# A Framework for Creating Global Schema Using Global Views from Distributed Heterogeneous Relational Databases in Multidatabase System

GJCST Computing Classification
C.2.4, H.2.1 & H.2.8

Ali Mohammad Ghulam
Post Graduate Studies and Research
Indian Institute of Technology Kharagpur, West Bengal 721302, India
ali_iit@yahoo.com , ali@hijli.iitkgp.ernet.in
Murtuza Mohammad Ghulam
mgm_rsc@yahoo.co.uk

***Abstract*- Creating a global view, this is an alternative way of looking at the data in one or more tables from different geographical located heterogeneous local component databases with better security. A view is simply a stored query accessible as a virtual table that can be used similar to a table. In this paper, we propose a framework to integrate heterogeneous relational local component databases using Global View and construct a Global Schema with a set of Global Views in multidatabase system. We have also illustrated the method of integration with an example. Query submitted on Global View will be decomposed into a set of sub queries and will be executed at various remote sites and finally sub-queries will be composed and will get back result to concerned user. We also discuss some issues concerning distributed databases security in the Multidatabase System.**

*Keywords*- Multidatabase, Global Schema, Relational Database, Database View.

## I INTRODUCTION

Keeping in mind the progress in communication and database technologies (concurrency, consistency, and reliability) has increased the data processing potential. Various protocols are proposed and implemented for network reliability, concurrency, atomicity, consistency, recovery and replication. The current demand is to access data from various existing databases distributed among sites in a network. If any organization has headquarter in any country and has many branches across the globe, wants efficient and quick retrieval of information for any kind of decision supports, the proposed framework will meet this requirement in this paper. This paper has addressed this issue nicely by implementing highly coupled federated multidatabase system.

The multidatabase system has gaining attention of many researchers that attempts to logically integrate several different independent distributed DBMSs while allowing the local DBMSs to maintain complete control of their operations. It means all existing databases are autonomous and evolve over times. In multidatabase system, it should be possible to address data in more than one database by a single query. On the other hand, it should be possible for different users to have different interpretation of the same data. Thus, the demand on a multidatabase is to provide an interpretation of data with same or similar meaning which have different representations. A multidatabase system is a database system that resides on top of existing component local database systems and presents a single database illustration to its users [1,2]. The Multidatabase System usually maintains a single global database schema, which is integration of all local component databases schemas and against which users issue queries and updates. Multidatabase System maintains only global schema and the local component database system actually maintains all user data. Creating and maintaining the global schema, which requires the use of database integration techniques, is a critical issue in the multidatabase system.

Variety of approaches to schema integration have been proposed e.g. [3,4,5,6,7,8,9,15]. We propose a framework to integrate multiple local component relational databases using database global view technique and query on global view will be translated to number of queries on physical remote distributed relational databases. However, effort for constructing the global view may be expensive with respect to the frequency of usage.

## II DATABASE ARCHITECTURAL FRAMEWORK

The ANSI/X3/SPARC architecture [10,13,14] as shown in figure 1 is claimed to be based on the data organization in DBMS standardization. It recognizes three views of data:

### A. *Local internal schema/view*

Local (internal) schema/view shows how the data is stored on each site. The format of the internal schema is dependent on the LDBMS of each site.

### B. *A global conceptual schema/view*

A global (conceptual) schema describes the data throughout a network and shows what data is at what site. The global schema usually stored in a global directory.

### *C. A User external schema/view*

A user (external) schema/view shows how user will view and manipulate the data.

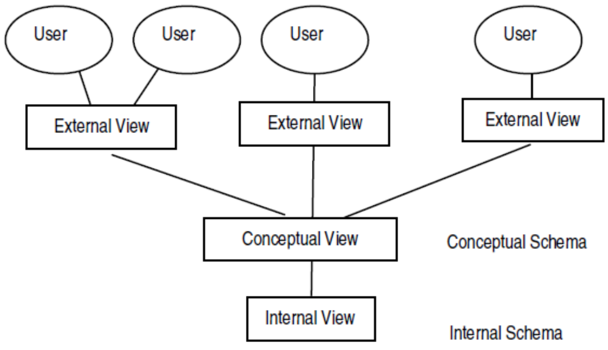


**Figure 1. ANSI/SPARC Architecture**

## III PROPOSED FRAMEWORK

In our proposed framework, the Multidatabase system will control multiple gateways and will access to remote databases through these gateways. The proposed multidatabase system manages and retrieves data from multiple sites within a single graphical user interface based application or web-based application and resides on top of the Global Schema while providing complete autonomy to individual remote database systems.

