Study of SOA and Distributed Database Technology and its Applicability in Software Tools - Evaluating the performance of:

A Thesis submitted to

The Maharaja Sayajirao University of Baroda

for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Ms. Swapna Salil Kolhatkar

Research Guide

Prof. Harshad J. Patel

Department of Computer Science and Engineering
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390 001

December 2014

Dedicated to

Jyoti

My mother who brought me up to be the woman I am today and whose strength gives me inspiration always.

Declaration

I, Swapna Salil Kolhatkar, hereby declare that the work reported in this thesis entitled “Study of SOA and Distributed Database Technology and its Applicability in Software Tools - Evaluating the performance of :” submitted for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

is original and was carried out by me at the Department of Computer Science and Engineering, Faculty of Technology and Engineering, The M. S. University of Baroda, Vadodara. I further declare that this thesis is not substantially the same as one which has already been submitted in part or in full for the award of any degree or academic qualification of this University or any other Institution or examining body in India or abroad. Due acknowledgements have been made in the thesis for all other sources referred.

_____

Ms. Swapna Salil Kolhatkar
Department of Computer Science and Engineering
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390001.

Certificate

This is to certify that Ms. Swapna Salil Kolhatkar has worked under my guidance to prepare the thesis entitled “Study of SOA and Distributed Database Technology and its Applicability in Software Tools - Evaluating the performance of:” which is being submitted herewith towards the requirement for the degree of Doctor of Philosophy in Computer Science and Engineering.

_____

Prof. Harshad J. Patel
Research Guide
Department of Computer Science and Engineering
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390 001

_____

Dr. Anjali G Jivani
Head
Department of Computer Science and Engineering
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390 001.

_____

Prof. A. N. Mishra
Dean
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390 001.

Approval Sheet

This thesis entitled “Study of SOA and Distributed Database Technology and its Applicability in Software Tools - Evaluating the performance of:” submitted by Ms. Swapna Salil Kolhatkar in Computer Science and Engineering is hereby approved for the degree of Doctor of Philosophy in Computer Science and Engineering.

EXAMINERS:

_____

_____

GUIDE:

_____

(Prof. Harshad J. Patel)

Acknowledgements

I thank all who in one way or another contributed in the completion of this thesis. First, I thank God for guiding me always and driving me towards newer challenges

It is a matter of privilege and pleasure to express my appreciation towards my guide Prof. Harshad J. Patel for his motivation, technical guidance and constant support throughout this work. His vision and direction helped me select this subject of research and his critical comments and innovative suggestions encouraged me to complete this work successfully.

I am highly indebted to Prof. Ms. Anjali G. Jivani, the Head of my department for her sincere support and co-operation during this period. Her silent and strong assistance in providing me all the required facilities has made it possible for me to complete this work in time. Her encouraging words have always made me stay optimistic.

My sincere gratitude to my parents who have been instrumental in making me truthful, independent and responsible in life and to my in laws who have been always present to support and encourage me. My husband's critical suggestions have been very helpful. Together, their unconditional love and moral support are the main reasons for all that has kept me going. I take this opportunity to thank my colleague, A K Ajay who has supported me during this research. I also thank my friends – Manjiri and Manisha in their consistent encouragement.

My college department – Department of Computer Science and Engineering has been a place of worship as well as a place of great joy and encouragement for me because of all my teachers as well as non-teaching staff members. I would like to thank all of them for their cooperation and assistance during this work.

I would like to express my special gratitude and thanks to Mr. Amrut Marathey and my father for their timely suggestions and interest in my work.

Swapna S Kolhatkar

Abstract

The ever developing and demanding world of Information Technology along with the dynamic businesses has seen significant changes in the way applications are developed, deployed and maintained. The evolution is seen not only in developing and maintaining applications; but also in computing and storing the data.

Distributed Database technology is extensively used for many systems to store the large amount of data. A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network. A distributed database management system is the software that manages the distributed database and provides an access mechanism that makes this distribution transparent to the users. Distributed database system is distributed database and distributed database management system, together.

An administrator for a distributed database is also called the Global Database Administrator. He is responsible for work being done smoothly on distributed database servers. Work that originates on the server system can be monitored in the same way as a local system by using the same tools but with different kinds of desired information that may or may not be located on the single system.

Service-oriented architecture is a flexible set of design principles used during the phases of systems development and integration in software development process. A software system based on SOA will package functionality as a suite of inter-operable services that can be used within multiple separate systems from several business domains. It is an evolution of distributed computing based on the request / reply design paradigm for synchronous and asynchronous applications.

This thesis / research basically involves 1) the study of Service Oriented Architecture for software development and Distributed Database Technology

for distributed database administration tool and 2) the applicability of service oriented architecture in a software tool for distributed database oriented system and to evaluate its overall performance. The tool – Heterogeneous Data Migration Tool is based on the SOA principles and is intended for the data migration between heterogeneous distributed database sources.

Some papers have been published on distributed database administration and service oriented architecture. Few papers on current web technologies have also been published keeping in view their implementation for real world problems amenable to service oriented aspects. A paper has also been published on the cloud services for managing the data related to higher education.

List of Figures

List of Tables

Chapter 1: Introduction

In today's world, computers are common place with variety software and large number of applications available readily for variety of tasks. In the past, though, this scenario was different; the computers were used mainly for and by the scientists for complex calculations. Very soon, the potential of computers was realized by the businesses / people. As a result of which, along with the hardware developments, the software development saw a tremendous change and consequently user friendly software and hardware were developed. The requirements needed to be captured and programmed.

Different languages and various development methodologies assisted and aided the development and documentation of software. The networking of computers leads to the different system architectures. Also, the businesses expanded themselves to be visible globally. All this lead to the generation and exchange of large amount of data which gave rise to commensurate technology of storage and distribution of these data. The storage and distribution of data becoming very important due to which the demand for reliable hardware and the customized software increased. The ever developing and demanding world of IT along with the dynamic businesses has seen significant changes in the way applications are developed, deployed and maintained. The evolution in developing and maintaining large and complex applications made it imperative to develop sophisticated technology of computing and storing and distributing the data.

1.1 Need for flexibility

According to John Charles Olamendy in 'Strategies for Managing Computer Software Upgrades' - to be competitive, companies have to be agile. They constantly had (have) to adapt their organization and environment to increasingly frequent technological changes, mergers and acquisitions. This necessitated the development of IT systems which are integrated and reactive, and therefore solutions need a loosely coupled integration, with the

possibility to progressively improve migration of legacy applications to newer platforms. Business processes are extremely complex because of the variety of technologies in the IT infrastructure, such as the operating system, application and enterprise information systems. Integrating such heterogeneous technologies is very costly, time-consuming and very complex.

The businesses are a witness to the ever expanding dynamic global market. Adapting to such a situation is crucial and at the same time difficult. Large amount of money is spent in developing, maintaining and providing security to the systems or applications and data thereof. Business process changes need to be quickly reflected in the systems which support these processes. Nowadays there is a great effort for Enterprise Applications Integration which involves a large number of companies and developers.

1.2 Current Software Scenario

With the world becoming a global village, the demand for information across the world increases tremendously. Information systems need to evolve so as to keep pace with the ever increasing software systems. The demand in software systems and the revolution and evolution of software tools and methodologies has been excellently given by Terry Woods as shown in Figure 1.1. Development in network and database technology, organizational and economic growth, interconnection of existing databases, incremental growth, reliability and availability and reduced communication overhead – all lead to the existing IT systems of today.

The relatively new Service Oriented Architecture (SOA) methodology is a recent advancement in the IT sector to meet the ever changing situation as regards the information generation and dissemination in the IT sector and businesses which reflects on storage, processing and communication of relevant data. SOA is based on the concept of services. It is a software design that is independent of product or technology. SOA is a vast area and its implementations are different as per client need or project requirement.

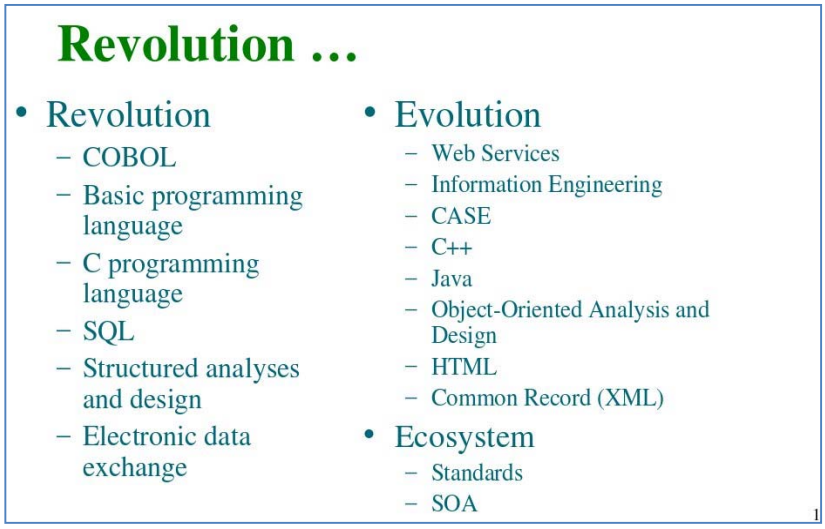


Figure 1.1 Revolution vs Evolution

Services are available at different layers like software as a service, platform as a service, infrastructure as a service etc, which has already spread into the IT industry. In the article titled “Software-as-a-Service: The Spark That Will Change Software Engineering?” Gartner’s Norton mentions that the transition to SaaS-based architectures is still in its early phase though by 2010, 15 percent of large companies would start projects replacing their ERP backbone with a SaaS offering.

SOA, promotes the idea that IT applications are service providers and consumers acting together to support the overall business goals of an organization. This new approach to enterprise architecture as posted by John Charles Olamendy is a loosely-coupled solution in which software components are very independent. He also provides an overview of how .NET supports the goals and principles of SOA. The main approach of .NET is explained along with the concepts of SOA and how .NET and the technologies associated fit into the strategy of SOA.

The evolution in software system development is excellently supported by the development in the Data Storage systems. The databases have evolved from flat files to the NoSQL databases of today. The distributed databases are also used by number of organizations to handle the vast volume of data distributed over the network and transparently accessed by the system components. The technical data storage details like the database management component, the

data communication component, the data dictionary and the distributed database component for the administration of distributed database, make the distributed database as it is today. In short, the Information Technology scenario in today's world is majorly concerned with the availability and accessibility of data through systems that are flexible and adaptable to the changing business environment and the customer demands.

1.3 Service Oriented Architecture

SOA has evolved from the use of DCOM or Object Request Brokers based on the CORBA specification. It is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed. According to Thomas Erl, SOA is a software design and software architecture design pattern based on distinct pieces of software providing application functionality as services to other applications. It can also be interpreted as a flexible set of design principles used during the phases of systems development and integration in software development process. A software system based on SOA will package functionality as a suite of inter-operable services that can be used within multiple separate systems from several business domains. It is an evolution of distributed computing based on the request / reply design paradigm for synchronous and asynchronous applications.

1.3.1 Need for SOA

The reality in business enterprises as explained by Raghu R. Kodali, is that IT infrastructure is usually heterogeneous across operating systems, applications, system software, and application infrastructure. Some of the existing applications are used to run current business processes, so starting from scratch to build new infrastructure is not an option. Enterprises need to quickly respond to business changes with agility; leverage existing investments in applications and application infrastructure to address newer business requirements; support new channels of interactions with customers, partners, and suppliers; and feature an architecture that supports organic

business. SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments.

1.3.2 Concept

An application's generic business logic or individual functions as mentioned by Raghu R. Kodali are modularized and presented as services for consumer / client applications as shown in Figure 1.2. The key to these services is their loosely coupled nature; i.e., the service interface is independent of the implementation. Application developers or system integrators can build applications by composing one or more services without knowing the services' underlying implementations. For example, a service can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.

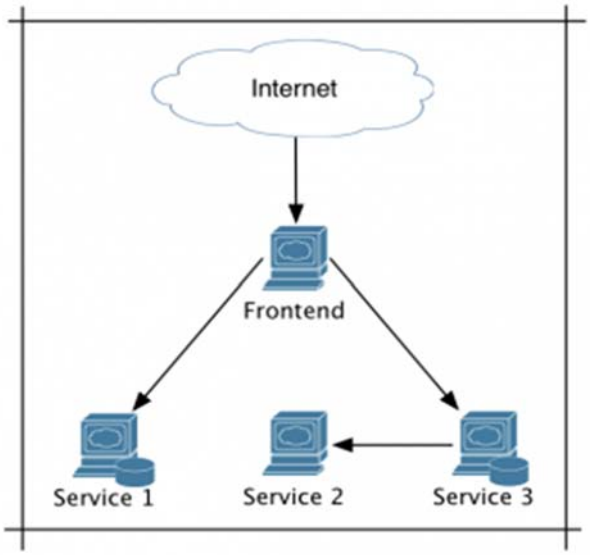


Figure 1.2 Simple SOA where services are called or communicate with each other.

SOA also generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. For example, several disparate departments within a company may develop and deploy SOA services in different implementation languages; their respective clients will benefit from a well understood, well defined interface to access

them. XML is commonly used for interfacing with SOA services, though this is not required.

SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality as shown in Figure 1.3 by by Debu Panda. An *endpoint* is the entry point for such a SOA implementation.

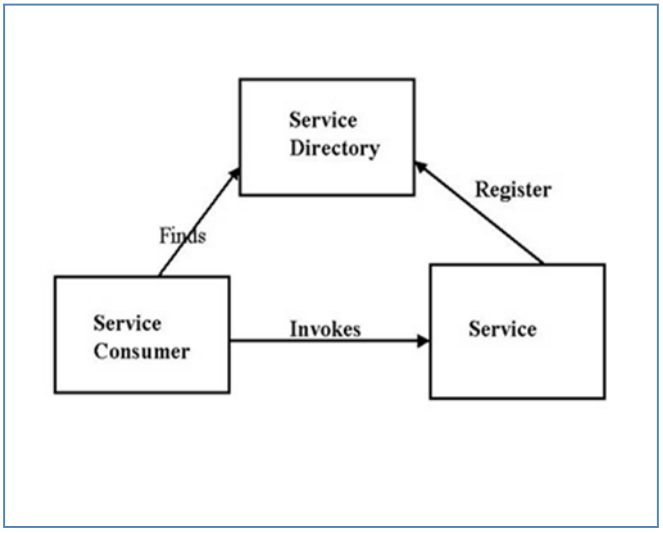


Figure 1.3 SOA from a Java Developers' Perspective

Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services.

Considering all these facts, I have implemented the SOA concept in development as well as deployment. I have implemented the SOA concepts through services that are software services and cloud services. Specifically, for the development and deployment purpose, I have considered the infrastructure

as a service, platform as a service, software as a service and application as a service available for open source tools and technologies.

1.3.3 Benefits of SOA

The benefits of SOA are realized over a period of time. It brings better re-usability of existing systems or applications or functionality and allows for the creation of applications that can be built on top of new and existing applications. SOA enables changes to applications while keeping clients or service consumers isolated from evolutionary changes that happen in the service implementation.

Most importantly, SOA provides enterprises better flexibility in building applications and business processes in an agile manner by leveraging existing application infrastructure to compose new services. The performance evaluation of SOA software systems based on the developed and implemented tool has been discussed in this study in chapter 5.

1.4 Distributed Database

The definition of Distributed Database (DDB) given by Ozsu M Tamer and Patrick Valduriez says that DDB is a collection of multiple, logically interrelated databases (DB)s distributed over a computer network. A DDB management system (D–DBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution transparent to the users as shown in Figure 1.4. Distributed database system (DDBS) is DDB and D–DBMS, together. Collections of data can be distributed across multiple physical locations. A DDB can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks. The replication and distribution of DBs improves DB performance at end-user worksites.

To ensure that the DDB are up to date and current, there are two processes: replication and duplication. Replication involves using specialized software that looks for changes in the DDB. Once the changes have been identified,

the replication process makes all the DBs look the same. Duplication on the other hand is not as complicated. It basically identifies one DB as a master and then duplicates that DB. The duplication process is normally done at a set time after hours. This is to ensure that each distributed location has the same data. In the duplication process, changes to the master DB only are allowed.

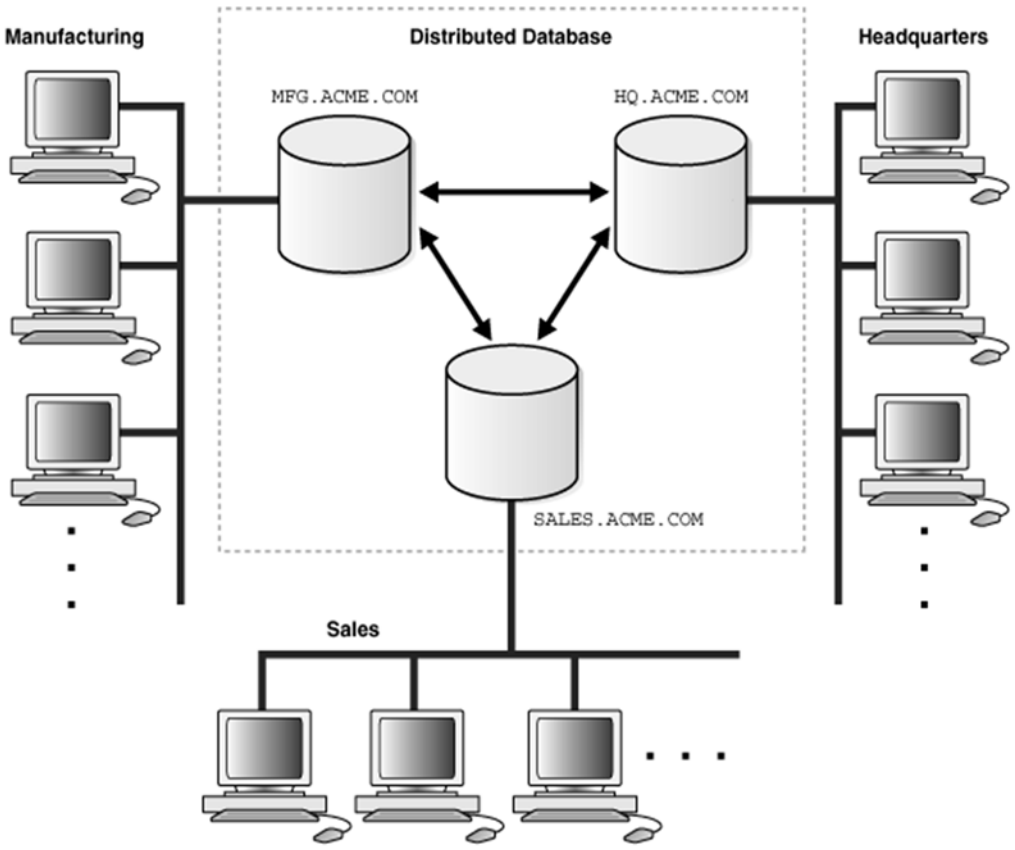


Figure 1.4 Distributed Database.

To store data in DDB, there are two principal approaches i.e. Replication and Fragmentation / Partitioning. In replication, the system maintains several identical replicas of the same relation in different sites so as to realize the benefit of data availability. In Fragmentation, the relation is fragmented into several relations in such a way that the actual relation could be reconstructed from the fragments and then the fragments are scattered to different locations. There are basically two schemes of fragmentation - Horizontal fragmentation and Vertical fragmentation. Considering the various issues for DDB, the DDB administration becomes very important as discussed in 1.5.

Kossman in 2000 has pointed out that, even though good ideas have been presented by research initiatives on distributed database management systems (DDBMS), the prototypes that have been developed did not make it to commercial tools. He believes that such research may have taken place ahead of the ideal time, mainly because of the lack of a communications infrastructure as stable and inexpensive as the Internet. With the growing availability of network communication resources, through the Internet, along with growing quality of service, some of the obstacles in the path of developing distributed databases have vanished.

1.5 Distributed Database Administration

The DB administration is about managing a DB server related to its databases, users, memory, access rights, handling system or power failures etc. The DBA is a person responsible for such administration. He is accountable for the local server, whereas for the DDB administration, there are various DBAs involved so as to coordinate the DDB administration activity. Therefore, the DB concept changes from local to global ie instead of managing a local DB site, one of the DBA or specifically appointed for DDB administration; GDBA is given the responsibility of managing all the global DBs involved in the set up of DDB. The general architecture of D-DBMS given by Ozsu M Tamer and Patrick Valduriez is as given in Figure 1.5.

These DDB systems operate in computer networking environments where component failures are inevitable during normal operation. Failures not only threaten normal operation of the system, but they may also destroy the consistency of the system by direct damage to the storage subsystem. To cope with these failures, the GDBA has to manage the DB based on site autonomy, DDB security etc. The DBA has several choices for tools to use when managing an DDB system. Few tools have been discussed in chapter 3 with respect to their functionality, architecture, dependence etc.

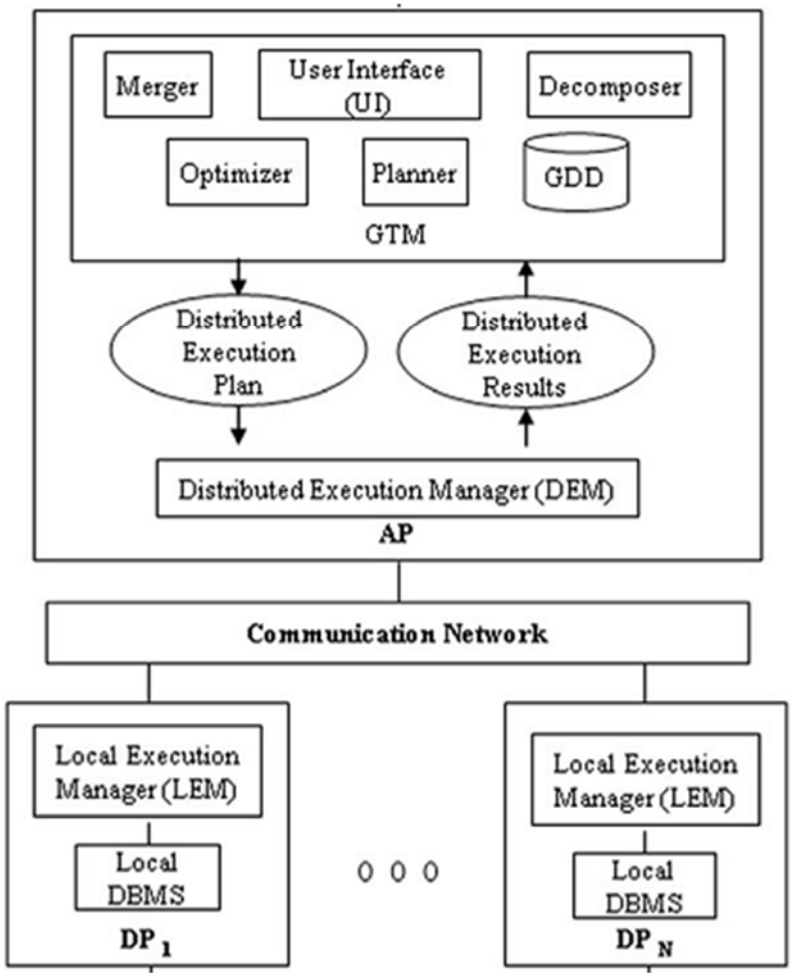


Figure 1.5 Generic D-DBMS architecture

In general, the DDB administration is required for various purposes. The administration for DDB is mostly required for monitoring DB activity in order to know of the performance and health of a DB. It may also involve data migration, operating remote servers (in special cases), job accounting, auditing, communicating with other DBAs in case of any problems related to system or hardware failure, deadlocks, long running queries, job failures etc. All these tasks or processes require some architectural model for developing a software system to aid the DBA with the job. This has been researched by others as discussed next.

According to Breno Mansur Rabelo and Clodoveu Augusto Davis Jr. as the volume of distributed data grows, the use of D-DBMS increases in

importance. One can build an integrated database from isolated and independent segments. The cost of high-capacity servers and the operational decentralization of several types of organizations could be one of the reason. Recent technological elements like XML, SOAP, web services and SOA can be put together, in a review of DDBMS concepts, using SOA to implement Internet based data exchange. As mentioned by the authors, recent works by Campbell in 2005; Tok and Bressan in 2006, have presented initiatives related to the integration of SOA and DBMS, thus proposing a service-oriented database architecture (SODA). From SODA the focus is now further on a new architecture for Internet-based distributed database management using SOA-based communication.

1.6 Layout of Thesis

The businesses are a witness to the ever expanding dynamic global market. Adapting to such a situation is crucial and at the same time difficult. Large amount of money is spent in developing, maintaining and providing security to the systems or applications. The motivation behind this study was to apply SOA principles in the development of a DDB administration oriented tool to be primarily used by the DBA / GDBA and to identify performance parameters and understand the benefits of SOA based system development. This study would enable the IT Managers to understand the merits or demerits in making any decisions related to the tools to administer the DDB that handles the functioning of the businesses or its processes.

This thesis / research basically involves 1) the study of Service Oriented Architecture for software development and Distributed Database Technology for distributed database administrator and 2) the applicability of SOA in the software tool for large data oriented system to evaluate its overall performance. The tool is developed for the purpose of data migration related to heterogeneous DB. Chapter 2 is about the literature study related to SOA and DDB administration. It also mentions the latest web technologies that provide appropriate development functionality.

Chapter 3 is about the DBA, GDBA, their role and responsibilities and the DBA tools used for their purpose. I have discussed the tools with respect to their functionality and features and also got to know that a lot of work has been done in the DBA tool area. There are a variety of tools to choose from; related to the various aspects of administration. I have finally narrowed down on the aspect of data migration and identifying the development of a relevant distinct tool by keeping SOA concepts in view. I have also selected two DBA tools for comparison with respect to their architecture and their functionality.

Chapter 4 contains the information about the heterogeneous data migration tool for DDB, its architecture and its features which will aid the DBA in the administration task related to the data migration in case of DDB. The work done on this tool is very useful for the administrators in situations as discussed in the chapter. It also mentions a case study for dynamic situations where the requirements and expectations change frequently thereby pointing to the fact that the software system can be best developed by an adaptable methodology such as SOA. Also the fact that the large amount of data related to educational sector may be distributed in nature so as to imply the importance of DDB administration in context of the case study. Hence, I have developed the heterogeneous data migration tool for DDB keeping the SOA approach in view. This chapter also contains the user interface of the tool. The tool is based on SOA concepts for its development and deployment. These concepts have helped the identification of overall performance parameters related to SOA software system as discussed in chapter 5. The major factors related to the overall performance are based on the parameters suggested by software engineering and have been discussed with respect to the tool and SOA approach. The sub factors also mention the specific case of SOA system development and deployment with the developed tool in view. I have also attached the tool code at the end for reference.

Chapter 2: Preliminary Study

I have mentioned a few important papers in this chapter which have relevance in my work of study. There are a few other related papers that have been mentioned in bibliography. The mentioned papers are on various topics on SOA, DDB, software tool development, software performance metrics, cloud services, current technologies etc. This chapter also contains information about software tool development technologies for their usage and relevance in current times. The literature review in these broad areas defined my topic under study at various stages. Along with these areas, I also have done a study of various DDB administration tools and their architecture. As the information is focused in topic but vast in content, I have mentioned this in a separate chapter as a study of DB or DDB administration tools in chapter 4. The resources referred are also mentioned in bibliography. I have also implemented a SOA based tool using open source technologies based on my study. The papers / works / chapters / articles that proved important for study along with their essence have been mentioned below.

The importance and popularity of software evolution increase as more and more successful software systems become legacy systems as mentioned by Tong Li. Therefore, the software evolution process, the inter-discipline of software process and software evolution, becomes a key area in software engineering. He also mentions that an approach to measuring software evolution processes should be proposed. The metric objects should include: interaction, efficiency, concurrency, operability, repeatability, decomposability, reach-ability etc to name a few. He says that in doing so the metric products will greatly promote the design, analysis and improvement of software evolution processes. The importance of software engineering and development in small independent services for a SOA based system, is truly interesting and motivating.

The vast and important work done by Thomas Erl in this area for introducing the design principles that comprise the service-orientation design paradigm and further exploring various aspects and effects of applying service-orientation in the real world is very useful for developing and maintaining complex software systems of current times. With such a broad idea in view and SOA as the broad topic, my preliminary study continued. Most of the resources referred were online refereed journals and articles. There are a number of researchers and authors who have made significant contributions in this area as mentioned in this chapter. The initial focus has been software engineering based topics like SOA, Web Services, eXtensible Markup Language, Cloud Services etc. along with the problems faced during software development and maintenance. Later on I narrowed on software tool development using open source technologies and its overall performance evaluation.

I got a very good opportunity to study an immensely related paper titled "SODDA – A Service-Oriented Distributed Database Architecture" by Breno Mansur Rabelo and Clodoveu Augusto Davis Jr. published in ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Volume DISI, Barcelona, Spain, June 12-16, 2008 with ISBN 978-989-8111-36-4. The authors have taken the service oriented database architecture (SODA) as a reference and proposed their work related to service oriented distributed database architecture (SODDA). The paper mentions that after the advent of interoperability standards, such as XML, and of service-based networking which laid the ground for allowing new alternatives for the implementation and deployment of distributed databases. The paper proposes the adoption of elements from service-oriented architectures for the implementation of the connections among distributed database components, thus configuring service-oriented distributed database architecture. SODDA merges DDBMS concepts with new technologies and initiatives associated to SOA and the Internet. SODDA uses Web services to coordinate operations among distributed database nodes. Each node includes a Web service to coordinate the local database, and which is capable to respond to a client

data provider, called the SODDA Hub, when the node receives requests for queries or other database operations. SODDA Hub can be seen as a common connectivity middle ware. It runs on the client's side. All operations that are submitted to database nodes are conducted through the SODDA Hub, which is also capable of accessing the distributed database's global catalog. In SODDA, the global catalog is replaced by a catalog service, which provides information about the location of Node Wrappers. Additionally, they have proposed services like Distributed Query Processing (comprising of various processes), Distributed Transactions Manager (DTM), the Replicated Data Manager (RDM), and the Database Recovery Service (DRS). To validate proposed architecture, they have implemented a SODDA prototype as a Microsoft .NET data provider. They conclude that SODDA intends to use some of the most interesting features of SOA to implement distributed databases. Expected benefits include easier implementation, lower communications costs, and greater accessibility.

"AutoGlobe: An Automatic Administration Concept for Service-Oriented Database Applications" published by Stefan Seltzsam, Daniel Gmach, Stefan Krompass, Alfons Kemper in Proceedings of the 22nd International Conference on Data Engineering (ICDE'06) IEEE Computer Society. This paper asserts the fact that future database application systems will be designed as Service Oriented Architectures (SOAs) like SAP's NetWeaver instead of monolithic software systems such as SAP's R/3. They mention a problem that the decomposition in finer-grained services allows the usage of hardware clusters and a flexible service-to-server allocation but at the same time increases the complexity of administration. In case, overloaded service instances are detected, the situation is remedied by either starting new service instances or by moving instances to more powerful servers. With this approach of automatic runtime adaption, the authors mention a tool – AutoGlobe that reduces administrative overhead and achieves a reduction of total cost of ownership as either more users can be handled with the existing hardware or because less hardware is required initially. One basic aspect of AutoGlobe is that services are virtualized, i.e., they are not running on a fixed server. Thereby, available resources are shared between all services as

appropriate for a particular situation. The paper later on presents the architecture of AutoGlobe, which is based on the team of authors' ServiceGlobe platform for location-independent execution of Web services. They next describe the foundations of fuzzy controllers and the fuzzy controller of AutoGlobe. Finally, they mention the simulation study results. For implementation, the simulation environment models a realistic SAP system with the corresponding hardware. The authors have described simulated services and servers using their own declarative XML language like real existing services and servers. This paper gives importance to the architecture of the system by showing three tier architecture of the simulated SAP installation. The three layers are the presentation layer, the application server layer, and the database layer. In this system, the end-users communicate with the SAP installation using clients in the presentation layer. Their software system installation comprises three subsystems in the application and database layer: Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Business Warehouse (BW), each running its own dedicated database and central instance.

I studied XML technology for its relevance in my work and published a paper[1] in an International journal. This paper is about removing the diagramming ambiguity that may arise in case of representation of Data Flow Diagrams (DFD. In the world of Information Technology, the working of an information system is well explained with the use of DFD. DFDs are one of the three essential perspectives of the Structured Systems Analysis and Design Method (SSADM). The sponsor of a project and the end users are briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. But, various practical problems exist with the representation of DFDs. Different tools are available in the market for representing the DFDs. These tools are user friendly and based on the object oriented features. The diagrams drawn using

1 "XML Based Representation of DFD" in International Journal of Advanced Computer Science and Applications (IJACSA). This paper has received citation in “Data Flow Sequences: A Revision of Data Flow Diagrams for Modelling Applications using XML” by James PH Coleman

these tools can be sent over the network for communicating with others. On the other hand, the XML is platform independent, textual information and is totally extensible. XML is structured and is an excellent way to transfer the information along with the meta data over the network. XML can be used to transfer the information related to DFDs, thereby removing the problems related to understanding the diagrammatic notations and concentrating more on the information flow in the system. This paper is aimed at understanding the problems related to DFDs and representing it in XML format for their storage and communication over the network. The discussion is divided into four main topics –introduction to XML and DFD, problems related to DFD, an XML representation for DFDs and finally the conclusion.

The paper titled “System Development As A Wicked Problem" by Raymond T. Yeh published in International Journal of Software Engineering and Knowledge Engineering (IJSEKE), discusses problems and principles based on experiences from good practices in the field to deal with the attributes of wicked problems. As a result of these principles, new process paradigms are needed. Automation of the alternative process and further research on downsizing mainframe applications to distributed applications are also discussed here. According to Albert L Lederer (University of Kentucky, USA) and John Benamati (Miami University, Ohio, USA) in a book titled 'Strategies for Managing Computer Software Upgrades' and edited by Neal G. Shaw; the challenges for implementing new information technologies and their effects need to be understood for their implementation. They have discussed their views in second chapter of the book related to continuous development in hardware, software and data management tools and the consequent challenges that have to be faced by organizations. The most important is that the technical people (project managers, system analysts, database administrators etc) are not aware of the challenges themselves as they have not used them before and are yet unaware of their end results. The maintainability of software as per the demands of the emerging new technologies burdens not only the IT people but at the same time leaves the customer frustrated due to poor software quality. In “Decomposing Composition: Service-Oriented Software Engineers,” M. Brian Blake proposes

an incremental development approach that integrates software engineers with subject matter experts as a first step toward development environments for service-centric software systems. Different stakeholders must have different roles in the development process, and Blake identifies two in his article: service-oriented software engineer and service-oriented software integration engineer. He concludes that by creating new life cycles and better educating software engineers about SOA methodologies will enhance next-generation distributed enterprise systems. Industry and research organizations will need to assimilate these enhanced software life cycles and evaluate their effectiveness at the end of each phase which may be different for different domains if the heterogeneity of service-centric environments is considered. However he also mentions that strategic models should be developed that help architects navigate the variations across domains while preserving the modularity of SOA environments.

The white paper on “Architected to Last: The Expanding Relevance of Service-Oriented Architecture” sponsored by: IBM, authored by Stephen D. Hendrick in April 2011; explores SOA's relationship with and relevance to IT and business. The paper mentions the need for SOA, SOA key constructs, role of SOA in enterprise and the observed benefits of SOA. It further identifies the challenges and opportunities for SOA. To start with, SOA in itself was an evolution related to the distributed computing and componentization. The key principles that guided the development of SOA include service modularity, interoperability, standardization, identification, and provisioning. These principles remain operative today because they reflect an elegant balance between being specific enough to drive adoption yet abstract enough to be extensible. The evolving software marketplace, which is transitioning from software to services, provides a context for better understanding the role of SOA in application development and deployment (AD&D) today and over the next five years. The author mentions that a tactical emphasis on data and application integration is often what brings organizations to SOA but the real value of SOA is realized as organizations standardize their approach to application development and adopt modern application development

techniques that share a service and business process focus and never lose sight of the architectural discipline that SOA brings to IT activities. In practical terms, SOA is seeing use in businesses to address integration tasks. The real benefit of SOA is that it is able to appeal to organizations on any level. There is value for both IT and the business, the architecture of SOA allows it to relate effectively to both legacy and modern applications, and SOA can support a wide array of activities ranging from simple system-to-system interoperability to being the underlying architecture for new system design with high levels of component reuse. A more recent implication of this transition from development to deployment is cloud computing. The most compelling characteristics of cloud computing are its shared service and self-service models. *The information given in this paper proved very useful to realize that SOA approach would be very appropriate for me to continue my study on my topic and also the fact that I got introduced to the importance of cloud services.*

"Software-as-a-Service Revenue Models" by Arto Ojala published in the IEEE Computer Society May/June 2013. The author gives an overview of the three main revenue models, focusing on the advantages and disadvantages for SaaS providers and their customers along with the identification of the most effective revenue model for particular situations. In accordance with Michael Armbrust and his colleagues, the author has considered cloud computing to cover not only software applications delivered through the Internet but also the hardware and system software that data centers use to provide these services. The work in this paper mentions Infrastructure as a service (IaaS) for providing computation and storage capacity, platform as a service (PaaS) for providing software development tools and an application execution environment, and SaaS that provides applications on top of a PaaS, IaaS, or private data center. The author has used the term “revenue model” in an operational sense, referring to how a firm collects revenue from its customers and has also discussed the advantages and disadvantages of traditional software licensing in comparison to pay-per-use and software renting models. An assumption made is that the SaaS providers sell the software using a

public cloud, although in some cases, the SaaS providers can rent the software for use in a customer's private cloud. On the other hand, smaller companies with limited financial resources might find it more cost-effective to use rental or pay-per-use models. Based on the comparison of revenue models from the software provider's viewpoint and the criteria for selecting the most advantageous model for software purchase and the level of impact, the author suggests that if the target segment for the software is narrow, traditional licensing or software renting can work well. By contrast, the pay-per-use model is better suited to a large customer group, with further possibilities for positive network externalities. From the customer's viewpoint, software renting can provide a trade-off between traditional licensing and the pay-per-use model. It has the particular advantage of making it possible to estimate the software costs and to buy the software without separate budgeting or complicated decision processes. *I found this paper relevant in my work. I have used a cloud service – vpsdime for the purpose of renting and availing cloud space for implementing database services. I found the discussion useful for the purpose of software tool implementation on a cloud.*

In "Realizing Service-Centric Software Systems" published by Olivier Nano and Andrea Zisman in IEEE Computer Society 0740-7459/07/2007; the authors have mentioned that to achieve adaptable, flexible, interoperable, and maintainable service-centric software systems, there is a need to define processes and create methods, tools, and techniques to support cost-effective application development and use. There must be interoperability among the tools and approaches to support development life cycles; service discovery and composition; service deployment, binding, and monitoring; quality of services and security; service-level agreements; and service description languages. They have presented five articles that tackle service-centric software systems. The first by Philippe Lalanda and Cristina Marin, is on two tools to support service composition. The first tool supports application specification on the basis of syntactic descriptions of concrete services. The second tool allows domain-specific environments to be developed according to rules specified by domain experts. The second article by Danilo Ardagna,

Marco Comuzzi, Enrico Mussi, Barbara Pernici, and Pierluigi Plebani describes a flexible approach for executing Web services. They base their approach on business processes that support adaptation both at design time and runtime, and they include several case study applications. The third article titled “Improving Web Service Discovery with Usage Data,” Aliaksandr Birukou, Enrico Blanzieri, Vincenzo D’Andrea, Paolo Giorgini,and Natallia Kokash present a system for improving service discovery on the basis of data provided by service clients rather than service owners. The fourth article by Tomas Vitvar, Michal Zaremba, Matthew Moran, Maciej Zaremba, and Dieter Fensel describe architecture to support service integration and interoperation on the basis of semantic languages and service models. This article also surveys Semantic Web technology for Web services. In the fifth article titled “Decomposing Composition: Service-Oriented Software Engineers,” M. Brian Blake proposes an incremental development approach that integrates software engineers with subject matter experts as a first step toward development environments for service-centric software systems.

The article on “Service-Oriented Architecture and Microsoft .NET”, posted by John Charles Olamendy on January 21, 2006; provided an overview of how Microsoft .NET and related technologies support the goals and principles of Service-Oriented Architecture. SOA is an architecture which evolved from the distributed and the components architectures and has become the new one to solve complex problems around the enterprise environment. The article gave an insight as to the approach of Microsoft .NET and its alignment according to SOA strategy. The importance of SOA approach and hence the alignment of the development environment according to this approach is aptly explained in this paper.

I was motivated to explore and understand other topics like design patterns and its applicability in .NET. I published an article titled “From Data Handling to Presentation of Data: Encapsulating the App.config in .NET Applications through Design Patterns” in International Journal of Computer Applications (0975 – 8887) Volume 3 – No.2, June 2010. This article is about studying the .NET based application development and the implementation of the design

patterns. Various papers[2] were referred for this purpose to show how certain database related processes can be considered as regular patterns in order to simplify the application development and project management and concentrate more on logic and presentation layer than the actual handling of data. After the introduction of Object Oriented concepts, systems were developed keeping in mind not only the data and its behavior but also laying stress on the object oriented analysis and design methodologies. Currently, there is a need to have a pattern or a service based approach to most of the software tools and applications to introduce loose coupling and ease of maintenance. In the problems related to .NET and database related operations, there can be a possibility of a pattern or a service which handles the data. This service can be thought of as an entity above DAL (Data Access Layer), to relieve the developer of syntax errors and the data format errors.

In the article titled “Successful Software Product Line Practices” by John D. McGregor, Dirk Muthig, Kentaro Yoshimura and Paul Jensen; published in IEEE Computer Society 0740-7459/10/2010, the authors have cited some software product line differences and commonalities followed by many of the aspects of software product line development that they have identified. The authors have also mentioned the successful software product line organizations like Cummins, Hitachi, Hewlett-Packard, Overwatch, SystemForge etc in order to describe their software development approach and methodologies; architecture and their challenges. They mention that a software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way in place. Organizations adopting product development strategies that include a software product line have achieved impressive results, reducing product cycle time and increasing productivity by an order of magnitude.

