Adopting a Proxy Database to Prevent Direct Access to Distributed Transaction Databases Ensuring Information Security

Mohammad Ghulam Ali [a*]

[a] Indian Institute of Technology, Kharagpur, West Bengal, India.

Author's contribution

The sole author designed, analyzed, interpreted and prepared the manuscript.

Original Research Article

Abstract

In this scientific and technical practical-oriented research paper, it is proposed a concept of Proxy Database to prevent direct access to distributed transaction databases and also ensuring the data and information security. The same approach has already been implemented in our one live project. A Proxy Database is created using a set of proxy tables and views from all relational distributed databases running for all different domains for any organization/institution. Creating a proxy table, this is an alternative way of looking at the data in one or more tables and this is also an alternate way of looking at the data through one or more views from relational distributed databases. Proxy table provides indirect access to the data in the distributed table(s) of the distributed databases. Location transparency of the local data at the local distributed databases will definitely be maintained by creating a proxy table. A proxy table will be mapped with the local distributed table and tables in the form of view. A schema for all proxy tables created, that resides in the Proxy Database, but the data resides in each local distributed databases. It is proposed here a practical framework to construct a Proxy Database with a set of proxy tables created from all local relational distributed databases which are being used for all different domains. Two or more proxy tables may further be merged to get results from two or more local distributed tables located in the respective distributed databases. Illustrated some methods of creating a Proxy Database with some examples. Illustrated about the query submitted on the proxy table without join, union or intersect and this is straightway go to the respective local distributed table(s) or tables in the form of view and sending back results to the concerned client user. Illustrated about the query submitted using join, union or intersect on two or more proxy tables, query decomposition with a set of sub

_____

Corresponding author: Email: ali@hijli.iitkgp.ac.in;

queries going to the respective local distributed tables or tables in the form of view, execution of each individual sub-query at each local distributed database server and finally about the query composition of all sub-queries and sending back results to the concerned client user. Discussed well about the network security proposing a viable solution by implementing a software firewall. Discussed about the local distributed databases structure modifications and propagation to the Proxy Database so that all the local distributed databases and the Proxy Database remain consistent.

1 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 the network reliability, concurrency, atomicity, consistency, recovery and replication. The current demand is now to access data from various existing distributed databases of all the different domains over the network in a very secured manner and also the system if proposed is to be considered more scalable, more resilient. It must also provide 'intelligent' load balancing. All existing distributed databases are autonomous and evolve over times. If any organization/institution maintains distributed database for all different domains in one or more database servers, wants efficient and quick retrieval of information for any kind of decision supports. A proposed practical framework is meeting this very requirement through this scientific and technical research paper with a better security by adopting a Proxy Database [1] as it is shown in Fig. 1 below. A proposed framework is concentrated on the web-based information system to retrieve data from all different local distribute databases for all different domains through a Proxy Database. For the security and consistency reasons, a web- based interface will not have database updates feature and also web-based clients will not directly access to local distributed transaction databases. Each local distributed transaction databases will be maintained locally using their own DBMS software either may be a Client-Server DBMS or a Web-based System. In this paper, the said issue is addressed very nicely by implementing a Proxy Database on set of local distributed databases of different domains. The section 1 itself Illustrating Information Security Framework with a Proxy database including creating a proxy tables, mapping between the proxy database and each local distributed databases and about the different aspects of the query processing, section 2 discussed about the basic of the database architecture framework, section 3 discussed in detail about the prosposed scientific, technical and practical oriented framework, section 4 discussed on the query processing including query parsing, execution, decomposition and composition, section 5 discussed on the novel invention and it is also copyrighted pertaining to maintenance of the distributed databases environment to be remained consisetent and coherent, section 6 discussed on the database, network and information security, and finally section 7, 8 and 9 are pertaining to acknowledgement, declaration of interest, conclusion including results and discussion.
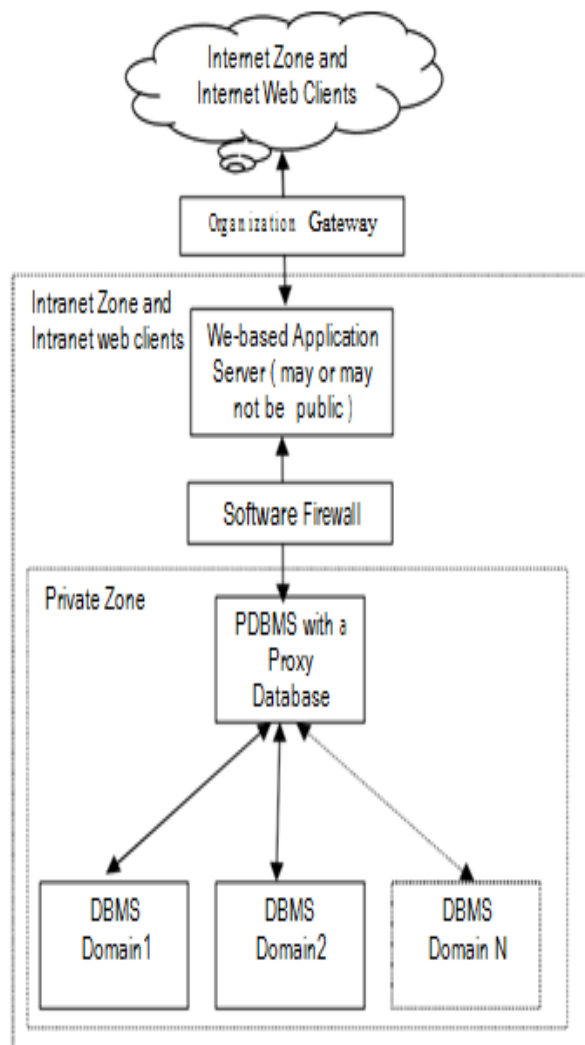
In the Fig. 1 below, it also includes a software firewall between a Web-based Application Server which may or may not be public that depends upon the policy of the Institution/Organization and a Proxy Database Server which is in the private and secured network to make local Distributed Database Servers which are also in the private network, more secured from the web clients and from unauthorized users. A Proxy Database Server and all Distributed Local Database Servers are kept behind the firewall and in the private and secured network. The Application Server either may or may not be kept on the public network and but always kept behind the gateway of the Institution/Organization. Implementing a firewall, will ensures second level data and information security as the first level data and inormation security is enfored by implementing a Proxy Database.