The proposed framework is divided into four layers (as shown in Figure 2) based on a classical example (as shown in Figure 1) of a data-based architecture is the ANSI/SPARC model by Tsichritzis and Klug [10,13,14]. See the details work of the Multidatabase System as 4-tiered client-server model in the distributed databases [16].

***Layer 1-*** Users are sitting on layer 1 that is top of the Multidatabase System and will submit queries and updates though a single multidatabase system application of the Global Multidatabase System to remote database servers. A single query can retrieve information from many remote sites table and can update many remote sites table through global views.

***Layer 2-*** Layer 2 is an Application Server where a web based application or client-server application resides. If the application is a web-based then user will access application through specific URL and will submit queries and updates. If the application is a client-server, then the application will be launched through citrix metaframe under thin-client platform. Layer 2 contains business logic, API and access to Multidatabase system server.

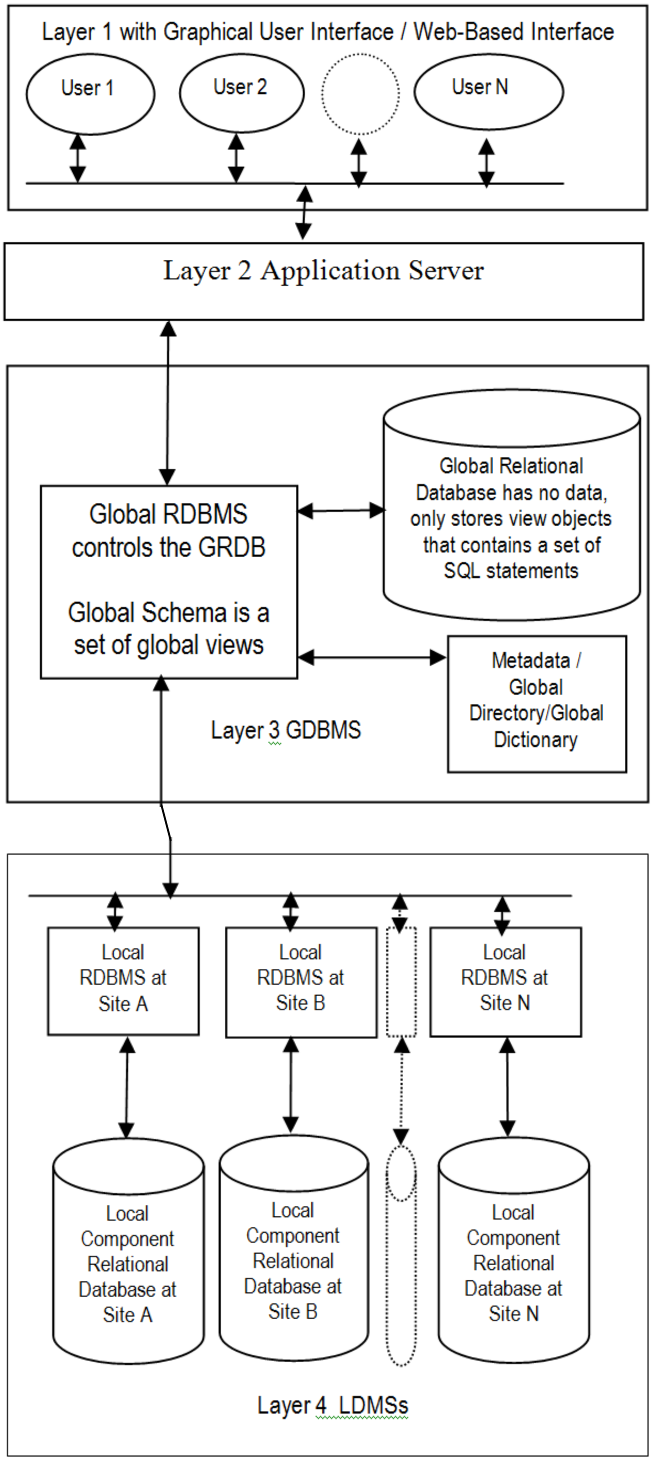


**Figure 2 Proposed Framework**

***Layer 3-*** In layer 3, there is a Global Database Management System (GDBMS) where Global Schema is created with a set of global views and is stored in a Global Database (GDB). User will submit a query to Global Schema specific to any global view with the help of GUI or web-based interface, the GDBMS will scan, parse and validate query. During this process, the system will also use Global Directory/Global Dictionary/Metadata. The same query will be decomposed into a set of sub-queries and will go to respective remote local component relational database servers. All sub-queries will be executed locally and at that time sub-queries will use local Directory/Dictionary/Metadata information of the local database servers and all sub-queries will produce responses. In case of information retrieval, sub-results coming from individual local schemas are then to be composed and send back result to respective user.
Similarly, through view user can update remote sites with applying SQL query on global view. However, there will be some restrictions in updating remote site databases through view. We are not taking into consideration this issue in this paper. The global view in the Global Relational database will continuously grow based on the requirements of the organization and expansion of the remote databases over time.
In our proposed framework, when global views are created within the multidatabase system, at the same time metadata from the remote sites is stored within the multidatabase system tables as we named as Global Directory/Global Dictionary. This metadata can be queried locally to quickly obtain information about views. This information will include column attributes, index definitions and what data objects exist at which site(s) in the multidatabase system. Now global and local query optimization is another issue in the multidatabase system. We are not taking into consideration this issue in this paper.