2 Various papers and articles [34] to [43] of bibliography were referred for the design patterns

"ReST-Based Web Access to Learning Design Services" by Juan Manuel Dodero and Ernie Ghiglione published in IEEE Transactions On Learning Technologies,Vol. 1, No. 3, July-September 2008. This paper proposes a Representational State Transfer (ReST) architectural style of accessing a learning service and its constituent resources from a Web-based environment. The authors have also described a methodology to guide the design of learning service access. They have applied a method to integrate operations of a generic wiki service in the client interface of a learning activity and validate on actual wiki services providing ReST-based programming interfaces. The current interest in Web services appears to be deeply influenced by the architectural style as given by M. zur Muehlen, J.V. Nickerson, and K.D. Swenson. The paper starts with defining ReST as an architectural style based on the HTTP protocol that describes a navigational, resource-oriented style of design. This means that, for each resource, a URI must be provided to let client applications navigate from it to other resources. ReST is usually confronted by the Service-Oriented Access Protocol (SOAP), which actually represents a procedural, function-oriented style of design in which the service is provided with a set of procedures or functions that can be called to access the service public operations. The paper mentions an interesting solution that interposes a software wrapper between the learning design player and the service provider that instantiates and enacts the service, multiplexes service calls, and compiles interesting data from the service operation. This generic service integration approach enables switching the service if required and solves the trade-off about fine versus coarse service call granularity needed for the learning design being aware of individual operations, since the wrapper is responsible for data compilation in either case. The proposed ReST solution is simple, since it is based on few principles and well-defined operations, scalable, since it relies on stateless protocols and distributed state, and layered, allowing for any number of intermediaries between activities and services. The service access requirements of open integration, fine grain control, and user interface decoupling are fulfilled by the architectural style and design methodology. Providing activities with a user interface to access ReST-based learning services is still a matter of configuration. The authors have only considered

the integration of service-specific primitive operations within learning design systems. They suggest that the actual list of functionalities that need to be integrated must cover the complete life cycle of the service, from authoring and enactment to monitoring to interoperability facilities. This is an ongoing effort that has been undertaken in the FLEXO project, funded by the Spanish Ministry of Industry and Commerce, and realized in cooperation with several Spanish companies and universities and the LAMS Foundation. This paper is very informative for web services and the issues involved therein. The web services are an important aspect for SOA.

In Tsinghua Science And Technology, a paper titled “Methodology for Web Services Adoption Based on Technology Adoption Theory and Business Process Analyses” by An Liping, Yan Jianyuan, Tong Lingyun; the paper presents a methodology that supports the deployment of web services to meet strategic company goals. Two of the most important advantages of the web services architecture are its openness and modularity. Web services are currently seen as the primary solution for dynamic integration of business functionality across the web. Web services are an architectural innovation that has the potential to alter the means by which organizations utilize information systems. The methodology presented in the paper helps organizations launch web services initiatives to make timely adoption decisions and formulate appropriate plans. It also suggests that strategies, business areas, and functions within an organization should be considered based on the existing organizational information technology status during the process of adopting web services to support the business needs and requirements. The authors have referred concepts by Chen who created a model for XML and web services technology adoption to analyze the factors affecting various adopters at various stages of e-business standards adoption. Chen has considered that companies with higher levels of IT sophistication required fewer resources to adopt web services technologies. The paper also mentions the situations for using the web services and the three types of integration used by companies ie business to business (B2B) integration, internal integration, and multi-channel implementation via web services.

At about the same time, I worked with a colleague from the management faculty and we both published a paper[3] "Transforming the Point of Sale to Point of Service - Applying SOA in the Indian Retail Scenario" in Journal of Business and Retail Management Research. In this paper we have organized the content into various sections like introduction to retailing, literature review where important papers[4] have been referred; trends in retailing and scope for Information Technology, SOA, a case study, SOA based solutions and finally the conclusion. Unorganized retailing is by far the prevalent form of trade in India – Constituting 98% of total trade, while organized trade accounts only for the remaining 2%. The retail industry in India is of late often being hailed as one of the sunrise sectors in the economy. However, in order to find its feet in the Indian environment a lot needs to be done in terms of efficiency enhancement. This paper is an attempt in exploring the benefits of applying SOA in the organized retail industry, for achieving business efficiency by transforming the point – of – sale into a point – of – service through SOA and thus ensuring maximum customer satisfaction. The paper aptly addresses the various aspects involved in retail and the IT support needed in its efficient functioning. It also shows the need for the IT related additional services in the current retail scenario by considering two supporting case studies and the possible solutions for taking care of the problems stated by the case studies. Customer sustainability and customer satisfaction can be achieved by combining the Information Technology related SOA concept with the Retail Management concept and converting the point-of-sale to point-of-service. The existing applications that are used by most of the retail outlets, need not be changed but only needs to incorporate new service(s) for handling the problems of customers and salespersons and ensuring smooth functioning of the mall/outlet for a better customer experience. A better customer experience will ensure customer sustainability which in turn boosts the profitability of the business venture. In the long run, such an approach goes on to attract more and more customers.

3 "Transforming the point of sale to point of service - applying SOA in the Indian retail scenario" in Journal of Business & Retail Management Research

4 Papers from [29] to [33] in Bibliography were referred for literature review purpose

"Cost Analysis of the Service Migration Problem between Communication Platforms" in 2008 IEEE published by Dayyan Shayani, Carmen Mas Machuca, Monika J¨ager and Andreas Gladisch. The paper is about service migration which is the process of moving currently active services from a telecommunications transport platform to another. In this paper, the source platform is referred to as old platform, while the destination is referred to as new platform. Because emerging services pose special requirements towards the networks, operators have not only to extend the capacity of their transport networks, but also need to optimize the architecture for differentiated services with advanced packet switching technologies, referred to as Next Generation Networks (NGN). During network evolution with disruptive technology changes, a general problem faced by network operators is the service migration of active services from an existing platform into a newly setup platform. Increased competition in the telecommunications industry pushes network operators to rationalize their total costs. Models and simulations appear as major tools aiming at better understanding the expenditures and to allow a more realistic network planning. However, their focus was, in the past, mostly limited to infrastructure investments. Their current work investigates strategies for service migration at minimized overall operational costs. A novel service migration cost model based on queuing theory and hill climbing optimization is provided in order to tackle the employee allocation optimization problem. Based on this framework, a thorough techno-economic analysis of costs and benefits is presented. They have presented four study cases based on this model. They are (i) the savings implied by the migration are taken into account to predict the most beneficial migration scenario among a number of parameters; (ii) the migration order (i.e. the criterion to choose which services should be migrated first) is discussed; (iii) the impact of services with different migration conditions (e.g., different marketing and installation procedures) is analyzed; (iv) personnel contracting modalities (e.g., full-time or part-time own staff or sub-contracted staff) are also compared. The paper gives an understanding of the issues related to service migration and their conclusions. The main findings of the case studies presented are that migrating running services between platforms is a costly project that should be planned carefully for minimizing its costs. A higher number of employees performing the

migration shortens the overall migration time, consequently reducing the costs of maintaining parallel platforms running. In contrast, the shorter the migration time is, the more services need to be migrated before they reach their end of life, which increases the overall migration effort. Also, the order in which services are migrated, regarding their life-time, have a strong influence on the total number of services migrated and the migration costs.

"Automating Cloud Services Life Cycle through Semantic Technologies" published in IEEE Transactions On Sercies Computing, VOL. 7, NO. 1, January-March 2014 by Karuna P. Joshi, Yelena Yesha, and Tim Finin. The authors describe a new integrated methodology for the life cycle of IT services delivered on the cloud and demonstrate ways in which it can be used to represent services and service requirements and so automate service acquisition and consumption from the cloud. They have defined an integrated ontology for processes needed to automate IT services life cycle on the cloud. They also mention that this is the first such effort, and it is critical as it provides a holistic view of steps involved in deploying IT services. Their focus for this framework is the life cycle for virtualized cloud services, where the services are composed of combining preexisting components. Hence, this life cycle does not include any requirements analysis or design phases. They have divided the IT service life cycle into five phases of requirements, discovery, negotiation, composition, and consumption and detail each phase and describe the ontologies that they have developed to represent the concepts and relationships for each phase. For demonstration, they have described a cloud storage prototype that has been developed by them based on their methodology that can be referenced by organizations to determine what key deliverable they can expect at any stage of the process. The tool (they have shown a user interface for discovering cloud storage service by specifying constraints) demonstrates how their methodology can be used to significantly automate the acquisition and consumption of cloud-based services thereby reducing the large time required by companies to discover and procure cloud-based services. As part of their ongoing work, they are working on automating complex service negotiation process where the

negotiation is on a range of values for a constraint. While developing the ontology, they have referred to NIST's (National Institute of Standards and Technology) cloud computing reference architecture to identify the key stakeholders in the life cycle. They have divided the virtualized service life cycle on a cloud into five phases. In sequential order of execution, they are requirements (where two main classes are the Specification class and the "Request for Service" class), discovery (for service certification), negotiation (for defining the service data, delivery mode, agent details, quality metrics, and cost of the service), composition (to combine the various components into a single service) and consumption (where consumer will require tools that enable service quality monitoring and service termination if needed).

"Comparison of Approaches to Service Deployment" by Vanish Talwar, Qinyi Wu, Calton Pu, Wenchang Yan, Gueyoung Jung and Dejan Milojicic; published in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS'05). The authors compare various situations and software tools in context of development and deployment. They conclude by relating the maintainability with the number of lines of code. They start with mentioning that the scale and complexity of current IT systems and services makes them increasingly difficult and expensive to administer and deploy where a service is defined as a standalone software component that encapsulates and presents useful functionality, is installed in a computing environment, and can be composed into an overall system or application. Services in this broad sense include business services as well as modules such as transaction services or databases. They can be realized as Web or Grid services or even as component services in an operating system. The authors analyze service deployment where deployment means an action to download, configure, activate, and maintain the life cycle (e.g., react to failures, terminate, and restart) of services. They quantitatively and qualitatively compare the approaches to service deployment in terms of scalability, complexity, and susceptibility to change. They have identified cases where each approach is best suited along with their hypotheses. They have also given examples of various software tools like Nixes (a tool used to

install, maintain, control, and monitor applications on PlanetLab), SmartFrog (a framework for service configuration, description, deployment, and life-cycle management) and Radia (a change-and-configuration management tool, employs a model-based approach). Their hypotheses are: First, manual deployment is likely to be well suited for simple, small scale deployments because it has small barrier to entrance. Second, for larger scale, yet still simple deployments, script-based deployment presents advantages over the manual approach, even though scripts may require learning the scripting language and then the development and maintenance of scripts. Third, for large scale, complex deployments, a language-based approach such as SmartFrog should be the best though it requires some additional investment in understanding the complexity of the framework and developing the template and configuration descriptions consisting of dependencies. Finally they mention that the environments involving a lot of dynamic, run-time changes are best suited for model-based deployment. Their results indicate that the number of steps and the number of lines of code for deployment are reduced with the introduction of more sophisticated deployment tools. Maintainability and document-ability are proportional to the number of lines of code. Manageability is proportional to the number of the lines of code modified and number of steps added in response to changes in system configuration. All these findings have benefited me for the purpose of my work.

I was fortunate to study an important paper on “Global Viewing of Heterogeneous Data Sources” by Silvana Castano, Valeria De Antonellis and Sabrina De Capitani di Vimercati published in IEEE Transactions on Knowledge and Data Engineering. *This paper provided a very good orientation for tool development that I was planning for my work. I also got an insight into the reference data model and the way the paper explains the situation using mathematical concepts.* The paper mentions the problem of defining global views of heterogeneous data sources to support querying and cooperation activities. The problem is becoming more and more important due to the availability of multiple data sources within complex organizations and in

global information systems. Global views are defined to provide a unified representation of the information in the different sources by analyzing conceptual schemas associated with them and resolving possible semantic heterogeneity. In the paper, the authors have proposed an affinity-based unification method for global view construction. They have also explored the concept of clustering to group all schema elements that have affinity in the source schemas. In the method: 1) The concept of affinity is introduced to assess the level of semantic relationship between elements in different schemas by taking into account semantic heterogeneity; 2) Schema elements are classified by affinity levels using clustering procedures so that their different representations can be analyzed for unification; 3) Global views are constructed starting from selected clusters by unifying representations of their elements. The authors have also described their experience of applying the proposed unification method and the associated tool environment ARTEMIS on databases of the Italian Public Administration information systems. The authors have then presented an affinity-based unification method for the identification and reconciliation of elements with affinity in heterogeneous data schemas. The proposed method can be fruitfully exploited with multiple data sources in order to reach the intended objectives of facilitating the unification process in cooperation scenarios, improving query processing and supporting formalization efforts in heterogeneity analysis and resolution. The experimentation in the approach has been conducted in the Italian Public Administration domain on a set of entity-relationship conceptual schemas of the databases of the Ministry of Labor and of the Ministry of Finance. They mention that the extensions of the approach will be in the direction of a deeper investigation of the issues related to extensional aspects in the unification method and of the analysis of semi structured data sources, such as the Web and XML document sources, to provide them with a uniform query interface. They have developed a prototype tool ARTEMIS in Java to support the process of creating global views. ARTEMIS provides functionalities for affinity-based analysis and clustering of source schema elements and for the unification of candidate clusters into global views. The work in this paper was beneficial to me for representing my work related to the developed tool; to be presented in a mathematical model with the help of relational algebra

expression. *I have narrowed down on three sub processes that are crucial to the working of my tool. These sub processes have been represented considering a relational database management system and is discussed in chapter 4.*

"An Enhanced Extract-Transform-Load System for Migrating Data in Telecom Billing" published in IEEE ICDE 2008 by Himanshu Agrawal, Girish Chafle, Sunil Goyal, Sumit Mittal, Sougata Mukherjea. The authors have put forth their discussion in light of Data Migration project undertaken by IBM Global Services for a large European Telecom Service Provider. The paper starts with the importance of data migration that has become a priority in many industries, spawned by a variety of business needs. Most of the existing tools for Extract, Transform and Load (ETL) process of data migration are piece-meal and do not present a complete solution. Moreover, while research has focused on the problem of schema mapping, a key step in the ETL process, most of the current algorithms do not perform well on real-world data. Their study shows that previous researchers have suggested the use of Domain Knowledge to enhance schema mapping. In this paper, the authors use domain knowledge in an innovative manner to improve schema mapping in an 'actual' industrial setting. Data Migration is an important problem in many domains and companies spend millions of dollars on such projects. In this paper the authors have presented DMT, an enhanced Extract-Transform-Load system for migrating data in Telecom Billing. Following are the major contributions of this paper: a comprehensive view of the Data Migration problem and describe an end-to-end system for ETL, utilizing domain knowledge in innovative ways to enhance schema mapping and presenting experimental results from an actual industrial setting to demonstrate the utility of domain knowledge using the authors' approach. They mention that their system combines existing tools for various stages of the ETL process and utilizes connectors, wherever applicable, to reduce the manual intervention required. To improve the accuracy of schema mapping, they have used domain knowledge to first group attributes related to each other, followed by using these algorithms to map each group on the source side to the

corresponding group on the target side. Similarly, they have used the sets of related attributes to filter the mappings generated by running existing algorithms on the source and target schema. The domain knowledge for Telecom includes ontology, a thesaurus and several domain-specific rules. To evaluate their system, they have used data from a Data Migration project and achieved significant improvement in schema mapping precision. Their results validate the utilization of domain knowledge for industrial settings. They also mention that although they have focused on Telecom billing during their experiments, their techniques can be applied to other domains as well.

"Towards a Better Understanding of Software Evolution: An Empirical Study on Open Source Software" by Guowu Xie, Jianbo Chen and Iulian Neamtiu, published in 2009 IEEE Proc. ICSM 2009, Edmonton, Canada. This paper discusses software evolution is a fact of life where researchers have proposed hypotheses on how software changes, and provided evidence that both supports and refutes these hypotheses. The authors performed an empirical study on long spans in the lifetime of seven open source projects in order to explain the software evolution process. Their analysis covered 653 official releases, and a combined 69 years of evolution. They have first tried to verify Lehman's laws of software evolution and their findings indicate that several of these laws are confirmed, while the rest can be confirmed depending on the laws' operational definitions. Second, they have analyzed the growth rate for projects' development and maintenance branches, and the distribution of software changes. They have found similarities in the evolution patterns of the programs under study, which resulted in constructing rigorous models for software evolution. The authors also mention a few laws related to the verification of each of Lehman's software evolution laws on their test applications. The first law postulates that a program must continually adapt to its environment, otherwise it becomes progressively less useful. The second law postulates that as a program evolves, its complexity increases, unless proactive measures are taken to reduce or stabilize the complexity. They mention a suggestion by Lehman that the evolution of large software systems is a self-regulating process, i.e., the system will adjust its size throughout its

lifetime. The next law, also known as “invariant work rate”, stipulates that the rate of productive output tends to stay constant throughout a program’s life time. The next law suggests that incremental system growth tends to remain constant (statistically invariant) or to decline, because developers need to understand the program’s source code and behavior. Another law stipulates that programs usually grow over time to accommodate pressure for change and satisfy an increasing set of requirements. Another law stipulates that over time software quality appears to be declining, unless proactive measures are taken to adapt the software to its operational environment. All the applications studied by the authors have points in their history where the development has forked into a development branch and a stable (maintenance) branch. The authors mention an important factor in program evolution; that is to understand which parts of the code change, and how frequently. Analyzing the reasons that lead to these hot spots i.e., parts that change frequently; can actually facilitate evolution. From the study, the authors furnish results that indicate that Continuing Change, Increasing Complexity, Self Regulation, and Continuing Growth are still applicable to the evolution of today’s open source software. They also mention that different branches of open source programs evolve in parallel, which confirms the parallel evolution hypothesis proposed by other researchers. *I found the paper valuable in terms of not only looking at tool development under the dynamic situation but also in considering the importance of open source technologies. The work done for handling dynamic situation in software development truly made me clear on my service oriented approach. At the same time I started identifying open source technologies for my work on tool development.*

"A Multidimensional Weighted-Attributes Framework (MWAF) For Evaluating Agent-Oriented Software Engineering Methodologies" by Abdel-Halim Elamy and Behrouz Far in IEEE CCECE / CCGEI, Ottawa, May 2006. This paper explains Agent-Oriented Software Engineering (AOSE) and is concerned with the use of agents in the development of distributed systems. The situation considered is especially in open and dynamic environments that require autonomous processes with communicative and cooperative features. The

explain that for deciding what AOSE methodology is the best to adopt for developing a potential multi-agent system depends on which aspects are considered more important for the prospective application under consideration. However, till the publishing of the paper, there was no industry-wide agreement on the kinds of concepts a methodology should support. This lack of standardization made the research in this area an open issue with accompanying challenges. Evaluation is a crucial and critical task here to identify the differences between several AOSE methodologies. In this paper, the authors propose the Multidimensional Weighted-Attributes Framework (MWAF), a new framework that provides a reliable solution with accurate results based on applying state-of-the-art statistical procedures to evaluate AOSE methodologies and come up with a set of measures that help in selecting the most appropriate methodology for developing prospective agent-based applications. The authors have also proposed the Multidimensional Weighted-Attributes Framework (MWAF) as a general-purpose framework that can be adopted to evaluate software products based on the user feedback. The main idea of MWAF is to define the most common and important criteria (i.e., dimensions) of the software being evaluated, identifying the attributes describing each of these dimensions, and then evaluating each dimension through its attributes against all the potential software products that are given for evaluation. The evaluation data have been collected by assigning two parameters to each evaluated attribute: a weight to identify the importance of the attribute, and a rate to measure its strength or effectiveness. Further, they have analyzed the data by means of statistical procedures to rank the evaluated methodologies according to the effectiveness of their attributes. *I found the paper very useful in order to identify and later evaluate the performance parameters for my SOA based DDB administration tool. The authors have designed questionnaire and performed necessary statistical analysis. The paper has shown the identification of attributes depending on various factors like Agency-related attributes, Modeling-related attributes, Communication-related attributes, Process-related attributes, Application-related attributes, User perception attributes, Upgrading-related attributes and Supporting properties like ontology, security and collaborative services.*

"System of Systems for Quality-of-Service Observation and Response in Cloud Computing Environments" published in IEEE Systems Journal by Paul C. Hershey, Shrisha Rao, Charles B. Silio and Akshay Narayan. This paper presents a System of Systems (SoS) approach to enable Quality of Service (QoS) monitoring, management, and response for enterprise systems that deliver computing as a service through a cloud computing environment. The authors mention that the traditional approaches for defining and evaluating quality, fall short in addressing the challenges faced by service providers and operations centers that struggle to monitor and manage quality of service. This is because they examine QoS from a limited perspective rather than from a SoS perspective applicable to a net-centric enterprise system in which any user from any location can share computing resources at any time. The authors have provided a concrete example for application of this new SoS approach to a real-world scenario (viz., distributed denial of service). Their simulated results confirm their approach. The work starts with mentioning that QoS specification and monitoring for cloud services is a complex and challenging issue as there are few universal benchmarks or standards. The usual quality metrics such as uptime and reliability may still be considered applicable in the context of cloud systems. Apart from these parameters, it is less clear what the QoS parameters unique to such systems are and how they should be applied in specific contexts. A lack of common metrics to be applied to cloud offerings from various providers is also a barrier to standardization of cloud offerings from different providers. The approach presented in this paper introduces a SoS to provide a clear and concise view of QoS events within cloud computing environments that proactively informs enterprise operators of the state of the enterprise and, thereby, enables timely operator response to QoS problems. The authors define various steps like 1) Define a SoS for monitoring, management, and response 2) Derive framework for QoS monitoring, management, and response in cloud computing environments 3) Identify Cloud computing metrics 4) Measuring performance metrics 5) Identify suitable locations within the cloud computing environment for metric detection and 6) Identify potential implementation schemes from which to collect and analyze the cloud computing QoS metrics. The authors also describe a real-world scenario in which the EMMRA CC (Enterprise

Monitoring, Management, and Response Architecture (EMMRA) for Cloud Computing Environments) SoS approach is applied to a complex cloud computing environment that is exposed to a cyber security threat (i.e., distributed denial of service - DDoS). In this scenario, the Cyber Security Plane has been used to observe cyber security threats across all domains in order to detect and enable proactive response to a DDoS security breach within any of these domains that could compromise the transactions and cause potentially devastating consequences to the end user. The authors built a prototype of an online transaction processing application where each layer of the application was hosted on a different cloud instance; the cloud itself was deployed in a private network which allowed them to demonstrate the dependence of performance metrics on the component-induced degradation. They also note that they could hardly feel the presence of a network-induced delay in the transactions because the setup was deployed in a private gigabit Ethernet supported network. Network-induced delay is a significant component of induced degradation in enterprise systems that have a global presence. They also notice that throughput is additive in nature. The new approach presented in this paper enables cloud computing service providers and operations centers to meet committed customer QoS levels using a trusted QoS metric collection and analysis implementation scheme that extends traditional monitoring, management, and response for IaaS and SaaS to a complete SOA stack that includes business logic (BaaS) and governance (GaaS). This paper was immensely beneficial to me in my work in identifying the performance parameters for the SOA based tool developed.

"The Case for Cloud Computing" in IEEE Computer Society 2009 published by Robert L. Grossman. The author distinguishes between clouds that provide on-demand computing instances and those that provide on-demand computing capacity. On-demand services and resources have been available over the Internet for some time, but the increased focus on cloud computing is due to three important differences: sale, simplicity and pricing. He mentions that a storage cloud provides storage services (block- or file-based); a data cloud provides data management services (record-, column-, or object-

based); and a compute cloud provides computational services. The author also emphasizes that due to cloud computing, the “unit of computing” has moved from a single computer or rack of computers to a data center of computers. Not only has cloud computing scaled computing to the data center, but it has also introduced software, systems, and programming models that significantly reduce the complexity of accessing and using these resources.

This work is relevant to my study on cloud services and I could identify the various cloud services for the tool under development. I have also studied the cloud services available in context of the education sector on which an article[5] *was published. This article on this study includes cloud and its types and the layers involved in cloud architecture. It also looks into the scope of applications in education sector. There are also details of existing technologies used in the education sector like Microsoft Live@edu, Microsoft Office Web Apps and Google Apps. The work also gives a comparison of features provided by companies and its suitability for education. Lastly, the article gives the cloud computing architecture which mentions the facts on recent times which has seen the focus shifting from the application development to the application deployment. The various stakeholders (that we consider based on the role) participate in this system differently. This cloud also consists of the most important services ie SaaS, PaaS and IaaS that are used to automate the education sector.*

As seen in the Figure 2.1 Education Cloud Architecture*, the students, that constitute the bulk of stakeholders, and the lecturers are allowed to use the software services (SaaS) and at the same time, they have the facility for storage of data (IaaS).*

5 Applicability of cloud computing in Acedemia in IJCSE. This paper has been cited in “Cloud Computing Support to University Business Processes in External Collaboration” by Imre Petkovics, Pere Tumbas, Predrag Matković, Zoltan Baracskai

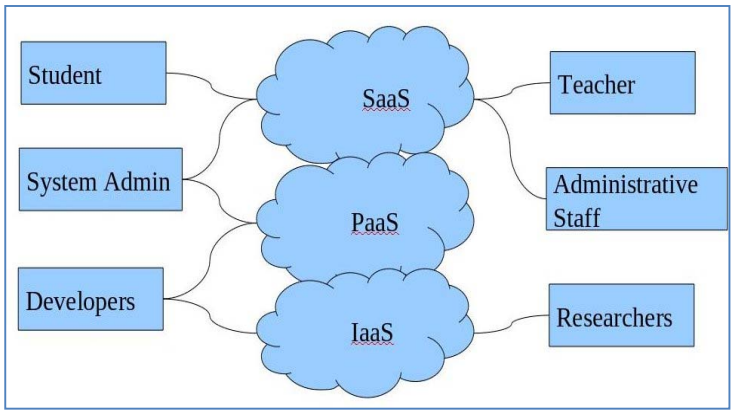


Figure 2.1 Education Cloud Architecture

This is evident from the Table 2.1 which shows a comparative of the most widely used applications. They are involved in the teaching learning process and hence their requirement on the cloud and the applications is limited to service usage and storage. Along with them, there are administrative staff who are involved in the maintaining of the documents and hence can be viewed as users of services and storage.

Another paper[6] about the need of cloud based System for higher & technical education in India as a scenario and proposing architecture for the implementation of cloud computing in educational system has been published. Finally, it concludes with the comparative study of various cloud based services available at different levels of cloud architecture for academia. The Indian Education sector has seen a tremendous rise in the field of higher education which has led to the demand for the automation of education sector at all the levels in order to cater to the need of information of various stakeholders. Due to burst in the field of communication technology everyone expects the access of relevant information (what they should); in fast, accurate and anytime any where manner. Information management of the educational sector including statutory bodies for the purpose of transparency and control through various information systems is the need and expectation of stakeholders.

6 "Need of Cloud based System for Higher & Technical Education in India" is [23] of bibliography

Table 2.1 Comparative data of various cloud services

Sr. No	Company	Apps Name	Feature	Suitability
1	Google	Google Apps	Word Processing, Presentaion, ...	Office work, Assignments, Sharing data
2	Google	Google Drive	5GB Free	Storage and Syncing (Windows & Mac, Mobile)
3	Dropbox.com	Dropbox	2GB	Storage and Syncing (Window, Linux, Mac, Mobile)
4	Microsoft	SkyDrive	-	-
5	Microsoft	Office 360	-	Office work, Assignments, Sharing data
6	Adrive.com	aDrive	50GB	Storage
7	Box.net	Box	5GB	Storage
8	YouTube	Youtube	-	Video Hosting
9	Google	Picasa	-	Image Hosting
10	Yahoo	Fliker	-	Image Hosting
11	Microsoft	Microsoft Live@edu	-	Word Processing
	Microsoft	Microsoft Office Web Apps	-	Office work, Assignments, Sharing data

These stakeholders belong to different diversified background and have different perspectives and information needs for their participation. The technological development in the abstraction and encapsulation of the IT resources has been successfully implemented with the help of cloud architecture. This technology not only caters to the various stakeholders; it also ensures the sharing, availability, security and reliability of information involved. The cloud computing architecture gives so many options for the benefit of efficient use of data, memory and computing power so as to make green computing a reality. At the same time, I have also studied the issues related to cloud computing as explained below.

"Understanding Cloud Computing Vulnerabilities" by Bernd Grobauer, Tobias Walloschek and Elmar Stöcker Copublished by The Ieee Computer And Reliability Societies March/April 2011. The authors define indicators based on

sound definitions of risk factors and cloud computing and its vulnerability. They refer to the Open Group's risk taxonomy, where vulnerability is the probability that an asset will be unable to resist the actions of a threat agent. Vulnerability exists when there is a difference between the force being applied by the threat agent, and an object's ability to resist that force. The authors have discussed further in reference to three examples of such vulnerabilities in context of cloud specific vulnerability. They are virtual machine escape, session riding and hijacking, and insecure or obsolete cryptography. They refer to NIST that describes five essential cloud characteristics: on-demand self-service, ubiquitous network access, resource pooling, rapid elasticity, and measured service. In relation to these characteristics, the authors have given examples of vulnerabilities with root causes in one or more of these characteristics. The examples are unauthorized access to management interface, internet protocol vulnerabilities, data recovery vulnerability and metering and billing evasion. They continue further that most services are likely web services, which share much vulnerability with web applications. Indeed, the web application layer might be realized completely by one or more web services such that the application URL would only give the user a browser component which shows that the supporting services and API functions share many vulnerabilities with the web applications layer. Such problems when known in advance can surely be handled and they also prove useful to people adopting or incorporating cloud computing.

"Expanding the Criteria for Evaluating Socio-Technical Software" in IEEE Transactions On Systems, Man, And Cybernetics—PART A: Systems And Humans, Vol. 38, NO. 4, JULY 2008 published by Brian Whitworth, Victor Bañuls, Cheickna Sylla, and Edward Mahinda. This paper compares two evaluation criterion frameworks for socio technical software. This work benefits my work in identifying parameters for performance evaluation of a tool. The authors mention about research on the technology acceptance model (TAM) which confirms that perceived usefulness and perceived ease of use are relevant criteria for users evaluating organizational software. However, information technology has changed considerably since TAM's 1989 inception, so an upgraded evaluation framework may apply. They say

that the web of system performance (WOSP) model suggests eight evaluation criteria, based on a systems theory definition of performance. Their paper compares WOSP and TAM criterion frameworks in a performance evaluation experiment using the analytic hierarchy process method. The authors find that subjects who used both TAM and WOSP criteria preferred the WOSP criteria, were more satisfied with its decision outcomes, and found the WOSP evaluation more accurate and complete. They suggest that as socio technical software becomes more complex, users may need or prefer more comprehensive evaluation criterion frameworks. In the discussion on the subject related to current technology evaluation criteria, the technology evaluation suggests the three variable types: 1) System variables (useful, easy to use, secure), 2) User variables (Age, gender, experience, attitude) and 3) Organizational variables (Corporate goals, technology infrastructure, social structures, normative influences). The system performance goals given in this paper are security, flexibility, extendability, reliability, functionality, usability, connectivity and privacy for WOSP. These papers provided the necessary thought process for my work later on as discussed in chapter 5.

"On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software" by Muhammad Umair Ahmed Khan and Mohammed Zulkernine in 2009 33rd Annual IEEE International Computer Software and Applications Conference. The major contributions of this paper are a comparison of various secure software development life cycle (SSDLC) processes and secure software development (SSD) methods for requirements engineering which can be helpful for software developers in selecting an appropriate SSDLC process, security specification language, and security requirements engineering process according to their needs. The properties (like specification of SSD activities for the requirements engineering phase, design phase, implementation phase, security assurance phase, resources available to the developers etc) identified by the authors to compare various software security requirements specification languages can be used to choose a particular language depending on the application requirements. The authors also mention that the identified properties can

guide software developers in designing a mechanism for translating one language into another which is useful as the tools developed for the target language can also be used to for the source language (after translation). They summarize their analysis on various SSDLC processes and SSD methods for requirements along with the related open issues. The authors also mention that to avoid security vulnerabilities, there are many secure software development efforts in the directions of secure software development life cycle processes, security specification languages, and security requirements engineering processes. They have compared and contrasted various secure software development processes based on a number of characteristics that such processes should have to analyze security specification languages with respect to desirable properties of such languages. Furthermore, they have also identified activities that should be performed in a security requirements engineering process to derive comprehensive security requirements.

"Performance and cost evaluation of an adaptive encryption architecture for cloud databases" published in IEEE Transactions on Cloud Computing by Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti. This article had been accepted for publication in the mentioned journal. It is about cloud database as a service which is a novel paradigm that can support several Internet-based applications, but its adoption requires the solution of information confidentiality problems. The authors propose a novel architecture for adaptive encryption of public cloud databases that offers an interesting alternative to the trade-off between the required data confidentiality level and the flexibility of the cloud database structures at design time. They have implemented the first proxy-free architecture for adaptive encryption of cloud databases. They also propose the first analytical cost estimation model for evaluating cloud database costs in plain and encrypted instances from a tenant’s point of view in a medium-term period. It takes also into account the variability of cloud prices and the possibility that the database workload may change during the evaluation period. Their model is instanced with respect to several cloud provider offers and related real prices. This paper is concerned with database services and takes an opposite direction by evaluating the

cloud service costs from a tenant's point of view. Their approach is quite original because the previous papers that they have referred are mainly about evaluating the pros and cons of porting scientific applications to a cloud platform, such as focusing on specific astronomy software and a specific cloud provider (Amazon), and presenting a compo-sable cost estimation model for some classes of scientific applications. The authors propose an architecture design which has been handled in three sub section ie adaptive encryption schemes, meta data structure and encrypted database management. The authors also mention the encryption algorithms that have been organized into structures called onions, where each onion is composed by an ordered set of encryption algorithms, called (encryption) layers. For the cost estimation of cloud database services the authors mention three main parameters: i.e. time, pricing and usage. The time identifies the time interval for which the tenant requires the service. The pricing refers to the prices of the cloud provider for subscription and resource usage. The usage denotes the total amount of resources used by the tenant. Their work also covers the cloud pricing models and usage estimation. They also evaluate the performance to verify whether the overheads of adaptive encryption represent an acceptable compromise from the performance point of view for guaranteeing data confidentiality in cloud database services. To this purpose, they have designed a suite of performance tests that allowed them to evaluate the impact of encryption and adaptive encryption on response time and throughput for different network latencies and for increasing numbers of concurrent clients. The authors note that there are two main tenant concerns that may prevent the adoption of the cloud as the fifth utility. They are data confidentiality and costs. The authors have addressed both issues in the case of cloud database services. The data confidentiality concerns have been addressed by proposing a novel cloud database architecture that uses adaptive encryption techniques with no intermediate servers. This scheme provides tenants with the best level of confidentiality for any database workload that is likely to change in a medium-term period. Moreover, they also propose a model and a methodology that allow a tenant to estimate the costs of plain and encrypted cloud database services even in the case of workload and cloud price variations in a mid-term horizon. By instantiating the model

with actual cloud provider prices, they say that they can determine the encryption and adaptive encryption cost of data confidentiality.

During this phase, work was carried out on a few technical topics like xml, SOA, web services etc. along with an application oriented retail management topic which were published in a few journals. *A study of such interesting work helped me immensely in formulating my own ideas on current software technologies and SOA and also the applicability of cloud services in Indian Higher Technical Education system for which various papers have been published in journals. After a couple of years, I was able to define my focus of work and come up with the basic concept and idea of a desired software tool. The idea struck due to work carried out as part of job responsibility on a governing body website. I then weighed the possibility of implementing the idea in the situation and finally designed an architecture of a tool with a purpose and implemented it as a software tool. In context of this tool, I narrowed my focus on using open source technologies for the purpose of development and deployment. The technologies that I have used are listed below with their features.*

- **LAMP**

According to Linux Solutions, the acronym LAMP refers to a solution stack of software, usually free and open source software, used to run dynamic Web sites or servers. The original expansion is as follows:

- Linux, referring to the operating system;
- Apache, the Web server;
- My SQL, the database management system (or database server);
- P (one of several scripting languages: Perl, PHP or Python. *I have used PHP for the software development*)

The combination of these technologies is used primarily to define a web server infrastructure, define a programming paradigm of developing software, and establish a software distribution package. Though the originators of these open source programs did not design them all to work specifically with each other, the combination has become popular because of its low acquisition cost

and because of the presence of its components which come bundled with most current Linux distributions. When used in combination they represent a solution stack of technologies that support application servers.

- Very low cost compared to other technologies
- GNU public licensing
- LAMP available as free software
- Best for Web applications
- LAMP has been proven faster, more flexible, and easier than any alternative

Choosing LAMP is the best way to gain complete control and power over the website. Each of the components in the LAMP stack is an example of Free or Open Source Software (FOSS). The benefit of the FOSS approach is three-fold. First, the nature of FOSS software means applications are available for free download, making them readily available to a wide range of people without payment. Second, licenses are open and thus have few restrictions on their use and the deployment of applications based on the FOSS technology. It is possible to develop and deploy LAMP-based projects without paying any license fees for distributing the software which makes it popular. Third, and a major reason for the growth and use of FOSS technology (including LAMP), is that because users have access to the source it is much easier to fix faults and improve the applications. In combination with the open license, this simplifies the development process for many enterprises and gives them flexibility and many other benefits as discussed below.

- **Customization**: As the LAMP components are open source, they have built up a huge array of additional components and modules that provide additional functionality. The open source approach enables any technical person to do the same, customizing components and functionality to suit the needs.
- **Ease of Development**: Powerful applications can be written using LAMP technology in relatively few lines of code which is straightforward enough that even non programmers can modify or extend the application.

- **Ease of Deployment**: There are neither licensing issues nor the need to compile applications, due to which deployment is often as easy as copying an application to a new host. Most hosting services provide LAMP-based environments as standard, or they can be deployed using a Linux distribution, such as Fedora or Debian.
- **Security**: With many developers developing the software and years of use by a wide range of users and community groups, LAMP technology is secure and stable where problems are normally fixed very quickly, and without the need for a costly support contract. Help and support during the development and deployment of LAMP-based applications is available.

PHP ie PHP Hypertext Preprocessor can generate dynamic page content; create, open, read, write, delete, and close files on the server; collect form data; send and receive cookies; add, delete, modify data in database; restrict users to access some pages on the website and encrypt data. With PHP, a person is not limited to output HTML. The output can be images, PDF files, and even Flash movies along with any text, such as XHTML and XML. It also has benefits as given below.

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is easy to learn and runs efficiently on the server side

- **JSON**

During my study on SOA, I came across XML technology for the purpose of implementing independent and complete services. Along with XML, I also read about JSON. There are many differences between JSON and XML as they were designed for different purpose. Once compressed, both formats take about the same space.

The most important aspect of JSON was that it had simple syntax, which results in less "markup" overhead compared to XML and it is easy to use with JavaScript as the markup is a subset of JS object literal notation and has the

same basic data types as JavaScript but it supports only a handful of different data types. In short, XML is about the document and JSON is about data. While both JSON and XML are used to represent data in hierarchy or tree structure, XML is far more descriptive and verbose. The serialized syntax of JSON has less redundancy than the syntax of XML. In case of XML, every element in the tree has a name (the element type name), and the element must be enclosed in a matching pair of tags ie XML has data and meta data, both. By contrast, JSON expresses trees in a nested array type of notation similar to that of Javascript. In fact, a JSON document can be parsed as Javascript to result in the corresponding array. XML may be more natural and useful for large, structured documents, while JSON is often the simpler and less verbose format for transmitting data sets. The JSON object is built in two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

In its simplest form, a JSON object is just a comma delimited set of name/value pairs. For example:

```
{"name one":"value one","name two":"value two"}
```

The data model of XML and JSON is different. Apart from the lack of markup, JSON also doesn't have all the other XML features, such as Processing instructions, Comments and Attributes. Hence, not only the data itself is more lightweight, but also any library implementing the data model. This makes rendering, parsing, processing JSON with non-JavaScript languages a lot faster. Also JSON is actual JavaScript code, and thus much more lightweight for a JavaScript parser to process. For the purpose of data-interchange, XML is not well suited. It carries a lot of baggage, and it doesn't match the data model of most programming languages. Though XML is difficult to do away with as it is already in use, newer technologies must surely be considered as

per the situation. Comparing XML and JSON on various factors given below, appropriate selection between the two can be done.

- **Simplicity**

XML is simpler than SGML, but JSON is much simpler than XML. JSON has a much smaller grammar and maps more directly onto the data structures used in modern programming languages. The JSON data model is simpler; it has fewer different kinds of object and they have fewer properties.

- **Extensibility**

JSON is not extensible because it does not need to be. JSON is not a document markup language, so it is not necessary to define new tags or attributes to represent data in it.

- **Interoperability**

JSON has the same interoperability potential as XML. The JSON data model is closer to the data models available in common programming languages such as Javascript, so less data conversion is needed when using these languages.

Openness

JSON is at least as open as XML, perhaps more so because it is not in the center of corporate / political standardization struggles.

- **JQUERY**

jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on the website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that can be called with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation. The jQuery library contains features like HTML/DOM manipulation, CSS manipulation, HTML event methods, Effects and animations, AJAX and Utilities. jQuery is used because of its advantages given below.

- **Lightweight –** jQuery is a lightweight library and because of its lightweight characteristics it provides high performance
- **Cross browser support –** jQuery supports cross browser, means while working with JavaScript the most important factor we have to consider is browser whereas while working with jQuery there is no need to worry about browser because jQuery functions are completely browser independent
- **Easy Dom traversing –** With the help of jQuery, the parent or child of any HTML element can be found at any time without writing much code.
- **Easy Plugins –** jQuery is highly extensible. It allows the extension of the existing behavior or add new behaviors.