A Proxy Database based information system has gaining attention of many researchers that attempts to logically integrate several different independent distributed relational databases and to import schemas from all distributed relational databases while allowing the local DBMSs to maintain complete control of their operations (autonomos). It means all existing distributed databases are autonomous and evolve over times. In a Proxy Database based system there is a possibility to address data at more than one distributed databases by issuing a single query.

A Proxy Database is a database that resides on top of the existing local distributed databases and presents a single database illustration to its users. A Proxy Database usually an importation of structure of selected distributed tables and tables in the fom of views from distributed local databases and which is required in a proposed web-based information system for any kind of decision supports and against which users will issue queries and updates (updates based on the policy decision).

It is proposed here a practical scientific and technical framework to create a Proxy Database using multiple local distributed databases using a proxy table technique [1-3].

Proxy tables are key to location transparency. A proxy table is a local table in the Proxy Database containing metadata / directory / dictionary / catalog that points to local table(s) or tables in the form of view of the local relational distributed databases.
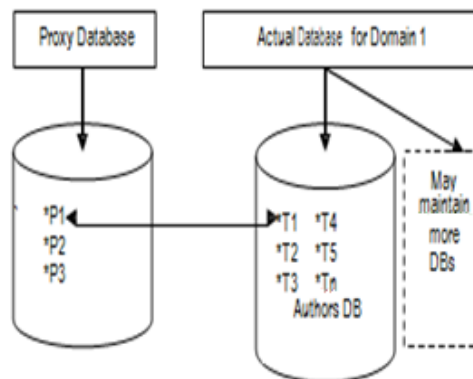

Fig. 1. Information security framework with a proxy database

When any client issue a SQL query through a web-based information system on a Proxy Database involving proxy tables using join i.e. where clause or union or intersect, a Proxy Database Management System (PDBMS) will open connections to all local distributed database servers and will pass part of client query (sub- queries) involved to all local tables pointing to local distributed database servers, where sub-queries will be executed and all results will be sent back to the Proxy Database System. The results will be stored in the Proxy Database Server Data Cache or Working Storage area and finally user will see the appropriate results.

Any updates to a local proxy table, the update command will then be sent to the local distributed database server and the table(s) will be updated there accordingly. However, a proposed web-based information system will not have this feature as it is also mentioned above.

