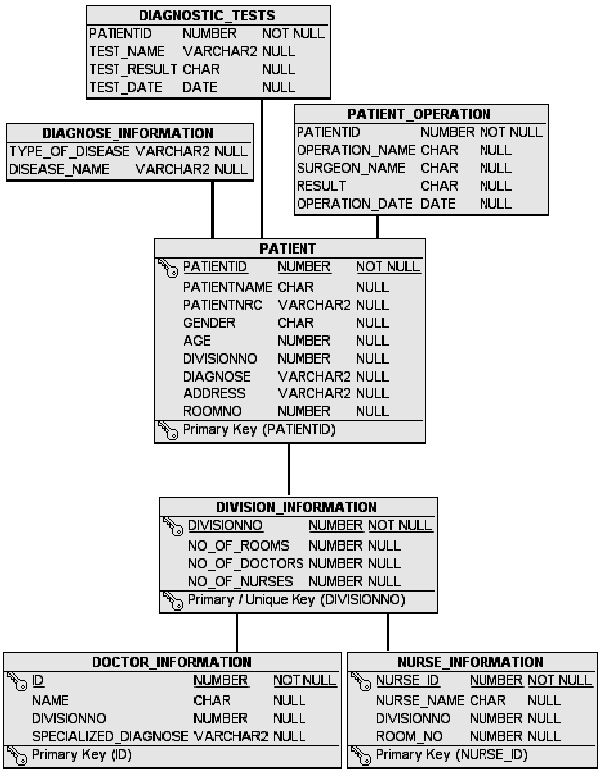

Figure.4.1 Database Design for Oracle Database

4.2 Database Design for MySQL Database

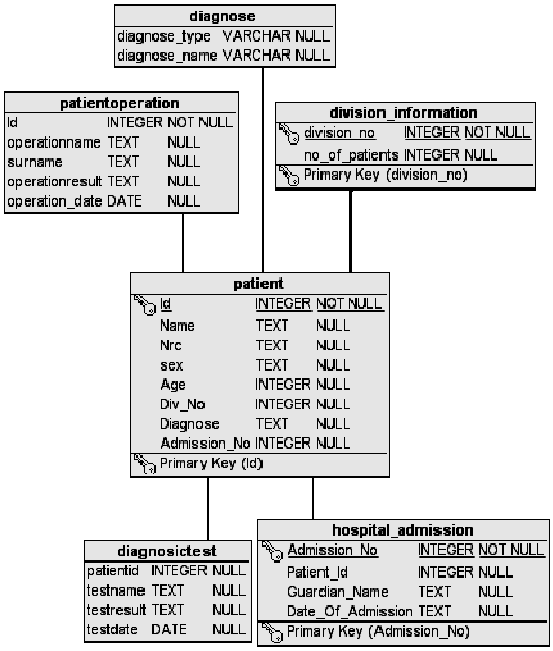

Figure.4.2 Database Design for MySQL Database

4.3 Sample Export Schema 1

Table: 4.1 ES1

Data base Schema	Oracle	Oracle	Oracle
Table Schema	PATIENT	PATIENT	PATIENT_ OPERATIO N
Column Name	PATIENTID	PATIENTNAME	RESULT
Column Type	number	Varchar2	char
Column Length	22	20	10
Is Nullable	no	yes	yes

Table.4.1 shows sample export schema for oracle database. It contains information of attributes from Oracle database. In this sample table, some attributes in Oracle database are showed as sample.

4.4 Sample Export Schema 2

Table: 4.2 ES2

Data base Schema	MySQL	MySQL	MySQL	MySQL
Table Schema	patient	patient	patient operation	patient operation
Column Name	Id	Name	surname	operation_ name
Column Type	int	text	text	text
Column Length	6	20	20	40
Is Nullable	no	yes	yes	yes

Table: 4.2 shows sample export schema for MySQL database. It contains information of attributes from MySQL. In this sample table, some attributes in MySQL database are showed as sample.