Chapter 3: Current Database Administration Tools

DB administration is a very important activity for the smooth functioning of the software system. The administrators have a well defined role and responsibility for implementing as well as working with the DB technology. To aid the job of a Data Base Administrator (DBA), a number of tools have been developed to provide necessary assistance and support. The Global Data Base Administrator (GDBA) is a DBA who is having the responsibility of administering all the servers that are a part of the DDB. Various tools have been developed for the GDBA for the purpose of administering DDB servers. As the GDBA is basically a DBA, the tools are known as DBA tools or DB tools. There are a number of DBA tools; involved in DB or DDB administration, having individual features and purpose and are based on their respective architecture. Out of these tools, only the relevant tools are considered for the study purpose. Specifically, this chapter gives a brief idea of the activities and responsibilities of the DBA and GDBA, followed by the differences in DBA and GDBA, in context of DDB. It also mentions important features of some DBA Tools and compares two DB administration tools which have architectural differences. This chapter is important in realizing key features of DBA tools and their benefits along with the architecture of these tools. This study is useful for identifying the requirements of the tool that is the essence of chapter 4.

3.1 Introduction to DBA Activities

The DBA for any application is positioned in middle – top management in an organization. The importance of their role varies according to the complexity and number of DBs in an organization. A DBA is involved in a large number of tasks like design and organization (physical structure, data definition), security (failure conditions, intrusion) and system performance (performance tuning).

To assist the DBA, a large number of tools are available. These tools are related to administration and monitoring tasks which together ensure the proper functioning of the DB. Another important task performed by the DBA is the evaluation and comparison of DBMS's so that the correct product can be selected to meet the DB and customer specifications. Apart from the pricing and support agreements, in general, the DBA should consider technical details like:-

- Data structures supported – The DBA must select the data model to be used. Different models like relational, object oriented, network or hierarchical are available which may be selected after business analysis, data analysis and logical design.
- Performance – Different factors such as software implementation, hardware support, CPU power, main memory, disks required, volume of data etc have to be considered before any DBMS being implemented in the organization.

Along with the technical and economic considerations for the DBA, there are many tools available, in addition to a DBMS, that are required for report generation, data analysis, data migration, data administration, performance monitoring etc for easing the work of DBA.

3.2 DBA v/s GDBA

A DBA is more concerned with the administration of a single server where as a GDBA is having the responsibility of administering all the servers that are a part of the DDB application. The study involves SOA and DDT and hence focuses more on the combination of DBA and GDBA activities for the purpose of developing a software tool. The following information is useful in understanding the differences in the two situations.

3.2.1 DBA Tasks

The job of a DBA as mentioned in most of the Database Administrator's Guide is primarily related to all DB activities as mentioned below.

- Evaluate the Database Server Hardware
- Install the Database Software
- Plan the Database
- Create and Open the Database
- Back Up the Database
- Enroll System Users
- Implement the Database Design
- Back Up the Fully Functional Database
- Tune Database Performance
- Download and Install Patches
- Roll Out to Additional Hosts

3.2.2 GDBA Tasks

I found useful resources with IBM Knowledge Center which have presented excellent contents on the GDBA. The GDBA is defined as a DBA who is additionally assigned the responsibility of planning and setting up a DDB or could be a additionally appointed person who is responsible for the planning and setting up of DDB and following tasks :-

- ***Plan and design*** – Design a DDB that will best meet the software system needs. The needs and goals of the enterprise must be considered when making the decision to use a DDB. Few operations that require particular attention when forming the strategy are networking protocol, system security, job accounting, problem analysis, backup and recovery processes.

- ***Set up a distributed relational DB*** – prepare the application requesters and application servers to send and receive work. When the operating system is installed, DDB support is installed. However, some setup work may be required to make the application requesters and application servers ready to send and receive work.

I had set up a DDB as part of my study. In case of a DDB server environment, the registration of DB servers is done under a particular server group. The DB

software provides for the registration of such DDB servers and allows the GDBA to access information related to the DB Servers. The registration can be done either by registering single server at a time or by registering the DB server group file. The majority of these server group files are XML based with the following structure as in Figure 3.1. I had worked on registration up DDB and creating server group file with Microsoft SQL Server Management Studio.

The MS SQL Server related sample .regsrvr file consists of following XML elements :

```
<?xml version="1.0" encoding="utf-8"?>
<registeredservers>
  <servertype id="" name="">
    <server name="">
      <connectioninformation>
        <servertype></servertype>
        <servername></servername>
        <authenticationtype></authenticationtype>
        <username/>
        <password/>
      </connectioninformation>
    </server>
  </servertype>
</registeredservers>
```

Figure 3.1 Sample .regsrvr file

The study of registering DB servers either individually or through server group file led to the implementation of registration service and the service as shown in Figure 3.2 to show the performance of a server in .NET, more specifically ASP.NET and C#.NET. I have published a few papers[1] related to XML and service implementation using C#.NET.

```
protected void BtnShowXML_Click(object sender, EventArgs e)
  {
   ……...
```

1 The details of the papers published are mentioned in [25] & [26] of bibliography

```
    strfilename = "C:\\Program Files\\Microsoft SQL Server\\swapna.regsrvr";
    XmlDocument rserver = new XmlDocument();
    rserver.Load(strfilename);
    int i = 0;
    XmlNodeList elements =
        rserver.SelectNodes("//Export/ServerType/Server");
        foreach (XmlElement element in elements)
        {
                string servername = element.GetAttribute("name");
                txtXMLFile.Text = servername;
                ……
        }
        myfunction();
}

protected void myfunction()
{
        string strcon = "Data Source=COMP-20\\ADVANCED_DB;Initial
        Catalog=trial_DB;Integrated Security=True";
        SqlConnection sqlConn = new SqlConnection(strcon);
        try
        {
        sqlConn.Open();
                string strcom = "insert into t_server (server_no, server_type,
                server_name, server_ip) values (3,'SQLServer','" +
                DropDownList1.Items[1]+"','')";
        SqlCommand sqlcom = new SqlCommand(strcom,sqlConn);
        sqlcom.ExecuteNonQuery() ;
        sqlConn.Close();
        }
        catch (SqlException se)
        {
                txtXMLFile.Text = se.Message;
        }
}
protected void Page_Load(object sender, EventArgs e)
{
    ConnectionInfoBase coninfo = new SqlConnectionInfo();
   …..........
```

```
      ((SqlConnectionInfo)coninfo).UseIntegratedSecurity = "false";
      TraceServer trace = new TraceServer();
      trace.InitializeAsReader(coninfo, "");
   }

 protected void processPerformance()
   {
      PerformanceCounter objMemperf = new PerformanceCounter("Memory", "Available
Bytes");
      PerformanceCounter objProcperf = new PerformanceCounter("System", "Processes");
     PerformanceCounter objComperf = new PerformanceCounter("System", "Threads");
     Label1.Text = string.Format("{0:#,###}", objMemperf.NextValue()) + "Byte";
     Label2.Text = objProcperf.NextValue().ToString();
     Label3.Text = objComperf.NextValue().ToString();
     if (!Page.IsPostBack)
     {
         foreach (PerformanceCounterCategory objPer in PerformanceCounterCategory.
GetCategories())
        {
           ListBox1.Items.Add(new ListItem(objPer.CategoryName));
        }
     }
   }
```

Figure 3.2 Sample services developed in .NET

- ***Administration and operations tasks*** – The GDBA has to perform routine administration and operation tasks that involve monitoring DB activity, operating remote servers (in special cases), job accounting, auditing etc. In addition, the GDBA is required to communicate with other DBAs in case of any problems related to system or hardware failure, deadlocks, long running queries, job failures etc. In addition, the GDBA must monitor the network, server and DB for improving the performance of the DDB servers.

A study of various monitoring tools was carried out and in my work I also implemented a service in .NET for fetching the performance or load related information with the assumption that the server under study is a dedicated DB server. There are system tables that are used in evaluating the performance

of DB server and which can be accessed to get the performance of the DB server. As a result, the entire process of registering individual DB servers and finally fetching their performance information proved useful in understanding the GDBA role and responsibilities.

- ***Communications*** – A GDBA must plan the networking and communication issues as per the software system under consideration so as to take care of the type of information communicated or the number of connections required for the software system or application to function smoothly.

"This aspect of 'type of information communicated' interested me and I started concentrating on data migration and its issues". Another fact that came across was related to data migration and its constraints which requires necessary authorization and authentication for the actual data to be migrated between distinct systems. The Indian Higher Education sector related Information system software gave me an opportunity in understanding the data upload process and its pre-conditions. I focused my attention to the data migration and its issues as discussed in **chapter 4.** At the same time, for getting full advantage of SOA for agile systems, I also worked on applicability of cloud and its services. This study led to the identification of problem of data exchange between distinct organizations and their distinct system implementation. Few papers[2] have been published with respect to higher technical education related software system and cloud services.

- ***Security*** – A GDBA is faced with two security issues to resolve. First, system to system protection; and secondly, identification of users at remote sites. Also, the data must be available at all times and be protected from intruders. A GDBA may additionally take care of the performance improvement by fine tuning the situation and resolve problems of intrusion detection, blocking certain IP addresses etc.

2 The papers are [23] & [24] of bibliography

This requirement of security along with the presence of cloud services, I explored in data services available on the cloud. The study[3] involved various types of cloud services and their use in implementing any user developed service. All these aspects of cloud services and application in education sector along with the above five responsibilities of GDBA gives an idea of the needs and applications of GDBA tool. They are as follows:-

- **Needs** :

A GDBA needs web based monitoring, real time performance data and alerts**.** There are software systems or applications or services available online, alternatively software can be downloaded for the purpose of monitoring and getting performance alerts. A paper[4] was published related to development of services in .NET for registration and monitoring the DB servers.

This paper presents some important issues in the administration of distributed databases and identifying key factors to be considered for these purposes and finally proposing the possibility of adopting SOA approach in developing software as services for the administration of distributed databases. The paper mentions that the software community is facing a lot of challenges to ensure that the software has value added characteristics related to automation of processes and also having organization of the automation in order to keep pace with the demands of the situation. It mentions that a number of applications related to banking, retail, telecommunication, e-governance, automobile, defense etc have been successfully implemented with distributed databases. Administration issues are of vital importance to organizations adopting distributed database technology in most of the above mentioned areas. On the other hand, SOA, in the software industry establishes an architectural model for software engineering that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in

[3] The details of the study have been mentioned in the article - Applicability Of Cloud Computing In Academia i.e. [24] of bibliography

[4] The details of the paper are published in [26] of bibliography

support of the realization of the strategic goals associated with service-oriented computing.

After the implementation of registration and monitoring aspect as services, I focused on other applicability aspects as discussed below.

- **Applications** :

Various applications are required for GDBA activities like registering servers to make a server group, monitoring individual servers, replicating to a server in case of any problems, migration of data, sending alerts or emails to other DBA etc. I studied a few DBA / GDBA tools aimed at such activities as discussed in following sections and concentrated on the application of **data migration**, though the above GDBA tool related features are present in most of the well known DBA tools as the following study shows.

3.3 Few Features of Some DBA Tools

There have been gradual as well as prominent advancements in the tools related to the DBA activity. Most of the DBs have their own monitoring and administration tool like MySQL Workbench (MySQL), SQL Monitor 4 (MS SQL Server), TOAD (Oracle) etc. They have many features required by DBA and also allow for the creation of registered server groups and their administration.

For understanding the evolution history of such DBA tools, it is important to understand that advancements in DB technology happened because of the demand to meet availability of reliable data anywhere anytime as a result of increasing requirements of expanding businesses and commensurate software developments. This facilitated the growth of DDB for which the technological advancements in DBA tools can be categorized as shown below.

- **Networks** – A DDB system needs a high speed network for smooth functioning. The network is independent of the geographical components of the system. With advanced networking systems, the DDB system has

been implemented more effectively. At the same time, the need for monitoring and administration of DDB has increased due to the demand for availability of reliable data in addition to problems related to server congestion or network traffic.

- **Intranet based administration tool** – The need to support a variety of DBs, led to the development of a DBA software tool that could be downloaded and installed on a company designated server of its premises. The tool would identify and register the various heterogeneous DDBs that were part of the server groups and would then be used for administration and monitoring activity.

- **Online monitoring tool** – With the focus being shifted from maintaining software infrastructure at the company location to the use of data centers or cloud services, the online monitoring tool was developed. This idea was developed to provide all the benefits of online software utility. This facility combined with the applicability of cloud services, has many benefits.

The above three categories along with the main focus of study ie DDB (homogeneous and heterogeneous) system, gives the below mentioned types of DDB management tools.

3.3.1 Intranet based Homogeneous DDB Management Tool

The homogeneous DDB refers to the fact that only one DB Technology (DBT) is considered. The administration and monitoring of such a DDB system requires LAN connection for identifying and finally registering the independent individual DB servers for the purpose of administering and monitoring them.

3.3.2 Intranet based Heterogeneous DDB Management Tool

The Intranet based heterogeneous DB management tool was developed with an intention to manage and administer data and data servers, centrally. The tool had to be installed on a server and data servers had to be registered for the purpose of data manipulation, monitoring and administration.

3.3.3 Web based DDB Management Tool

The next advancement i.e. the web based heterogeneous DDB management tool was developed to meet the ever growing demands of DB administration using a tool that is hosted online. These tools have been developed with their individual architectures and methodologies.

3.3.4 Few DBA Tools

Based on the broad categories of DDB management tools, there are many tools available in the market, some of which are mentioned below. These tools are meant for the DBA to work on various DDB administrative aspects which can be local or global to the DBA / GDBA. I have discussed these tools with respect to their special features, but concentrated more on the data migration and architecture aspect.

3.3.4.1 Toad

Toad is a powerful and robust tool for the DBA. It provides tool sets designed to manage all data sources and keep DB professionals productive. Toad has different product or plug ins for various DBs though it does not support any non-Microsoft environment. It gives a facility to browse and manage schema and objects inside schema.

The Toad StatsPack Browser (DBA module required) takes beginning and ending snapshots of DB performance statistics and then offers graphical trending and time-series analyses based on those stats. The result is a gallery of useful charts for interpreting DB performance. Toad has a batch mode interface which scans the code and highlights the line that needs attention for DBA.

TOAD offers an advanced security management system and the DBA can use TOAD Group Policy Manager as a central point for the DB administration to navigate to every DB simultaneously, and drill down into any of them to administer DB and schema objects and also perform DB Health Check on the selected DB.

It provides DB performance testing tool that allows DBA to conduct DB workload replay, industry-standard benchmark testing, and scalability testing. It enables the DBA to deploy changes to your DB environment with confidence by reducing the risks associated with patches, upgrades, migrations (as shown in Figure 3.3), and adjustments to virtual machine configurations through the incorporated load testing tools.

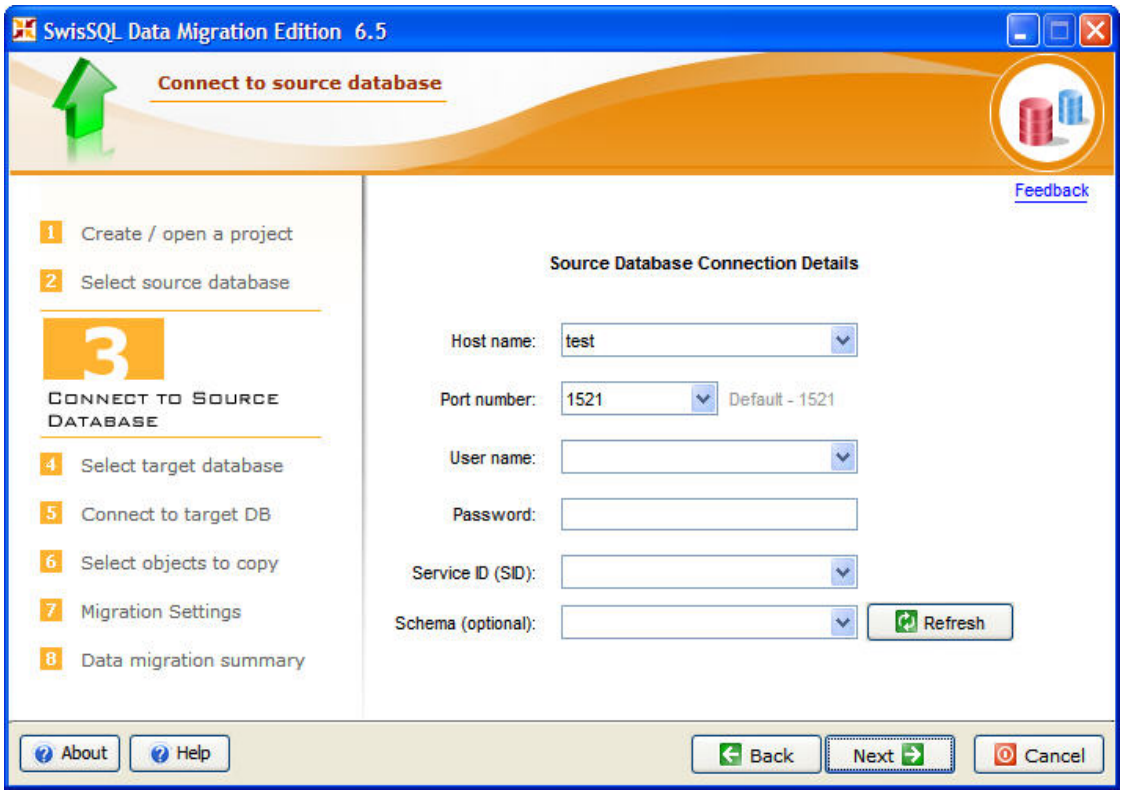


Figure 3.3 SwisSQL Data Migration Tool

Specifically, DBA can do data migration through Toad Software like SwisSQL which helps the migration and transfer of DB schemas as shown above. The migration done is at the DB level ie entire DB is migrated from source to destination. In case of data migration of more specific entity like tables or records, the DBA has to follow a different route of migration which requires his expertise in fetching the records and verifying them for names and data type information before the actual migration.

3.3.4.2 DbVisualizer :

I have considered DbVisualizer's in understanding the features for better software awareness. Its strength is in visualizing the structure in the DB, its

DB objects and most importantly, table data. DB objects are simple to create using powerful actions in DbVisualizer requiring no manual SQL to be written. DbVisualizer runs in the background so that Admin can work with other tasks. All active background tasks are monitored in a task manager with interrupt support by the DBA.

DbVisualizer is based on the JDBC toolkit. DbVisualizer supports connecting to local, remote and embedded DBs. Connection management is flexible, with options to set up a connection via the step-by-step connection wizard or by manually defining the connection data. It supports a number of DBs and runs on Windows, Mac OS X, Linux/UNIX.

The DB Object Management gives facility to browse the DB objects and administering objects such as tables, procedures, functions, triggers, packages, modules, etc that is it gives DBA visual support to create, alter and modify characteristics for DB objects. The DB Browser is Tree based navigation through DB objects. The Table Data Management is a facility to support for editing table data including binary/BLOB and CLOB data types.

3.3.4.3 AnySQL Maestro

AnySQL Maestro Group offers powerful DB administration, development and management tools for DBs like MySQL, Oracle, SQL Server, DB2, SQL Anywhere, PostgreSQL, SQLite, Firebird and MaxDB. AnySQL Maestro provides all the tools DBA needs to manage and navigate DB objects. DB Explorer is the basic feature of AnySQL Maestro which allows DBA to perform practically all necessary operations upon DBs and their objects. The Object Manager is a tool for operating on DB objects designed as an extension of the DB Explorer with advanced features, such as multiple selecting of objects (for copying, dropping, etc.).

3.3.4.4 SQuirreL SQL

The SQuirreL SQL Client is a database administration tool. It uses JDBC to allow users to explore and interact with databases via a JDBC driver. It provides an editor that offers code completion and syntax highlighting for

standard SQL. It also provides a plugin architecture that allows plugin writers to modify much of the application's behavior to provide database-specific functionality or features that are database-independent. As this desktop application is written entirely in Java with Swing UI components, it runs on any platform that has a JVM.

It is free as open source software that is distributed under the GNU Lesser General Public License. It provides a simple graphical interface to relational DBs. it is built using Java, it can access any JDBC – compliant DB running on any machine, allowing remote access to multiple DBs. A SQuirreL user can easily view and edit data in any JDBC-compliant DB, view the DB's meta-data, work with multiple DBs on both local and remote machines, use a single, consistent interface to work with different DB engines, and expand the tool's capabilities and include DB-specific functionality using plugins.

3.3.4.5 ETL tools

ETL tools follow ETL process to extract data, mostly from different types of systems, transform it into a structure which is more appropriate for reporting and analysis and finally load it into the DB and or cube(s). The data is extracted from different internal and external sources. The data is put in a staging area (SA), usually with the same structure as the source. Once the data is available in the SA, it is all on one platform and one DB for further processing ie transform. The transform step of the ETL process, allows users to check on data quality, pivot to another structure and clean the data if necessary. Finally, data is loaded into a data warehouse, usually into fact and dimension tables in the load process. From there the data can be combined, aggregated and loaded into data marts or cubes as is deemed necessary. The entire ETL process is aimed at loading the data into data warehouse for the purpose of data mining. This requires various transformations on the data for better uniformity. I have considered Informatica and Microsoft SQL Server Integration Services in this case to understand their uses.

- **Informatica** is a very good example of ETL tool for extracting the source data and loading it into the target after applying the necessary

transformations as shown in Figure 3.4. The entire process of ETL is required for the purpose of a data warehouse and this activity is scheduled to occur regularly as per the business requirements.

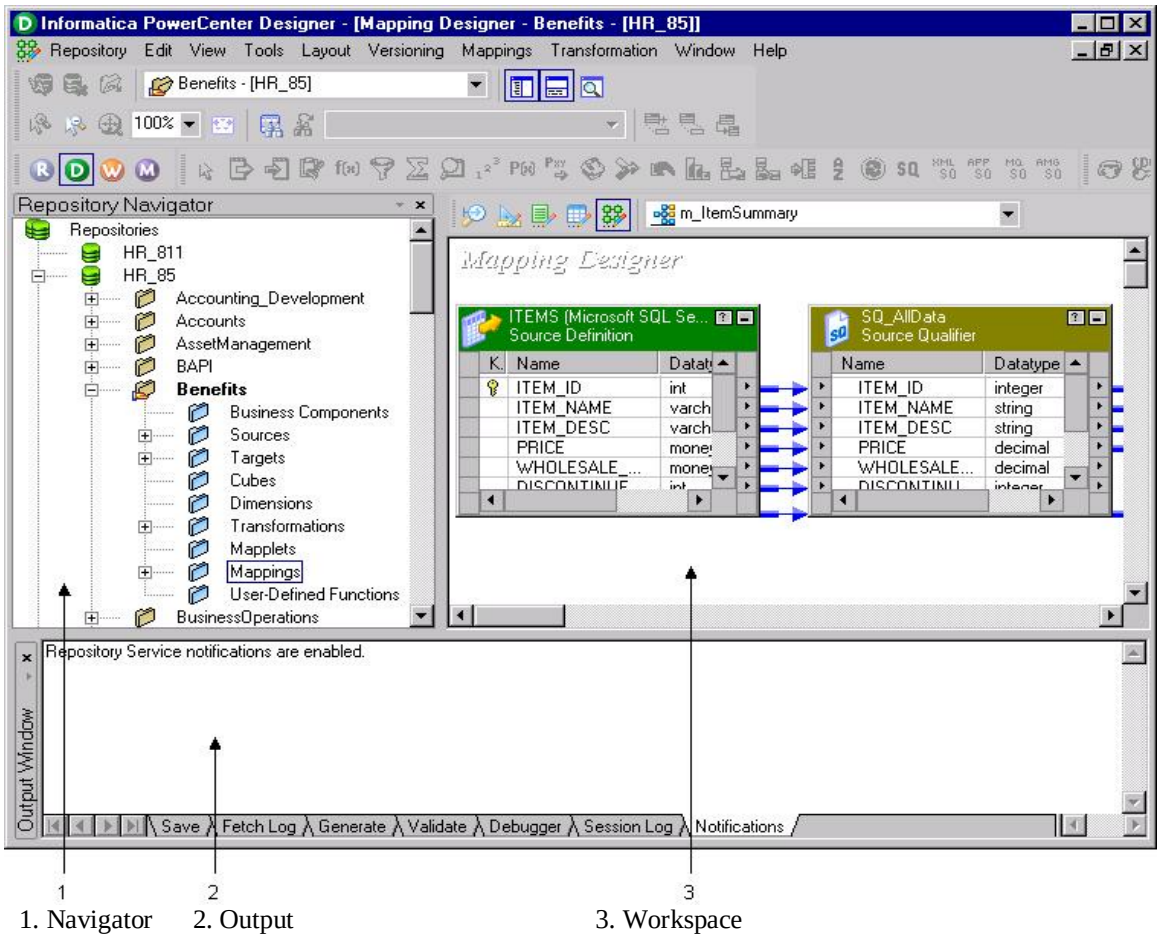


Figure 3.4 Informatica Tool

- **Microsoft SQL Server Integration Services** (SSIS) is a platform for building high performance data integration solutions, including ETL packages for data warehousing as shown in Figure 3.5. SSIS includes graphical tools and wizards for building and debugging packages; tasks for performing workflow functions such as file transfer protocol operations, executing SQL statements, and sending e-mail messages; data sources and destinations for extracting and loading data; transformations for cleaning, aggregating, merging, and copying data; a management service, the Integration Services service for administering package execution and

storage; and application programming interfaces (APIs) for programming the Integration Services object model.

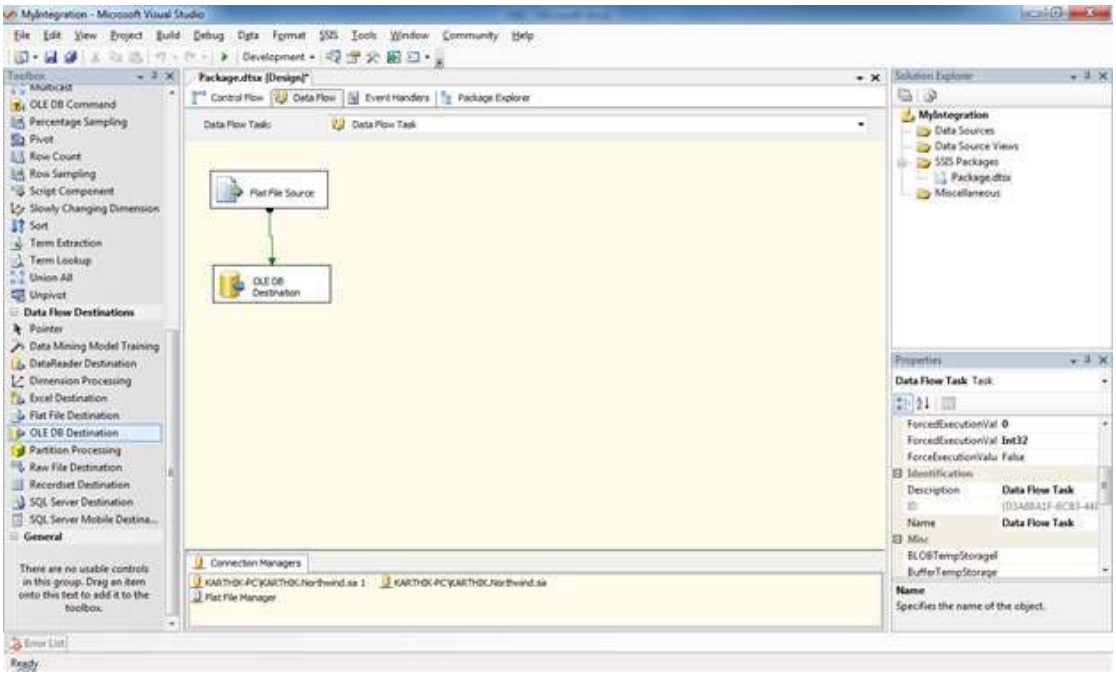


Figure 3.5 Microsoft SQL Server Integration Services

During this time I also got an opportunity to work as a user for a particular governing body website in uploading the relevant data. The situation is discussed in chapter 4 in detail which made me think of data migration where the DBA's expertise is needed. In such a case the focus is not on ETL but more on the fetch records, map onto destination DB and migrate.

The above listed DBA tool examples from 3.3.4.1 to 3.3.4.5 are among a wide variety of products available to be selected as per the need and demand of the situation. My consideration for them has solely been for the purpose of knowing the solutions available to the DBA for the administration task. But at the same time I have kept my focus on the data migration aspect related to DBA. **The tool discussed in 3.3.4.1 gives a good idea of data migration application**. There are other data migration tools also applicable at different levels or for different situations. They are Host-based file-level migration for static unstructured content, Host-based block-level migration for migrating one

storage array to another, Network-based file-level migration, Network-based block-level migration for moving data online and Array-based block-level migration. After my study on such DBA tools and keeping data migration in view, I moved ahead for understanding the architecture of a few selected tools.

3.4 Comparison of any two DBA tools

In order to limit the scope of study and focus on the main aspect applicability of SOA to DBA tools, I have selected two tools based on the networking and DDB types. They are:-

3.4.1 Aqua Fold - Aqua Data Studio

Aqua Data Studio is a complete Integrated Development Environment (IDE) for DB query, administration and development. It is not aimed at monitoring but for enabling the DBA to manage manage the DBs. It supports all major DB vendors and works on all major operating systems. Aqua Data Studio scales with your ever growing datasets and can query and manipulate Big Data effortlessly. It is used for DB management by over 300,000 IT professionals in 98 countries, across Fortune 1000, public enterprise, science and technology sectors.

3.4.2 Redgate SQL Monitor

Redgate SQL Monitor is a web-based monitoring and alerting tool for SQL Servers. It displays real-time data about the current performance of all host machines, SQL Server instances, and DBs that you choose to monitor, and raises alerts when problems occur. SQL Monitor comprises three main components that need to be installed. They are Web Server, Base Monitor service, and Data Repository DB as shown in Figure 3.6.

The Web Server and Base Monitor service can be installed on the same computer, or on different computers on the network. The SQL Server DB for the Data Repository can be hosted on any SQL Server instance, but the Base Monitor computer needs access to this DB. Installation of the Web Server and

Base Monitor service on the same computer that's hosting the SQL Server instances to be monitored is not recommended.

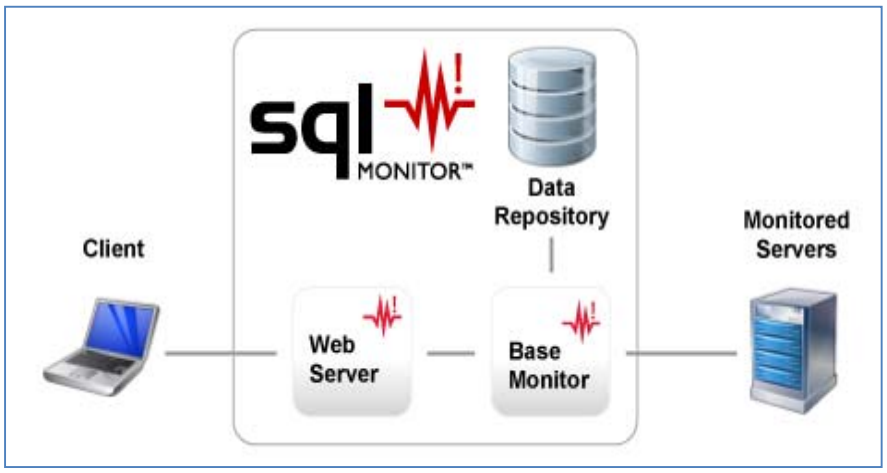


Figure 3.6 Redgate SQL Monitor Architecture

These tools have a number of functions and facilities for the benefit of the DBA. The selection of any of the tool depends on company requirements and business policies. I have shown a comparative of these tools considering their evolution, features and facilities and also their inclination towards open source development. The Table 3.1 is as follows :

Table 3.1 Comparison of Aqua Data Studio with Redgate SQL Monitor

Sr. No.	Parameter	Aqua Fold - Aqua Data Studio 14.0	Redgate SQL Monitor 3
1	Purpose	IDE for DB query, administration and development.	SQL Monitor is a web-based monitoring and alerting tool for SQL Servers.
2	Brief History	Version 6.5 in 2007 for Version Control Support for Subversion & CVS, Pivot Grid & Graphs, Grid Graphs, Grid Aggregate Functions, SQL [History] Archive, Results Compare Tool, Sybase & Oracle Debuggers to version 14.0 in 2013 for NoSQL DBs.	Version 1.0 in 2008 for SQL server monitoring to Version 4.0 for web-based monitoring and alerting for SQL Servers
3	DB support	DB2, MS SQL Server, SQL DB (Azure), MongoDB, MySQL, Netezza, Oracle, ParAccel, PostgreSQL, SQLite, Sybase,	SQL Server

		Teradata, Vertica, Apache Cassandra, Apache Derby, Apache Hive, Greenplum, Informix	
4	DDB Type	Heterogeneous (purpose is managing individual DB technicalities)	Homogeneous
5	Platform	Windows, Linux (JDK / JRE 1.6), Mac (Java 1.6)	Windows platform with .NET 3.5 SP1
6	Download	Respective 32 / 64 bit version	Installs three main components of SQL Monitor on a PC that is not a server itself
7	Network / Accessibility	Intranet	Internet. [SQL Monitor is a web application accessed using a web browser within your LAN.]
8	Open Source	$499 USD per user [AquaFold provides no-cost licensing to qualified individual OSS (open source software) developers].(as on 31/03/2014)	1 license $1,119. SQL Monitor is licensed and priced per server, with volume discounts available.(as on 31/03/2014)
9	Registration of DB Servers	Yes	Yes
10	Replication	Yes	Yes
11	Migration	Yes	No
12	Monitoring	Yes	Yes
13	Messaging	Yes	Yes
14	Email	Yes	Yes
15	User Roles	Yes	Yes
16	Security	Yes	Yes
17	Portability	No	Yes

The Table 3.1 gives a good idea as to the features and environment of the two tools. As per the study of the above DB administration and monitoring tools, we get a comparative of two existing DBA tools. The technical factors (9 to 17 of Table 3.1) in the comparison show that the DBA benefits immensely by way of monitoring the DB activity and taking prompt and necessary decisions for the smooth functioning of the system. The role and responsibility of the DBA is clear, precise and extremely important. ***This aspect directed me towards an architecture that is SOA based and would surely help in the development of more such functionality.***

The first step was in identifying the key components of the Intranet based DBA tool. The software for this tool had to be downloaded as per the platform and then installed on a machine to carry out further DBA activities. Normally, the user realizes that the architecture is client server based where information about various DB servers has to be registered and maintained. In order to identify the DB servers, the DB client needs to be present on the machine on which the tool is installed. The overall architecture can be summed up as shown in
Figure 3.7.

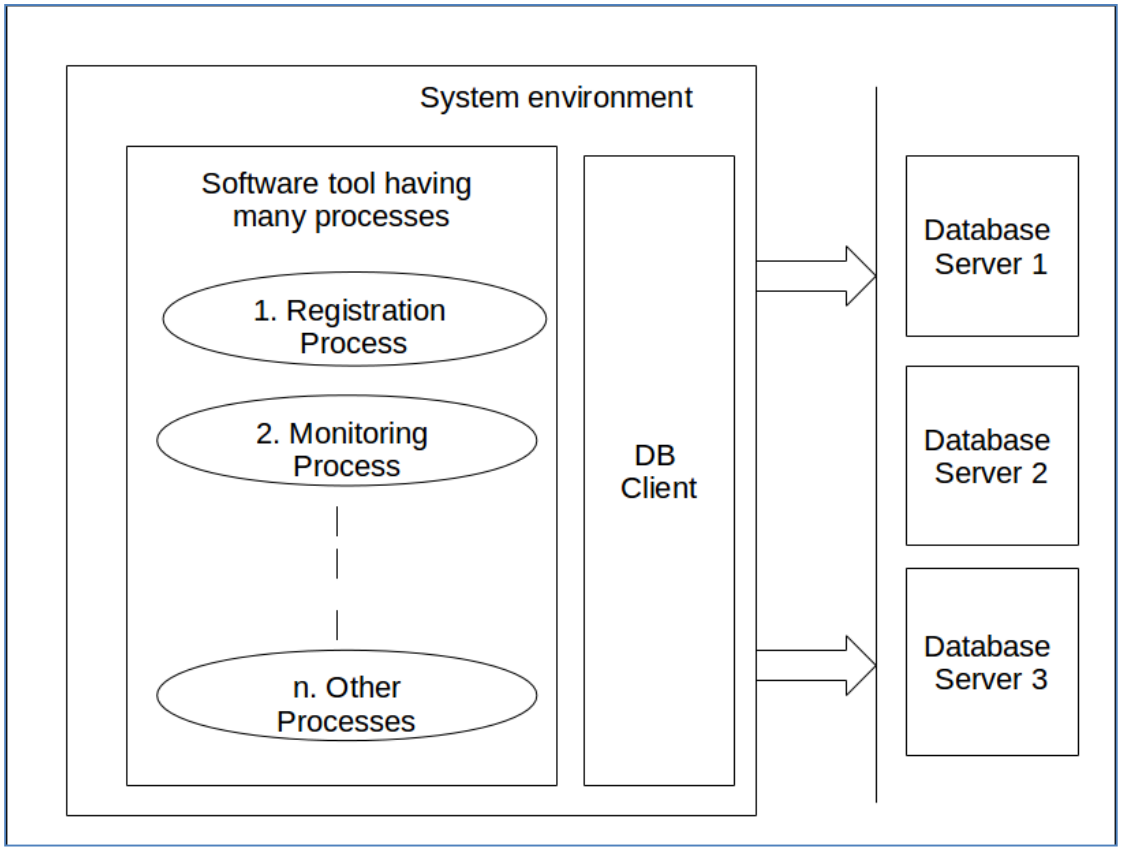


Figure 3.7 Two Tier Architecture based DBA Tool

The above architecture in
Figure 3.7 shows a few services that are of use to the DBA for the administration and monitoring activity of DDB servers. Similarly, for a web based architecture for DBA tool as shown in Figure 3.8, the user realizes the

use of some service, Web Server and Data Repository. This tool is used to monitor the DB servers from any client machine that is accessible to the web server. It requires either a DB client or a xml based server group file to identify all the DB servers for further registration and administration.

In this architecture, the DBA needs to arrange for the three prerequisites mentioned in Figure 3.6 and then connect to the web server from any client in the company network. The base monitor service that is installed on the web server or other computer, needs to connect to the various SQL servers to be monitored and also connect to the data repository for storing data related to the SQL monitor. Finally, the client machine will display all the DBA related activities like monitoring.

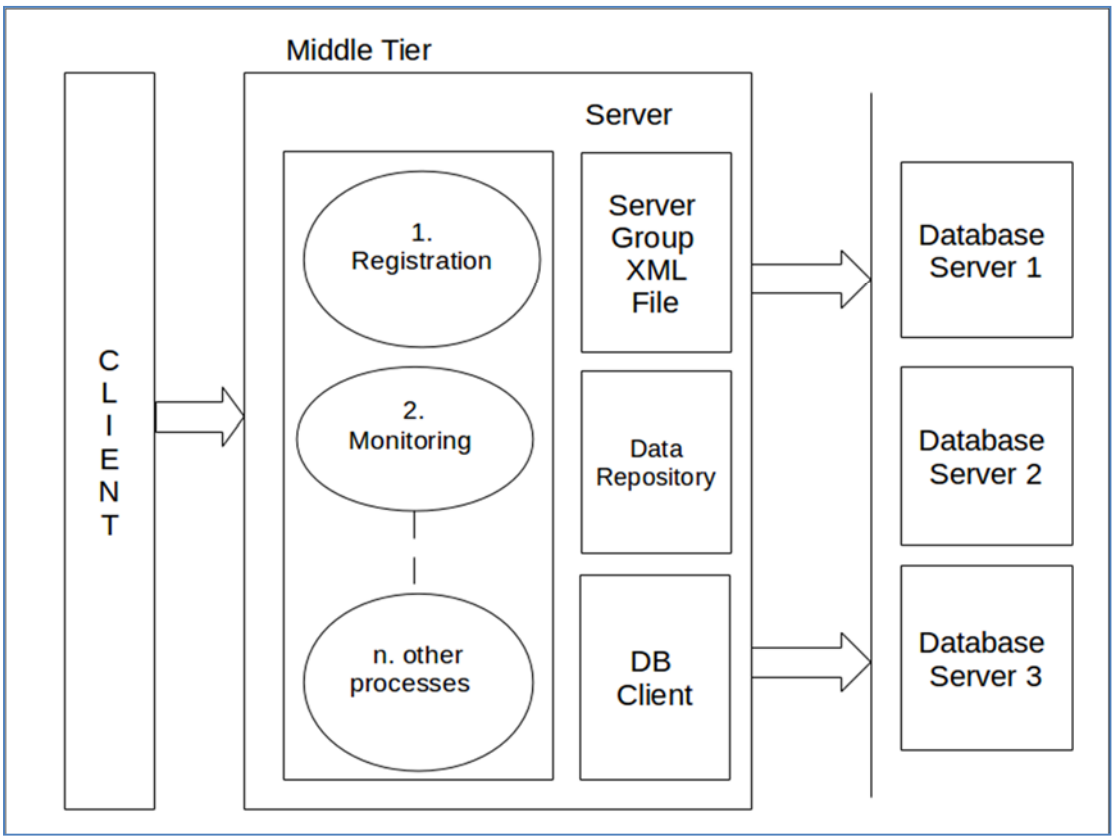


Figure 3.8 Three Tier Architecture based DBA Tool

Taking the reference from the architectures shown in Figure 3.7 and Figure 3.8*, I have worked upon the SOA based architecture for the planned (discussed in chapter 4) tool in order to migrate data of heterogeneous DDB*

under consideration. The main requirements of the architecture are: 1) web server where services are deployed, 2) a DB or data repository where the data related to the tool and its activity can be stored and 3) the client from where the system can be accessed.