Illustrated some examples of proxy tables and their mappings local table or tables in the form of view to the local distributed database. The same problems and solutions are worked-out in the similar way in the Multidatabase System [3].

Example 1: One to one mapping between a proxy table of the Proxy Database and a table of the local distributed database as it is shown in Fig. 2 below.
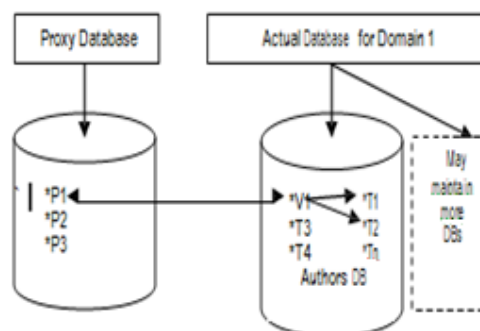

Fig. 2. One to one mapping between a proxy table and the local distributed database table

Creating a proxy table and establishing mapping between a local proxy table and a local distributed database table.

Create proxy_table P1 external table At "Domain1.Authors.dbo.T1";

Example 2: One to one mapping between a proxy table of the Proxy Database and a view of the local distributed database where view is created by using two tables as it is shown in Fig. 3 below.
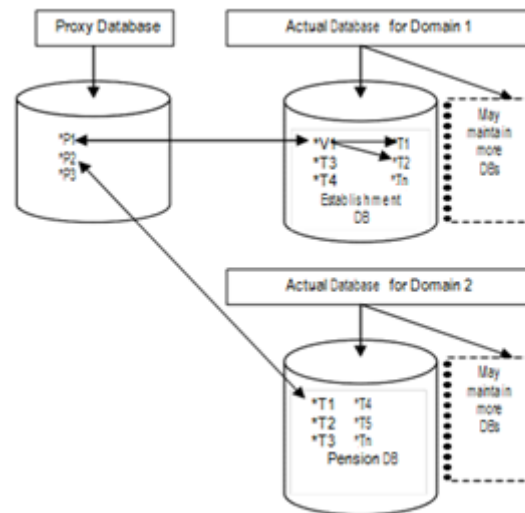

Fig. 3. One to one mapping between a proxy table of a proxy database and a view of the local distributed database

Creating a proxy table and establishing mapping between the proxy table and view, where a view is created from two local tables of the same distributed database.

Create proxy_table P1 external table At "Domain1.Authors.dbo.V1";

Similarly, proxy tables can be created from all distribuuted database pertaining to different domains to form a proxy database as per the requirements for a proposed web-based information system to retrieve information using a single query.

Example 3: Query execution using two proxy tables where

Fig. 4. Query execution using two proxy tables pointing to tow distributed local database servers

1st proxy table is directly mapped with the local view of the distributed database server located at Domain 1 where view is again created by using two tables. 2nd proxy table is directly mapped with the local table of a distributed database server located at Domain 2 as it shown in Fig. 4 above.

Lets us we want a list of persons of the organization/institution with Person Id and their's Name from two different locally located distributed database servers. The following query will get back the apporpriate results:
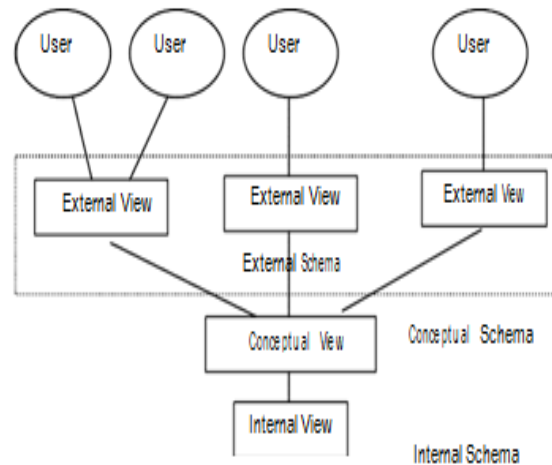
select PersonId, PersonName from P1 union select PersonId, PersonName from P2;

where a proxy table P1 points to a view V1 at the local distibuted database at Domain 1 in the Establishment Database and a view V1 is created from two tables T1 (Regular Employees) and T2 (Adhoc Employees) in the same database server. A proxy table P2 points to a table T1 at Domain 2 local distributed database server in the Pension Database.