4.5 Sample Local Object 1

Table: 4.3 LO1

Objects Name	**Patient Operation**
Table name	PATIENT, PATIENT_OPERATION
Available Attributes	PATINETID (*number*) PATIENTNAME (*char*) PATIENTNRC (*varchar2*) GENDER (*char*) AGE (*number*) DIVISIONNO (*number*) DIAGNOSE (*varchar2*) ADDRESS (*varchar2*) ROOMNO (*number*) OPERATION_NAME (*char*) SURGEON_NAME (*date*) RESULT (*char*) OPERATION_DATE (*date*)

Table: 4.3 shows sample local objects for oracle database. It contains available objects for oracle database. One object contains one or more related tables based on export schema. Assume that LO1 table contains "Patient Operation" object because *primary key* (PATIENTID) of PATIENT table is containing in PATIENT_OPERATION table as foreign key. Attributes of PATIENT and PATIENT_OPERATION tables are shown in figure 4.1.

4.6 Sample Local Object 2

Table: 4.4 LO2

Objects Name	**Patient Operation**
Table name	patient, patientoperation
Available Attributes	Id (*integer*) Name (*text*) Nrc (*text*) sex (*text*) Age (*integer*) Div_No (*integer*) Diagnose (*text*) Admission_No (*integer*) operationname (*text*) surname (*text*) operationresult (*text*) date (*date*)

Table: 4.4 shows local objects for MySQL database. It contains available objects for MySQL database. One object contains one or more related tables based on export schema. Assume that LO2 table contains "Patient Operation" object because *primary key* (Id) of **patient** table is containing in **patientoperation** table as *foreign* key. Attributes of patient and patientoperation tables are shown in figure 4.2.

4.7 Sample Object Equivalence

Table: 4.5 TOE

Available Objects	**Patient Operation**
Available Attributes in Oracle	PATINETID (*number*) PATIENTNAME (*char*) PATIENTNRC (*varchar2*) GENDER (*char*) AGE (*number*) DIVISIONNO (*number*) DIAGNOSE (*varchar2*) ADDRESS (*varchar2*) ROOMNO (*number*) OPERATION_NAME (*char*) SURGEON_NAME (*date*) RESULT (*char*) OPERATION_DATE (*date*)
Available Attribute in MySQL	Id (*integer*) Name (*text*) Nrc (*text*) sex (*text*)

	Age (*integer*) Div_No (*integer*) Diagnose (*text*) Admission_No (*integer*) operationname (*text*) surname (*text*) operationresult (*text*) date (*date*)

Table: 4.5 shows equivalence objects by matching objects from oracle and MySQL databases. It contains object names and attributes for this objects. In this sample, "Patient Operation" object is containing in both databases. So, this table contains attributes of "Patient Operation" object for both databases.

4.8 Data Dictionary

Table: 4.6 Synonyms stored in Data Dictionary

Default Word	Synonyms
PATIENT_ID	Patientid, patientid, id, Id, patient_id, PATIENT_ID
PATIENT_NAME	Patientname, patientname, Name, name, patient_name, PATIENT_NAME
PATIENT_NRC	NRC, Nrc, Patient_Nrc, PATIENT_NRC
PATIENT_AGE	Age, age, AGE, PATIENT_AGE
GENDER	Gender, gender, sex, GENDER
DIVISIONNO	Divno, Division_no, div_no, DIVISIONNO
ROOMNO	Room, room_no, Room_No, ROOMNO
PATIENT_DIAGNOSE	Disease_name, Diagnose, name_ of_diagnose, diagnosename
PATIENT_ADDRESS	Address, ADDRESS
ADMISSION_NO	Admission_No
OPERATION_NAME	OPERATION_NAME, operationname
OPERATION_DATE	OPERATION_DATE, date
OPERATION_RESULT	RESULT, operationresult
SURGEON_NAME	surname, SURGEON_NAME

Table: 4.6 shows synonyms stored in data dictionary. Data Dictionary contains attributes of separated databases and default word for each. In Attribute Matching stage, data dictionary is used to match attributes in objects.