The web server and DB may be rented from a cloud space renting company so as to get the benefits of SOA totally. The benefit of scalability is also realized with this facility. *At the end of this chapter, I continued to concentrate more on the SOA based architecture and efforts in data migration. At around the same time, I got an opportunity to use a governing body software application for the student data upload facility. This really motivated me to think more and the further discussion is in the next chapter.*

Chapter 4: SOA based Heterogeneous Data Migration Tool

4.1 Motivation behind HDMT

The Indian Education sector has seen a tremendous growth in the last decade with the Ministry of Human Resource & Development (MHRD) giving permission for private institutions / universities for higher education. This required the setting up of various statutory bodies like All India Council for Technical Education (AICTE), National Board of Accreditation (NBA), National Assessment and Accreditation Council (NAAC), Directorate of Technical Education (DTE) etc. These bodies and their policies enabled the establishment and formation of various institutes in almost every state which required their monitoring & control, to ensure uniformity of environment; both administrative as well as technical. Effectively, they provide the necessary rules, regulations and guidelines in the formation and running of any technical educational institution so as to be a world class organization in leading the technological and socioeconomic development of the country. This is done by enhancing the global competitiveness of technical manpower and by ensuring high quality technical education to all sections of the society. Various MIS / BI reports are published to gain more insight into the sector. Such reports are generated by gathering relevant data of various institutes or colleges. These reports show the growth and volume of courses and educational institutes.

This data is made available by respective colleges or institutes to the governing body website in their prescribed format for better control and monitoring of the environment. The data is mainly related to the institute or college infrastructure, financial investments, teaching and non teaching staff, student admitted, courses conducted etc. This data is obtained and published as various reports for the purpose of better human resource planning and development by the governing body.

On the other hand, the colleges or institutes try to maintain the data of such large volume by implementing software systems or by other methods at the college level. Currently, few of the colleges or institutes have their individual MIS or ERP system implemented at their end so as to handle vast amount of data and cater to the demand of any management reports. The software systems are designed to be implemented as per the specifications of the colleges and need to provide for any future changes in the current implementation. Many colleges use their respective software systems or other software methods which means that their entities or data attributes are very much similar and the schema may vary in naming convention, data type and data size because of vendor and requirement differences. But, the information that they want to exchange with the governing bodies is definitely the same which needs to be converted into prescribed format before upload process at governing body website.

Currently, these two systems communicate or exchange their data with each other by using some middle entity like excel for the data upload. The excel sheet has to be as per the prescribed details as shown Figure 4.1. The column headers, certain standard and defined input like course have be selected from the options given. The sequence of columns and the format of the data is to be followed as prescribed.

Title	First Name	Middle Name	Surname / Family Name	Mother's Name	Father's Name	Res Phone
Miss	ANU	MUKUND	BILONIA	ARCHANA	MUKUND	24563214

Figure 4.1 Sample of format of excel sheet

The instructions mentioned for the upload process is also mentioned in the user manual provided by the governing body. A sample of the instructions is as given in the Figure 4.2, Figure 4.3 and Figure 4.4

i) Script ActiveX controls marked safe for scripting* >Enable
j) Initialize and script ActiveX controls not marked as safe for scripting >Enable

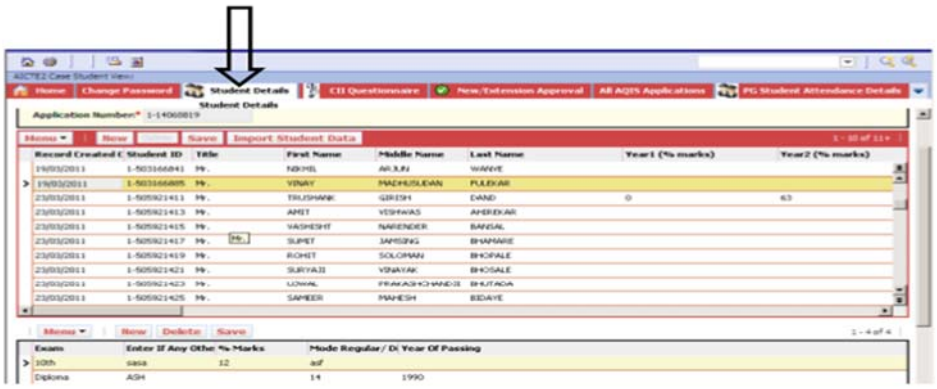
iv. The file should be placed on the Drive(D:\). It should have the path as "**D:\FacultyExcel.xlsx**"
v. All data where column header is in Red is mandatory data

vi. Wherever dropdown list is given, please select value from the dropdown. No other values will be accepted. Please don't copy & paste or drag and drop in the excel sheets.
vii. For checking valid data, in the excel toolbar, go to Data tab >Data Validation >Circle Invalid Data. Correct the circled data except for the Headers.
viii. One data row has be entered as a sample data row, for reference
ix. Please select Programme first and then Course
x. Enter the Res Phone No without STD code(5-10 digits)
xi. FY/Common Subject Teacher should be entered only if First Yr teacher is Y
xii. If FY/Common Subject Teacher is Y then enter FY/Common Subject
xiii. After one attempt of using import facility close the browser and reopen to use import functionality again.
xiv. When clicked on the Import Button, if prompted for replacing existing file, click on "OK".
xv. When clicked on the Import Button, if prompted for running ActiveX controls, click on "OK".
xvi. Once the data is imported check the excel sheet for the "LogSheet", for information about import status. For further import of data in the same tab, delete the "LogSheet", and then proceed

Figure 4.2 General Instructions for accepting and sending data using Excel sheet

Student Details:

1. The institute log's in to the portal.
2. Navigate to – **Student Details** View.
3. Click on the new button to create new student record.
4. To create new students for academic year 13-14 click on New button. To search for the students for academic year 12-13 you can search in the existing view for the student as the record has been already created previous year.

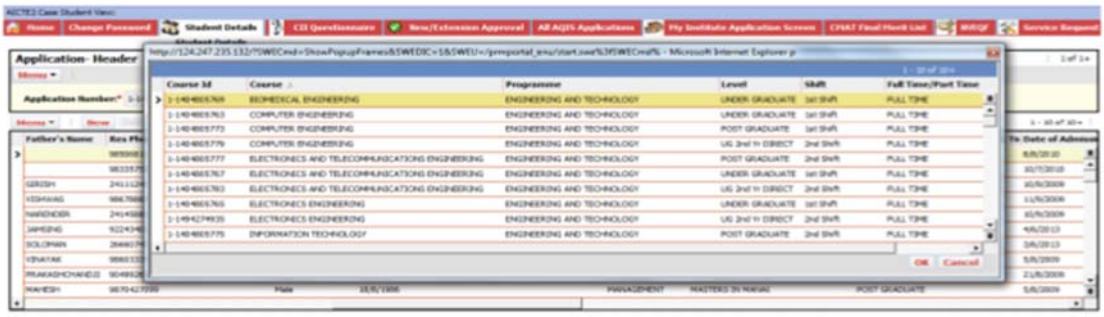


5. In Student Details view the fields that are mandatory are, First Name, Last Name, Title Father's Name, Mother's Name, Student Status, Date of Birth, Date of Joining, Course, Programme, Permanent Address Line 1, Home District of the student, Home State of the Student. Once student selects the course from pick applet Course Id, Programme, Level, Shift gets populated automatically.

Figure 4.3 Instructions for adding student details

Help Manual for PG Approval Process

6. The Institute can select the approved course from the pick applet and its corresponding programme, level, shift, full time/part time and course id get populated automatically.



7. Institutes can import student records in bulk by clicking on the button 'Import Student data'. You need to save the Student Excel Sheet in D:\StudentExcel.xls Update the StudentExcel.xls with all student records that need to be created and click on 'Import Student data button'. This will create records in Student Details View.

   Student Excel Sheet for Importing Student Details is available in AICTE Website -> Students ->Scholarships -> PG Scholarship GATE/GPAT -> Format for importing student Data

Figure 4.4 Instructions for importing student details from excel sheet

After the upload process, there is a report on the number of records sent and updated. This report helps in identifying the success of the upload operation. Any errors or exceptions are mentioned in the report and have to be accordingly handled by the colleges or institutes.

The entire process of data gathering, data formatting and data uploading or sending, generated an interest in me. My work on one of the module for sending data from college to governing body on their website motivated me to think about the situation and the complexities therein. The most important question that came to me was that if these two systems are having exactly the same data then there should be no need of such middle ware like excel. Data should be transferred by some easy methods.

Alternately, in cases where colleges are using the software systems, then, for the data uploading, the college has to download the data, convert in required format from the software system that they are using and then upload it on the governing body location. The trouble with this middle ware is the need to download the data for its exchange purpose; which spoils the security, authorization, privacy etc. that has been implemented on the MIS system to maintain above data. This sequence of activity is an additional burden to the DBA and ideally should not require an intermediate data source for the data communication.

All these facts along with the discussion in 3.2.2 (d) related to the communication helped me in identifying the features of a tool for the above situation. My study about various DBA tools as discussed in chapter 3, gave an idea about the existing migration tools and its features and also about the existing ETL tools. These tools are complete in themselves with their respective objectives and features. As of now, the entire DB migration is already being done successfully by many DB tools like SwisSQL Data Migration Tool as discussed in 3.3.4.1. So, I have focused on the requirements of the data migration / exchange tool with respect to education sector or any other sector in which data is to be migrated between distinct companies / organizations with some mutual agreement to share details related to the data exchange. **The tool is a variation of existing migration tools where DBA can actually select the attributes and records and not carry out the entire DB or table migration.** I have also taken into consideration the fact that different companies / organizations maintain data using different technology. This means that the proposed tool should implement data exchange / migration between heterogeneous DBs.

The understanding about SOA and Cloud services, together; also pointed to the fact that there could be a service made available online for the data transfer or data uploading in such cases. The service can greatly benefit all those involved in the data exchange activity. Viewing this requirement as additional services to be made available to DBA of institutes or person handling DB activities of institutes or institutes' software, an additional layer of

service was planned as it involves large amount of data transfer between the affiliating institutes and governing bodies. **This transfer necessitates in most of the cases, the format and data type transfers which becomes a bottleneck for DBA. The tool is handy in such situations.** Keeping all this in view, *I have implemented a SOA based Heterogeneous Data Migration Tool (HDMT), which will aid the DBA in the task of exchanging or migrating the required data.* The HDMT tool aims to provide a general utility for the easy exchange of data among the distinct organizations or governing bodies who have similar data and need to exchange. The tool has been developed with SOA approach in mind. The large amount of data may involve a distribution of data as in DDB. The data in DDB could be kept vertically or horizontally fragmented. All these are the requirements for a DDB administration tool for DDB which have been implemented in HDMT. The HDMT tool proves the heterogeneous data migration process for DDB and can be tested and implemented further for practical benefits.

4.2 HDMT Purpose

The main purpose of HDMT is to demonstrate the migration of data between distinct heterogeneous DDB servers where the data can be selected as to the most detailed column and record level for migration. This tool facilitates the DBA to migrate the data from a source to more than one DB name / Table name / Column names / Compatible types without writing any configuration code. It implements the data migration in terms of horizontal and vertical fragmentation which is an important feature of DDBs. This facility is different from the existing DBA tools used for data migration by the DBAs and no middle entity like excel is required. It is also different from existing ETL tools as it requires data transformations with respect to the data type only. Also, the fact that ETL tools are existing for the purpose of data warehouse and mostly are scheduled to run at given time, which is different than the HDMT. In order to migrate the data in case of DDB, the HDMT requires few inputs which have to be given to get the desired results of data migration. The HDMT user interface is as shown in Figure 4.5 and Figure 4.6.

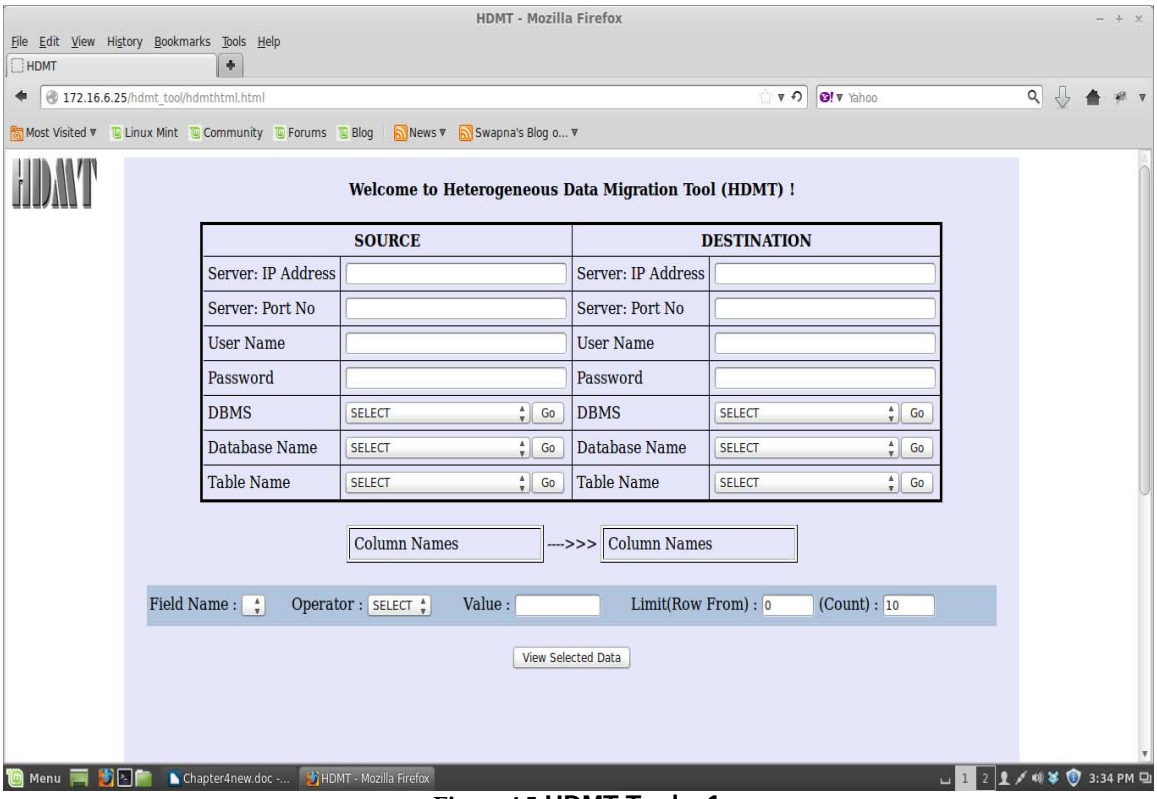


Figure 4.5 HDMT Tool – 1

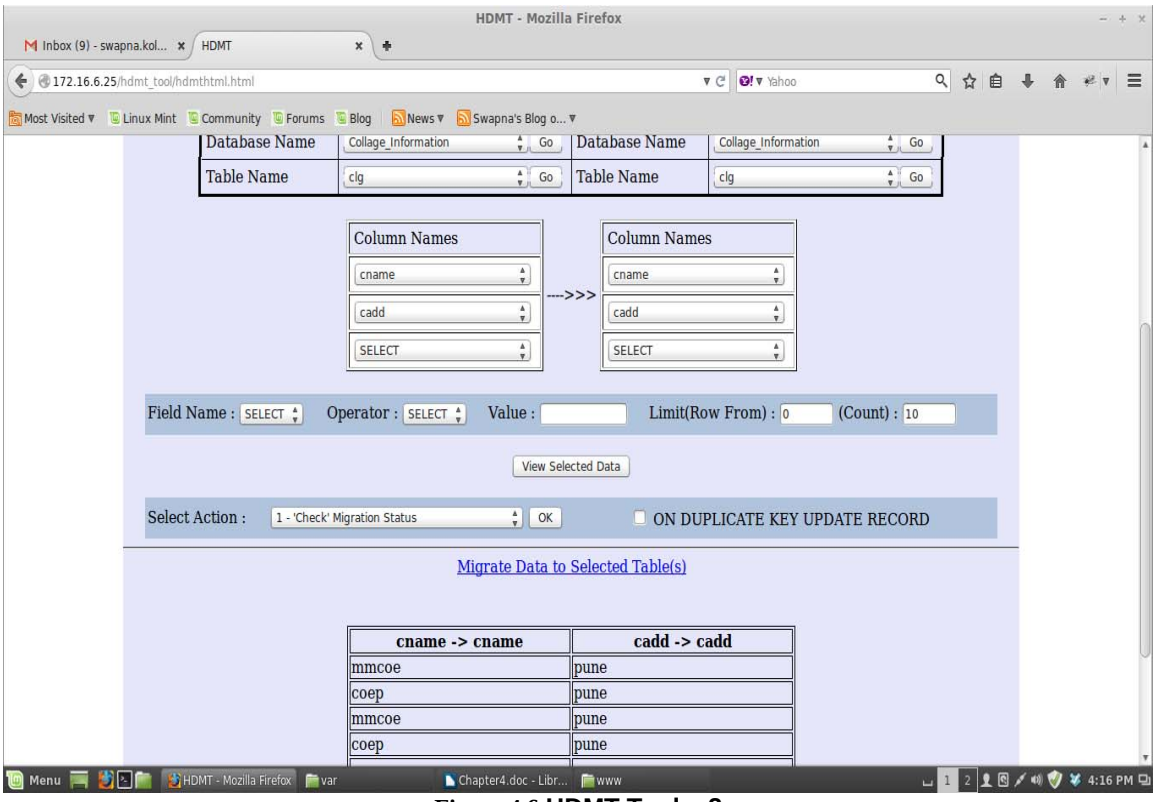


Figure 4.6 HDMT Tool – 2

4.3 Mathematical Model for HDMT

The HDMT is developed for the purpose of demonstrating heterogeneous data transfer in a DDB environment. It is based on the core functionality of selecting the desired data from source (so as to define our focus for horizontal fragmentation), transforming the selected data according to compatible data types (this is needed due to the data migration between heterogeneous databases) and finally migrating it to selected destination table(s) (in case of vertical fragmentation done in DDB). Accordingly, the mathematical model based on relational algebra expression is shown for the three functionality or processes as given below.

4.3.1 Selection of source data

Let us suppose that,

T = *table name*

D = *database name*

S = *database server name*

c = *condition*

CL = *column list or attribute list*

S is of a given RDBMS such as MySQL / PostgreSQL. S has many databases denoted by D. A database has many tables denoted by T. Hence, $T \subset D \subset S$ is true.

Let A be the sum of all data D_i (*0 to n; where n is the number of records*) belonging to the set of selected attributes (denoted by CL) and tuples (denoted by the condition c) of T which is a subset of D which is a subset of S.

$$A = \sum_{i=0}^{n} D_i \in (\prod_{CL} \sigma_C(T \subset D \subset S)) \quad \text{.....} \quad \textbf{Equation 4.1}$$

Let B be the sum of all data D_j (*1 to m; where m is the number of records*) belonging to the set of selected table name, T of database D which is a subset of server S.

$$B = \sum_{j=1}^{m} D_j \in \prod_{T} (D \subset S) \quad \text{.....} \quad \textbf{Equation 4.2}$$

Let C be the sum of all data (*1 to l; where l is the number of records*) belonging to the set of selected database D of server S.

$$C = \sum_{k=1}^{l} D_k \in \prod_{D}(S) \quad \text{..... Equation 4. 3}$$

Then for the entire data that can be accessed through A, B and C, the following is true.

$$A \subset B \subset C \quad \text{..... Equation 4. 4}$$

4.3.2 Transformation

This process requires transforming the selected data according to compatible data types so as to evaluate whether such compatibility on data type is possible or not. If it is possible then user can proceed to next step discussed in 4.3.3 else user cannot proceed to next step mentioned in 4.3.3 with appropriate message and the possible reason of incompatible data type conversion. In case of compatible data type conversion it is onto the user to check whether to proceed with the migration. So user has to check that unnecessary truncation of data does not take place even if data type is compatible. This is to say that small int stored as big int is possible but vice versa may not be appropriate due to data truncation. Hence, compatibility has been given for similar data types as shown below.

a) Transformation for Integer data type

Various data types for storing the integer values are mapped to a single key ie 0 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} int \\ integer \\ smallint \\ bigint \end{matrix}\right\} \rightarrow key"0"$$

b) Transformation for Character data type

Various data types for storing the character values are mapped to a single key ie 1 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} char \\ varchar \\ blob \\ text \end{matrix}\right\} \rightarrow key"1"$$

c) Transformation for Date data type

Various data types for storing the date values are mapped to a single key ie 2 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} date \\ datetime \end{matrix}\right\} \rightarrow key"2"$$

d) Transformation for Integer, Int and Real to other data type

The Integer, Int and Real data type on the source side can be saved as string or text data type on the destination side. This transformation does not lead to loss of data. It may require type casting in future for the defined mathematical or other operations. The mapping function is as shown below.

$$integer \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases} \quad int \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases} \quad real \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases}$$

e) Transformation for Date to other data type

The Date data type on the source side can be saved as string or text data type on the destination side. This transformation may not lead to loss of data if the container has the required memory to hold it. It may require type casting in future for the defined mathematical or other operations. The mapping function is as shown below

$$date \rightarrow \begin{cases} string \\ varchar \\ text \\ varchar2 \end{cases}$$

4.3.3 Insertion of data in DDB environment

This is the final process of migration of the selected data obtained from the source to the destination table(s). I have shown the use of more than one table to indicate a DDB environment. The source data gets migrated to more than one table as shown in the Figure 4.7.

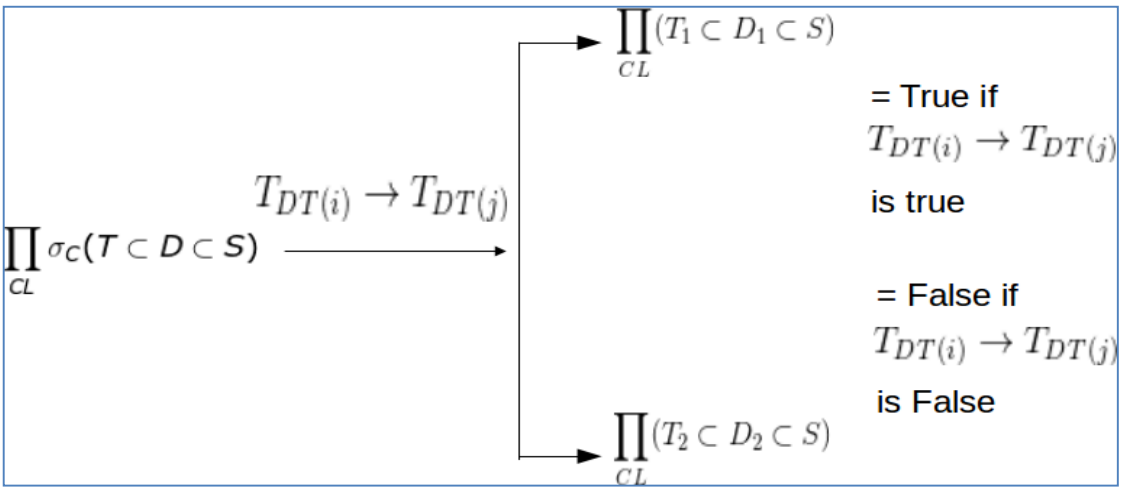


Figure 4.7 Data Migration in DDB environment

As shown in the Figure 4.7, the list of selected attributes from the table and filtered using a condition c, undergo a data transformation from DT (i) to DT (j) for as many tables selected from the database of a given server where DT is the data transformation.

4.4 Situations for HDMT implementation

Various existing DBA tools are used by DBAs for different purposes. The HDMT focuses on the maximum level of details for data migration under the situation that DBAs are able to share some minimum amount of connection information. They need to allow for data communication from the IP address of the tool server. Also, this tool is different from other migration tool in a way that the DBAs can exactly select the data attributes to be migrated and also that it requires ETL operation to the minimum. This service of data exchange provided by HDMT can be most appropriate in the following two situations:

4.4.1 Data Communication Between Inter Connected Organization

When two or more independent organizations but working towards the same benefit, maintain the data and which can be shared, then the HDMT can be used. For example, the colleges are required to send the student data to governing body and also other departments or cells which provide financial relief to the students. In such a situation, the governing body can directly migrate the authenticated student data to the concerned departments or cell or vice versa. The same situation exists for examination data being migrated directly by examination department to the governing body.

4.4.2 Maximum attributes of entities are similar

The HDMT can be used to communicate maximum possible non key attributes for data migration and for further processing. Under this situation too, the appropriate data attributes can be selected for data migration. For example, the examination related performance details of a student can be directly taken from the DB of the concerned institute authority and migrated to the DB of governing body. In this case, only a part of the student information that is related to the examination gets migrated to the governing body DB server.

4.5 HDMT Features

In order to handle the above two situations, HDMT provides the following features.

a) Connection to remote database
b) Pre - Assessing the DBA action
c) Data Type Compatibility Checking
d) Compatible Data Type Conversion
e) Data Migration Report (in some cases it may vary from Pre–Assessment report which may be a result of constraints at the destination side)

Additionally, HDMT also allows for the selection of subset of data depending on an attribute to implement horizontal fragmentation and the selection of multiple destination tables to implement the vertical fragmentation. In order to

implement the above features, HDMT has the following user interface details and users details.

4.5.1 HDMT User Interface

i) Source – The source is the DB server at the college (or company branch) end from where information will be sent to the governing bodies (or company).

ii) Destination – The destination is the DB server at the governing body side (or company). In case of customized tool (as mentioned in 4.7.2), this location will have the connection parameters already specified. I have implemented the tool considering a general purpose tool.

4.5.2 HDMT Users

HDMT can be used in general, by anybody wanting to migrate the data. Online availability of the service also means that internet connection should be available. But specifically, the tool is useful for people working with DBs.

i) DBA – This user is the person at the college end (or company) who is looking after the college level software system data security and maintenance. This user has to provide necessary database login credentials in order to connect to the source (known to the college) and destination (provided by governing body) database servers. The DBA is involved only in the data uploading and does not have any other privilege on the destination side.

ii) GDBA – This user could be a DBA at the governing body side. He is responsible for the data security and maintenance of software system at the governing body side. There is large amount of data involved in such systems as a large number of Indian educational institutes provide their data to them. This data may be distributed on various database servers. So, the GDBA needs to monitor the smooth functioning of all the database servers during any data transfer from various colleges. The GDBA has all privileges on the destination side DB servers and is involved in monitoring the DB servers. There are various monitoring tools available to check the DB performance of

DDB, so HDMT is implemented for DDB data migration where data is sent to multiple locations.

4.6 HDMT Specification

a) The specification of this tool basically involves the input of various DB connection parameters for the data exchange. This is possible only when the concerned organizations are agreeable to take appropriate security precautions and share necessary connection details.
b) The DBAs play a crucial role in ensuring security by defining appropriate user and their roles for the data migration process.
c) The DBAs also need to allow data communication from the IP address or server name on which the HDMT is hosted for usage.
d) HDMT provides facility to map the columns one to one and initiates the services for assessing the DBA action before executing it. The assessment report shows the possibility of data being successfully migrated to the destination as well as the possible reasons (data format, constraints on data or column) in case of unsuccessful migration. HDMT also allows converting the source data format to compatible destination data format wherever possible. After completion of action, HDMT stores and displays the summary of action executed.
For the data type compatibility checking and conversion, the tool refers to a php file containing associated array with the name – value pair mentioned in it.

4.7 HDMT Assumptions

HDMT has been developed to demonstrate the migration of data between heterogeneous data sources. It is based on the following assumptions.

1. The DBAs of involved organizations for the data communication are agreeable to the sharing of some basic information like IP address of database server, type of database server and user name and password. This information is required for the server connection which is finally required for data migration. The source gives the data to be inserted into the destination

tables. In order to handle security issues, the DBAs can work with specific user name having specific role and responsibility.

2. SELECT is a suggestion for the user to select a column or attribute name. It is NOT an attribute in itself. This imposes the restriction that column name cannot be SELECT.

3. The selection of columns is lateral ie source column *n* is mapped to destination column *n* and so on. There could be c1 columns on the source side and c2 columns on the destination side. The tool allows a user to select a subset of the columns from the source to be sent to the destination. For proper mapping, equal number of columns need to be selected from the source and destination side.

4.8 HDMT as a Utility

HDMT can be implemented by companies for their use and by themselves on their own servers or it can be implemented as a generalized tool on a server accessed by people under different policies. The generalized tool option will require the source and destination related information to be filled in entirely for the actual data communication to happen. The inputs required are the IP address of the database server or the server name, type of database, user name and password. After this, the other inputs like the database, tables and columns can be simply selected to be finally sent from source table to destination table. I have considered the generalized tool for the implementation.

4.9 HDMT Pr-requisites

HDMT tool is based on LAMP platform. It needs a LAMP server for its implementation. Also for implementing the heterogeneous database environment, I have considered MySQL (version MySQL 5.5) and PostgreSQL (version Postgresql-server9.1) database servers. These DB drivers have to be installed on the LAMP server for the migration tool related services to connect to various DB servers. Also, the DB servers should allow

for data communication from the migration tool server for which the DBA needs to apply certain environment settings as shown in Figure 4.8. The changes related to database settings have to be applied in the configuration file of the database server. The configuration file for PostgreSQL is postgresql.conf and that for MySQL is my.cnf. Both the files need to be modified for the purpose of allowing communication from the server where HDMT is deployed.

For Postgresql

File name : postgresql.conf

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost', '*' = all
                                # (change requires restart)
```

For MySQL

File name : my.cnf