Now, it is proposed here a viable framework in this paper for creating of a Proxy Database. In the proposed practical framework, there is a Proxy Database Management System (PDBMS) and a web-based information system accessing a Proxy Database where a Proxy Database is created with a set of proxy tables. User will submit a query on a Proxy Database specific to any proxy table, the PDBMS will scan, parse and validate query. During this process, the system will also use Directory / Dictionary / Metadata / Catalog of the Proxy Database. The same query will then straightway go to the respective local distributed table(s) or view created frrom table(s), the same query will then be executed locally and at that time a query will use Directory / Dictionary / Metadata / Catalog information of the local distributed database, and finally query will produce response and send back results to the concerned user. If user submits a join, union or intersect query on the Proxy Database using two or more proxy tables, a query will then be decomposed into a set of sub-queries and will go to the respective local distributed database servers and each sub-query will then produce a response. The sub-results coming from individual local distributed database server will then be composed and send back results to the respective user. Similarly, through a proxy table, user can update local table with applying SQL query. In this paper, not given more emphasis on updates as local distributed databases are being maintained by their own DBMS software as it is also mentioned above. Number of proxy table will continuously grow based on the requirements of the organization and expansion of the local databases with time.

2 Database Architecture Framework

It is also necessary to discuss over heare the data organization in the DBMS standarization. The ANSI/X3/SPARC architecture [4-6] as shown in Fig. 5 is claimed to be based on the data organization in DBMS standardization. It recognizes three views of data:

Fig. 5. ANSI/SPARC Architecture

2.1 Local internal schema/view

Local (internal) schema/view shows how the data is stored in all distributed databases of different domains. The format of the internal schema is dependent on the DBMS of each domain.

2.2 A conceptual schema/view

A conceptual schema describes the data throughout a network and shows what data is at what domain. The conceptual schema usually stored in a directory / dictionary / catalog / metadata.

2.3 A User external schema/view

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

3 A Detailed Scientific, Technical and Practical-Oriented Framework and Method Proposed

In the practical proposed framework, a Proxy Database Management System (PDBMS) will control multiple gateways and will access to local distributed databases through these gateways. The proposed PDBMS manages and retrieves data from multiple distributed databases throgh a web-based application and that resides on top of the Proxy Database system while providing complete autonomy to individual local distributed database system.

The proposed framework is divided into four layers (as shown in Fig. **6** below) based on a classical example (as shown in Fig. 5 above) of a data-based architecture is the ANSI/SPARC model by Tsichritzis and Klug [4-6]. Please also see the details of work pertaining to the Multidatabase System as 4-tiered client-server model in the distributed databases [7].
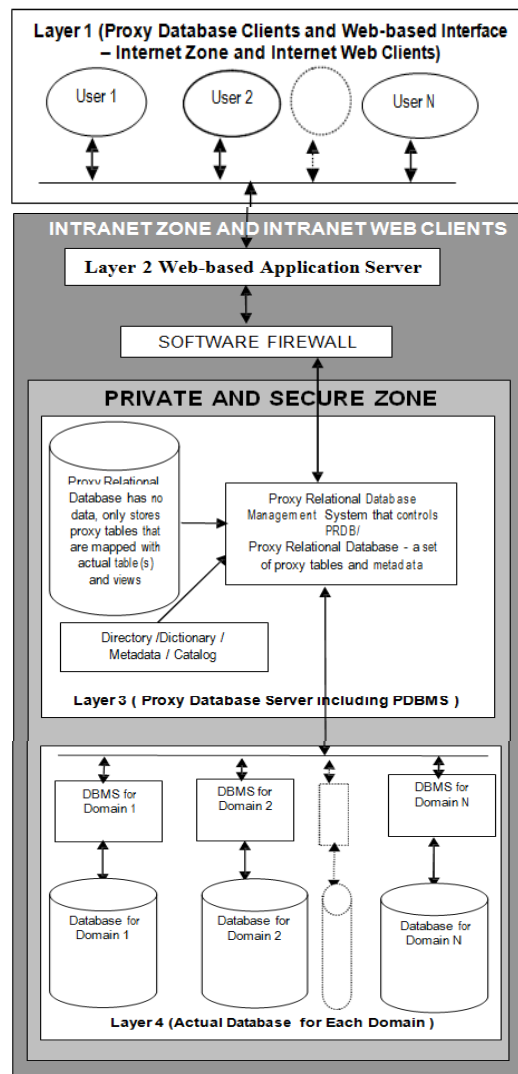
In the proposed framework, when proxy tables are created within the Proxy Database system, metadata from local distributed databases are stored within the Proxy Database system tables as we named as Directory / Dictionary / Catalog. This metadata can be queried locally to quickly obtain information about proxy tables and the corresponding distributed tables and viwes. This information will include column attributes, index definitions and what data objects exist at the distribted database servers in the distributed database system.

