"Knowledge Scholar"
An International Peer Reviewed Journal Of Multidisciplinary Research
Volume: 02, Issue: 02, March–April 2015 *eISSN NO. 2394-5362*
Impact Factor (IIFS) - 0.776

A FRAMEWORK FOR FEDERATED DATABASE SYSTEM FOR AUTONOMOUS APPLICATIONS

Dr. Moussa Demba
Assistant Professor,
Aljouf University,
Kingdom of Saudi Arabia

Mohammed Afzal Bashu
Senior Lecturer,
Aljouf University,
Kingdom of Saudi Arabia

Abstract

The system maps of federated database can change multiple autonomous database systems into a single federated database. A federated database system provides a logically integrated view of existing heterogeneous and distributed databases. A framework for federated database has to be done for the development of environmental databases. This article proposes the modern technical challenges associated with classification of environment and Federated technology based on levels of local and compound databases integration and the extent of services are offered by the federation, it also shows the overall system with local and federated database. More generally, the framework provides a step in the development of a new generation of guidance techniques such as local and federated

database, schema integration, and data integration. Finally, the implications for practice and for further researches are discussed below.

Keywords : Federated Database system,

Definition

The architecture for FDB shows the

Introduction

A federated database system is a collection of meta-database management system local and component database systems that are autonomous and possibly heterogeneous. Now days the organizational data spread over various parts of the organization there is a need to provide a common platform over these localized databases that support global data processing applications. The architecture for FDB management systems is one solution for common platform in global application processing for different application. We then define a graphically representation for developing the popular frames works architectures of an FDBS. Finally, we discuss issues related to the present database system and tables that shows the different software's uses different database system including the merit and demerit of database system with FDBS.

General Architecture for FDB management systems

Federated database systems can provide a uniform user interface enabling users and clients to store and retrieve data in multiple noncontiguous databases with a single query or multiple query. The three important components of an FDBS as pointed are autonomy, heterogeneity and distribution. The federated database is combination of autonomous database schema like local , component , import , export and federated , external schemas. Moreover, these local databases are heterogeneous, that is, managed by different database management systems (DBMS) based on different models (like hierarchical, network, relational, object oriented). With the organizational data spread over various parts of the organization there is a need to provide a common platform over these localized databases that support global data processing applications. In general the architecture is to be design according to the current challenges. Let is consider the basic structure that shows the database with common various information.