```
# bind-address = 127.0.0.1  # comment this line out
```

Figure 4.8 DBA settings for allowing HDMT to access DB Server

For the development and implementation, I have selected the open source technologies for which the pre requisites are –

Development : Linux Apache Mysql PHP (LAMP), AJAX, JSON, JavaScript, JQUERY. All the technologies have their advantages in web development.

Deployment : Openshift PaaS by RedHat. This is a cloud server for PaaS.

I have also taken into account the cloud service (VPSDime at https://vpsdime.com/) for MySQL server and Postgresql server installed on CentOS Linux 6.5 and webmin version 1.700.

4.10 HDMT Architecture

The HDMT is a three tier architecture software tool which can be accessed from website as a service. It is SOA based and is hosted on LAMP server which can be on a cloud. The various databases that the tool accesses may also be hosted as part of cloud services. The HDMT architecture can be viewed from different aspects as mentioned below.

4.10.1 HDMT Software & Utility View

The DBAs are the users of this tool and can access HDMT through the internet as shown in the Figure 4.9. The DBA / GDBA connects to the data migration service, HDMT available online as a user. When used, the tool connects to the DB servers on proper authorization and authentication details of the source and destination. The figure also shows that the database servers may be maintained on different cloud services but communicate with each other through HDMT for the purpose of heterogeneous data migration in a DDB environment.

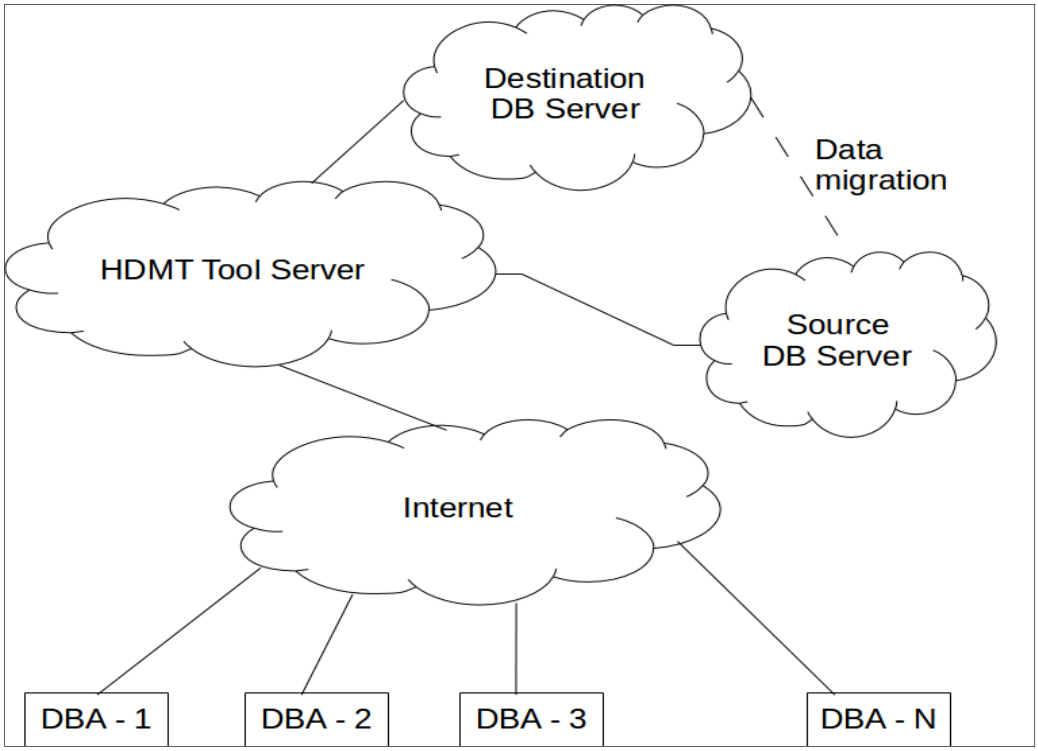


Figure 4.9 HDMT Tool, DB Servers and DBA as users

4.10.2 HDMT Services View

HDMT itself has been implemented as a service available to college DBAs for the purpose of transferring / migrating their data to governing body. The HDMT Tool consists of various services which are called as per the DBA action. These services may in turn call other sub services like transformation and migration so as to complete the DBA action as shown in Figure 4.10. The figure also shows the data migration between heterogeneous database servers in a DDB environment. The DB servers may be implemented and maintained on different cloud servers. The DBA / GDBA can carry out the data migration through HDMT which is itself available online as a service.

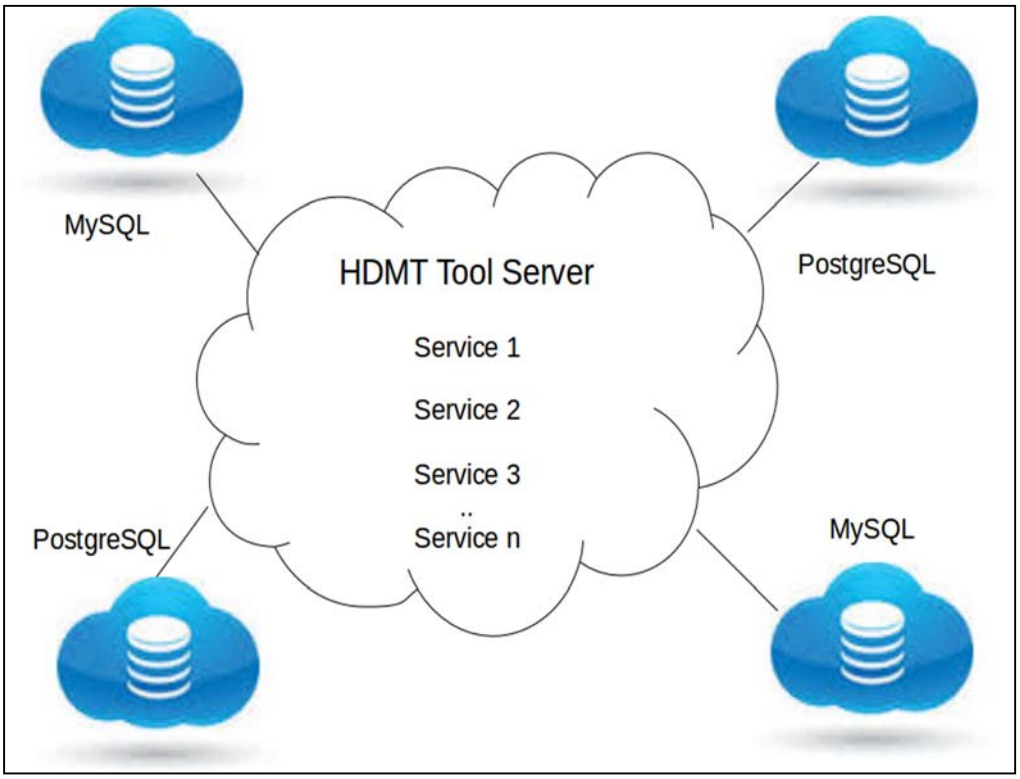


Figure 4.10 HDMT Tool Services

4.10.3 HDMT Three Tier Architecture View

From the point of view of architecture, the HDMT tool can be represented as shown in Figure 4.11. There are a few services shown in the LAMP servers.

These services may in turn call other sub services. I have attached the sample code at the end for reference. The DB servers shown in the Figure 4.11 can be on a LAN or cloud. They must allow data communication to happen from the HDMT tool server. The three tier architecture comprises of the client which works with the presentation layer, the HDMT server which provides the necessary business logic and interface of the data migration tool and the data repository which is used by various database servers to communicate for the purpose of data migration through the HDMT.

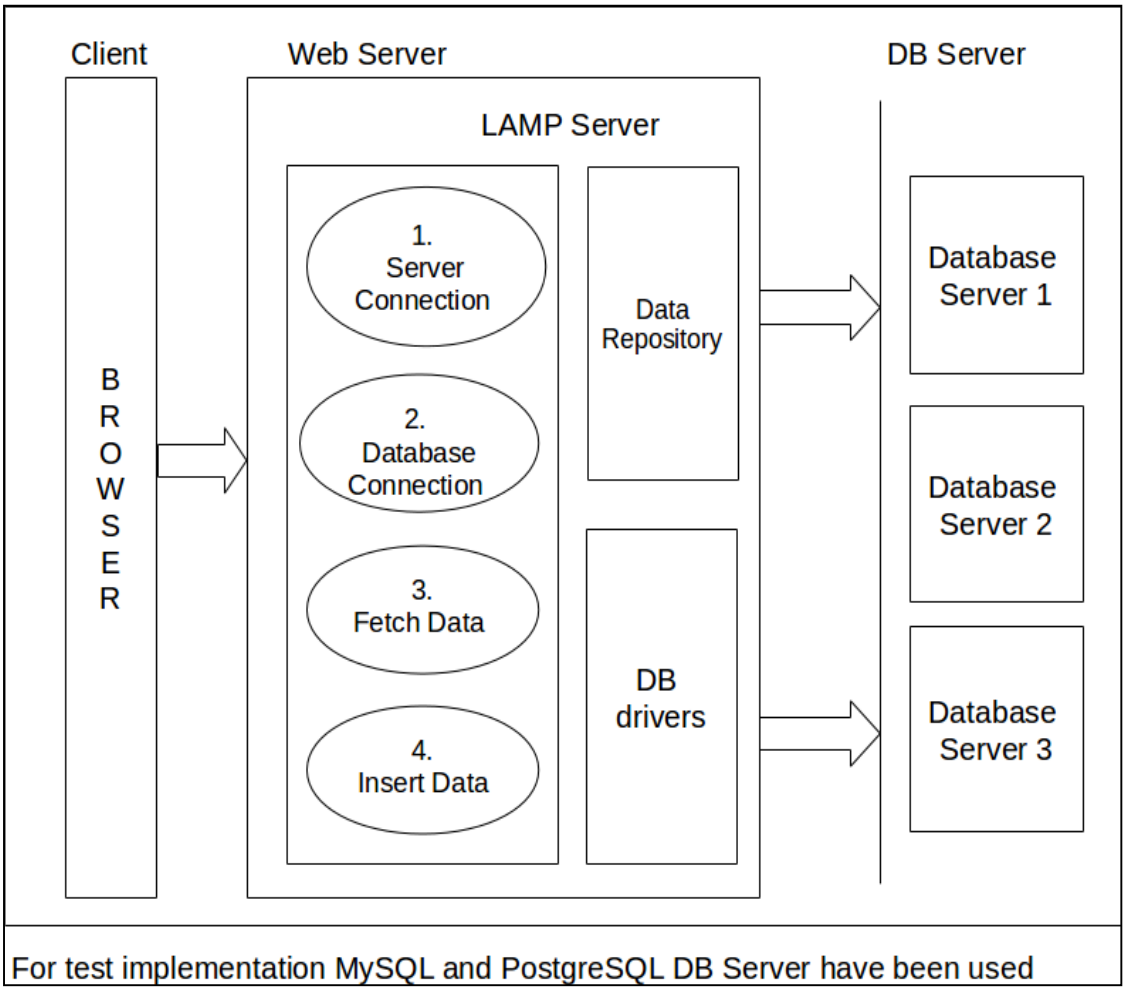


Figure 4.11 Three Tier Architecture based HDMT Tool

4.10.4 HDMT SOA View

The use of cloud services architecture for the tool development and implementation considering the hardware to the application layer can be as

shown in Figure 4.12. Various services like infrastructure as a service, platform as a service, software as a service etc are involved for the development and implementation of the HDMT. Also, HDMT is plugged in with cloud services. The role and responsibility of the data center is maximum at the hardware layer and decreases as we move up the layers. This has been shown with the help of the triangles (standing and upside down).

There are six layers in the tool architecture from the point of view of SOA. The first layer is the hardware layer that is provided by the data center. For the demonstration of HDMT, I have implemented using a cloud service on a rental basis. The data center (openshift PaaS provided by RedHat) provides facility related to the memory and processing units. The second layer is the platform layer which provides the necessary OS and drivers. The environment provided by the data center is as per the client demands. I have used an open source OS (CentOS). The third layer is the application development and deployment layer where there is a need of servers, compilers or software necessary for the developer to work with for software development. The fourth layer is the security layer where anti-virus, firewall, encryption, authentication environment needs to be provided by the data center and developer. The fifth layer is the change management layer which is most useful for any software updations or software change by the developer. For SOA based software system, the focus is on small services to make up a large service. Managing changes in small services or functions which are complete and independent is possible without affecting the overall system functionality. The last layer is the application layer which is the user's layer for working with the system.

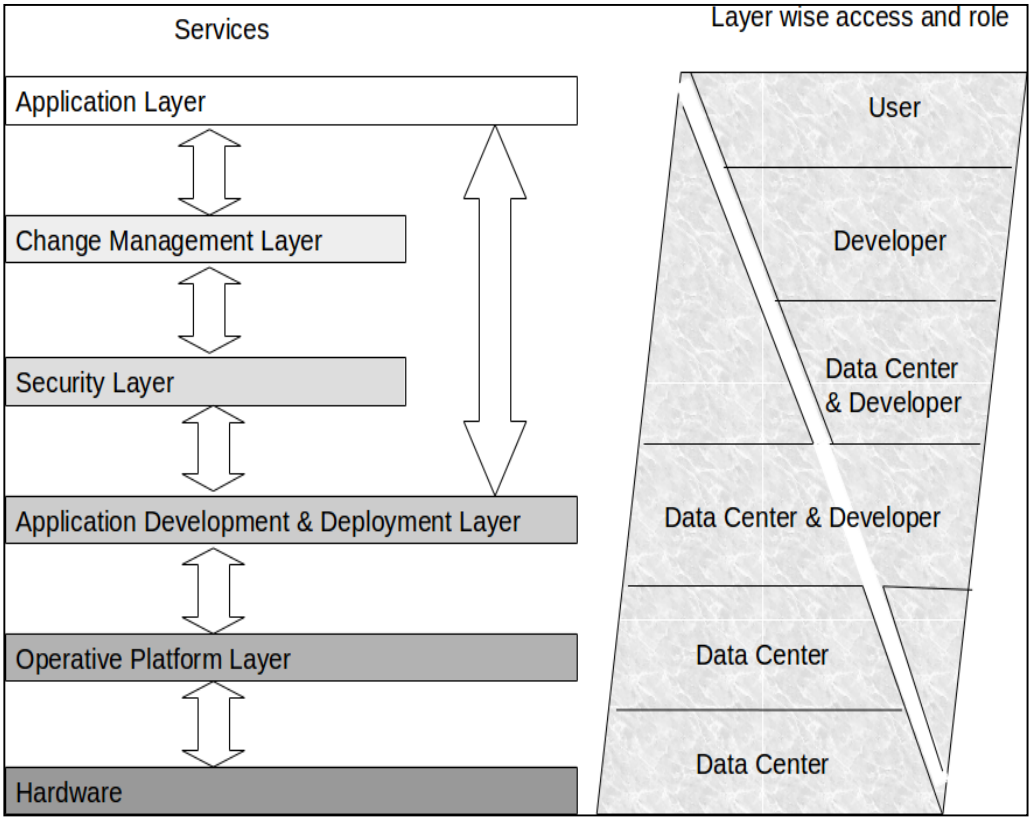


Figure 4.12 Cloud services based tool architecture

The facility provided by the data center depends on the agreement with the clients. The users work with the application layer at the top of services hierarchy as per their role. In HDMT the DBAs are the users of the tool. The DBAs work at the application layer and access HDMT as a service for the data migration in DDB environment. The DBAs need to allow for communication from the HDMT server and then provide the connection and authentication details of the source and destination database for the purpose of data migration.

The service approach for the deployment of HDMT makes the migration service available to all for the purpose of heterogeneous data migration in a DDB environment. The services related to the database also show that the actual data communication takes place at (or between) the data center(s) which is very fast as supported by the underlying hardware and network. All the available cloud services and the service approach, facilitates the job of the DBA in various ways as given below:

- **Location Independence** – The DBA can avail the HDMT service from any location through a web browser and internet connection
- **Migration in DDB environment** – The DBA / GDBA avail the HDMT service for migrating the selected data to multiple destination tables so as to store the data as per the horizontal or vertical fragmentation
- **Heterogeneity** – The DBA is not burdened with the DBMS and can carry out heterogeneous data migration in a DDB environment
- **Reliability** – The DBA is made aware of the pre assessment report before proceeding for the data migration which ensures reliability of the operation

The developer works with the change management and application development and deployment layer as a user of the facilities that are provided by the data center. For example, the data center provides a PHP environment for the developer to use it for software development. The developer and data center play an important role in the security of the system. Few services related to security aspects are provided by the data center whereas the developer also handles the security aspect related to authentication or encryption etc. The hardware and operative platform service is provided and maintained by the data center.

4.11 HDMT Implementation

HDMT is a open source tool and basically requires the LAMP server. The LAMP environment can be implemented on the following depending on the company's needs. For demonstration, I have implemented the LAMP server using cloud services.

4.11.1 Company / Organization Server

This requires a server on which the LAMP environment can be set up. In case of company server implementation, the responsibility of setting up and maintaining the server is with the company. The HDMT tool can be utilized only in case of internal company operations. It will not be accessible online to others outside the company. The security of HDMT is as provided by the web server / LAMP or company policy and standards.

4.11.2 Cloud Server

In general, the HDMT can be implemented on a cloud and accessed through the internet. This service can be accessed as per the terms and policies of the implementer. The security of the HDMT is as provided by the cloud infrastructure or data center.

4.12 HDMT Services

I have worked upon the identification and implementation of following services for SOA based HDMT functionality and its development. The services have been developed using PHP, Javascript, Jquery, AJAX and JSON for client and server side processing. For rich user experience, AJAX technology has been implemented along with JSON that is used primarily to transmit data between a server and web application, as a layer above XML.

a) **Authentication** – I have implemented this functionality as a service to authenticate the credentials based on the user role. The GDBA is the administrator of the tool and will have the highest responsibility. This concept is implemented for the purpose of data exchange between heterogeneous DBs and meta data exchange between web server and the database.

b) **Pre – Assessment service** – This service is implemented to read schema detail and check for possible obstacles that will occur during the execution of query for migration of data.

c) **Data Type Compatibility Checking** – This service shows the data type incompatibility in case of variation in data type between the mapped columns.

d) **Compatible Data Type Conversion** – this service is used to customize the insert query as per the compatibility expected by Destination Database by considering the same as on source database.

e) **Data Migration Reporting** – this service is implemented to show the status of the migration and the number of rows migrated successfully. In case of any database related constraint error, the report will reflect the same.

Continuing the discussion on features further, I have identified a few services and sub services for the implementation of the features. These services along with the user interface are as follows :-

4.12.1 Server Authentication & DB Name Display

This service takes parameters like the IP address, port number, user name and password. The service is implemented individually for distinct DB servers. The LAMP server must have necessary DB driver installed for the implementation and execution of the service. The installation of the drivers is done as per the server specifications.

This service gives the list of DB names as an output that is shown in the control used for displaying the DB names on successful authentication of the DB server. The output ie the list of DB names as shown in Figure 4.13, is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message as shown in Figure 4.14.

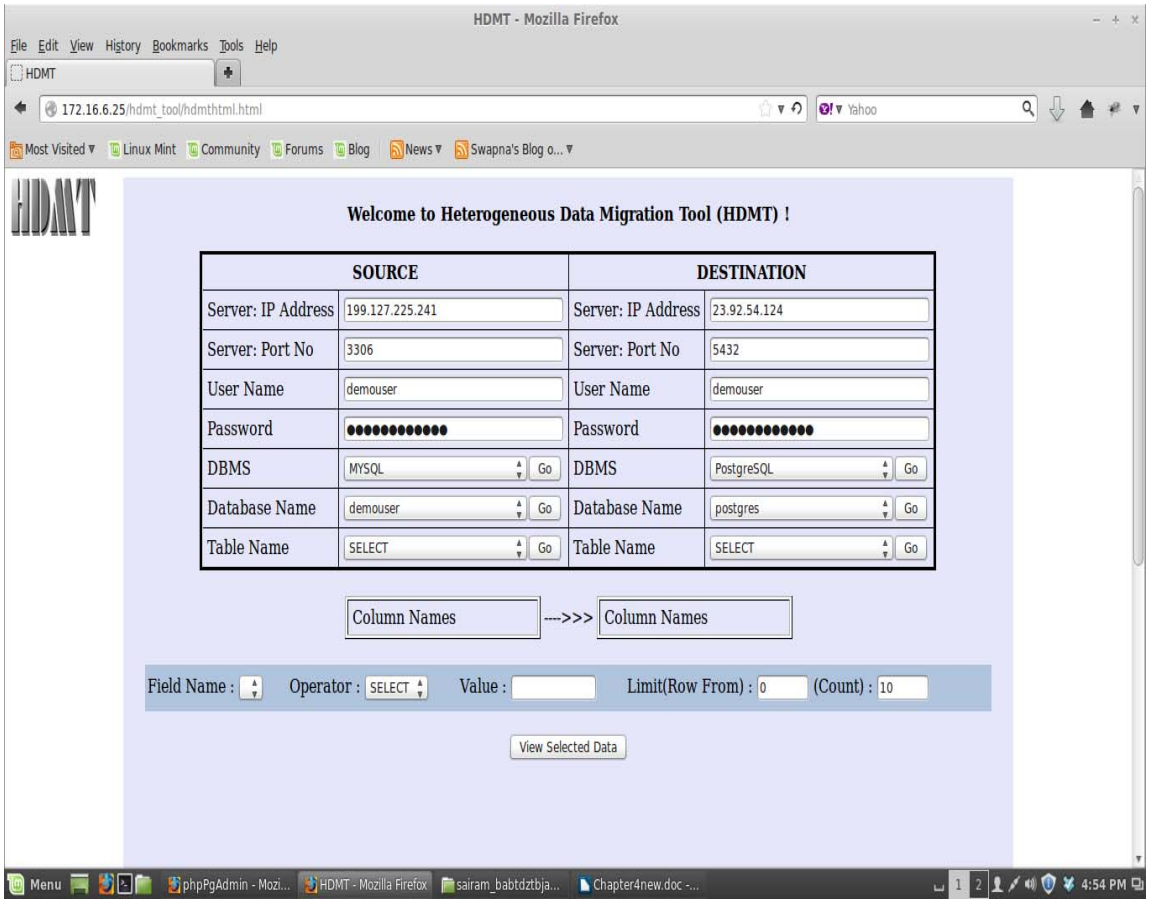


Figure 4.13 HDMT server connection and DB name display

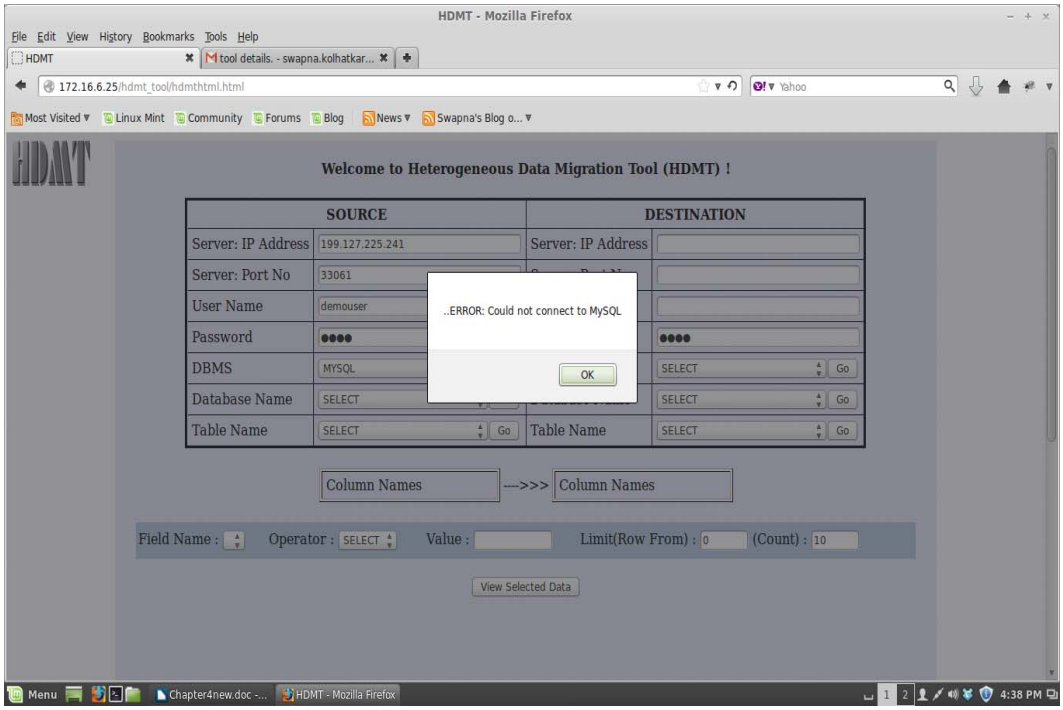


Figure 4.14 HDMT unsuccessful server connection

4.12.2 Database Authentication & Table Name Display

This services takes parameters like the IP address, port number, user name and password and DB name. The server side service is implemented individually for distinct DB servers. This service gives the list of table names as an output that is shown in the control used for displaying the tables on successful authentication of the DB server and DB. The output ie the list of table names, is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message.

As shown in the Figure 4.15, we are connecting to MySQL and Postgresql DB servers by giving the input details. The Figure also shows a demouser created for the purpose of demonstrating the working of the tool.

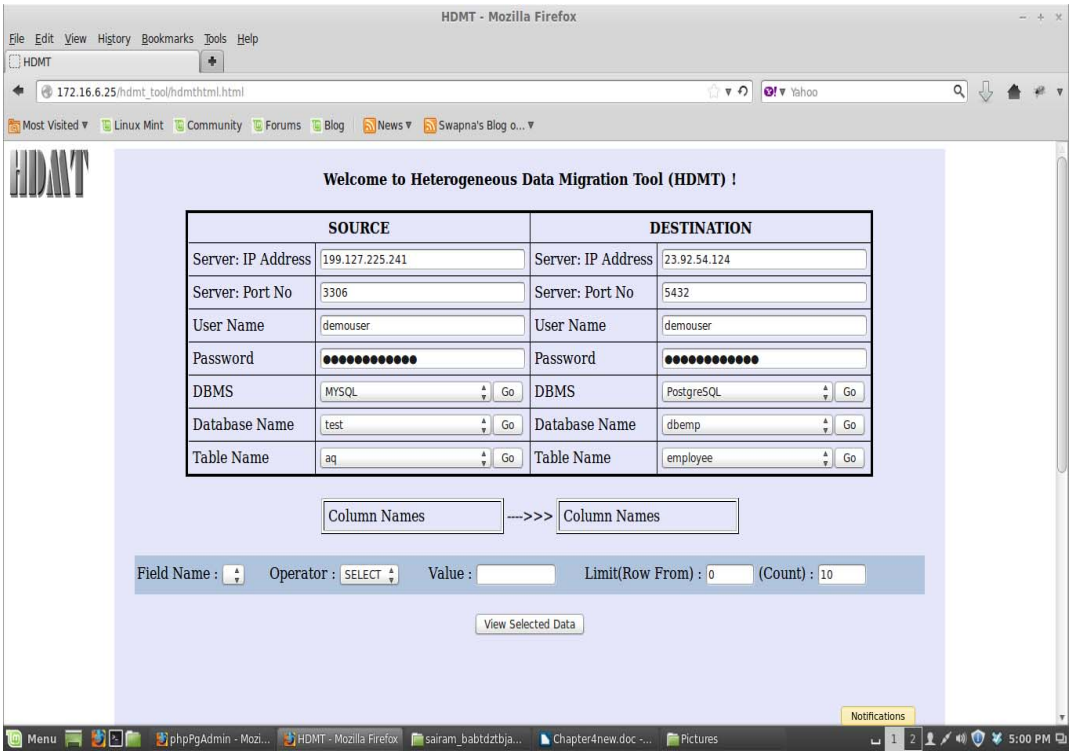


Figure 4.15 HDMT DB connection and table name display

4.12.3 Table Authentication & Column Name Display

This service takes parameters like the IP address, port number, user name and password, DB name and table name. The server side service is implemented individually for distinct DB servers. This service gives the list of column names as an output that is shown in the control used for displaying the columns on successful authentication of the DB server, DB and table as shown in Figure 4.16. The output ie the list of column names is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message.

The Figure 4.16 shows root as a user for accessing MySQL and Postgresql DB servers for the purpose of heterogeneous data migration. The column names can vary in number for which an independent functionality written in Javascript is called to display the appropriate number of columns. The function dynamically generates the html select control, populating each and every control with the list of column names for the user to select.

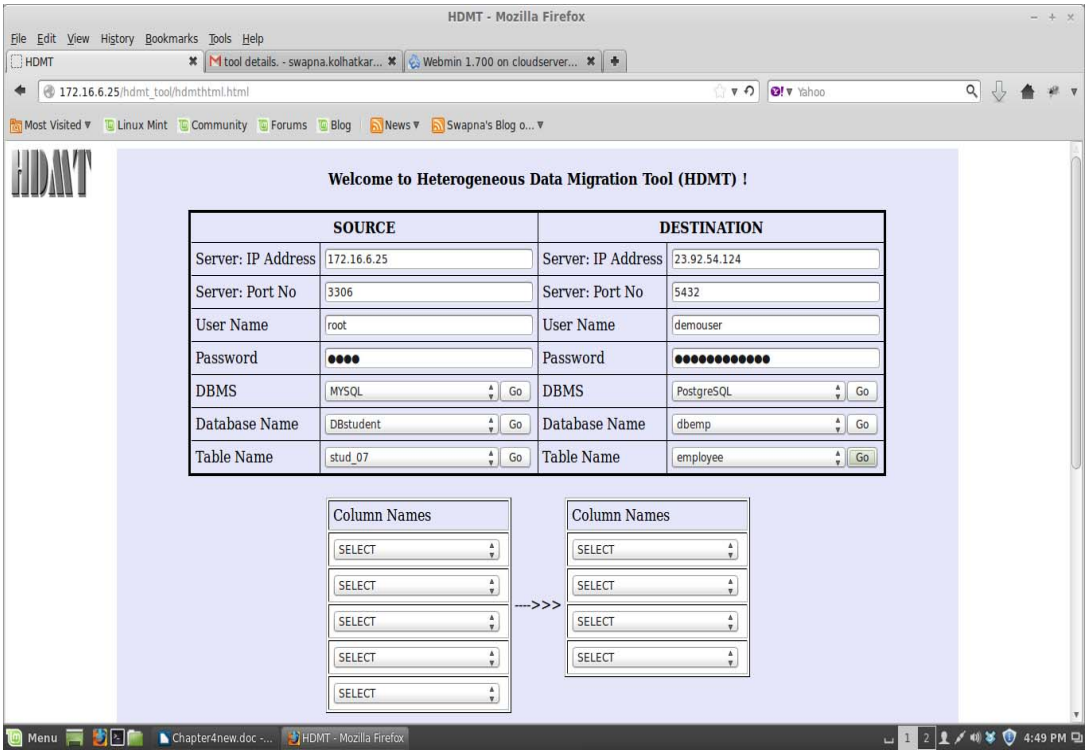


Figure 4.16 HDMT DB connection and column name display

4.12.4 Column Name Selection & Validation

This is a server side process. It composes the selected column names in csv format on the client side and sends the source and destination column name list for further validation. The server side function compares the number of columns on both sides and checks for duplication. In case of problems, appropriate messages are displayed. In case of unequal selection of the number of columns on the source and destination side, following message is shown as shown in Figure 4.17. This checking is done under the assumption mentioned previously for mapping columns.

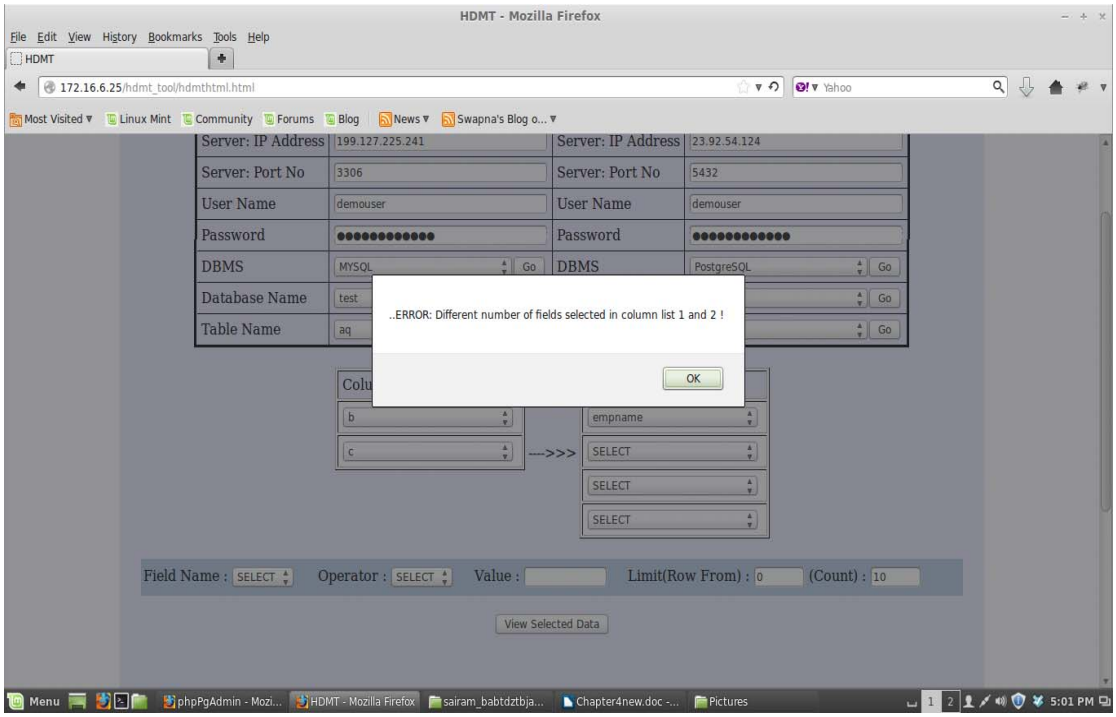


Figure 4.17 HDMT unequal number of column names error display

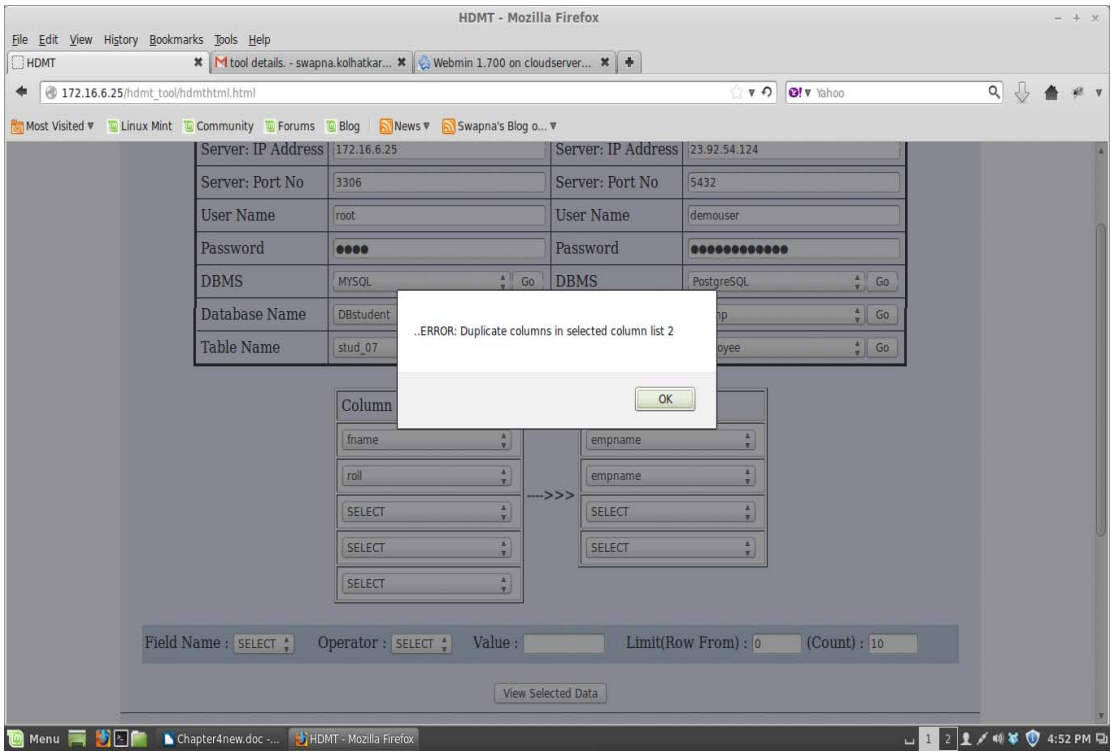


Figure 4.18 HDMT duplicate column selection error message

In case of duplicate column names being selected, the error message is displayed as shown in Figure 4.18. On selecting correct number of columns as shown in Figure 4.19, we proceed for further processing.

The column names may be different as the name difference is immaterial for the data migration process.

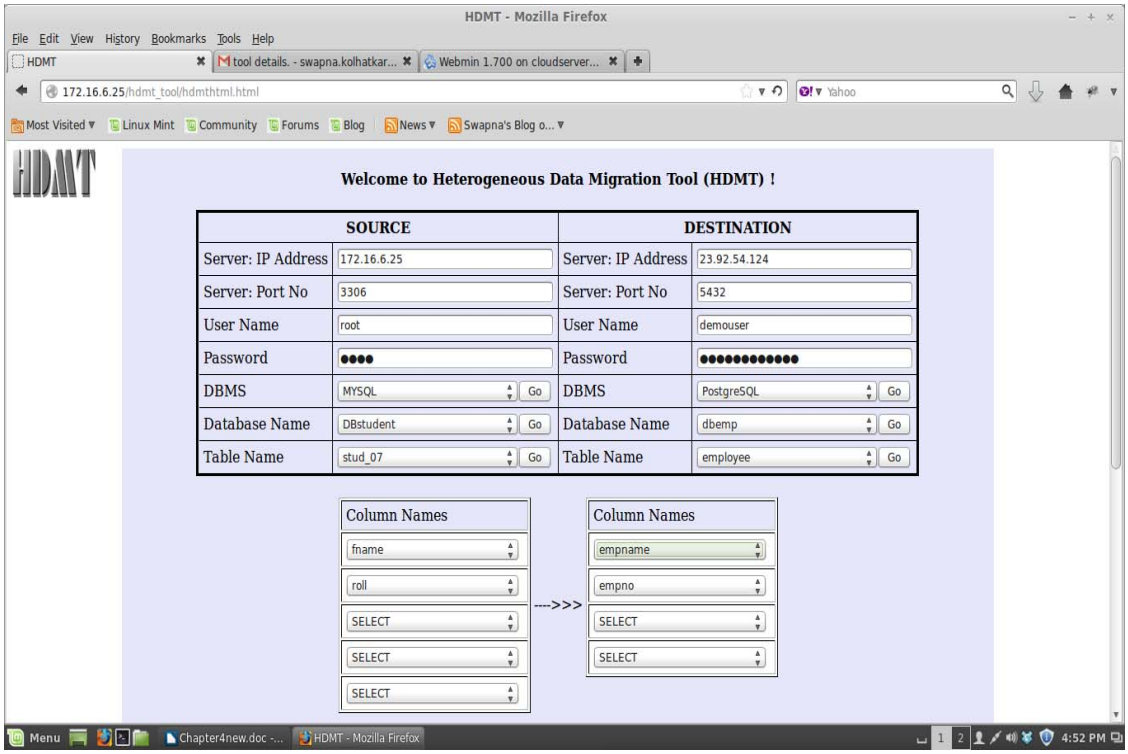


Figure 4.19 HDMT column names selection

4.12.5 Display Selected Data

This is a server side process called on the click of 'view selected data' button and it takes all the connection parameters as input and selects the records from the source side for displaying on the html table. The records sent to the client side are in JSON format which is handled by jquery and sent to the html table. Standard PHP functions have been used to display the data into a suitable format. Various sub services like fetching the data type of the attributes present in the DBA selected list of columns is also called for completing the process of data display. As shown in Figure 4.20, the number of data displayed at a time in the table is as mentioned by the DBA. Here, the

count mentioned is 10 but if kept blank then all the records from the source table is displayed in the web browser to the user. Simultaneously, the other interface related to the migration is made visible for the DBA to decide upon the next course of action.

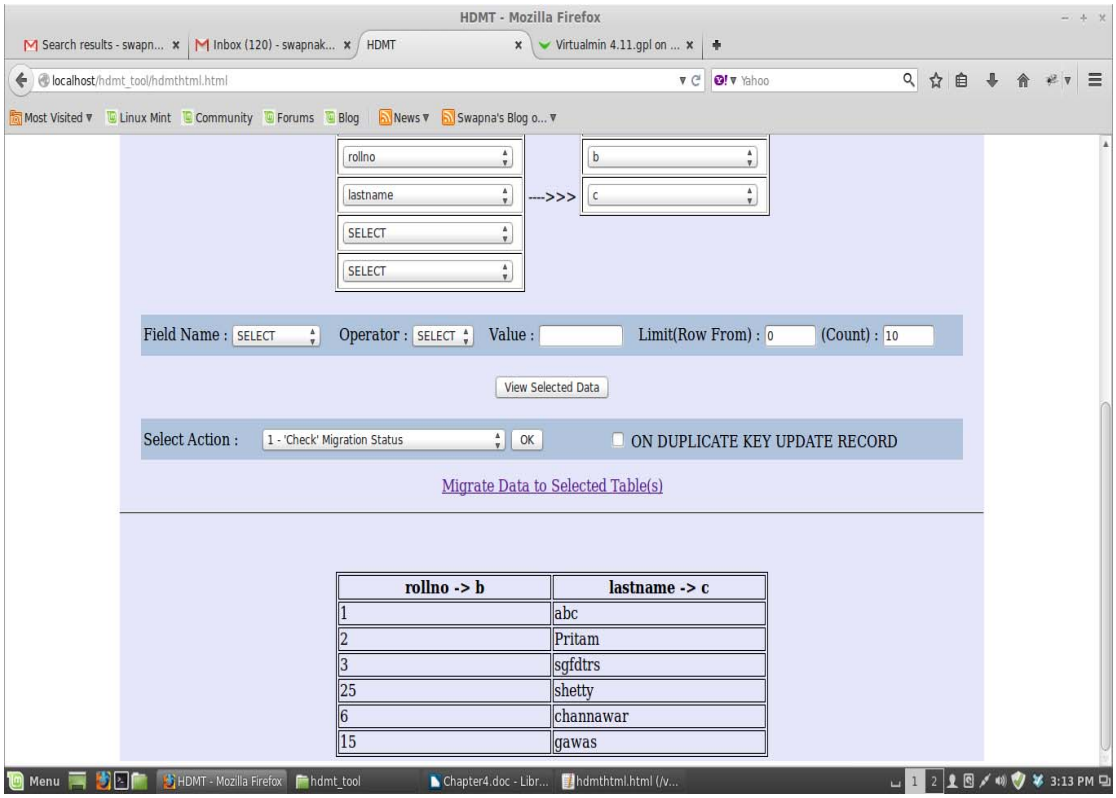


Figure 4.20 HDMT data display

4.12.6 Pr Assessing DBA Action

HDMT provides for some pre assessment of the DBA action ie migration. Before the migration process, the DBA has an option to avail a sub service for deleting any records from the html table by clicking on the record as shown in Figure 4.21. The deletion is done from the html table and not the DB table as shown in Figure 4.23 and Figure 4.24. This deletion may be required for filtering out certain records due to any foreseen problems. The usage of this facility is optional. If all displayed records are to be sent for migration, then deletion is not required. In case the DBA does not delete the record as shown in Figure 4.22, then it is highlighted for the DBA from the reference point of view. The highlighting is just for visibility. The HDMT is a facility that does not disturb the structure or content of the database.

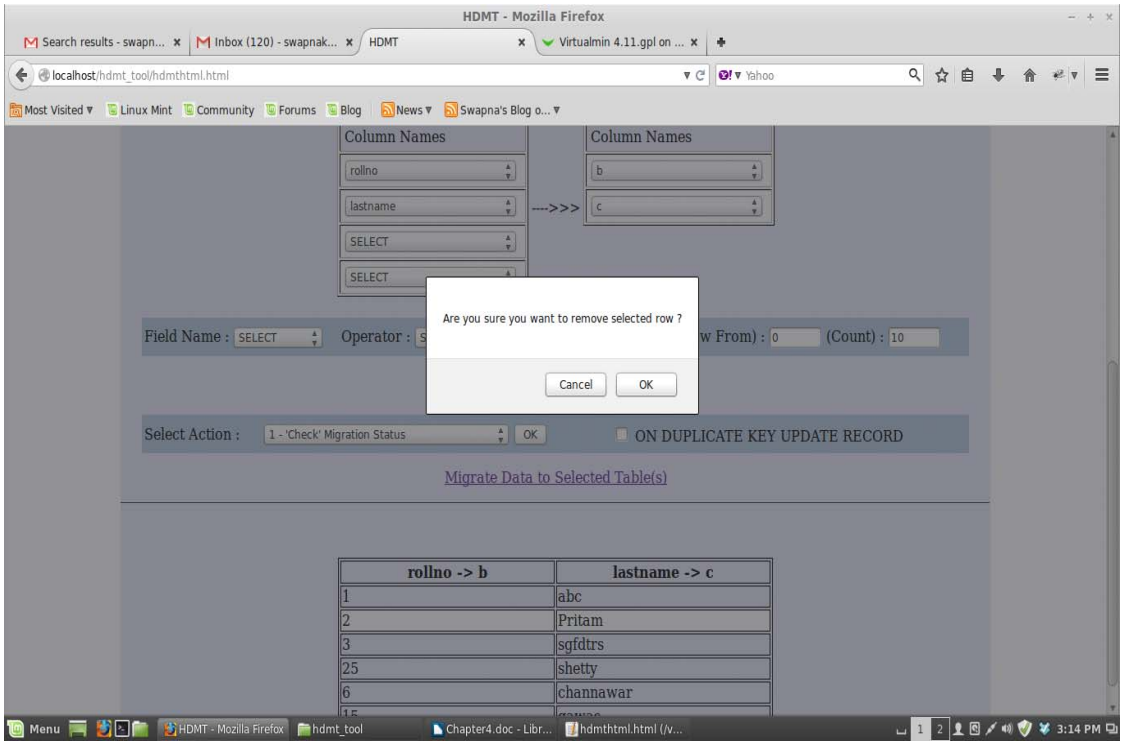


Figure 4.21 HDMT html data deletion facility

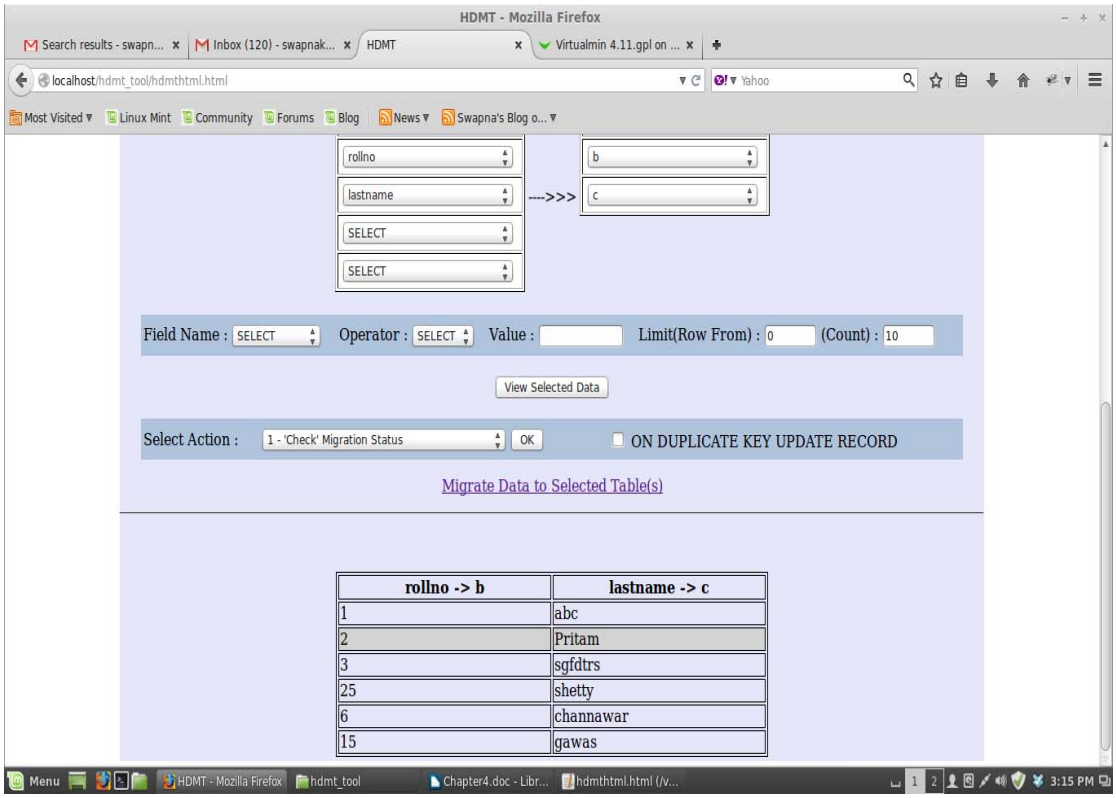


Figure 4.22 HDMT html data deletion facility (data not deleted on selection of 'Cancel')

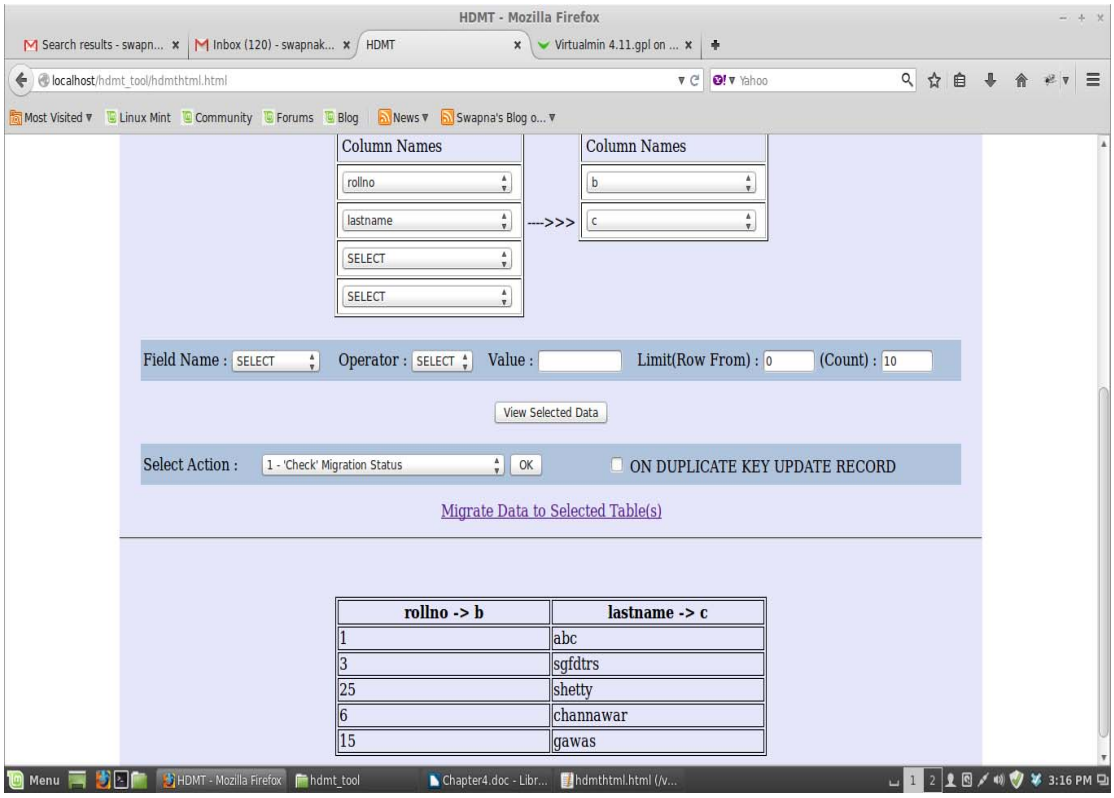


Figure 4.23 HDMT html data deletion facility (data deleted from html table)

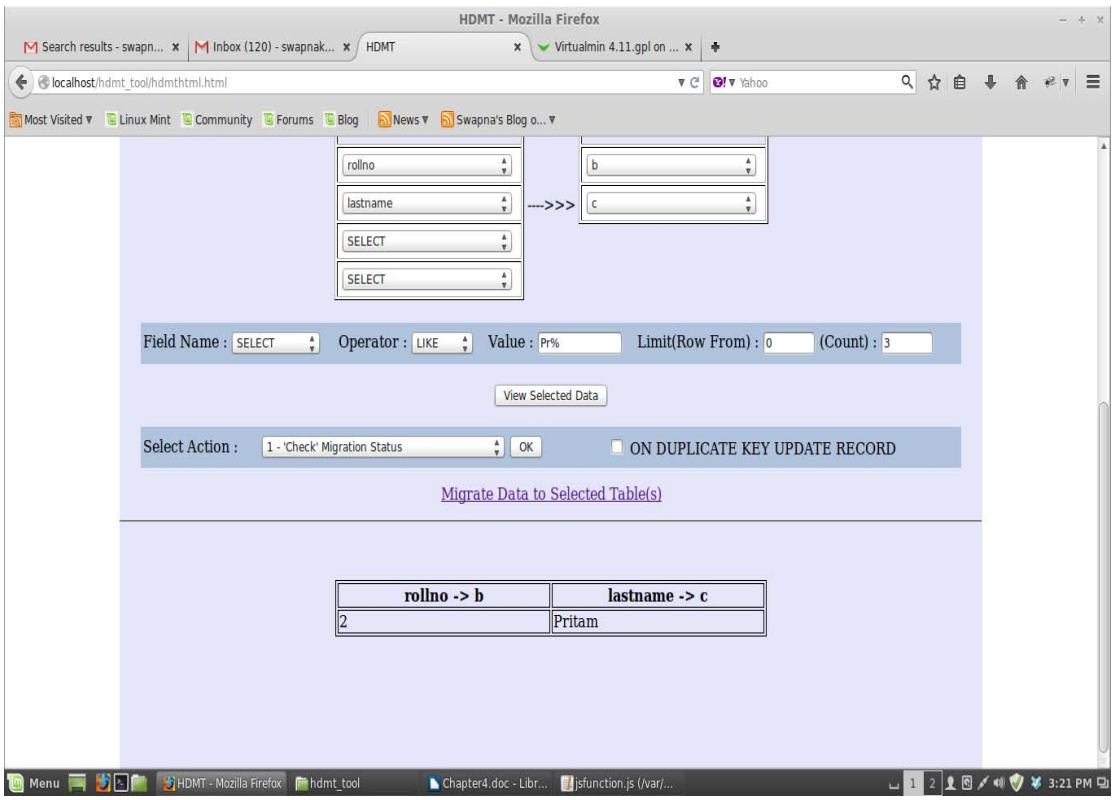


Figure 4.24 HDMT html data deletion facility (data not deleted from DB table)

4.12.7 Data Type Compatibility Checking

This is a server side process written in php which compares the data type of the mapped selected columns between the source and destination. The source and destination has different RDBMS ie MySQL and Postgresql to demonstrate the heterogeneous data migration process. The data type system of MySQL and Postgresql has its differences. These data types are interpreted by PHP to check the data type compatibility.

In case the data type compatibility check is as per the functionality provided by the tool, then the DBA can send the data for migration by selecting the check box for migration. The data type compatibility and convertibility report is as shown in Figure 4.25 and Figure 4.27. The Figures 26, 28 through 31 shows the facility to add other tables for migration in a queue or destroy the queue in case the migration is not to be done for the selected tables added to queue.

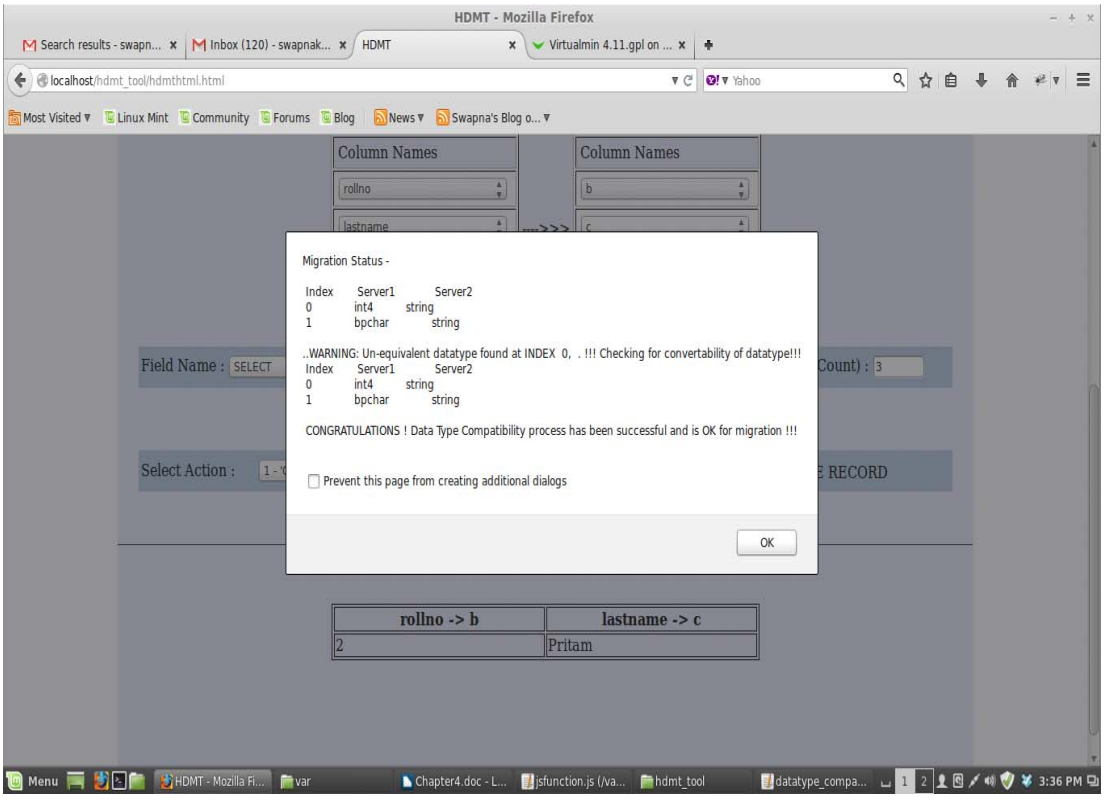


Figure 4.25 HDMT data type compatibility & convertibility checking

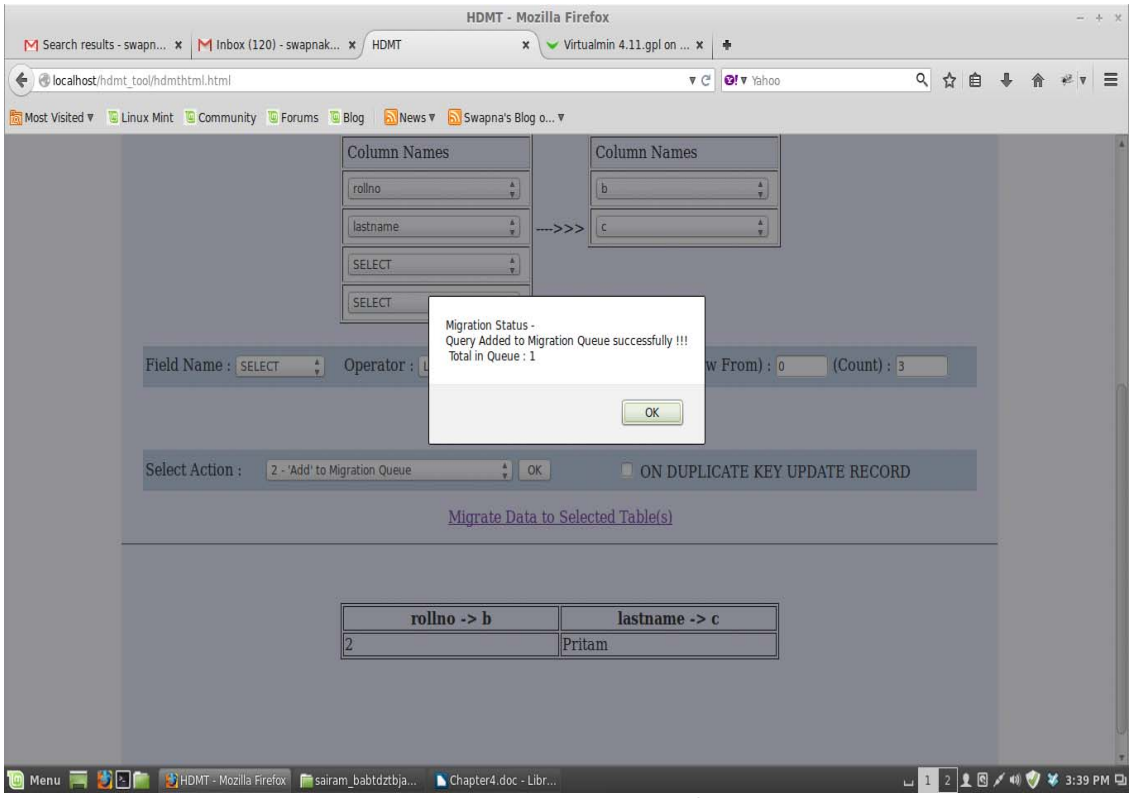


Figure 4.26 HDMT Add Table to queue for migration

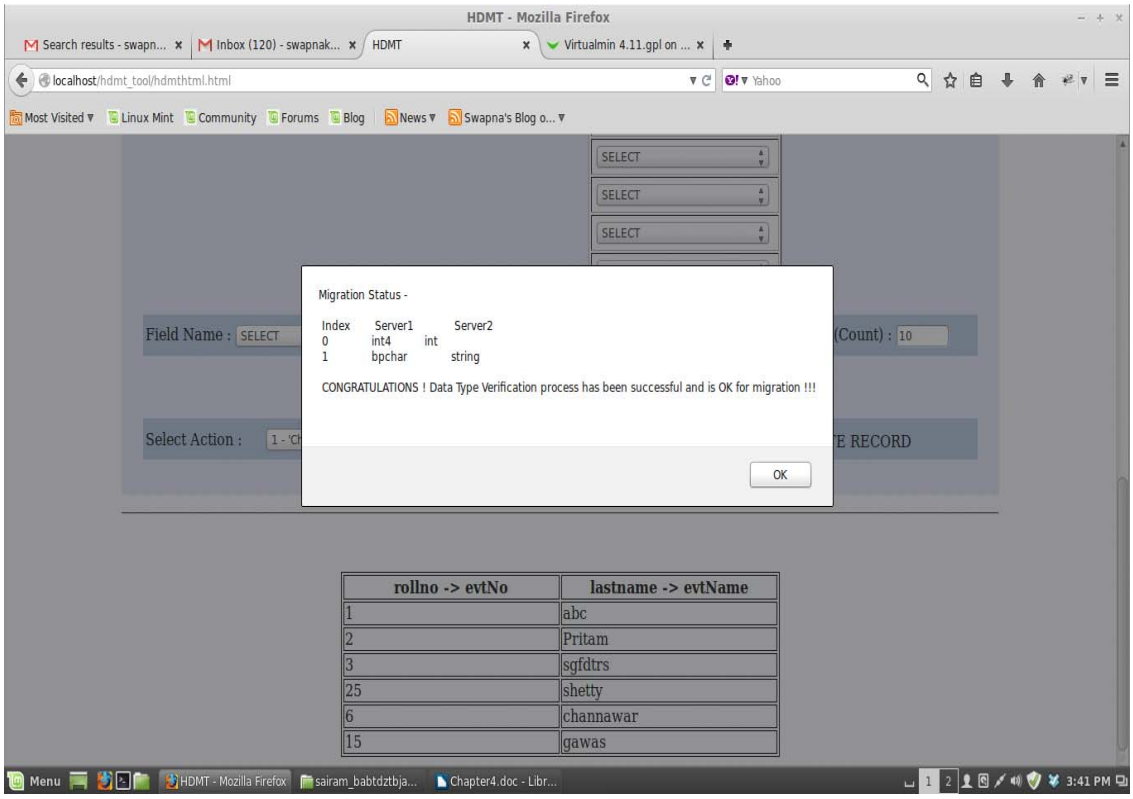


Figure 4.27 HDMT data type compatibility checking for another table

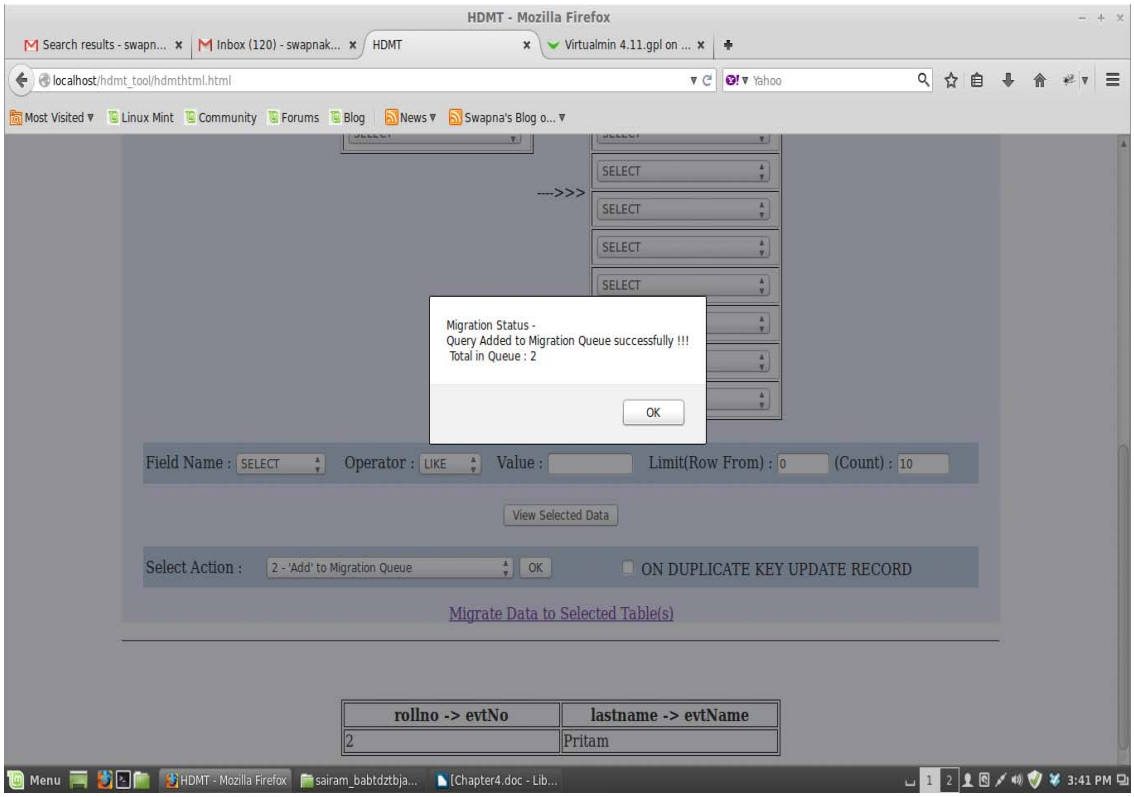


Figure 4.28 HDMT Add another table to queue for migration

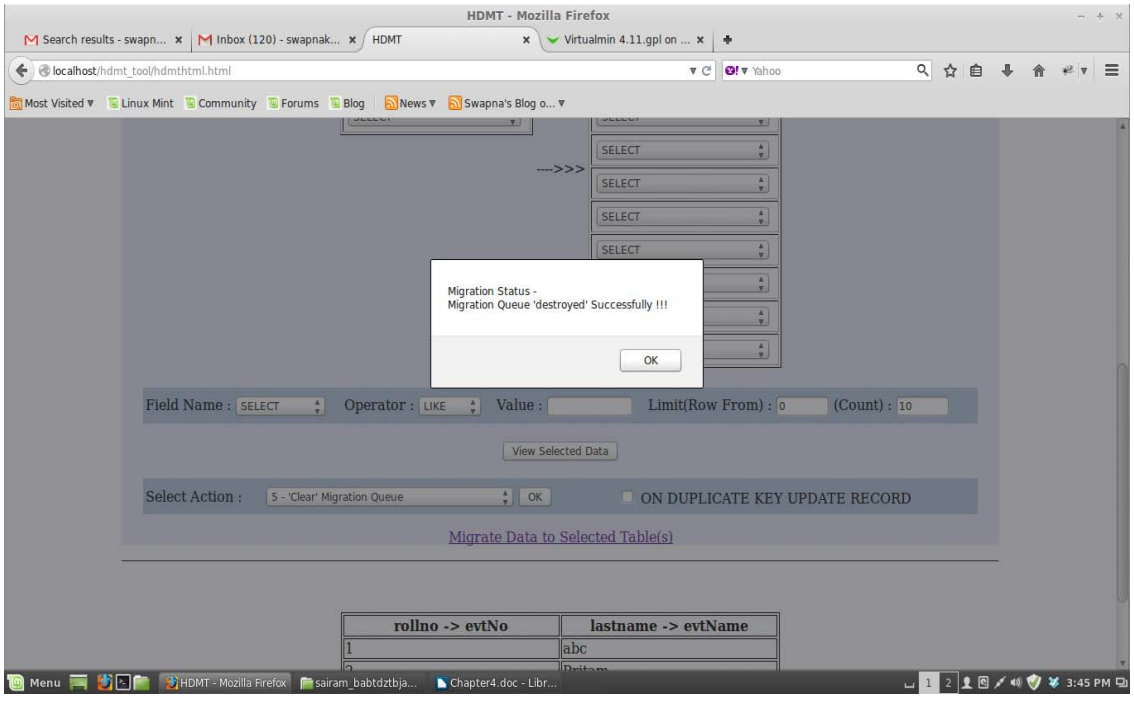


Figure 4.29 HDMT – Destroy the migration queue

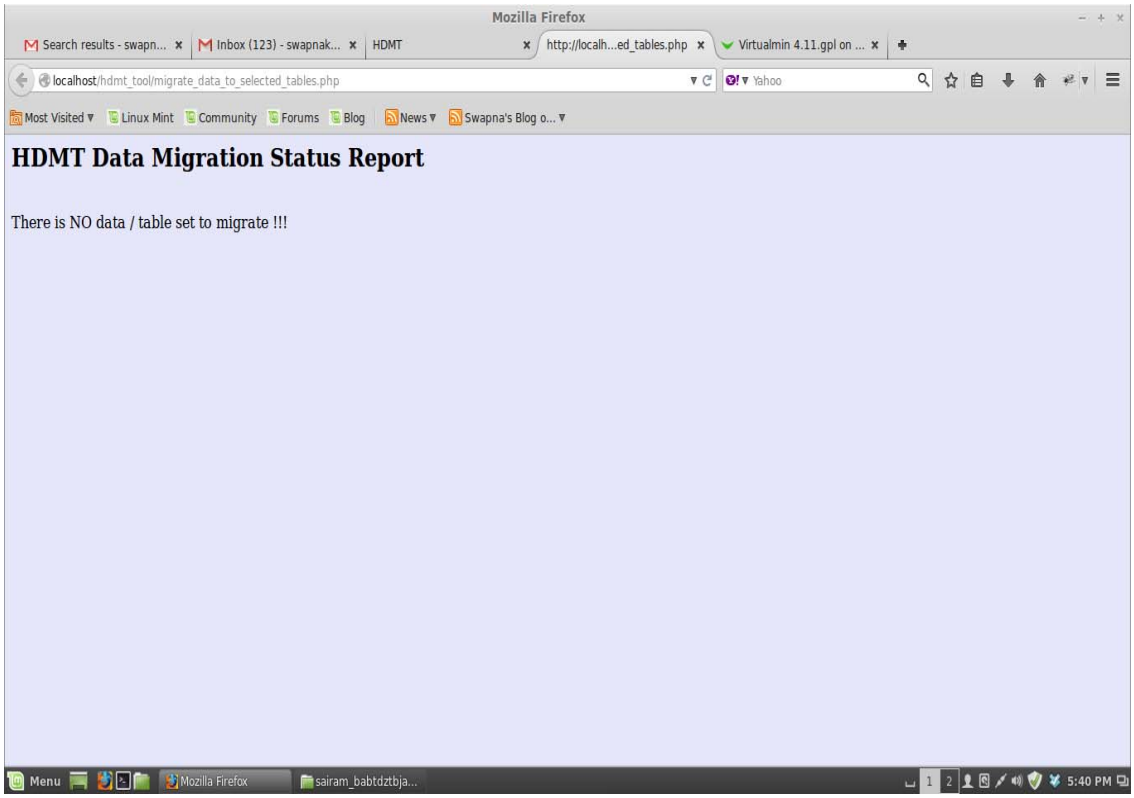


Figure 4.30 HDMT – Data Migration Status Report if no data to be migrated

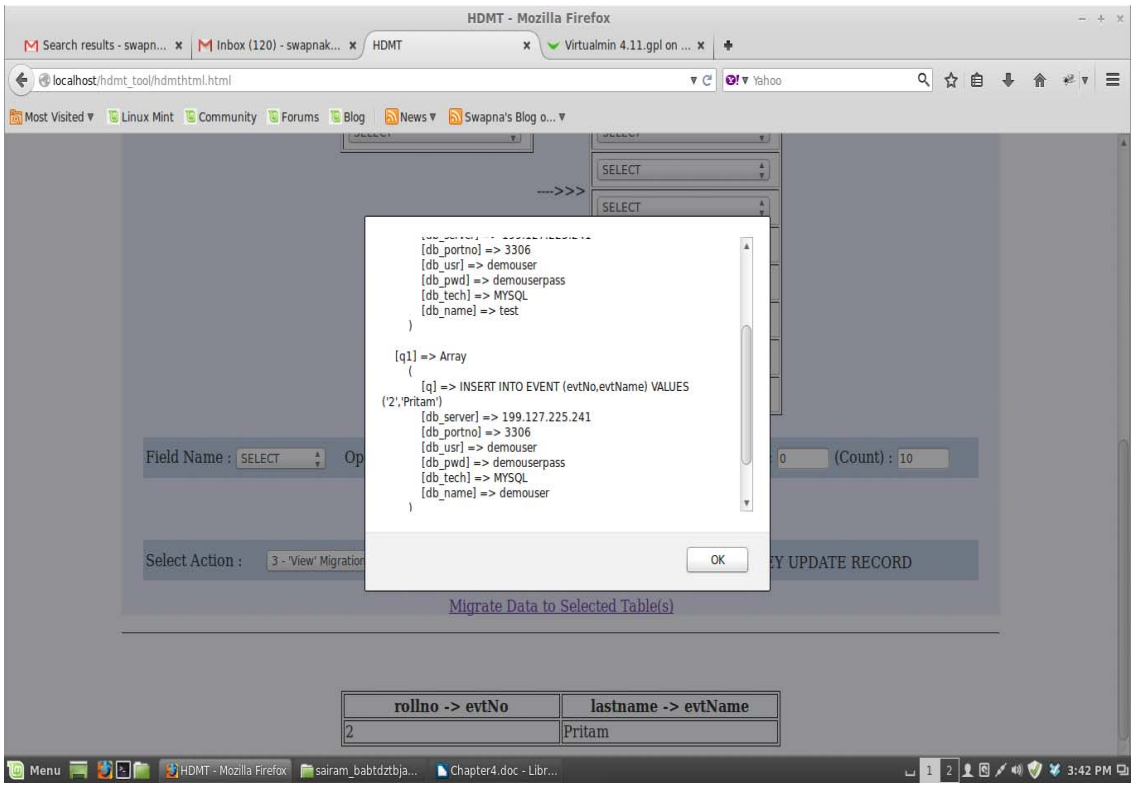


Figure 4.31 HDMT – View the migration queue

4.12.8 Compatible Data Type Conversion

The differences in data type are shown to the DBA. Though the data types are compatible, they may have certain data type differences which restrict the migration of data from source to destination for example small int, int, big int, integer etc. A homogeneous DB would have made the data migration simple due to the similarity in data type, but the same does not hold true for the heterogeneous DB.

The data type compatibility difference in such a heterogeneous environment has been handled with HDMT. It has been handled with the help of a data repository which can be modified appropriately as per the need to handle the wide range of data types, though care must be taken in identification of compatible (equivalent and convertible) data types. The following Figure 4.19 shows a sample code for handling the data type compatibility conversion.

The code uses the associative array of PHP in order to identify the value key pair for defining the compatible data type values as shown below.