The maintenance of this metadata now becomes a critical issue, as the local distributed schemas evolve over times and are autonomus, there should be a mechanism to import and synchronize metadata for a proxy table and the proxy database. Please see the details of work of [8-10] pertaining to the multidatabase system. The

critical issue is worked-out and it is also copyrighted. Please see the details of work [11]. Similarly, it can be implemented for a proxy database here.

A Proxy table can also point to a view on the local database, and it needs not come from a single table.

Through this framework, any user sitting on a Proxy Database system can access single or many local distibuted database tables located at different distributed database servers using a single query. A single query can retrieve information from many local distributed database servers table using more than one proxy tables. A single query can also update local distribted database server's table through a proxy table and it is not considered in this paper and the reason as stated earlier. Any user can retrieve information from two or more local distributed database servers' tables applying join, union, intersect and in various ways of different SQL statements query on proxy tables. Any user can also transfer data from one local database servers' table to another local database servers' table through a proxy table.

Fig. 6. A detailed proposed framework

Following technical steps are involved in a creation of a Proxy Database using proxy tables from distributed local databases in a Proxy Database system.

3.1 Local database server interface

It is required to first define local distributed database server interface definition at the Proxy Database Management System (PDBMS) through which a Proxy Database server will be able to interact with all local distributed database servers over the network within the premises. Following parameters needed for the distributed database server interface definition which is as bellows:

Database Server Interface Driver Definition

Syntax:

Local Server Name
Local Server IP Address
Local Server Port
Communication Protocol

Example:

Local Server Name: Domain1
Local Server IP Address: 144.16.192.211
Local Server Port: 5000
Communication Protocol: TCP

3.2 Distributed local database gateway

Database Gateways address the needs of data access in a distributed environment. Gateways make it possible to integrate any number of database servers. Proxy table only can be created on the basis of using SQL gateway otherwise system will not understand where to go during creation of Proxy table. In the following way, the Database Gateway is defined for the distributed local Database:

Syntax:

Local Server Name.Database Name.Database Owner Name.Object Name

Where object's name is a name of table or view created at the distributed local database server by dbo.

Example:

Domain1.HumanResourceB.dbo.Employees

3.3 Method for mapping distributed local objects to a proxy table and creating a proxy database

Since local tables and views are located at the distributed local database servers, it is needed to provide access methods over a network, resolve pathname i.e. distributed local gateway, and syntax for this, which will be transparent to the Proxy Database system users. It is already discussed in 3.1 and 3.2 above, how to define distributed local server's interface and distributed local database gateway to resolve pathname of the distributed local database server.

As it is discussed earlier that a Proxy Database is a set of proxy tables. Now, illustrating here, how to create a proxy table in a Proxy Database System and how to map proxy table to the distributed local objects i.e. table(s) of view.

create proxy_table <table_name> [external table] at "LocalDatabaseGateway";

This will create proxy table in the Proxy Database system and this will derive tables' structure based on metadata it obtains from distributed local database server. The command **create proxy_table is used** only if the

table does exist at the distributed local database server, external **table** parameter defines that object is distributed local table(s) or view.

Example 1:

Create proxy_table EmployeesDomain1 external table at "Domain1.HumanResource.dbo.Employees";

Here, a proxy table EmployeeDomain1 is created from local distributed table(s) or view which is/are already located at the distributed local Database Server for domain 1 in the Human Resource Database with the name Employees.

Example 2: Alternate ways
create existing table EmployeesDomain1 (EmployeeCode Int not null, Name varchar(35) not null, phone char(15) not null, address varchar(60) null, city varchar(25) null, state char(2) null, country char (3) null, zip char (7)) at "Domain1.HumanResource.dbo.Employees", "table";

A command create existing table can only be used if the table does exist at the distributed local database server. The same data definition language (DDL) is used for the distributed local object.