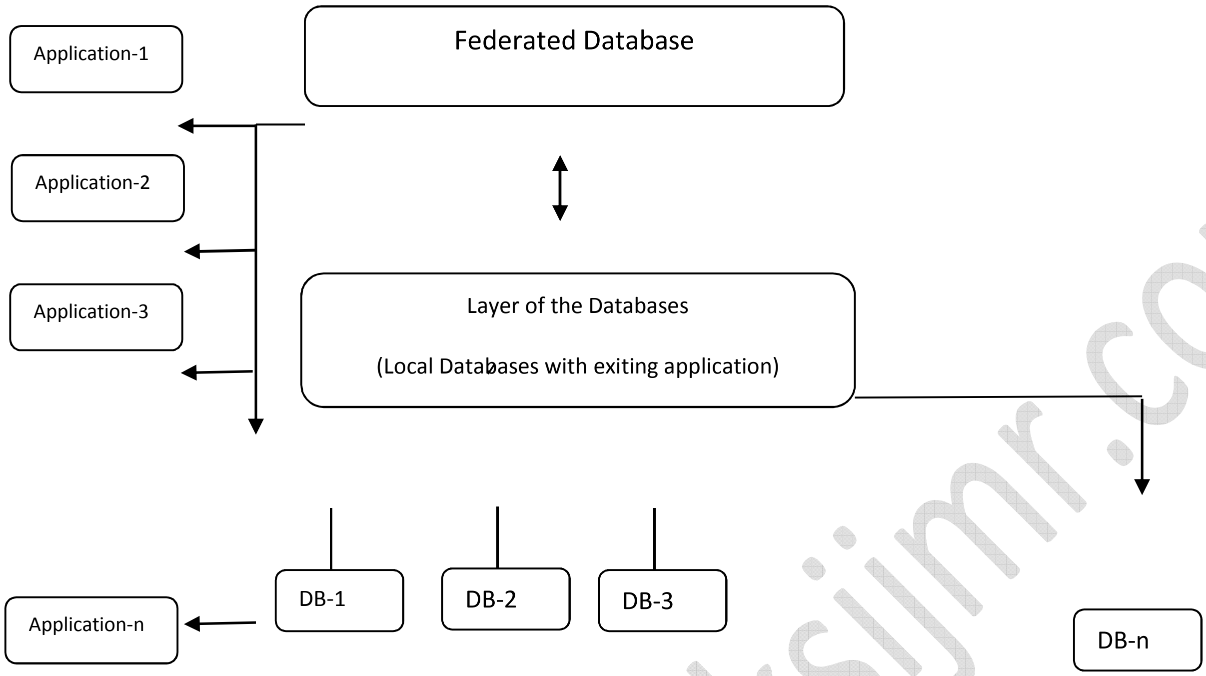

Fig-1 General frame FDBS

The above figure shows the general frame for the Federated database system. It consists of the various applications that will handle by the FDBS and the information that will share from the local DB-1.DB-2…DB-n respectively. There are three phases to be performed: requirement analysis and specification, conceptual and logical design, and physical design (Figure 1). This process is mainly iterative. In the requirement analysis and specification phase, business domain experts select strategically relevant operational database attributes and specify the purpose to use them. During this phase, the users express their view of the universe of discourse in order to produce specifications of the application requirements which will be obtained through the future database. More precisely, two types of requirements are specified: data requirements and operational requirements. Data requirements impact the content of the database, and operational requirements are concerned with the utilization of the database by users and programs. The second phase called conceptual and logical design or data modeling phase involves formulating the data objects resulting from the previous stage in terms of data modeling formalisms. The resulting logical schema is built on the data and the operational requirements. It actually contains all the information needed to meet the operational requirements .Moreover, depending on the model, it represents some business rules related to these data. If these constraints cannot be modeled by the schema, they will be embedded in programs. In relational databases this phase is split into two steps. The first step leads to a conceptual schema using Entity-Relationship (ER), Extended Entity-Relationship (EER) or Unified Modeling Language (UML) formalisms. The second step maps the conceptual schema to a relational schema. The last phase deals with implementation issues, considering object structures, and inter-object links. The logical schema is transformed into a physical one by taking into account storage considerations and performance aspects imposed by the DBMS under consideration.