```
/* This file is used for providing the data type compatibility information. All the compatible data types will be having the same value and data type as key ie data type name will be key and its value as integer value. This file has been included in hdmtinsertselect.php file for the check_datatype_compatibility() function. */
<?php
$arr_equivalent_datatype_set =
array("integer"=>"0","int"=>"0","number"=>"0","int2"=>"0","int4"=>"0",
"char"=>"1","varchar"=>"1","text"=>"1","string"=>"1","character"=>"1","blob"=>
"1","bpchar"=>"1",  "date"=>"2","datetime"=>"2","year"=>"2", "real"=>"3"
 );
$arr_convertible_datatype_set = array("integer"=>array ("char","string",
"varchar","varchar2"),
"int"=>array("char","string","varchar","varchar2"),
```

```
"real"=>array("text","varchar","varchar2","blob","bpchar"),
"string"=>array("text","varchar","varchar2","blob","bpchar"),
"int4"=>array("char","string","varchar","varchar2"),
"bpchar"=>array("text","varchar","varchar2","blob","string"),
"date"=>array("text","varchar","varchar2","blob","bpchar")
);
?>
```

Figure 4.32 HDMT data type compatibility conversion code

4.12.9 Data Migration

This is the final service provided by HDMT for the purpose of the heterogeneous data migration process for DDB. In case of DDB data migration, the DBA / GDBA can select data to be migrated from the source table and migrate it to more than one table of a DB server on the destination side in a single transaction. This facility of migrating to more than one table in a transaction is required for data that is vertically partitioned in DDB environment. There are four options provided for the DBA / GDBA in order to know the number of tables selected on the destination side for the data migration.

I have used MySQL and PostgreSQL for the demonstration purpose. These DB servers have their differences related to the insert or update of data. The selected records have been successfully migrated between MySQL and Postgresql as shown in Figure 4.33. This Figure shows the execution and completion report of the number of tables in the migration queue.

The DBA can additionally avail the facility of sub service of updating in case of duplicate records related to MySQL DB server.

On again attempting to insert a duplicate set of records into the DB, results in an exception by the DB where the primary key constraint is mentioned as shown in Figure 4.34. The DBA can choose to avail the facility of 'on duplicate key update', but this facility is not supported by Postgresql.

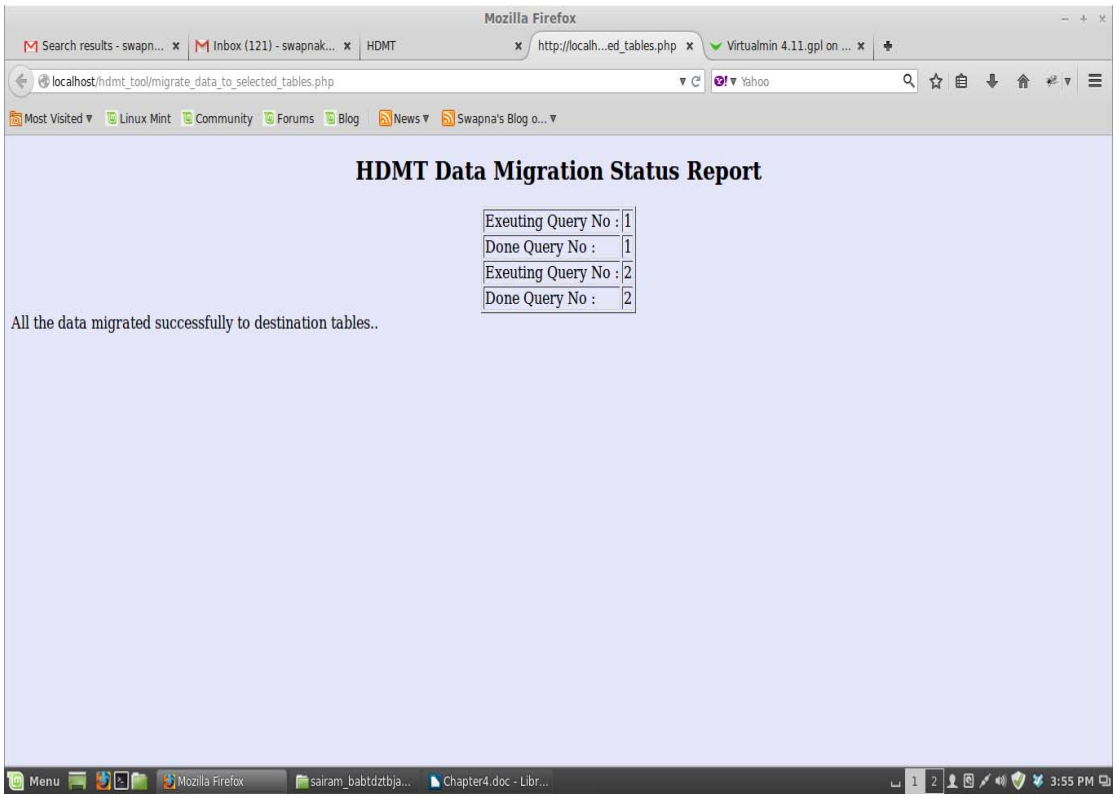


Figure 4.33 HDMT data migration report

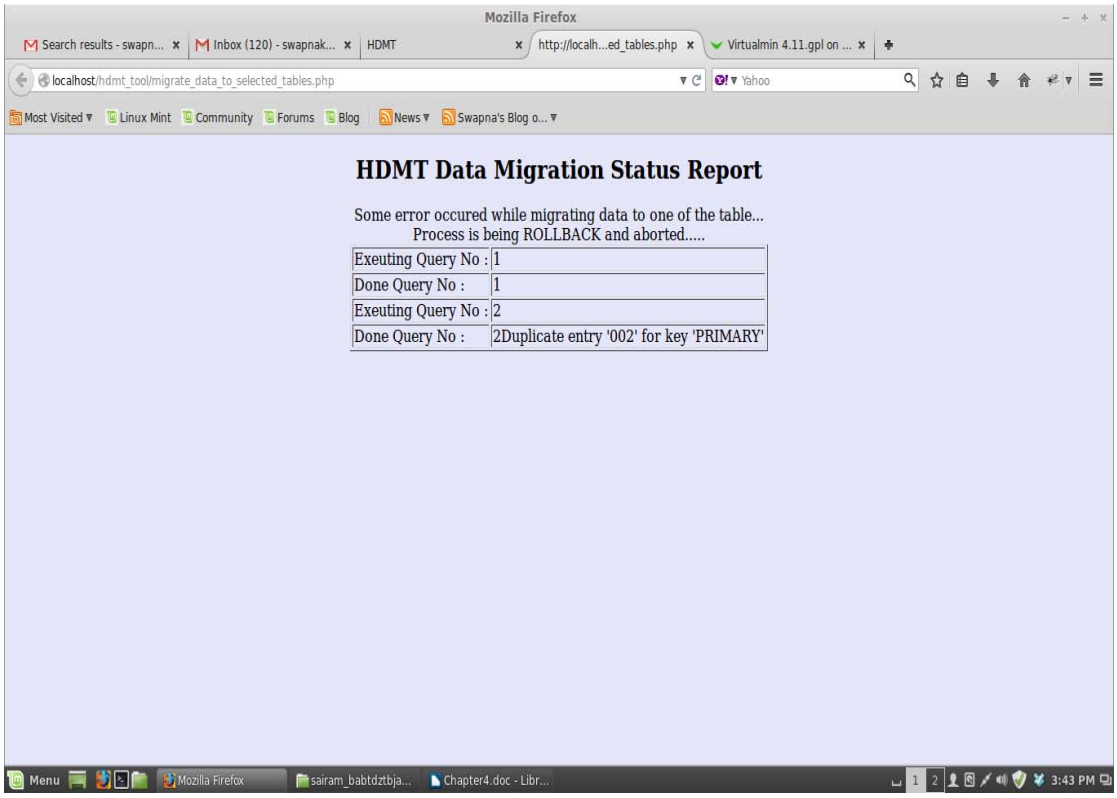


Figure 4.34 HDMT data migration report in case of constraints

On again adding the same set of records to the DB, the DBA gets an exception of duplicate entry where one of the attribute to be migrated is defined as a primary key on the destination side. This exception is applicable only in case of primary key defined on the respective table. In such cases, the data migration report may be different from the pr assessment report as can be seen by comparing the messages of Figure 4.34 and Figure 4.25. The difference in the report is due to constraints defined on the destination DB. In such cases, the destination DB will report the exceptions whenever any rule is violated.

The functionality of HDMT is as shown in all the section 4.12. The purpose of SOA based HDMT is realized along with the use of cloud services.

Chapter 5: Overall Performance Evaluation of SOA Based System

The software systems developed for various user applications have evolved from time to time due to users' expectations and needs for problem solving or data processing. In the process, specific patterns emerged over a period of time in the construction of software systems. The patterns in due course represented various architectures. One of the architecture that became popular was SOA. So now, after the demonstration of the SOA based HDMT, my work concludes with identifying the parameters for evaluating SOA based software system in various contexts so as to benefit others for software system development, for which I have taken reference of SOA based HDMT. Accordingly, on a broader categorization, I have identified the performance parameters for SOA based software systems and mention them in the overall performance of SOA based HDMT from the development and deployment point of view.

The benefits of SOA have already been discussed in chapter 1. But a look at the more detailed technical benefits of SOA given by The Open Group, it is found that SOA has more benefits at the level of service, messages, virtualization and model level implementation. On the other hand, according to TIBCO Software Inc, the benefits of SOA are vast as mentioned in 'Extending the benefits of SOA beyond the Enterprise'. Here they mention that SOA implementation and infrastructure works not only for an organization but also beyond, through partnerships, B2B commerce, collaborations etc. Another important conclusion[1] mentioned by Paul C. Hershey, Shrisha Rao, Charles B. Silio and Akshay Narayan, is that the presence of a network-induced delay in the transactions could hardly be felt in their cloud based application because the setup for SOA application and using cloud services; was

[1] The details of the paper are mentioned in reference [68] of bibliography

deployed in a private gigabit ethernet supported network. Considering the above facts, my effort is directed towards primarily identifying the various parameters on which performance of SOA based system can be analysed and secondly, mentioning various aspects that are distinct for SOA based system. Hence, I have identified and mentioned overall performance of SOA based system from the point of view of software development and deployment of HDMT. Following are the performance parameters that have been identified to consider the various aspects involved in SOA based software systems.

5.1 Development

HDMT has been implemented using open source tools and technologies as mentioned in chapter 4 considering their utility as mentioned in chapter 2. The development parameter of SOA based HDMT can be evaluated in detail as given below. This phase is assumed to begin only after previous processes of analysis and design have been completed. The choice of technology to be used and the service model to be implemented is also assumed to have been decided.

- **Functional Decomposition** – This factor refers to the process by which complex software can be broken down to smaller parts which are better to understand and maintain. With the work on ARTEMIS tool published by Silvana Castano, Valeria De Antonellis and Sabrina De Capitani di Vimercati, I was able to work on the decomposition of HDMT to identify and implement various functionalities to be organized as services. The decomposition could be for an algorithm or for object oriented design. Decomposing HDMT in this context, I have considered as per the work done i.e. the functionality as well as the objects. HDMT functionality has been developed keeping SOA approach in view. This approach along with the basic object oriented concepts made the functional decomposition of overall HDMT functionality, possible. Independent classes related to server, database and table information authentication have been developed that have data members and member methods. These methods have been developed for the

individual database technology or RDBMS and are independent of each other. The functionalities have been decomposed into smaller sub services. These services communicate with each otherl using JSON which provides a data transfer like xml technology. The Figure 5.1 gives an idea as to the various functionalities supported by HDMT tool.

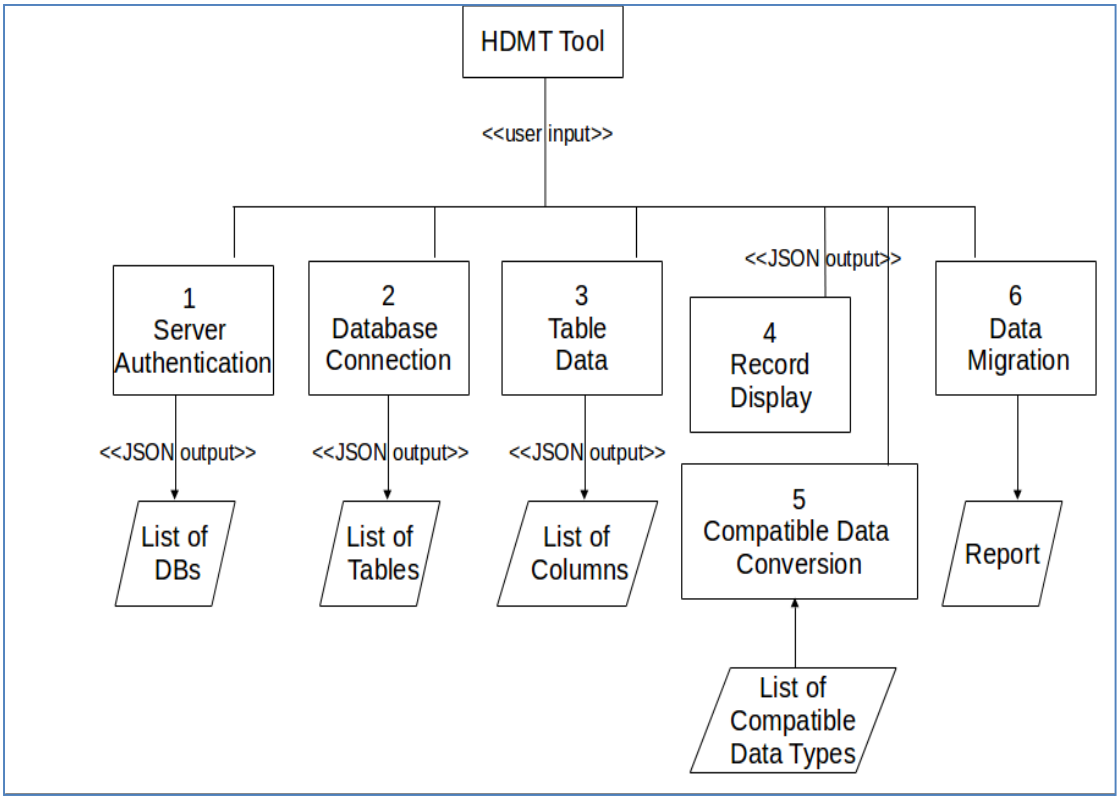


Figure 5.1 HDMT tool and its functional decomposition

The server authentication, database connection and fetching the table meta data are basic services for accessing the user desired data as shown in Figure 5.2. The other service i.e. of displaying the records i.e. data fetched is for the GDBA / DBA to view and then select for the purpose of migration. Before the final migration, the data is checked for compatibility through a service and if possible may be converted to a compatible data type as mentioned by the service. All these activities happen by presenting necessary reports and messages to the DBA / GDBA. The sub services related to MySQL and PostgreSQL are also shown as developed individually for the database technology. The actual migration may be unsuccessful in case of any database violations mentioned on the destination database.

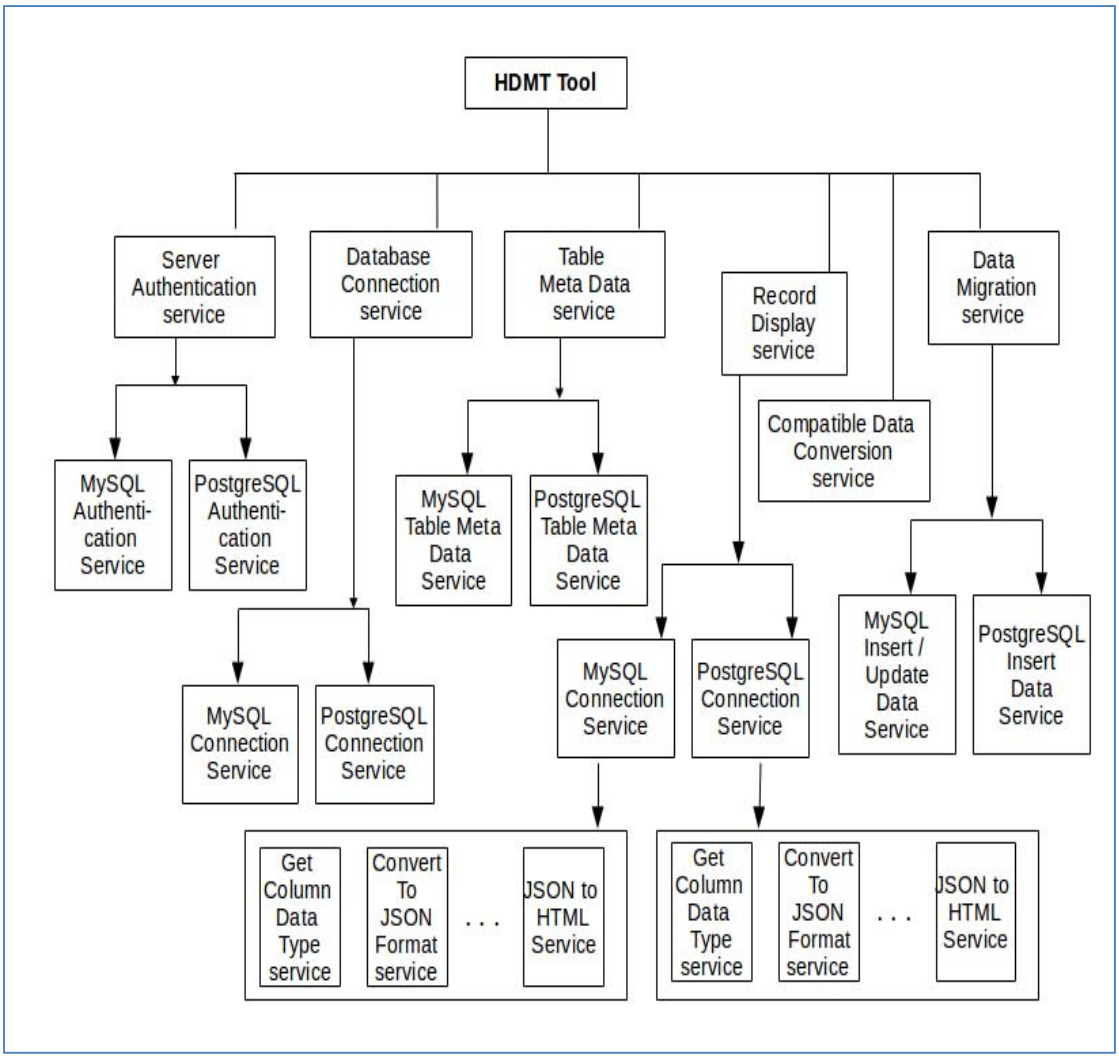


Figure 5.2 HDMT tool and its service decomposition

- **Services and related Functions** – The HDMT functionality has been implemented by developing and implementing different functions as services for their respective purpose. I have implemented HDMT functionality and its services using object oriented concepts too for the server side scripting. Hence, each class developed is independent and is called as a service where the data communication is JSON based. The purpose of the member function is taken into account and not the implementation code. This results in a number of complete member functions which can be reused in other situations. For the client side scripting, I have written small functions for better manageability. During the development phase of the HDMT tool, I realized that the number of

functions is more for SOA based architecture. As the number of functions is more, the lifetime of variables is shorter due to reduced number of lines of code.

- **Variables** – Current web technologies have developed a better data exchange system which makes the variables more manageable. The OOP methodology used in HDMT has made the development more manageable in terms of passing values from client to server side, function to function or within a function. There are many data members on the server side scripting to handle the variable concept. The client side scripting is also developed using small function and its variables. The concept of services was implemented using JSON and jQuery for which the variable definition and maintenance is efficient and manageable and exists for the lifetime of focused services.
- **Inter function communication** – This factor is compared for two situations. First for functions that are communicating and developed using same language; and second for communication between functions that are developed using different languages. In both the cases, the parameter passing for HDMT was done using JSON and jQuery. The same technique was applied for client side or server side scripting.