Example 3: Alternate ways

create table EmployeesDomain1 (EmployeeCode Int not null, Name varchar(35) not null, phone char(15) not null, address varchar(60) null, city varchar(25) null, state char(2) null, country char (3) null, zip char (7)) at "Domain1.HumanResource.dbo.Employees", "table";

Create table command creates the table in the distributed local database server and then creates the proxy table in the Proxy Database system.

Similarly, many proxy tables can be created based on different located distributed local database servers and in resulting of creating a Proxy Database in the Proxy Database Management System.

4 Submiting A Query On Proxy Table of The Proxy Database

We can apply **select**, **insert**, **delete** and **update** SQL statements on the proxy tables.

*Select * from EmployeesDomain1*

This will give a list of all employees from Domain 1 distributed database server to the PDBMS.

Let us suppose, we want a list of all persons of the same organization/institution, then we submit a query like

Select PersonID, Name from EmployeesDomain1union

Select PersonID, Name from PensionDomain2 union

Select PersonID, Name from AdhocDomain3

Similarly, a join query on the proxy table applied in the following way:

*Select * from ProxyTable1, ProxyTable2*
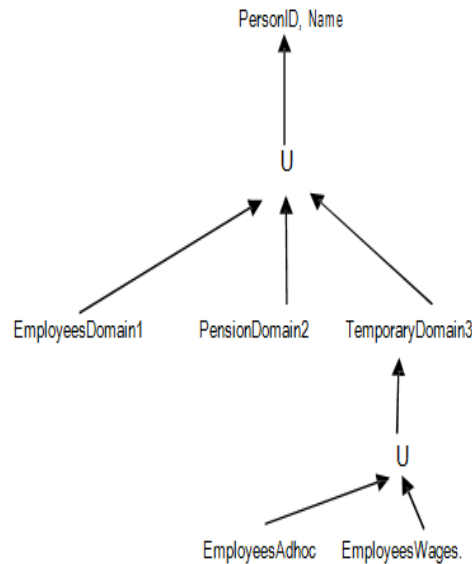
Where

ProxyTable1.Column1 = ProxyTbale2.Column1

The results can be further ordered by using an order by statement.

Similarly, data can be modified under the distributed local database using a single SQL query on the proxy table, the update command will then be sent to the distributed local database servers and table(s) to update the data there. The framework proposed in this paper will not have this feature.

An operator graph is illustrated here including query decomposition and data localization.

Fig. 7. Operator graph

Case: Suppose, if using more than one proxy tables and among these on proxy table is created and mapped with the view of the distributed local database server and the view at distributed local database server is again created with more than one tables at the same distributed local database server then how query will process and how we will get back the results to the Proxy Database system.

Proxy Table 1 – EmployeesDomain1 mapped with Employees table at Domain 1 distributed local database server.
Proxy table 2 – PensionDomain2 is mapped with Pension table at Domain 2 distributed local database server.

Proxy table 3 – TemporaryDomain3 is mapped with Temporary Employees view at Domain 3 distributed local database server where Temporary Employees view is created and mapped with two different tables locally Employees Adhoc and Employees Wages.

When submitting a query as

Select PersonID, Name from EmployeesDomain1 Union

Select PersonID, Name from PensionDomain2 Union

Select PersonID, Name from TemporaryDomain3;

A Proxy database based information system will obtain results as shown in the operator graph below from all distributed local database servers. Many algorithms are proposed and implemented in distributed query processing and query optimization and it is not taken into consideration this issue in this paper.

The operator graph for the above query will look like as it is shown above in Fig. 7.

5 Local Distributed Database Structure Modifications Propagation [Resolved the Critical Issue]

If any DBA, at the distributed and local database server, changes the structure of the existing table such as type of the attributes, size of the attributes, this will automatically be reflected in the Proxy Database in case of no join. But in case of join, how to deal with it, if adding new attributes, changing the type and size of the attributes, the same changes should automatically be reflected in the Proxy Database without manual intervention. Many researchers have addressed this issue in the multidatabase systems at the certain extent [8-10]. The highly demanded issue has already been worked-out and also copyrighted. Please see the details of work [11]. Similarly, the above can be implemented in the Proxy Database. This is not discussed in details here.