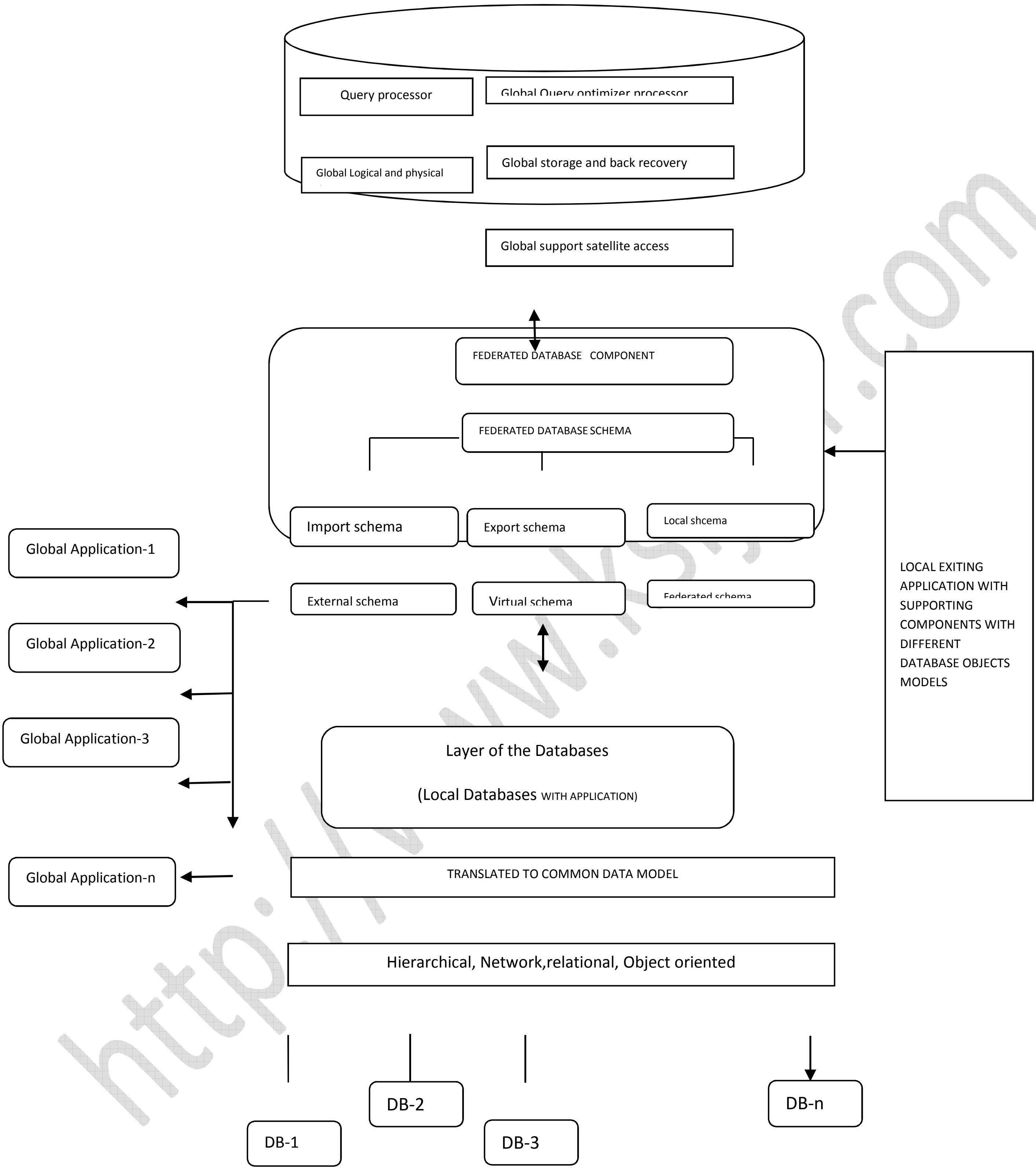

Fig-2 General frame FDBS

The above figure shows the general architecture of the Federated database system. It consists of details design of the FDBS. The federated information provides the decision support system design base on existing autonomous database system. The federated database system built from existing schema Local schema, component schema, import schema, export schema, external schema and federated schema science the federated schema built from export and import schema. It has various applications that will handle by the FDBS and the information that will share from the local DB-1.DB-2…DB-n respectively. In the Above the layer of the FDBS consist of design of a database is a long and fastidious process. A consensus is now existing about the main three phases to be performed: requirement analysis and specification, conceptual and logical design, and physical design (Figure 1). This process is mainly iterative. In the requirement analysis and specification phase, business domain experts select strategically relevant operational database attributes and specify the purpose to use them. During

Analysis of various databases with applications

Due to the data flow in various organizations that may have different applications or any modification updates with respective applications the data base will get damage or the data maybe isolated. If we use this frame work in FDB then a common database may follow.

For example: daisy software whose platform is java XML. And the backend is MySQL suppose if the organization wants to change to SQL server database means then all structure will change with respective program database. More over the common structure of this framework we use means all the performance with good features will be useful.

The below table shows different databases with different applications

Name	**Platform**	**Supported database**
Apache	Java	HSQL, MySQL, SQL server,DB2
Amery's CMS	Java	HSQl, MySQL, SQL server,DB2
Apache	Java , XML	MySQL
Daisy	Java , XML	MYSQL
Composite CI	ASP.NET	SQL Server,MYSQL,Firebird
Umbraco	ASP.NET	SQL Server , MySQL
Blosxom	Perl	Flat file database
Cyclone3	Perl	MySQL, PosterSQL, Oracle
WEbGUI	Perl	MYSQL
AdaptCMS	PHP	MYSQL
Atutor	PHP	MYSQL
Dokuwiki	PHP	Flat file database
Dotclear	PHP	MySQL,PostSQL,Oracle , SQL server

Modern data users demand control over how data is presented; their needs are somewhat in conflict with such bottom-up approaches to data integration.

Conclusions

This article describes the designers to develop the Federated database system. Hence the database users store and retrieve their information using various soft wares and databases and for that they have to do major changes whose impact sphere is an important part of the system for each type of change; for that a frame work has done which elaborates that if we create a single and federated database then we may able to use that single federated database system which defines one or several techniques to be used in order to ensure that the evolution is propagated in its system and documentation. We produced a framework matching three dimensions of changes, and database life-cycle phases to evolutionary techniques such as forward engineering, backward engineering migration process, etc. Further work will consist in defining a guidance tool in order to help database designers and maintenance teams to manage database in an unanticipated evolution. This guidance tool will be based on our changed typology and on our framework. Prototyping this tool will allow us to validate this typology and enrich it.

References

- Akoka J, Comyn-Wattiau I, MeRCI: An Expert System for Software Reverse Engineering, Fourth World Congress on Expert Systems, Mexico, 1998
- Atzeni P., Ceri S., Paraboschi S. and Torlone R., Database Systems: Concepts, Languages and Architectures. The McGraw-Hill Companies editor, 1999.
- Batini C., Lenzerini M. and Navathe S., A Comparative Analysis of methodologies for Database Schema Integration. ACM Computing Surveys Journal 18 (4), 1986.
- Management of ER'99,LNCS 1727, Paris, France, 1999.
- Yugopuspito P. and Araki K., Evolution of Relational Database to Object-Relational Database in Abstract Level. Proceedings of the InternationalWorkshop on Principles of Software Evolution, July 1999, pp 103-107.
- Zicari R., A Framework of Schema Updates. Proceedings of the 7 □_ International Conference on Data Engineering, Japan, 1991.