```
$.post("hdmttableclass.php",{ipaddress:ip, portno:prtno, dt:dbtech, un:username, pwd:password, db:database, tab:table})
```

Figure 5.3 Ajax – jQuery page call using POST method

```
$table_list=json_encode($table_list);

echo $table_list;
```

Figure 5.4 Echo json data to display html table on client side (web browser)

- **Exception handling** –In SOA based HDMT tool, the dependency of one service on another was taken into account to either activate or deactivate a service. As I have developed independent services for individual database technology, the handling of exceptions is easy and customized error messages have been given. The development

language also has error handling functionality as suggested for OOP.

- **Inter application communication** – According to w3c.org, XML is plays an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. Also JSON is a format similar to XML and is a lightweight data-interchange format. The HDMT handles jQuery or JSON data and makes inter application communication simple and reliable as applicable in SOA.
- **Concurrent development activity** – Parallel development activity is fast as compared to sequential development. The SOA based approach also proves this fact. Most of the services and sub services have been developed independently in my work. As these services communicate using JSON data, the problems related to language and platform dependence have been taken care of. I was able to work on various HDMT sub services at a given time during the development phase. These sub services led to the development of a larger service.
- **Team size** – The team comprises of various job roles and their contribution towards application development could vary depending on SOA based development and deployment. Although the team size during the HDMT development was single, the SOA approach showed that there is a need for variety of job roles to work on wide range of technologies and platforms as needed or as decided during the analysis and design phase. The team size hence becomes a dynamic factor. More members are needed for software development during the initial company steps towards SOA, but gradually as the SOA approach begins to take hold, the team size may be comparatively less.
- **Agility of software development** – This is to know how fast a working function can be developed. For SOA based system development, this factor is most flexible as services comprising of sub services can be developed in lesser time as compared to entire function development. A system comprising of many small well defined and focused services is easy to develop and maintain than a system having limited number of services performing a similar task. In HDMT development this factor was utilized for its advantages.

5.2 Maintenance

Software maintenance is a regular and expected activity for the purpose of meeting the demands in situation, technology, business etc. According to Lehman's software evolution law; that programs usually grow over time to accommodate pressure for change and satisfy an increasing set of requirements, it becomes evident that software comes under maintenance for various reasons that involve not only the change in requirements but also the changing implementation technology.

- **Patch size** – This factor is mostly dependent on the decomposition in development. It could also depend on the technological differences and its up gradation. The SOA based approach allows for software system maintenance by incorporating the necessary patches into its existing services or allowing for the development of new required services. The downtime of the software system in this case is minimized. The HDMT maintenance also saw a few patches which were incorporated as needed within the services.
- **Change Management** – This is an important factor that comes under maintenance as well as well as a topic in itself due to the involvement of the online services. Hence it is explained in section 5.4 along with its sub factors also. It basically works on the concept of assessment, implementation and evaluation. An SOA based software system is made up of well defined services and sub services which are possible for being assessed with respect to the changes and its effect. The implementation may require technological expertise which is available as there is a combination of technical experience in the team. At the same time the SOA manages to incorporate new or existing sub services without changing the meaning of existing services. Such software systems are evaluated as per the functionality without affecting the existing functions or services. The principle of SOA to compose, consume and expose is thus utilized.
- **Concurrent Patching –** This factor helps in faster maintenance of software systems. More concurrent patching is possible in SOA based applications due to smaller size of functions / services. Various team

members adept in different technologies can carry out concurrent patching at the same time. The HDMT patching did not see concurrent patching but a simultaneous patching done on different services.

5.3 Deployment

The concept of deploying the application at client side by installing the executable file has changed drastically to the current scenario of selecting the cloud structure for the deployment of applications. The HDMT has also been deployed on a cloud. The concept of services is not just in producing or consuming the services, but also in availing the services available on the cloud such as IaaS, PaaS, SaaS etc which is the essence of SOA.

- **Portability** – This refers to the usability of software in different environment. Programs written in traditional languages are more environment dependent than those written as services which are more complete. In case of HDMT, the focus was on open source technologies which have their own benefits as discussed in chapter 2. HDMT has a three tier implementational architecture which requires LAMP server, data repository and client to access the system software. In this case only LAMP server requires a compatible platform which can be availed from the PaaS cloud service. The data repository is a php file which is executed by the LAMP server. Hence portability in SOA based HDMT can be done effortlessly and efficiently.
- **Server side / client side dependency** – HDMT does not require any third party tool or any other customized software for the server or client side machine. The server needs to have open source LAMP installed for the implementation of HDMT.
- **Hardware dependency** – It refers to the part of an operating system that varies across microprocessor boards and is comprised notably of device drivers and of boot code which performs hardware initialization. HDMT has been implemented on openshift (PaaS) provided by RedHat, on which CentOS Linux 6.5 has been installed. It can be

implemented on any other comparable hardware.

- **Software dependency** – This factor needs to be specified if the software requires a specific browser version, plug-ins, or other software for execution. HDMT requires a web browser for its execution. It has been tested using commonly used web browsers like chrome and Mozilla Firefox.
- **Platform dependency** - It typically refers to applications that run under only one operating system. Sometimes, it means the same as hardware dependent and refers to applications that run in only one hardware series with the operating system not being relevant. HDMT requires Linux operating system for its deployment. The HDMT has been deployed using cloud service.
- **Program file size** – The total program file size for SOA based HDMT is 800KB, which is very easy and efficient for the purpose of deployment.

5.4 Change Management

Change in software system should reflect minimum change effect to user in terms of the working or functionality of the application. The change management process for SOA based systems should be thought in terms of all the services involved therein. According to Bernd Grobauer and co authors there are essential cloud characteristics: on-demand self-service, ubiquitous network access, resource pooling, rapid elasticity, and measured service. All these characteristics point to the fact that below mentioned factors should be considered for change management.

- **Down Time** – The higher the up time and lesser the down time means more availability and reliability of the application which results in better user loyalty and response. Increase in down time of a given service also means that the consumer of the service may switch to some other services in case of SOA based applications. I have mentioned this factor not specific to HDMT, but in general.
- **Risk Factor in change management** – This factor is about a

technology or architecture incorporating the risk factor in change management. Errors and exceptions are expected for change management, but what could be the best mechanism to provide a buffer for such changes. These were the thoughts that I felt were necessary for SOA based systems in general.

- **Simulation** – This factor refers to knowing whether we can virtually test the new SOA based system in general. I realized this factor while changing a few services of HDMT according to their new requirements. The process of simulation proves useful in understanding any exceptions that may cause software system failure.
- **UI Redesign** – Another important aspect of building good software is the user interface which ideally should be simple and easy for the user. In case of any redesign, the SOA approach allows for maximum flexibility and adaptability to future changes. While adapting HDMT to certain changes, I realized that SOA approach is easy for adapting the software according to the new changes.

5.5 Usability

ISO defines usability as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." Usability is the ease of use and learn-ability of a human-made object. The term *user friendly* is often used as a synonym for usable. The term user friendly is customer / client / situation dependent. SOA based software systems can be easily customized as per the preference which gives rise to the following situations.

- **UI Customization** – today's user expects the user interface customization as per individual preference. The incorporation of this facility into existing or new applications could greatly depend on SOA where UI customization services can be implemented without changing the functionality.
- **User Guide** – The use of this document decreases as the UI customization increases. Software systems can be incorporated with

the tool tips for increased assistance. HDMT is implemented specifically for DBA / GDBA who are acquainted with the SQL terms. The tool gives appropriate messages for the completion of data migration. The GUI of the tool is simple and easy to understand for the IT aware people.

- **Response Time** – The better the response time to user actions, better is the user satisfaction towards the application. SOA based applications have their own benefits of decomposition in this regard. The network induced delay expected in services deployed on the cloud is hardly felt as mentioned by Paul C. Hershey. HDMT has been implemented keeping in mind this factor by allowing the mention of number of records to be migrated as a transaction to the destination DB server so as to give immediate response to the user.
- **User Interactions State** – this refers to the maintenance of the data entry state in case of exceptions during forms filling so as to aid the user in continuing his actions when the application resumes working. SOA based software systems can be provided with this functionality as an add-on service.

5.6 Re – usability

The service re-usability principle is a design principle that is applied within the service-orientation design paradigm, in order to create services that have the potential to be reused across the enterprise. These reusable services are designed in a manner so that their solution logic is independent of any particular business process or technology which is one of the core principles of SOA.

- **Functional Dependency** – this factor has differences in the way of data exchange for SOA based applications. The use of XML, JSON etc for the purpose of data exchange between heterogeneous functions in HDMT, greatly solves the problem of functional dependency.
- **Function Re-usability** - this is a very desired feature for most of the situations. The re-usability of the code for SOA based applications is

high. In HDMT, the common functionality has been reused as applicable.

- **Module Re-usability** – HDMT consists of three important processes of fetching the data from the source, transforming the data as applicable and migrating the data to destination side. The number of modules in HDMT is minimum.

There are other types of re – usability like configuration re-usability, documentation / help / manual re-usability and application re-usability. The HDMT facility is at its nascent stage and can be incorporated with these in future.

5.7 Security

This can be broadly classified into 2 factors

- **Application** – The security of an application greatly depends on its platform, architecture (client server or web), type of business, audience etc. Such security threats by various categories of hackers gives the following other detailed factors. The various layers involved at platform, operating system, architecture etc need either physical or software security. Corresponding to these layers, the IaaS and PaaS provide vulnerability tracking and handling at physical level and LAMP provides vulnerability tracking and handling at the software level. HDMT has been deployed using the cloud services so as to provide for these securities.
- **Data** – The security of data depends greatly on three factors. They are data exchange security within an application or inter application, inter device data communication / exchange security and encryption algorithm change management. The SOA based software system has number of small services or functions so code is manageable and security can be implemented within function for data exchange purpose through various encryption algorithms. Also to avoid any data loss, the encryption algorithm change management is very vital which can be handled through the services. HDMT has implemented only basic data security through the use of independent communication technologies.

5.8 Portability

The portability refers to operating system or platform level portability, hardware level and client side portability (applicable for browser, plug ins etc.). HDMT is implemented using LAMP and works with modern browsers. LAMP can be installed on any open source compatible environment.

5.9 Scalability

Scalability is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. This aspect indicates whether the software under consideration is scalable depending on the current situation. Scalability is of different types like Horizontal & vertical scaling, database scalability, etc. In my study on cloud services, I worked on this factor for deploying HDMT. I have found that there are options provided for factors mentioned below.

- **Resources** –This factor refers to hardware resources like hard disk and memory. These resources need to be easily scalable. Resources such as Apache, MySQL and hard disk, all have defined memory requirement. Though, the processing by CPU and capacity of RAM may need to be increased in case of increase in number of users. SOA based software systems can be provided this flexibility as per the facility provided by data center and provided the system is adaptable. For the purpose of deploying HDMT, I found this very useful as the scalability facility was provided by data center and HDMT is adaptable to it.
- **Users** – This factor refers to the number of users connected to a website at any given time. This number is dynamic and data centers must provide for scalability as per the increase or decrease in number of users. For the purpose of deploying HDMT, I found this very useful in context of the case study discussed in chapter 4, though HDMT

handles only limited number of users at a given time. Also, the users can be given access rights accordingly and is easy to incorporate in SOA based software systems.

- **Application** – Application scalability refers to the improved performance of running applications on a scaled-up version of the system. I have mentioned this factor for SOA based system in general. SOA based software system is easily scalable.
- **Service Instance Scalability** – This factor is discussed with respect to SOA in general. Scalable services utilize a scalable resource group to contain the application resources and a failover resource group to contain the network resources (shared addresses) on which the scalable service depends. The scalable resource group can be online on multiple nodes, so multiple instances of the service can be running at once. The failover resource group that hosts the shared address is online on only one node at a time. All nodes hosting a scalable service use the same shared address to host the service. Container creates instance as per the requirement.

5.10 Compatibility and Adaptability

The small, independent and well defined services of SOA based software systems together make the software system more adaptable to changes that may arise due to requirement changes or enhancing existing functionality. During the development and deployment of HDMT, I faced this situation and found that changing a few services and sub services or adding new services was manageable. The compatibility of the software systems was also easily managed as I had selected all open source tools and which had defined interfaces.

5.11 Modularity

The modularity of any software system is based on re usability and involves its manageability. The service provided to the user in the form of features is different from the service implementation at the system level. In SOA based software system, a service is more inclusive than component. The functions in

SOA based software systems are focused, easy to manage and do not affect other functions. As the functions are focused, they can be used anywhere. The small size also means processing of a function is fast and the life time of variables is less. All these factors give rise to functional dependence and independence where due to small functions, there is independence. As the functions have general scope they are less dependent and more independent. In HDMT, there are small functions and big functions (that consist of small functions).

5.12 Data Exchange / Interoperability

The case study discussed in chapter 4 mentions that data has to be downloaded or made available in the specified format and then imported or uploaded. HDMT makes data migration possible without any security risk or violation. This means that the data exchange is secure. Moreover, HDMT provides the facility to specify the number of records to be migrated. The default number is 10 which when not mentioned means all the records. HDMT also provides for the data compatibility through equivalent and convertible data types.

5.13 Durability

This factor is new in terms of software durability. Durability is the ability to endure. In terms of software, durability is defined for database systems which is one of the ACID properties. Software being durable is a new concept which refers to its life time ie the time for which a software can be used efficiently and effectively. Many a times software becomes obsolete though it is durable. This happens because occurs because a replacement has become available having more advantages. In case of software durability I have considered the following criteria for SOA based software systems.

- **Adaptability to changing environment** – The first law proposed by Lehman postulates that a program must continually adapt to its environment, otherwise it becomes progressively less useful. The software development environment has already seen a great change

till recent times; SOA approach being a big evolution in bringing about the revolution needed for software systems. Service centric software systems provide maximum adaptability to the changing environments as mentioned by Olivier Nano and Andrea Zisman by defining processes and create methods, tools, and techniques to support cost-effective application development and use. Such systems show better durability.

- **Maintainability** – Maintainability is as per need of :
  - *Changing business logic* – As published by Vanish Talwar, Qinyi Wu, Calton Pu, Wenchang Yan, Gueyoung Jung and Dejan Milojicic, the maintainability is related to the number of lines of code which is one of the reasons of changing business logic. SOA based software systems are made of small sub services to build a larger service. So maintainability required is per sub service to make the software system more durable.
  - *Changing operating environment* – This factor refers to the required adaptations made in the operating environment which may affect the software systems. Lehman's law stipulates that over time software quality appears to be declining, unless proactive measures are taken to adapt the software to its operational environment. An SOA based software system is based on small units called services which are implemented using various languages. These services communicate with each other using other technologies to make them independent. Hence SOA approach is suited for changing operating environment in order to be durable.
- **Replace-ability of software development components** – With changes in business and software development architecture scenario, there are new software development components developed. So this factor is important in making software durable.

5.14 Expandability

Expandability is a factor which may be related to hardware, network, software etc. But basically, it is the ability of a system to accommodate additions to its

capacity or capabilities. From a hardware point of view, expandability may include additional or larger hard disks, more memory, or faster video board. From a software point of view, it may include ability to support more network users, greater number of 'hits' from website visitors. This is different from extensibility which is the ability of a software system (such as a database system) to allow and accept significant extension of its capabilities, without major rewriting of code or changes in its basic architecture. Incorporating SOA Approach does make software system more expandable in the following cases.

- **Enhancing existing functionality** – When there are enhancements to be done on a software system which are not as a result of changing requirements but a result of minimum expectations for software adaptability, then SOA based approach is useful in case of this factor. The enhancements required to be done on existing software systems need to be checked for its effect on the existing software system and uniformity in its applicability too.
- **Incorporating new functionality / modules –** When requirements change, the software must be adaptable to incorporate the changes, but in case of expandability, the addition of new modules is about improving user experience or client expectation. SOA based approach makes such incorporation much easier through the use of services.

5.15 Collaboration

Collaboration is working with others to do a task and to achieve shared goals. It is a recursive process where two or more or organizations work together to realize shared goals. In management terminology, it is also about control and authority sharing. I have focused on the software aspect for the purpose of this discussion. Companies spend to protect data generated and managed at their level. But at times they also need to share as part of business or policy changes. An SOA approach is efficient only under the mentioned terms and conditions of the company. I have identified below mentioned factors to be considered for collaboration.

- **Data Sharing** – This term is new in terms of sharing data as the companies spend to protect data generated and managed at their level. New business or policy changes may require data sharing. Software systems need to be designed keeping in view the adaptability required in such scenarios.

5.16 Resource Utilization

In the software industry, the resource utilization is important as it deals with not only the hardware and software but also the talent involved in the working of the industry. As the demand for quick response increases, the software system and the cloud services should also accommodate for such a demand. The cloud services provide for scalability as per the terms and conditions with the client. As per the work published by Stefan Seltzsam & et al where they mention a problem that the decomposition in finer-grained services allows the usage of hardware clusters and a flexible service-to-server allocation. This factor is discussed for SOA based software systems.

- **Load Balancing** – I have added this factor with respect to SOA based systems. In computing, load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing usually involves dedicated software or hardware, such as a multilayer switch or a Domain Name System server process. A SOA approach in development and deployment is suited for load balancing as per the service decomposition and cloud services & its terms and conditions.
- **Elasticity (elastic computing)** – In cloud computing, elasticity is defined as the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an

autonomic manner, such that at each point in time the available resources match the current demand as closely as possible. Elastic computing is a concept in cloud computing in which computing resources can be scaled up and down easily by the cloud service provider. The elasticity of these resources can be in terms of processing power, storage, bandwidth, etc. The SOA based software systems show elasticity in deployment. Alternately another term i.e. Auto scaling (up and down) is also used. It is a cloud computing service feature that automatically adds or removes compute resources depending upon actual usage. Auto-scaling is sometimes referred to as automatic elasticity.

5.17 Integrity

The integrity for SOA based software systems is better. HDMT does not use any third party tools for data migration so as to ensure data integrity. Other SOA based systems can be additionally analyzed on the basis of functional and application integrity.

5.18 Reliability

As mentioned by Jiantao Pa, software reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability. It differs from hardware reliability in that it reflects the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of software reliability problems. Conversely, the simplicity of a system contributes to lesser software reliability problems. SOA based software systems are simple individually to accomplish the functionality of a complex software system and their reliability depends on following factors.

- **Availability** – A software system deployed using SOA approach is reliable as per the terms and conditions offered by the cloud services. In case of increase in the number of users, the cloud services do offer the scalability option which will make the software system available to

the users. Also the fact that availability is the ratio of up time / total time will depend on the data center. SOA based systems are more manageable because of the decomposition.

- **Up – time –** It is used as a measure of computer operating system reliability or stability, in that this time represents the time a computer can be left unattended without crashing, or needing to be rebooted for administrative or maintenance purposes, though conversely, long uptime may indicate negligence. Paul C. Hershey has mentioned that for SOA based software systems, the usual quality metrics such as uptime and reliability may be considered applicable in the context of cloud systems.

5.19 Supportability

It is easy to provide functional support in case of adaptable software systems. The client side environment detection is also an important aspect of supportability. HDMT is also implemented with jQuery which detects the browser and executes the functions accordingly so as to make HDMT browser independent. Hence if there is any change on client side then web based systems need to detect it and work accordingly which HDMT does.

5.20 Testability

To analyze any software for its effectiveness, it must be testable. SOA based software systems can be tested under various tools and methods to know its purpose realization. HDMT is testable as function size is limited.

5.21 Audit-ability

The conduction of an audit at any level needs expertise and an eye for detail. There are many levels like functional level, module level, code level and application level at which an audit can be conducted. The SOA based software systems are integrated with various development and deployment technologies so as to bring all the services to work in coordination with each other through well defined communication. HDMT is a simple and small utility

service for the DBA / GDBA to consume where the audit is easy. The audit itself can be carried out as internal / external audit by internal or external experts.

5.22 Localization and Internationalization

In computing, internationalization and localization are means of adapting computer software to different languages, regional differences and technical requirements of a target market. Internationalization is the process of designing a software application so that it can potentially be adapted to various languages and regions without engineering changes. Olivier Nano and Andrea Zisman mention that adaptable, flexible, interoperable, and maintainable service centric software systems are made possible through defining processes and creating methods, tools, and techniques to support cost-effective application development and use. Such methodology is the essence of SOA and is applicable.

5.23 Customization

New software services involved in the software customization can be incorporated into the existing software system so as to make the software system adaptable as per the client or customer demands. The services involved in the customization can be implemented at various levels like user level, software level, database level etc.

The HDMT can also be customized at the destination side. This option will cause considerable changes in the number of inputs required. The tool can be customized to take only the source database related information while hiding the destination connection details. This way the destination DB information can be hidden or not shared with the other DBA. Only the relevant tables can be displayed for the mapping purpose for the data communication. The tool can be customized so as to take only the college or institute database connection details to be sent to the governing body website.

In this way, the SOA approach is relevant to users in different ways. For the online services or facilities, the user can ask for customized things through services; whereas for the developers, SOA means better flexibility and manage-ability at the system level.

Chapter 6: Future Enhancements

The DBA / GDBA has a tremendous responsibility in maintaining, managing and administering the data. SOA based HDMT aids the DBA in such a task of migration of data for a distributed database environment. The services involved in the entire operation have been developed and deployed considering the various layers involved in the software system development and IT infrastructure for a generalized situation. More software services related to customization and authentication can be developed further for specific or customized situation and similarly, more layers can be added for adaptability to future needs or changes. The SOA approach will be beneficial in providing various DB administration services through software layers for more such software systems.

An article by Eifrem on ZDNet interested me. He mentions that companies are already past the point where a single database is capable of managing all data workloads — and it's misleading for any vendor to suggest it has the answer to all an enterprise's database problems. Overtime, the concept of data storage has evolved from flat file model to hierarchical and relational model and is evolving further into semi structured big database that is so large that it's difficult to process using traditional database and software techniques. It usually refers to information that doesn't reside in a traditional row-column database. These big databases also need big data analysis platforms and tools viz Hadoop, MapReduce, HPCC, GridGain etc. More such tools targeting general or specific situations can be developed in future as per the needs. “The Future of Hadoop and of Big Data DBMSs” by Rob Klopp mentions that the Hadoop eco-system builds all of the piece-parts of a very large database management system. There are the basics: a distributed file system in HDFS, a low-level query engine in Map/Reduce with an abstraction in Pig, and the beginnings of optimization, SQL, availability, backup & recovery, etc.

An article in The Economic Time, Education on “Accelerate private investments in India's education sector” dated 10 Oct, 2014, gives a good view of the Indian education sector. It mentions that the Indian education industry is growing at an average compound annual growth rate of 14 per cent and is expected to be $82 billion to $87 billion by the end of this year, up from $50 billion in 2011. Also, the amount of information with the education sector is large where the e resources are available either free or have to be subscribed. For the learning resources the nature of information is semi structured. Large amount of learning resources are shared online for the purpose of online viewing or downloading. Available tools facilitate this situation for the benefit of those involved in academic. More such tools for the maintaining and sharing of learning resources centrally, will help in the case of not only learning resources but also in the software to carry ahead the open source tools policy of the education sector.

Summary

The study of SOA and DDB administration and its respective tool comprises an important aspect of my research work. The situation in education sector for data uploading which presented itself was immensely responsible in focusing my work. Through this situation and work, I got interested in finding out a utility tool to handle the situation of data uploading. This led to the study on various DB administration tools and their features. A comparative of various existing database tools helped me in identifying the needs for HDMT. At the same time, I was able to develop service oriented architecture for HDMT because of work done in SOA, XML, web services etc. The academic focus on open source tools lead to the development specification of HDMT. Eventually, the various cloud services lead to the deployment specification of HDMT. The DDB environment existing in current times paved way for the SOA based HDMT to be implemented as a tool in context of heterogeneous data migration. At the end of implementation of HDMT, I focused on identifying and analyzing the various factors on which users / IT people / companies analyze SOA based applications.

The study and results published by researchers has given me an insight into the current work and direction of technical working in various fields and technologies. The SOA based architecture of HDMT for the administration of DDBs, specifically for the purpose of heterogeneous data migration would be usable not only by IT aware people of academics but also by IT experts of companies and computer science community. The performance parameters identified in chapter 5 for the purpose of analyzing them with respect to SOA based software systems will be useful to IT managers / experts / professionals involved in taking decisions related to such business and technical situations; to understand, evaluate and implement as per their needs.

Considering all DB administration tools, if a DBA / GDBA are given the opportunity of working with online DB administration tools, then the SOA

approach is most appropriate for dynamic and online services. The various technologies involved in the development of HDMT benefited me so as to provide an insight into their relevance and importance in software system development. The domain knowledge equips oneself with the issues involved in their working which can be evaluated further for their automation. I experienced this important aspect as I have taken a situation related to the education sector for which I have proceeded with automating the heterogeneous data migration for DDB through HDMT. This tool would greatly benefit the colleges approved by AICTE (or other governing bodies) to import their data to the governing body website by sharing some information required by HDMT so as to facilitate the data migration.

File 1:

Hdmthtml.html : HTML file that provides the user interface.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>HDMT</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<style type="text/css">
#left {
  margin-right: 15px;
}
#right {
  margin-left: 15px;
}
#left_wrapper {
  width: 10%;
  height:1000px;
  float: left;
  position: relative;
  z-index: 0;
  background-color:#FFFFFF;
}
#middle_wrapper {
  width: 80%;
  height:1000px;
  margin-right: -150px;
  float: left;
  position: relative;
  z-index: 2;
  background-color:#E6E6FA;
```

```
}
#right_wrapper {
  width: 10%;
  height:1000px;
  float: right;
  position: relative;
  z-index: 1;
  background-color:#FFFFFF;
}
</style>
        <script language="javascript" type="text/javascript" src="./libs/js/jquery-1.11.1.js"></script>
        <script language="javascript" type="text/javascript" src="jsfunction.js"></script>
        <script language="javascript" type="text/javascript" src="js_ajaxfunction.js"></script>
        <link rel="stylesheet" href="//code.jquery.com/ui/1.11.2/themes/smoothness/jquery-ui.css">
        <script src="//code.jquery.com/ui/1.11.2/jquery-ui.js"></script>

        <!--for JSON TO HTML TABLE--->
        <script type="text/javascript" src="./libs/js/jquery.jsontotable.min.js"></script>
        <!--for HTML TABLE TO JSON --->
        <script type="text/javascript" src="./libs/js/jquery.tabletojson.min.js"></script>
        <script type="text/javascript" language="javascript">
        function initially()
        {
                document.getElementById('divid_migrate').style.visibility = 'hidden';
        }
        </script>
        <style>
        .jsontotable table, .jsontotable th, .jsontotable td {
```

```
                border: 1px solid black;
                margin: 10px;
                width:50%;
        }
 code {
   white-space: normal;
 }
        </style>
</head>
<body onload="initially();">
 <div id="left_wrapper">
   <div id="left"> <img src="hdmtlogo" height="50" width="100">
   </div>
 </div>
 <div id="middle_wrapper">
   <div id="middle">
        <center><h4>Welcome to Heterogeneous Data Migration Tool (HDMT) !</h4>
        <table border="2" style="border-collapse:collapse; border:solid 3px black;" cellpadding="5">
                <tr>
                        <th COLSPAN="2">SOURCE</th>
                        <th COLSPAN="2">DESTINATION</th>
                </tr>
                <tr>
                        <td>Server: IP Address</td>
                        <td><input type="text" value="172.16.6.23" id="txtid_ip1" name="txtnm_ip1" size="30">
                        <td>Server: IP Address</td>
                        <td><input type="text" value="172.16.6.25" id="txtid_ip2" name="txtnm_ip2" size="30">
                </tr>
                <tr>
                        <td>Server: Port No</td>
```

```
                    <td><input type="text" value="3306" id="txtid_port1"
name="txtnm_port1" size="30">
                    <td>Server: Port No</td>
                    <td><input type="text" value="5432" id="txtid_port2"
name="txtnm_port2" size="30">
                </tr>
                <tr>
                    <td>User Name</td>
                    <td><input type="text" value='root' name="txtusername1"
id="txtusername1" size="30"/></td>
                    <td>User Name</td>
                    <td><input type="text" value='root' name="txtusername2"
id="txtusername2" size="30"/></td>
                </tr>
                <tr>
                    <td>Password</td>
                    <td><input type="password" value='root'
name="txtpassword1" id="txtpassword1" size="30"/></td>
                    <td>Password</td>
                    <td><input type="password" value='root'
name="txtpassword2" id="txtpassword2" size="30"/></td>
                </tr>
                <tr>
                    <td>DBMS</td>
                    <td><select id="sdbtname1" name="sdbtname1" style="width:
220px">
                            <option>SELECT</option>
                            <option>MYSQL</option>
                            <option>PostgreSQL</option>
                            </select><input type="button" id="btnid_ip1"
value="Go">
                    </td>
                    <td>DBMS</td>
```

```
                    <td><select id="sdbtname2" name="sdbtname2" style="width:
220px">
                        <option>SELECT</option>
                        <option>MYSQL</option>
                        <option>PostgreSQL</option>
                        </select><input type="button" id="btnid_ip2"
value="Go">
                    </td>
                </tr>
                <tr>
                    <td>Database Name</td>
                    <td><select id="sdbname1" name="sdbname1" style="width:
220px">
                            <option>SELECT</option>
                        </select><input type="button" id="btnid_db1"
value="Go"></td>
                    <td>Database Name</td>
                    <td><select id="sdbname2" name="sdbname2" style="width:
220px">
                            <option>SELECT</option>
                        </select><input type="button" id="btnid_db2"
value="Go"></td>
                </tr>
                <tr>
                    <td>Table Name</td>
                    <td><select id="stbname1" name="stbname1" style="width:
220px">
                                <option>SELECT</option>

                            </select><input type="button"
id="btnid_table1" value="Go">
                    </td>
                    <td>Table Name</td>
```

```
                    <td><select id="stbname2" name="stbname2" style="width:
220px">
                                        <option>SELECT</option>

                                    </select><input type="button"
id="btnid_table2" value="Go">
                    </td>
                </tr>
        </table>
<br>
        <table>
                <tr>

                    <td valign="top">
                        <table bgcolor="#FFFFFF" border="1" id="tdisp1"
name="tdisp1" cellpadding="5">
                            <tr>
                                <td width = "215" height = "15"
bgcolor="#E6E6FA">Column Names</td>
                            </tr>
                        </table>
                    </td>
                    <td>---->>></td>
                    <td valign="top">
                        <table bgcolor="#FFFFFF" border="1" id="tdisp2"
name="tdisp2" cellpadding="5">
                            <tr>
                                <td width = "215" height = "15"
bgcolor="#E6E6FA">Column Names</td>
                            </tr>
                        </table>
                    </td>
                </tr>
        </table>
```

```
<br>
	<table border="0" width='95%' height="40" bgcolor='#B0C4DE'>
		<tr><td>Field Name : <select name='ddlnm_fldName' id='ddlid_fldName' width="100"></select></td>
			<td>Operator : <select name='ddlnm_opName' id='ddlid_opName'>
				<option>SELECT</option>
				<option>LIKE</option>
				<option>=</option><option>!=</option>
				<option><</option><option><=</option>
				<option>></option><option>>=</option>
			</select></td>
			<td>Value : <input Type="text" name='txtnm_fldVal' id='txtid_fldVal' size="10"></td>
			<td>Limit(Row From) : <input Type="text" name='txtnm_start' id='txtid_start' value="0" size="5">
			(Count) : <input Type="text" name='txtnm_cnt' id='txtid_cnt' size="5" value='10'>
			</td>
		</tr>
	</table>
<br>
<input type="button" value="View Selected Data" id="btnquery" name="btnquery"/>
<br><br>

<div id="divid_migrate" name="divnm_migrate">

<table border="0" width='95%' height="40" bgcolor='#B0C4DE'>
	<tr>
		<td>
		Select Action :
		</td>
		<td>
```

```
                <select id="s_action_id" name="s_action_name" style="width: 300px">
                        <option value='1'>1 - 'Check' Migration Status</option>
                        <option value='2'>2 - 'Add' to Migration Queue</option>
                        <option value ='3'>3 - 'View' Migration Queue</option>
                        <option value ='4'>4 - 'Process' Migration Queue</option>
                        <option value ='5'>5 - 'Clear' Migration Queue</option>
                </select>
                <input type="button" id="btnid_ok" value="OK">
                </td>
                <td>
                        <input type="checkbox" id="chkid_migrate_update" name = "chknm_migrate_update"> ON DUPLICATE KEY UPDATE RECORD
                </td>
        </tr>
</table>

<div id="dialog" title="Basic dialog">
<p>  <a href='migrate_data_to_selected_tables.php' target='_blank'>Migrate Data to Selected Table(s)</a></p>
</div>
        <hr>
</div>
<br><br>
        <div id="jsontotable" class="jsontotable">[Preview Table will appear here]</div>
        </center>
  </div>
</div>
```

```
<div id="right_wrapper">
  <div id="right">
  </div>
</div>
</body>
</html>
```

File 2:

js_ajaxfunction.js : Javascript file that uses jquery and ajax technology for updating part of the user interface screen as per user action. The data sent by the server side file is in JSON format.

```
var norow1, norow2;
//jquery function to  call json to return the array of databases 1
$(document).ready(function(){
    $('#btnid_ip1').click(function() {
    ip=$('#txtid_ip1').val();
          prtno=$('#txtid_port1').val();
          dbtech=$('#sdbtname1').val();
          username=$('#txtusername1').val();
          password=$('#txtpassword1').val();
          $.post("hdmtipaddrclass.php",
{ipaddress:ip,portno:prtno,dbt:dbtech,un:username,pwd:password})
          .done(function( data ){
          if(data.search("ERROR:")==2)
                alert(data);
          else
                update_dom(data,"#sdbname1");
          });
    });
});

//jquery function to call json to return the array of databases 2
$(document).ready(function(){
    $('#btnid_ip2').click(function() {
```

```
    ip=$('#txtid_ip2').val();
        prtno=$('#txtid_port2').val();
        dbtech=$('#sdbtname2').val();
        username=$('#txtusername2').val();
        password=$('#txtpassword2').val();
        $.post("hdmtipaddrclass.php",
{ipaddress:ip,portno:prtno,dbt:dbtech,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#sdbname2");
        });
    });
});
//jquery function to assign the list of data in select control
function update_dom(data,ctrlname)
{
        $(ctrlname).empty();
        data=JSON.parse(data);
        for (i=0; i<data.length; i++)
        {
        $(ctrlname).append('<option>' + data[i] + '</option>');
        }
}
//jquery function to call json to return the array of table 1
$(document).ready(function(){
    $('#btnid_db1').click(function() {
        ip=$('#txtid_ip1').val();
        prtno=$('#txtid_port1').val();
    database=$('#sdbname1').val();
        username=$('#txtusername1').val();
        password=$('#txtpassword1').val();
        dbtech=$('#sdbtname1').val();
```

```
        $.post(                                                    "hdmtdbclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,db:database,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#stbname1");
        });
   });
});
//jquery function to call json to return the array of table 2
$(document).ready(function(){
   $('#btnid_db2').click(function() {
        ip=$('#txtid_ip2').val();
        prtno=$('#txtid_port2').val();
      database=$('#sdbname2').val();
        username=$('#txtusername2').val();
        password=$('#txtpassword2').val();
        dbtech=$('#sdbtname2').val();
        $.post( "hdmtdbclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,db:database,un:username,pwd:password})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
                update_dom(data,"#stbname2");
        });
   });
});
//jquery function to call json to return the array of column 1
$(document).ready(function(){
   $('#btnid_table1').click(function() {
        ip=$('#txtid_ip1').val();
        prtno=$('#txtid_port1').val();
```

```
        database=$('#sdbname1').val();
            table=$('#stbname1').val();
            username=$('#txtusername1').val();
            password=$('#txtpassword1').val();
            dbtech=$('#sdbtname1').val();
            $.post( "hdmttableclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,un:username,pwd:password,db:database,tab:table
})
            .done(function( data ) {
            if(data.search("ERROR:")==2)
                    alert(data);
            else
            {
                    data_arr = data.split("#");
                    data_col = JSON.parse(data_arr[0]);
                    data_type = JSON.parse(data_arr[1]);
                    cnt = data_col.length;
                    norow1=cnt;
                    genCols('tdisp1',cnt,data_col,data_type);
                    genFieldNames(cnt,data_col);
            }
            });
    });
});

//jquery function to call json to return the array of column 2
$(document).ready(function(){
    $('#btnid_table2').click(function() {
            ip=$('#txtid_ip2').val();
            prtno=$('#txtid_port2').val();
        database=$('#sdbname2').val();
            table=$('#stbname2').val();
            username=$('#txtusername2').val();
            password=$('#txtpassword2').val();
```

```
        dbtech=$('#sdbtname2').val();
        $.post( "hdmttableclass.php",
{ipaddress:ip,portno:prtno,dt:dbtech,un:username,pwd:password,db:database,tab:table
})
        .done(function( data ) {
        if(data.search("ERROR:")==2)
                alert(data);
        else
        {
                data_arr = data.split("#");
                data_col = JSON.parse(data_arr[0]);
                data_type = JSON.parse(data_arr[1]);
                cnt = data_col.length;
                norow2=cnt;
                genCols('tdisp2',cnt,data_col,data_type);
        }
        });
   });
});
//jquery function to call generatequery function on button click
$(document).ready(function(){
   $('#btnquery').click(function() {
        generatequery('tdisp1','tdisp2',norow1, norow2);
   });
});
```