***Layer 4-*** in layer 4 all remote distributed local relational database servers are running and also maintaining metadata information of local server in local dictionary/local directory.

## IV GLOBAL SERVER INTERFACE

We first define remote servers interface definition at Global Database Management System (GDMS) through which Global database server will be able to interact with the remote servers. The way of server interface definition is as follows:

***Database Server Interface Driver Definition***
Global Server Name
Global Server IP Address
Global Server Port
Communication Protocol

Example

Global Server Name: NewYork
Global Server IP Address: 144.16.192.211
Global Server Port: 5000
Communication Protocol: TCP

## V REMOTE DATABASE GATEWAY

Database Gateways address the needs of data access in a distributed environment. Gateways make it possible to integrate with any number of remote database servers. Global view can only be created on the basis of using SQL gateway otherwise system will not understand where to go during creating Global View. The way of Remote Database Gateway definition is as follows:
Global Server Name.Database Name.Database Owner Name.Object Name
Where object name is a name of relational table created at remote server by dbo.
Example
NewYork.StateDB.dbo.NameOfState

## VI INTEGRATION METHODOLOGY AND CREATION OF GLOBAL VIEW IN MULTIDATABASE

Now we show how to create a global view using N numbers of remote database server's relational tables.

```
create view View1

 (CountryCode, StateCode, StateName)

as

select CountryCode, StateCode, StateName from
NewYork.StateDB.dbo.NameOfState

union

select CountryCode, StateCode, StateName from
London.StateDB.dbo.NameOfState

union
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
union

select CountryCode, StateCode, StateName from
nCountry.StateDB.dbo.NameOfState
```

In the above example View1 is a name of Global view created in Global Schema. Attributes of the global view are CountryCode, StateCode and StateName. Attributes of remote relational tables are similarly CountryCode, StateCode and StateName and are mapped and that we are going to use in creating view. There may be more attributes in the remote relational tables. We may use N number of Remote gateway such as

NewYork,StateDB.dbo.NameOfState,

London,StateDB.dbo.NameOfState,

. . . . . . . . . .. . . . . . . . . . . . . . .,

nCountry,StateDB.dbo.NameOfState

Where NewYork is the name of remote local database server located in New York, USA. StateDB is the name of database server which is created in the remote server, dbo is an owner of object who created object and NameOfState is a name of relational table created in the StateDB of the Remote Server NewYork. Similarly for London based remote server and for n numbers of countries.

## VII DATABASE MODIFICATIONS PROPAGATION

If DBA at remote server changes the structure of the attributes such as type of the attribute, size of the attribute of the relation table will be automatically reflected in the Global View. Some user-defined functions may also be proposed to resolve schema conflicts. For example, attribute of one site‘s table is varchar and in other site it is char, then the user-defined function at the time of integration will solve this problem. I am not taking into consideration this issue in this paper. Many researchers have addressed these issues. Please see detailed work of [1,8,12,15].