4.9 Type Dictionary

Table: 4.7 Type stored in Type Dictionary

Default Type	Types
NUMBER	integer
DATE	date
VARCHAR2	text
NUMBER	number
VARCHAR2	varchar
VARCHAR2	char
VARCHAR2	varchar

Table: 4.7 shows types stored in Type dictionary. Type Dictionary contains types of attributes contained in separated databases and default type for each. In Attribute Matching stage, type dictionary is used to match types of attributes in objects.

4.10 Sample Attribute Equivalence

Table: 4.8 TAE

Attribute Name	Attribute Type
PATIENT_ID	NUMBER
PATIENT_NAME	VARCHAR2
PATIENT_NRC	VARCHAR2
GENDER	VARCHAR2
PATIENT_AGE	NUMBER
DIVISIONNO	NUMBER
PATIENT_DIAGNOSE	VARCHAR2
PATIENT_ADDRESS	VARCHAR2
ROOMNO	NUMBER
OPERATION_NAME	VARCHAR2
OPERATION_DATE	DATE
OPERATION_RESULT	VARCHAR2
SURGEON_NAME	VARCHAR2
ADMISSION_NO	NUMBER

Table: 4.8 shows equivalence attributes by matching attributes for each object in TOE. It contains attribute name, attribute type, and attribute length. System contains one or more TAE. Number of TAE is number of objects. In this sample table, attributes from TOE tables are matched by using ***Data Dictionary*** as shown in table 4.6 and ***Type Dictionary*** as shown in table 4.7.

5. Evaluation of the System

In our system, by creating data objects, foreign key problems are resolved. So user more easily access data from separated databases. When system performed attribute matching stage, it can reduce duplicated data. So, time costing of accessing data from separated databases is more than time costing of accessing data from integrated database. So, this system is more efficient for client.

6. Conclusion

When much information from separated sources is accessed by many organizations, data integration is needed. Because data are received from many places, data may be duplicated. So, data integration is important system for all organization. In this system, data are access by integrating based on schema based integration and data integration is based on export schemas from separated data sources. Foreign key problems are resolved by creating objects with related tables. It is time saving and reduces data duplication within many databases.

7. References

[1] Alapati, “Expert Oracle Database 11g Administration”, the expert's voice in Oracle Apresss, http://books.google.com/?id=tdRes4IdLiIC, Retrieved 2010-07-07.

[2] C.H.Goh, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems”, 1997.

[3] E.F. Codd, "A relational model of data for large shared data banks", In Communications of the ACM archive, Vol 13, Issue 6, pp.377-387, June 1970.

[4] E. Mena, "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies”, 1996.

[5] G.Petri, “A Comparison of Oracle and MySQL”, SELECT Journal, 2005.

[6] G.Z.Kantorski, “Heterogeneous Database Interoperability Using the WWW”, UFRGS, Federal University of Rio Grande do Sul, 2000.

[7] H. Wache, "Ontology-Based Integration of Information A Survey of Existing Approaches" , 2001.

[8] I.Lukovic, “Database Schema Integration Process – A methodology and aspects of its applying”, 2006.

[9] M.Lenzerini,"Data Integration: A Theoretical Perspective", 2002.

[10] R.Schumacher, "Dispelling the Myths". MySQLAB, http://dev.mysql.com/ tech-resources/articles/dispelling-the-myths.html, Retrieved 2007-02-10.

[11]. T.Lonstki, “Database Integration: Criteria and Techniques”, Senior Technical Consultant, UGC Consulting, 6200 S. Syracuse Way, Suite 222, Englewood.

[12] Watkins, "Look inside ASM disk groups with Oracle 10gR2's ASMCMD", 30 January 2007.

[13] W.S.Li, “Semantic Integration in Heterogeneous Databases Using Neural Networks”, Proceedings of the 20th VLDB Conference Santiago, Chile, 1994.

[14] Y. Arens, "Query Processing in sims information mediator", 1996.