6 Network and Data 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 [12].

A Network Security expert can better protect distributed local Database Servers by implementing a software firewall between a Web-based Application Server which may or may not be public and the Proxy Database Server which is in the private network and will examine each incoming packet coming to the Proxy Database Server from authorized web clients or from unauthorized users, will authenticate all incoming packets and will decide whether packets are to be denied, dropped or forwarded to the Proxy Database Server. Since the IP address, port and the type of the network service that the Web-based Application Server is using in communicating with the Proxy Database Server is known by the firewall policy rules, can easily forward, drop or deny every incoming packets. For unauthorized users, the intrusion detection and protection depends on the standard of the firewall policy rules. All local distributed Database Servers are also running in the private network and are also behind the same firewall and the same firewall will examine all incoming packets similarly particular to all local Database Servers. A server, where a software firewall is implemented, will have two network interface cards where 1st network card will be connected to the internet and will have public IP and the second network card will be connected to the private and secured network and will have private IP where a Proxy Database Server and all local distributed Database Servers are running. A proxy database server and all local distributed database servers will use private IPs and the Web-based application server will use either the public or private IP based on the policy. In the intranet zone it is not necessary that all computer machines' IP of the organization/institution are public i.e. accessible from internet zone. Also the DBA at Proxy Database Server and at each distributed local database server will provide a better database server and database object levels security. The System Administrator too at the Proxy Database Server and at each distributed local database server will provide a better OS level security. How to exactly tackle all these issues, this is not taken into consideration here.

7 Conclusion Including Results and Discussion

The main objective of writing this paper is to provide transparent access to autonomous and distributed local relational databases of any institution / organization adopting a proxy database to ensure data and information security. Users are accessing to a Proxy Database considering as a virtual database and are far away from local distributed databases over the network of the institute / organization premises. This is a very viable practical approach proposed in creation of the Proxy Database System and also in implementating a web-based information system in a more secured manner. The proposed approach has already been tested in a live project. It is also taken into consideration to discuss in brief about the second level data and information security by implementing a software firewall policy rules (Linux Iptables Packet Filtering Software Firewall Including IP Forwarding / IP Routing). The Network and Database Security have already been discussed. It also considered about the local schema structure modifications propagation so that the Proxy database and all local distributed

databases remained consistent and coherent. In future, the author is planning to address other issues involved in the development of the secured Information System in the distributed databases environment.

Disclaimer

This paper is an extended version of a preprint document of the same author.
The preprint document is available in this link: https://assets.researchsquare.com/files/rs-1382229/v1/20762422-5acd-47d8-a4c8-7546f17f5692.pdf?c=1645547068
[As per journal policy, pre-print article can be published as a journal article, provided it is not published in any other journal]

Acknowledgement

Competing Interests

Author has declared that no competing interests exist.

References

[1]   Component Integration Services Users Guide, Adaptive Server Enterprise 15.0.2.

[2]   Olson S. Distributed query processing using adaptive server enterprise and omni connect 11.9.2. Sybase, Inc; 1998.

[3]   Ali MG. A framework for creating Global Schema using proxy tables from distributed heterogeneous relational databases in Multidatabase system. IJCEE, Singapore; 2010.

[4]   Tsichritzis D, Klug A. The ANSI/X3/SPARC DBMS framework report of the study group on database management systems. Information Systems. 1978;1:173-191.

[5]   Tamer Ozsu, Patrick Valduriez. Principles of distributed database systems. Springer; 2010.

[6]   A. Umar. Distributed database management systems issues and approach. The University of Michigan, Technical Report; 1988.

[7]   Ali MG. Multidatabase System as 4-Tiered Client-Server Distributed Heterogeneous Database System". IJCSIS, USA; 2009.

[8]   Bright MW, Hurson AR, Pakzad SH. A taxonomy and current issues in multidatabase systems. IEEE Computer. 1992;25(3):50-60.

[9]   Rehab Duwairi M. A framework for generating and maintaining global schemes in hetrogeneous multidatabases systems. IEEE; 2003.

[10]   Motz R. Fankhauser P. Propagation of semantic modifications to an integrated schema. 3rd IFCIS International Conference on Cooperative Information Systems (Cat. No.98EX122). IEEE; 1998.
        DOI: 10.1109/COOPIS.1998.706278.

[11] Ali MG. Method and System for Local and Distributed Remote Schemas Structure Modifications Propagation and Maintenance of Metadata of Multidatabase and all subsequent Local Distributed Remote Databases" Copyright retained with IIT Kharagpur, Reg. No. L-84975/2019, Copyright Office, Government of India; 2019.

[12] Bertino E, Haas LM. Views and security in distributed database management systems; 1984.

_____