## VIII SUBMIT QUERY ON GLOBAL VIEW

If a user is submitting following query through their client application on global view of the multidatabase system:

Select * from View1

This will give a list of all states across n numbers of country using a single query. The federated query will be executed as follows:

Select CountryCode, StateCode, StateName from
NewYork,StateDB.dbo.NameOfState
Union

Select CountryCode, StateCode, StateName from
London,StateDB.dbo.NameOfState
Union

Select CountryCode, StateCode, StateName from
Delhi,StateDB.dbo.NameOfState
Union
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. .
Select CountryCode, StateCode, StateName from
nCountry,StateDB.dbo.NameOfState

Similarly, we can modify data of all remote sites using a single SQL query on global view.

## IX DATABASE SECURITY

The increased usage of databases to store large amounts of data has created new security problems. Typically a database contains data of various degrees of importance and levels of sensitivity. This data is shared among a wide variety of users with different responsibilities and privileges. It is therefore necessary to restrict users of the database to those portions of the total data that are necessary for their activities. Additionally, more control is needed over changes a user can make to data because of the many ways these changes can affect other users of the database [11]. The Network security expert at each remote site can better protect remote site database server by implementing a firewall between the Global Database Server and the local database server and will examine each incoming packets coming to local database server, will authenticate this and will decide whether this packet is to be denied, dropped or forwarded to local database server. Since the IP address, port and the type of network service that the Global Database Management System is using in communicating with the remote database server is known by the firewall policy rules, can easily forward, drop or deny incoming packets. The DBA at each local site will provide a better database server level and database object level security. The System Administrator at each local site will provide a better OS level security. How to exactly tackle all these issues, we are not explaining in details in this paper.

## X CONCLUSION

The main objective of the work is to provide transparent access and manage distributed, heterogeneous, autonomous, and relational databases. This is a viable proposed framework in the integration of distributed, heterogeneous, autonomous, and relational databases in the Multidatabase System and easy to maintain the Global Schema. In future, we plan to address other issues in the multidatabase system.

## XI REFERENCES


1) M. W. Bright, A. R Hurson, and S. H Pakzad, A Taxonomy and Current Issues in Multidatabase Systems. IEEE Computer, Vol. 25, No. 3, pp. 50-60, March 1992.
2) Pitoura, O. Bukhres, and A. Elmagarmid, Object-Oriented in Multidatabase Systems. ACM Computing Surveys, Vol. 27, No. 2, pp. 141-195, June 1995.
3) Czejdo, and M. Taylor, Integration of Database System Using an Object-Oriented Approach. Proceedings of 1st International Workshop on Interoperability in Multidatabase Systems, April 1991, pp. 30-37.
4) W. Gotthard, P. C. Lockemann, and A. Neufeld, System-Guided View-Integration for Object-Oriented Databases. IEEE Transactions on Knowledge and Data Engineering, Vol 4, No. 1, pp. 1-22, February 1992.
5) M. Kaul, K. Drosten, and E. J. Neuhold, ViewSystem: Integrating Heterogeneous

Information Bases by Object-Oriented Views. Proceedings of 6th International Conference on Data Engineering, 1990, pp. 2-10.
6) L. Koh, and A. L. P. Chen, Integration of Heterogeneous Object Schemas. in Entity-Relationship Approach, Springer-Verlag, 1994, pp. 297-314.
7) Chao Ching-Ming, Schema Integration between Object- Oriented Databases. Tomkang Journal of Science and Engineering, 2001.
8) Rehab M. Duwairi, A framework for Generating and Maintaining Global Schemes in Hetrogeneous Multidatabases Systems. *IEEE*, 2003.
9) Soon M. Chung, and Pyeong S. Mah, Schema Integration for Multidatabase Using the Unified Relational and Object-Oriented Model. ACM, 1995.
10) Umar, Distributed Database Management Systems Issues and Approach. The University of Michigan, Technical Report, July 1988.
11) Bertino, and L. M. Haas, Views and Security in Distributed Database Management Systems.
12) R. Motz, and P. Fankhauser, Propagation of semantic modifications to an integrated schema.
13) Tsichritzis, A. Klug, The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. Information Systems, 1:173- 191, 1978.
14) Tamer Ozsu, Patrick Valduriez, Principles of Distributed Database Systems.
15) M. G. Ali, Object-Oriented Approach for Integration of Heterogeneous Databases in a Multidatabase System and Local Schemas Modifications Propagation, IJCSIS, USA, 2009.
16) M. G. Ali, Multidatabase System as 4-Tiered Client- Server Distributed Heterogeneous Database System, IJCSIS, USA, 2009.