File 3 :

jsfunction.js : Javascript file with jquery and ajax to respond to user action of selecting columns. Output is the data received in table.

```
var col_list2 = "";//declared global to complete select and insert action for destination
var col_list1 = "";//declared global to complete select and insert action for source
//function to generate the columns for a table with the values
function genCols(tablename,tab_noc,columnNames,columnType)
{
```

```
var i, strcol;
strcol=tab_noc;
//identify the table in case of more than once click on the button
var table = document.getElementById(tablename);
//clearing the table in case of more than once click on the button
for(var i = table.rows.length - 1; i > 0; i--)
{
	table.deleteRow(i);
}
// Find a <table> element with id="myTable":
var table = document.getElementById(tablename);
for (i=0;i<strcol;i++)
{
	// Create an empty <tr> element and add it to the 1st position of the table:
	var row = table.insertRow(i+1);
	// Insert new cells (<td> elements) at the 1st and 2nd position of the "new" <tr> element:
	var cell1 = row.insertCell(0);
	// Add some ddl to the new cells:
	var ddA = document.createElement("select");
	//assign the name and id of the dynamically added controls using the table name also
	ddA.name = "name1";
	ddA.id = tablename+"id1"+i;
	ddA.options[ddA.length] = new Option("SELECT", "0");
	ddA.style.width="210px";
	for (j=0;j<strcol;j++)
	{
		//adds the columns names in the specified table and drop down list
		ddA.options[ddA.length] = new Option(columnNames[j], j+1);
	}
	//Add the dropdown to the parent node
```

```
                cell1.appendChild(ddA);
        }
}//end of function genCols()
//function to populate the column names for where condition
function genFieldNames(option_cnt,option_data)
{
        var ddl=document.getElementById('ddlid_fldName');
        var length = ddl.options.length;
        while(ddl.options.length > 0){
   ddl.remove(0);}
        ddl.options[ddl.length] = new Option("SELECT","0");
        for (j=0;j<option_cnt;j++)
                {
                        //adds the columns names in the specified table and drop down
list
                        ddl.options[ddl.length] = new Option(option_data[j], j+1);
                }
}//end of function genFieldNames
//function to send the column values to generate a dynamic query on server side file
function generatequery(tabledisp1, tabledisp2, tab_noc1,tab_noc2)
{
        //declare variables for function scope only
        var selectcol1, selectcol2, strcolumn1, strcolumn2, strtype1;
        var strtable2=document.getElementById("stbname2").value;
        var strtable1=document.getElementById("stbname1").value;
        col_list2 = "";
        //call a function to compose the string for selected columns
        col_list2=get_column_string(tab_noc2, tabledisp2);
        col_list1 = "";
        //call a function to compose the string for selected columns
        col_list1 = get_column_string(tab_noc1, tabledisp1);
        //Reading the db 1 conneciton details
        ip1=$('#txtid_ip1').val();
        prtno1=$('#txtid_port1').val();
```

```
database1=$('#sdbname1').val();
username1=$('#txtusername1').val();
password1=$('#txtpassword1').val();
dbtech1=$('#sdbtname1').val();
//Reading the db 2 conneciton details
ip2=$('#txtid_ip2').val();
prtno2=$('#txtid_port2').val();
database2=$('#sdbname2').val();
username2=$('#txtusername2').val();
password2=$('#txtpassword2').val();
dbtech2=$('#sdbtname2').val();
//added parameter for horizental filteration of rows
ffName = $('#ddlid_fldName option:selected').text();
ffOp =  $('#ddlid_opName option:selected').text();
ffVal =  $('#txtid_fldVal').val();
recFrom =  $('#txtid_start').val();
recLimit =  $('#txtid_cnt').val();
//ajax technology to send data to php file
$.post("hdmtinsertselect.php",
{queryaction:1,ff_name:ffName,ff_op:ffOp,ff_val:ffVal,rec_from:recFrom,rec_limit:r
ecLimit,fld_list1:col_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,ipaddress1:ip
1,
dt1:dbtech1,db1:database1,un1:username1,pwd1:password1,ipaddress2:ip2,dt2:dbtec
h2,db2:database2,un2:username2,pwd2:password2,portno1:prtno1,portno2:prtno2})
.done(function(response){
//call function to update dom or user form
////alert(response);
if(response.search("ERROR:")==2)
        alert(response);
else
        update_dom_table(response);
//make the migrate button visible for further process only if jsontable is
created
```

```
		if (response) document.getElementById('divid_migrate').style.visibility = 'visible';
		});
}

//function to assign the records to a jsontable pulugin
function update_dom_table(response)
{
		var data = get_jsondataformat(response);
		//remove all from container div
		$("#jsontotable").html("");
		//'tblid_jsontbl' has been set as id in html table created through json plugin
		$.jsontotable(data, { id: '#jsontotable', header: true });
}

//function to convert string in json readable format
function get_jsondataformat(response)
{
		response = response.replace('["','[[');
		response = response.replace('"]', ']]');
		var matchesCount = response.split('","').length - 1;
		for (var i=0; i<matchesCount; i++)
			response=response.replace('","', '],[');
		matchesCount = response.split("'").length - 1;
		for (var i=0; i<matchesCount; i++)
			response=response.replace("'", "");
		return response;
}
//ajax function to capture the click on the dynamically created json table.
$(document).ready(function(){
		$(document.body).on('click', '#tblid_jsontbl tr', function(){
		$(this).css('background', 'white')
				//use a javascript confirm control
```

```
                        if (confirm('Are you sure you want to remove selected row ?'))
{
                                $(this).remove();
                        } else {
                        $(this).css('background', 'lightgray')
                }
        });
});
$(document).ready(function(){
        $("#dialog").attr('visibility','hidden');
});

$(document).ready(function(){
                $("#btnid_ok").click(function(){
                selected_action = $('#s_action_id').val();
                var update_rec_json = $('#tblid_jsontbl').tableToJSON();
                var strtable2=document.getElementById("stbname2").value;

                var strtable1=document.getElementById("stbname1").value;
                if(selected_action == 4)
                {
                        $("#dialog").attr('display','block');
                        $("#dialog").dialog();
                        alert(selected_action);
                        exit;
                }
                update_rec_json=JSON.stringify(update_rec_json);
                //Reading the db conneciton details
                ip2=$('#txtid_ip2').val();
                database2=$('#sdbname2').val();
                username2=$('#txtusername2').val();
                password2=$('#txtpassword2').val();
                dbtech2=$('#sdbtname2').val();
                prtno2=$('#txtid_port2').val();
```

```
                //Reading the db conneciton details
                dbtech1=$('#sdbtname1').val();
                ip1=$('#txtid_ip1').val();
                database1=$('#sdbname1').val();
                username1=$('#txtusername1').val();
                password1=$('#txtpassword1').val();
                prtno1=$('#txtid_port1').val();
                //check for migration option
                chkmigrationstatus=$('#chkid_migrate').is(":checked");
                chkmigrationupdatestatus=$('#chkid_migrate_update').is(":checked");

                //ajax technology to send data to php file

        $.post("hdmtinsertselect.php",{check_m_u_status:chkmigrationupdatestatus,c
heck_m_status:selected_action,queryaction:2,update_json:update_rec_json,fld_list1:c
ol_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,
ipaddress1:ip1,ipaddress2:ip2,dt2:dbtech2,dt1:dbtech1,db1:database1,db2:database2,u
n1:username1,un2:username2,pwd1:password1,pwd2:password2,portno1:prtno1,port
no2:prtno2}).done(function(response){
                alert ("Migration Status - \n" + response);
                });
                //alert(selected_action);
        })
});

//function to migrate the data in html table to json and finally in the destination
database. DEPRICATED
$(document).ready(function(){
                $("#btnid_migrate").click(function(){
                var update_rec_json = $('#tblid_jsontbl').tableToJSON();
                var strtable2=document.getElementById("stbname2").value;

                var strtable1=document.getElementById("stbname1").value;
                update_rec_json=JSON.stringify(update_rec_json);
```

```
                //Reading the db conneciton details
                ip2=$('#txtid_ip2').val();
                database2=$('#sdbname2').val();
                username2=$('#txtusername2').val();
                password2=$('#txtpassword2').val();
                dbtech2=$('#sdbtname2').val();
                prtno2=$('#txtid_port2').val();
                //Reading the db conneciton details
                dbtech1=$('#sdbtname1').val();
                ip1=$('#txtid_ip1').val();
                database1=$('#sdbname1').val();
                username1=$('#txtusername1').val();
                password1=$('#txtpassword1').val();
                prtno1=$('#txtid_port1').val();
                //check for migration option
                chkmigrationstatus=$('#chkid_migrate').is(":checked");
                chkmigrationupdatestatus=$('#chkid_migrate_update').is(":checked");

                //ajax technology to send data to php file

        $.post("hdmtinsertselect.php",{check_m_u_status:chkmigrationupdatestatus,c
heck_m_status:chkmigrationstatus,queryaction:2,update_json:update_rec_json,fld_list
1:col_list1,fld_list2:col_list2,tbl1:strtable1,tbl2:strtable2,
ipaddress1:ip1,ipaddress2:ip2,dt2:dbtech2,dt1:dbtech1,db1:database1,db2:database2,u
n1:username1,un2:username2,pwd1:password1,pwd2:password2,portno1:prtno1,port
no2:prtno2}).done(function(response){
                alert ("Migration Status - " + response);
                });
        });
});
//function to compose the column string
function get_column_string(tab_noc, tabledisp)
{
        var strcol=tab_noc;
```

```
	var selectcol, strcolumn, col_list="";
	for (var i=0;i<strcol;i++)
	{
		selectcol = document.getElementById(tabledisp+'id1'+i);

		strcolumn = selectcol.options[selectcol.selectedIndex].innerHTML;
		//consider column names that do not start with the select keyword.
		if (strcolumn != 'SELECT')
		{
			col_list=col_list+strcolumn;
			col_list=col_list+",";
		}
	}
	//remove the trailing comma.
	col_list=col_list.substring(0,col_list.length-1);
	return col_list;
}//end of function
```

File 4 :

hdmtipaddressclass.php : PHP file to take server related input information and authenticate it and give the list of databases as json output.

```
<?php
//start of session
session_start();
//create object of class connect_to_server
$obj_ip = new connect_to_server();
//class connect_to_server
class connect_to_server
{
	//data members of connect_to_server for connecting to server
	var $ipaddress;
	var $portno;
	var $dbtech;
	var $username;
```

```
var $password;
//constructor
function connect_to_server()
{
        //to initialize values from session variables
        $this->ipaddress = $_POST['ipaddress'];
        $this->portno = $_POST['portno'];
        $this->dbtech = $_POST['dbt'];
        $this->username = $_POST['un'];
        $this->password = $_POST['pwd'];
        //function to check mysql or mssql server is used
        $this->check_dbtechnology();
}
//function to check mysql or mssql server is used
function check_dbtechnology()
{
        if($this->dbtech == "MYSQL")
        {
                //user defined function to connect to mysql
                $this->fconnect_mysql();
        }
        else if($this->dbtech == "PostgreSQL")
        {
                //user defined function to connect to mysql
                $this->fconnect_pgsql();
        }
        else
        {
                echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                return;
        }
}
//member function to connect to MySQL server
function fconnect_mysql()
```

```
	{
		// Create connection
		$con=mysql_connect($this->ipaddress . ":" . $this->portno,$this->username,$this->password) or die ("..ERROR: Could not connect to MySQL");
		// Check connection and exit in case of exception
		if (!$con)
		{
			echo "..ERROR: Failed to connect to " . $this->dbtech . ": " . mysql_error();
			exit;
		}
		else
		{
			//call mysql_query function to get the list of databases and store in result
			$result=mysql_query("SHOW DATABASES WHERE `Database` != 'information_schema'");
			//iterate through the result
			while($row=mysql_fetch_assoc($result))
			{
				//add the result to an array
				$db_list[]=$row['Database'];
			}
			//encode the array using json for data exchange instead of xml
			$db_list=json_encode($db_list);
			//close connection
			mysql_close($con);
			//send the db list to the client side
			echo $db_list;
		}
	}//end of function fconnect_mysql()
	//member function to connect to MS SQL server
	function fconnect_pgsql()
	{
```

```
                // Create connection
                $con = pg_connect("host= " . $this->ipaddress . " port = " . $this->portno . " dbname = postgres user= " . $this->username . " password= " . $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
                // Check connection and exit in case of exception
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to Database " . $this->dbtech . " error is : " . pg_last_error();
                        exit;
                }
                else
                {
                        //call pgsql_query function to get the list of databases and store in result
                        $result=pg_query($con, "SELECT datname FROM pg_database WHERE datistemplate = false;");
                        //iterate through the result
                        while($row=pg_fetch_assoc($result))
                        {
                                //add the result to an array
                                $db_list[]=$row['datname']; //datname as defined in pg_database
                        }
                        //encode the array using json for data exchange instead of xml
                        $db_list=json_encode($db_list);
                        //close connection
                        pg_close($con);
                        //send the db list to the client side
                        echo $db_list;
                }
        }//end of function fconnect_pgsql()
}//end of class connect_to_server
?>
```

File 5 :

hdmtdbclass.php : PHP file that takes the server and selected database information and gives the list of tables as json output.

```
<?php
//start of session
session_start();
//create object of class connect_to_server
$obj_db = new connect_to_database();
//class connect_to_server
class connect_to_database
{
	//data members of connect_to_server for connecting to database
	var $ipaddress;
	var $portno;
	var $dbtech;
	var $database;
	var $username;
	var $password;
	//constructor
	function connect_to_database()
	{
		//to initialize values from session variables
		$this->ipaddress = $_POST['ipaddress'];
		$this->dbtech = $_POST['dt'];
		$this->portno = $_POST['portno'];
		$this->database = $_POST['db'];
		$this->username = $_POST['un'];
		$this->password = $_POST['pwd'];
		//function to check mysql or mssql server is used
		$this->check_dbtechnology();
```

```
	}
	//function to check mysql or mssql server is used
	function check_dbtechnology()
	{
		if($this->dbtech == "MYSQL")
		{
			//user defined function to connect to mysql
			$this->fgettablelist_mysql();
		}
		else if($this->dbtech == "PostgreSQL")
		{
			//user defined function to connect to pgsql
			$this->fgettablelist_pgsql();
		}
		else
		{
			echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
			return;
		}
	}
	//member function to connect to MySQL server
	function fgettablelist_mysql()
	{
		// Create connection
		$con=mysql_connect($this->ipaddress . ":" . $this->portno,$this->username,$this->password) or die ("..ERROR: Could not connect to MySQL");
		// Check connection and exit in case of exception
		if (!$con)
		{
			echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error is : " . mysql_error();
			exit;
		}
		else
```

```
		{
			//call mysql_query function to get the list of tables and store in result
			$result = mysql_query("SHOW TABLES FROM " . $this->database );
			//iterate through the result
			while($row=mysql_fetch_row($result))
			{
				//add the result to an array

				$table_list[]=$row[0];
			}
			//encode the array using json for data exchange instead of xml
			$table_list=json_encode($table_list);
			//close connection
			mysql_close($con);
			//send the db list to the client side
			echo $table_list;
		}
	}//end of function

	//member function to connect to PostgreSQL server
	function fgettablelist_pgsql()
	{
		// Create connection for PostgreSQL
		$con = pg_connect("host= " . $this->ipaddress . " port = " . $this->portno . " dbname = " . $this->database . " user= " . $this->username . " password= " . $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
		// Check connection and exit in case of exception
		if (!$con)
		{
			echo "..ERROR: Failed to connect to " . $this->dbtech . " . Error is : " . pg_last_error();
			exit;
```

```
                }
                else
                {
                        //call mysql_query function to get the list of tables and store in
result
                        $result = pg_query("SELECT table_name FROM
information_schema.tables WHERE table_schema = 'public';");
                        //iterate through the result
                        while($row=pg_fetch_row($result))
                        {
                                //add the result to an array

                                $table_list[]=$row[0];//echo $table_list;
                        }
                        //encode the array using json for data exchange instead of xml
                        $table_list=json_encode($table_list);
                        //close connection
                        pg_close($con);
                        //send the db list to the client side
                        echo $table_list;
                }
        }//end of function
} //end of class
?>
```

File 6 :

hdmttableclass.php : PHP file to take server, database and table information and to give the list of columns as json output.

```
<?php
session_start();
//create an object of connect_to_table
$obj_table = new connect_to_table();

class connect_to_table
```

```
{
	//data members that are used for connecting and fetching
	var $ipaddress;
	var $portno;
	var $dbtech;
	var $database;
	var $table;
	var $username;
	var $password;
	function connect_to_table()
	{
		//in the constructor to set the data member values using session values

		$this->ipaddress = $_POST['ipaddress'];
		$this->portno = $_POST['portno'];
		$this->dbtech = $_POST['dt'];
		$this->database = $_POST['db'];
		$this->table = $_POST['tab'];
		$this->username = $_POST['un'];
		$this->password = $_POST['pwd'];
		//call function to proceed to mysql or mssql server functions
		$this->check_dbtechnology();
	}
	function check_dbtechnology()
	{
		//check if mysql is selected
		if($this->dbtech == "MYSQL")
		{
			//call function to get the list of columns.
			$this->fgetcolumnlist_mysql();
		}
		//check if ms sql server is selected
		else if($this->dbtech == "PostgreSQL")
		{
```

```
                        //call function to get the list of columns.
                        $this->fgetcolumnlist_pgsql();
                }
                else
                {
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                        return;
                }
        }
        //function to display the columns in drop down list from the selected table.
        function fgetcolumnlist_mysql()
        {
                // Create connection
                $con=mysql_connect($this->ipaddress . ":" . $this->portno,$this->username,$this->password) or die ("..ERROR: Could not connect to MySQL");

                // Check connection
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error is : " . mysql_error();
                        exit;
                }
                else
                {
                        //connect to database using mysql function
                        mysql_select_db($this->database);
                        //use mysql function to get the result of a query
                        $result = mysql_query("SHOW COLUMNS FROM " . $this->table );
                        //iterate the result of query to get the list of column names and its data type
                        while($row=mysql_fetch_row($result))
                        {
```

```
                    $column_list[]=$row[0];//column names
                    $column_type[]=$row[1];//column data type
                }
                //encode the array using json for data exchange instead of xml
                $column_list=json_encode($column_list);
                $column_type=json_encode($column_type);
                //close connection
                mysql_close($con);
                //send the db list to the client side
                echo $column_list."#".$column_type;
            }//end of database connection activity
        }//end of function

        function fgetcolumnlist_pgsql()
        {
            // Create connection
            $con = pg_connect("host= " . $this->ipaddress . " port = " . $this->portno . " dbname = " . $this->database . " user= " . $this->username . " password= " . $this->password) or die ("..ERROR: Could not connect to PostgreSQL");
            // Check connection and exit in case of exception
            if (!$con)
            {
                echo "..ERROR: Failed to connect to " . $this->dbtech . ". Error is : " . pg_last_error();
                exit;
            }
            else
            {
                //query string to connect to postgres to get the columns of a table
                $sql="SELECT column_name, data_type FROM information_schema.columns WHERE table_name = '" . $this->table . "'";
                //use mysql function to get the result of a query
                $result = pg_query($con,$sql);
```

```
                    //iterate the result of query to get the list of column names and
its data type
                    while($row=pg_fetch_row($result))
                    {
                        $column_list[]=$row[0];//column names
                        $column_type[]=$row[1];//column data type
                    }
                    //encode the array using json for data exchange instead of xml
                    $column_list=json_encode($column_list);
                    $column_type=json_encode($column_type);
                    //close connection
                    pg_close($con);
                    //send the db list to the client side to be updated through ajax
                    echo $column_list."#".$column_type;
                }//end of database connection activity
        }//end of function
}//end of class
?>
```

File 7 :

datatype_compatible_config.php : PHP file that uses the concept of associative array for data type compatibility comparison. This file acts as a data repository which can be added with new meaningful data type associations.

```
<?php
/* This file is used for providing the data type compatibility information. All the
compatible data types will be having the same value
and data type as key ie data type name will be key and its value as integer value.

This file has been included in hdmtinsertselect.php file for the
check_datatype_compatibility() function. */
$arr_equivalent_datatype_set =
array("integer"=>"0","int"=>"0","number"=>"0","int2"=>"0","int4"=>"0",
```

```
"char"=>"1","varchar"=>"1","text"=>"1","string"=>"1","character"=>"1","blob"=>"1
","bpchar"=>"1",
                                        "date"=>"2","datetime"=>"2","year"=>"2",
                                        "real"=>"3"
                                            );
$arr_convertible_datatype_set                                                    =
array("integer"=>array("char","string","varchar","varchar2"),

        "int"=>array("char","string","varchar","varchar2"),

        "real"=>array("text","varchar","varchar2","blob","bpchar"),

        "string"=>array("text","varchar","varchar2","blob","bpchar"),

        "int4"=>array("char","string","varchar","varchar2"),

        "bpchar"=>array("text","varchar","varchar2","blob","string"),

        "date"=>array("text","varchar","varchar2","blob","bpchar")
                                            );
?>
```

File 8 :

hdmtinsertselect.php : PHP file that takes all connection parameter inputs from source and destination database. It selects records from source table to be migrated to destination table. The selected records are displayed in the html table. At this stage few records may be deleted if so desired. The remaining records in the html table is saved in session for migration of data to multiple tables. There are few sub services also to fetch the column data types for compatibility comparison.

```
<?php
//start of session
session_start();
```

```
error_reporting(0);
//create object of class
$obj_dml = new data_insert_select();
//class definition
class data_insert_select
{
	//data members for connecting to source and destination servers
	var $ipaddress1;
	var $portno1;
	var $dbtech1;
	var $database1;
	var $username1;
	var $password1;
	var $col_list1;
	var $tbl1;
	var $ipaddress2;
	var $portno2;
	var $dbtech2;
	var $database2;
	var $username2;
	var $password2;
	var $col_list2;
	var $tbl2;
	//data member for taking the json table data
	var $update_rec_json;
	//data member for identifying the insert or select query
	var $queryaction;
	//data member for getting migration status of check box.
	var $check_m_status;
	//constructor to initialize data members with session values sent from js through jquery
	function data_insert_select()
	{
		$this->ipaddress1 = $_POST['ipaddress1'];
```

```
            $this->portno1 = $_POST['portno1'];
            $this->dbtech1 = $_POST['dt1'];
            $this->database1 = $_POST['db1'];
            $this->username1 = $_POST['un1'];
            $this->password1 = $_POST['pwd1'];
            $this->col_list1 = $_POST['fld_list1'];          //col list for
SELECT query
            $this->tbl1 = $_POST['tbl1'];

            $this->ipaddress2 = $_POST['ipaddress2'];
            $this->portno2 = $_POST['portno2'];
            $this->dbtech2 = $_POST['dt2'];
            $this->database2 = $_POST['db2'];
            $this->username2 = $_POST['un2'];
            $this->password2 = $_POST['pwd2'];
            $this->col_list2 = $_POST['fld_list2'];          //col list for
INSERT query
            $this->tbl2 = $_POST['tbl2'];          //col list for INSET query

            $this->queryaction = $_POST['queryaction'];
            $this->check_m_status = $_POST['check_m_status'];
            $this->check_m_u_status = $_POST['check_m_u_status'];
            //call function to check the query action parameter.
            $this->check_queryaction();
        }//end of constructor
        //function to check the query action parameter for either select or insert action
        function check_queryaction()
        {
            if($this->queryaction == 1)
                $this->select_records();
            else if ($this->queryaction == 2)
            {
                $this->update_rec_json = $_POST['update_json']; //get the
jsonhtml table for insert into destination
```

```
                    if ($this->check_m_status==1)        //check        dataypte
compatibility
                    {
                            $this->check_datatype_compatibility();        //exited
from there, if datatype not compatble
                    }
                    else if ($this->check_m_status==2)   //add to migration queue
                    {
                            $this->compose_query();       //This function composes
the query and save them to session
                    }
                    else if ($this->check_m_status==3)   //add to migration queue
                    {
                            //display the queryies & connection details saved in
session
                            print_r($_SESSION);
                    }
                    else if ($this->check_m_status==4)  //Process    to    migration
queue
                    {
                            //display the queryies & connection details saved in
session

        echo("<script>window.location='http://www.continue.com'</script>");

                    }
                    else if ($this->check_m_status==5)   //clear migration queue
                    {
                            session_destroy();       //To reset/delete all session data
                            echo("Migration Queue 'destroyed' Successfully !!!");
                    }
            }
            else
```

```
		{
			echo("..ERROR: Please click on [View Selected Data]");
			return;
		}
	}// end of function check_queryaction()

	//function to select the source records from the available connection information.
	function select_records()
	{
		if ($this->dbtech1 == "MYSQL")
		$rs = $this->fetch_mysql_records();
		else if ($this->dbtech1 == "PostgreSQL")
			$rs = $this->fetch_pgsql_records();
		else
		{
			echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
			return;
		}
		if(!$rs)
		{
			echo("..ERROR: No column details selected !!");
			return;
		}
		$num_rows = $this->get_number_records($rs);
		$colcnt = 0;
		//get col_list1 in array format to validate for duplicate columns
		$arrcol_list1  = explode(',', $this->col_list1);
		$count_col_list1  = count($arrcol_list1);
		$arrcol_list2  = explode(',', $this->col_list2);

		//validate for duplicate columns by calling check_duplicate_column($arrcol_list1)
```

```
            if($this->check_duplicate_column($arrcol_list1))  //exits in case on duplicate columns
            {
                  echo("..ERROR: Duplicate columns in selected column list 1");
                  return;
            }
            if($this->check_duplicate_column($arrcol_list2))  //exits in case on duplicate columns
            {
                  echo("..ERROR: Duplicate columns in selected column list 2");
                  return;
            }
            if(count($arrcol_list1)!=count($arrcol_list2))  //exits in case unequal columns
            {
                  echo("..ERROR: Different number of fields selected in column list 1 and 2 !");
                  return;
            }
            if ($this->dbtech2 == "MYSQL")
                  $arr_coltype = $this->fetch_mysql_destination();
            else if ($this->dbtech2 == "PostgreSQL")
                  $arr_coltype = $this->fetch_pgsql_destination();
            //json string with the col names of table1 & 2
            $col_list_json = $this->get_cols_json($count_col_list1);
            //returs 1D array/rows containing csv value of fields
            $arr_vals = $this->get_compose_string($rs, $arr_coltype, $count_col_list1,$col_list_json);
            //function call to return the data in format as per the json_html table

            $html_records = $this->display_records_htmltable($arr_vals);
            echo $html_records;
      }//end of function select_records()
      //function to return the number of rows in a result set
```

```
	function get_number_records($rs)
	{
		if ($this->dbtech1 == "MYSQL")
			$num_rows = mysql_num_rows($rs);
		else if ($this->dbtech1 == "PostgreSQL")
			$num_rows = pg_num_rows($rs);
		else
		{
			echo("..ERROR: Please select MYSQL and PostgreSQL only.");
			return;
		}
		return $num_rows;
	}//end of function get_number_records
	//function to get destination details for mysql
	function fetch_mysql_destination()
	{
		// Create connection
		$con2=$this->db_connect_mysql(2);
		//fetch columns data types
		$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . " WHERE 1=2";
		mysql_select_db($this->database2,$con2) or die("Unable to select db2");
		$rs2 = mysql_query($strquery,$con2);
		if(!$rs2)
		{
			echo("..ERROR: No record ie column list selected (msg1)!!");
			return;
		}
		$arr_coltype = $this->get_column_types ($rs1);
		mysql_close($con2);
		return $arr_coltype;
	}//end of function fetch_mysql_destination()
```

```
	//function to get destination details for postgresql
	function fetch_pgsql_destination()
	{
		// Create connection
		$con2=$this->db_connect_pgsql(2);
		//fetch columns data types
		//$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . "
WHERE 1=2";
		$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . "
WHERE 1=2";
		//echo "..ERROR: " . $strquery;
		$rs = pg_query($con2,$strquery);
		if(!$rs)
		{
			echo("..ERROR: No record ie column list selected (msg2)  !!" .
pg_last_error());
			return;
		}
		$arr_coltype = $this->get_column_types ($rs);
		pg_close($con2);
		return $arr_coltype;
	}//end of function fetch_pgsql_destination()
	//function to fetch source records for mysql
	function fetch_mysql_records()
	{
		// Create connection
		$con1=$this->db_connect_mysql(1);
		mysql_select_db($this->database1,$con1) or die("..ERROR: Could not
connect to MYSQL");
		//Filter criteria
		$ff_name = $_POST['ff_name'];
		$ff_op = $_POST['ff_op'];
		$ff_val = $_POST['ff_val'];
```

```
		$rec_from = $_POST['rec_from'];
		$rec_limit = $_POST['rec_limit'];
		$strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1 ;
		//apply where clause if specified by user
		if(($ff_name !="SELECT") && ($ff_op !="SELECT") && ($ff_val !==""))
		{
			$where_clause = " WHERE $ff_name $ff_op '$ff_val'";
			$strquery .= $where_clause;
		}

		//Apply the row limit.
		if(($rec_from !="") && ($rec_limit !=""))
		{
			$limit_clause = " LIMIT $rec_from,$rec_limit";
			$strquery .= $limit_clause;
		}
		$rs = mysql_query($strquery,$con1) or die("..ERROR: Could not execute MYSQL query");
		mysql_close($con1);
		return $rs;
	}//end of function fetch_mysql_records()
	//function to fetch source records for pgsql
	function fetch_pgsql_records()
	{
		// Create connection
		$con1=$this->db_connect_pgsql(1);
		//selecting the column list from table with columns specified in col_list1
		$strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1;
		//echo $strquery;
		$rs = pg_query($con1,$strquery);
		pg_close();
		return $rs;
```

```
	}//end of function fetch_pgsql_records()
	//function to check the selection of duplicate column names
	function check_duplicate_column($arrcol_list1)
	{
		$fld_cnt_before = count($arrcol_list1);
		$arrcol_list1  = array_unique($arrcol_list1);
		$fld_cnt_after = count($arrcol_list1);
		if($fld_cnt_before != $fld_cnt_after)
			return 1;
		else
			return 0;
	}//end of check_duplicate_column($arrcol_list1)
	//function returns array containg datatype of all the fields in the resultset
	function get_column_types ($result_id)
	{
		$info = array( );			# create empty array
		if ($this->dbtech1 == "MYSQL" || $this->dbtech2 == "MYSQL")
		{
			for($i=0;$i<mysql_num_fields($result_id);$i++)
			$info[] = mysql_field_type($result_id,$i);
		}
		if ($this->dbtech1 == "PostgreSQL" || $this->dbtech2 == "PostgreSQL")
		{
			for($i=0;$i<pg_num_fields($result_id);$i++)

			$info[] = pg_field_type($result_id,$i);
		}
		//if else
			//echo "..ERROR: Please select MYSQL or PostgreSQL only.";

		return ($info);
   }//end of get_column_types ($result_id)
```

```
        //function to fetch the column constraints as array
    function get_column_constraints ($result_id)
    {
                $info = array( ); //create empty array
     for($i=0;$i<mysql_num_fields($result_id);$i++)
        $info[] = mysql_field_flags($result_id,$i);
                //print_r ($info);
                return ($info);
    }//end of get_column_constraints
        //function to convert columns string into json compatible type
        function get_cols_json($count_col_list1)
        {
                $col_list1_arr = explode(",",$this->col_list1);
                $col_list2_arr = explode(",",$this->col_list2);
                for($i=0;$i<$count_col_list1;$i++)
                {
                        $col_list_arr[$i] = """ . $col_list1_arr[$i] . " -> " .
$col_list2_arr[$i] . """;
                }
                //convering single quoted array elevem to sq csv.
                $col_list_json = implode(",",$col_list_arr);
                return $col_list_json;
        }//end of function get_cols_json($count_col_list1)
        //function to encode the array values in json compatible format
        function display_records_htmltable($arr_vals)
        {
                $table_list=json_encode($arr_vals);
                return $table_list;
        }//end of function
        //function to compose the string for select query
        function get_compose_string($rs, $arr_coltype,
$count_col_list1,$col_list_json)
        {
                $arr_vals[] = $col_list_json;
```

```
                if ($this->dbtech1 == "MYSQL")
                {
                        while($row = mysql_fetch_array($rs))
                        {
                                $row_val_csv = "";
                                for($colcnt=0;$colcnt<$count_col_list1;$colcnt++)
                                {
                                        $row_val_csv .= "'". $row[$colcnt]."',";
        //json compatible
                                }
                        $row_val_csv = substr("$row_val_csv",0,-1);          //remove
last comman (,)
                        $arr_vals[] = $row_val_csv;
                        }
                }
                else if ($this->dbtech1 == "PostgreSQL")
                {
                        while($row = pg_fetch_array($rs))
                        {
                                $row_val_csv = "";
                                for($colcnt=0;$colcnt<$count_col_list1;$colcnt++)
                                {
                                        $row_val_csv .= "'". $row[$colcnt]."',";
        //json compatible
                                }
                        $row_val_csv = substr("$row_val_csv",0,-1);          //remove
last comman (,)
                        $arr_vals[] = $row_val_csv;
                        }
                }
                else
                {
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                        return;
```

```
                }
                return $arr_vals;
        }//end of get_compose_string($rs, $arr_coltype, $count_col_list1)
        //function called to insert records in mysql or postgresql
        function compose_query()
        {
                //echo($this->update_rec_json);
                $json_row = json_decode($this->update_rec_json,true);
                $row_count = count($json_row);
                $row_csv = array();
                $i=0;
                while ($i<$row_count)
                {
                        $row_csv[] = """. implode("','", $json_row[$i]) . """;
                        $i++;
                }
                $arr_vals = $row_csv;
                if ($this->dbtech2 == "MYSQL")
                        $this->compose_save_query_mysql($arr_vals);
        //call function to compose and save query details to session
                else if ($this->dbtech2 == "PostgreSQL")
                        $this->compose_save_query_pgsql($arr_vals);

        } // end of function insert_records()
        //function to insert data in case of mysql
        function compose_save_query_mysql($arr_vals)
        {
                // Create connection
                $con2=$this->db_connect_mysql(2);
                //Composing INSERT query with multiple values
                $i=0;
                $strquery = "INSERT INTO " . $this->tbl2 . " (" . $this->col_list2 . ")
VALUES ";
                $num_rows = count($arr_vals);
```

```
                while($i<$num_rows)
                {
                        $strquery .= "($arr_vals[$i]),";
                        $i++;
                }
                //remove last commas(,) from $strquery;
                $strquery = rtrim($strquery,',');
                if ($this->check_m_u_status=='true')
                {
                        $arrcol_list2  = explode(',', $this->col_list2);
                        $count_col_list2  = count($arrcol_list2);
                        $strqueryupdate = " ON DUPLICATE KEY UPDATE ";
                        $j=0;
                        while($j<$count_col_list2)
                        {
                                $strqueryupdate .= $arrcol_list2[$j] . "= VALUES(" .
$arrcol_list2[$j] . "),";
                                $j++;
                        }
                        $strqueryupdate = rtrim($strqueryupdate,',');
                        $strquery = $strquery . $strqueryupdate;
                }
                //saving to session
                if(isset($_SESSION['qNo']))
                        $_SESSION['qNo'] ++;
                else
                        $_SESSION['qNo'] = 0;
                $str_qNo = "q".$_SESSION['qNo'];
                $_SESSION[$str_qNo]['q'] = "$strquery";              //query
                $_SESSION[$str_qNo]['db_server'] = $this->ipaddress2;
                $_SESSION[$str_qNo]['db_portno'] = $this->portno2;
                $_SESSION[$str_qNo]['db_usr'] = $this->username2;
                $_SESSION[$str_qNo]['db_pwd'] = $this->password2;
                $_SESSION[$str_qNo]['db_tech'] = $this->dbtech2;
```

```
                $_SESSION[$str_qNo]['db_name'] = $this->database2;

                echo("Query Added to Migration Queue successfully !!!\n Total in Queue : ". ($_SESSION['qNo'] + 1));
        } // end of function insert_records_mysql()
        //function to insert data in case of pgsql
        function compose_save_query_pgsql($arr_vals)
        {
                // Create connection
                $con2=$this->db_connect_pgsql(2);
                //Composing INSERT query with multiple values
                $i=0;
                $strquery = "INSERT INTO " . $this->tbl2 . " (" . $this->col_list2 . ") VALUES ";
                $num_rows = count($arr_vals);
                while($i<$num_rows)
                {
                        $strquery .= "($arr_vals[$i]),";
                        $i++;
                }
                //remove last commas(,) from $strquery;
                $strquery = rtrim($strquery,',');
                if ($this->check_m_u_status=='true')
                {
                        echo ("..ERROR: ON DUPLICATE KEY UPDATE facility is not applicable for PostgreSQL.");
                        exit;
                }
                //saving to session
                if(isset($_SESSION['qNo']))
                        $_SESSION['qNo'] ++;
                else
                        $_SESSION['qNo'] = 0;
                $str_qNo = "q".$_SESSION['qNo'];
```

```
                $_SESSION[$str_qNo]['q'] = "$strquery";
                $_SESSION[$str_qNo]['db_server'] = $this->ipaddress2;
                $_SESSION[$str_qNo]['db_portno'] = $this->portno2;
                $_SESSION[$str_qNo]['db_usr'] = $this->username2;
                $_SESSION[$str_qNo]['db_pwd'] = $this->password2;
                $_SESSION[$str_qNo]['db_tech'] = $this->dbtech2;
                $_SESSION[$str_qNo]['db_name'] = $this->database2;

                echo("Query Added to Migration Queue successfully !!!\n Total in Queue : ". ($_SESSION['qNo'] + 1));
                pg_close($con2);
        } // end of function insert_records_pgsql()
        //function to check the data type compatibility of source and destination
        function check_datatype_compatibility()
        {
                // Create two connections for source and destination location
                if($this->dbtech1=="MYSQL")
                        $con11=$this->db_connect_mysql(1);
                else if($this->dbtech1=="PostgreSQL")
                        $con1=$this->db_connect_pgsql(1);
                else
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                if($this->dbtech2=="MYSQL")
                        $con2=$this->db_connect_mysql(2);
                else if($this->dbtech2=="PostgreSQL")
                        $con2=$this->db_connect_pgsql(2);
                else
                        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
                //fetch columns data types
                $strquery = "SELECT " . $this->col_list1 . " FROM " . $this->tbl1 . " WHERE 1=2";
                if ($this->dbtech1=="MYSQL")
                        $rs1 = mysql_query($strquery,$con11) or die ("..ERROR: No record ie column list selected (msg3) !!".mysql_error());
```

```
else if($this->dbtech1=="PostgreSQL")
        $rs1 = pg_query($con1,$strquery) or die ("..ERROR: No record ie column list selected (msg4)!!");
else
        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");
//fetch columns data types
$strquery = "SELECT " . $this->col_list2 . " FROM " . $this->tbl2 . " WHERE 1=2";
if ($this->dbtech2=="MYSQL")
        $rs2 = mysql_query($strquery,$con2) or die ("..ERROR: No record ie column list selected (msg5)!!");
else if($this->dbtech2=="PostgreSQL")
        $rs2 = pg_query($con2,$strquery) or die ("..ERROR: No record ie column list selected (msg6)!!");
else
        echo ("..ERROR: Please select MYSQL or PostgreSQL only.");

$arr_coltype1 = $this->get_column_types($rs1);
$arr_coltype2 = $this->get_column_types($rs2);
//count the number of elements in coltype array
$n = count($arr_coltype1);

///fetch arr_col_constaints. //
$constraint_arr1 = $this->get_column_constraints($rs1);
require_once("datatype_compatible_config.php");          //file contains the datacompatibility deffination
$flg = "";      //flag to show non-compatable data type
$equvalence_report = "\n Index \t Server1    \t Server2";
for($i=0;$i<$n;$i++)
{
        if($arr_equivalent_datatype_set["$arr_coltype1[$i]"]          != $arr_equivalent_datatype_set["$arr_coltype2[$i]"])
                $flg .= $i.",";
```

```
                $equvalence_report .= "\n $i \t\t" . $arr_coltype1[$i] . "\t\t" .
$arr_coltype2[$i] ;                    }
        echo($equvalence_report);
        if($flg !="")   //if datatypes are not equivalent the checking for
compatibility/convertibility
        {
                echo("\n\n..WARNING: Un-equivalent datatype found at
INDEX  $flg  . !!! Checking for convertability of datatype!!!");
                $flg = "";        //flag to show non-compatable data type
                $convertable_report = "\n Index \t Server1    \t Server2";
                for($i=0;$i<$n;$i++)
                {
                        $sub_arr                                        =
$arr_convertible_datatype_set["$arr_coltype1[$i]"];
                        $x=in_array("$arr_coltype2[$i]",$sub_arr);
                        if($x!=1)        //means unconvertable datatype
                                $flg .= $i.",";
                $convertable_report .= "\n $i \t\t" . $arr_coltype1[$i] . "\t\t" .
$arr_coltype2[$i] ;                             }
                echo($convertable_report);
                if($flg !="")
                {
                        echo("\n\n..ERROR: UnCompatable datatype found at
INDEX  $flg  . Migration Process has been aborted !!!");
                        exit;
                }
                else
                {
                        echo("\n \n CONGRATULATIONS ! Data Type
Compatibility process has been successful and is OK for migration !!!\n\n");
                }
        }
        else
        {
```

```
                    echo("\n \n CONGRATULATIONS ! Data Type Verification process has been successful and is OK for migration !!!\n\n");
                }
        }
        //function to connect to mysql database. location specifies the source or destination details. function returns connection.
        function db_connect_mysql($location)
        {
                if ($location == 2)
                {
                        $con = mysql_connect($this->ipaddress2 . ":" . $this->portno2,$this->username2,$this->password2,true) or die ("..ERROR: Could not connect to MYSQL 2" . mysql_error());
                        mysql_select_db($this->database2,$con) or die("Unable to select db2");
                }
                else if ($location == 1)
                {
                        $con = mysql_connect($this->ipaddress1 . ":" . $this->portno1,$this->username1,$this->password1) or die ("..ERROR: Could not connect to MYSQL 1" . mysql_error());
                        mysql_select_db($this->database1,$con) or die("Unable to select db1");
                }
                else
                {
                        echo ("Invalid Server");
                        exit;
                }
                //check if connected
                if (!$con)
                {
                        echo "..ERROR: Failed to connect to server : " . $location . ". Error is : " . mysql_error();
```

```
                exit;
        }
        return $con;
}//end of function db_connect_mysql
//function to connect to postgresql database. location specifies the source or destination details. function returns connection.
function db_connect_pgsql($location)
{
        if ($location == 2)
        {
                $con = pg_connect("host=" . $this->ipaddress2 . " port=" . $this->portno2 . " dbname=" . $this->database2 . " user=" . $this->username2 . " password=" . $this->password2) or die ("..ERROR: Could not connect to PostgreSQL" . pg_last_error());
        }
        else if ($location == 1)
        {
                $con = pg_connect("host= " . $this->ipaddress1 . " port = " . $this->portno1 . " dbname= " . $this->database1 . " user= " . $this->username1 . " password= " . $this->password1) or die ("..ERROR: Could not connect to PostgreSQL" . pg_last_error());
        }
        else
        {
                echo ("Invalid Server");
                exit;
        }
        //check if connected
        if (!$con)
        {
                echo "..ERROR: Failed to connect to server : " . $location . ". Error is : " . pg_last_error();
                exit;
        }
```

```
                return $con;
        }//end of function db_connect_mysql
}       //end of class
?>
```

File 9 :

migrate_data_to_selected_tables.php : PHP file that either commits or rollbacks the insertion of data into the destination table(s). It also displays appropriate reports to DBA / GDBA on completion of the operation.

```
<?php
        session_start();
        if(isset($_SESSION['qNo'])=="")
        {
         echo("<body bgcolor='#E6E6FA'><h2>HDMT Data Migration Status Report</h2><br>There is NO data / table set to migrate !!!<br></body>");
         return;
        }
        $dbtech2 = $_SESSION['q0']['db_tech'];
        $noq=$_SESSION['qNo'];     //number of queries in session. It has been set in htmtinsertselect.php through session
        if ($dbtech2 == "MYSQL")
        {
                echo ("<html><body bgcolor='#E6E6FA'><center><h2>HDMT Data Migration Status Report</h2><table border='1'>");
                $curr_con = db_connect_mysql("q0");          //fetch current connection.
                mysql_query("BEGIN");
                for($i=0;$i<=$noq;$i++)
                {
                        $queryId = "q".$i;
                        echo("<tr><td>Exeuting Query No : </td><td>" . ($i+1) . "</td></tr>");
                        $curr_query = $_SESSION[$queryId]['q'];  //fetch query connection.
```

```
                $db_name = $_SESSION[$queryId]['db_name'];
                mysql_select_db($db_name,$curr_con) or die("Unable to select" . $db_name . ":" . mysql_error());
                mysql_query($curr_query,$curr_con);
                echo("<tr><td>Done Query No : </td><td>" . ($i+1) . mysql_error());
                echo("</td></tr>");
                //check of last execution was successful or NOT.
                if(mysql_errno())
                 {
                        echo("Some error occured while migrating data to one of the table...<br> Process is being ROLLBACK and aborted.....");
                        mysql_query("ROLLBACK");
                        exit;
                 }
        }//end of for loop
        mysql_query("COMMIT");    //apply the change to database tables;
        echo("</table></center></body>All the data migrated successfully to destination tables..");
        session_destroy();
    }//end of if
    else if ($dbtech2 == "PostgreSQL")
    {
        echo ("<html><body bgcolor='#E6E6FA'><center><h2>HDMT Data Migration Status Report</h2><table border='1'>");
        $curr_con = db_connect_pgsql("q0");          //fetch current connection.
        pg_query("BEGIN");
        for($i=0;$i<=$noq;$i++)
        {
            $queryId = "q".$i;
            echo("<tr><td>Exeuting Query No : </td><td>" . ($i+1) . "</td></tr>");
```

```
                $curr_query = $_SESSION[$queryId]['q'];  //fetch query
connection.
                $db_name = $_SESSION[$queryId]['db_name'];
                //mysql_select_db($db_name,$curr_con) or die("Unable to
select db2 :" . mysql_error());
                pg_query($curr_query,$curr_con);
                echo("<tr><td>Done Query No : </td><td>" . ($i+1) .
mysql_error());
                echo("</td></tr>");
                //check of last execution was successful or NOT.
                if(pg_last_error())
                 {
                        echo("Some error occured while migrating data to one
of the table...<br> Process is being ROLLBACK and aborted.....");
                        pg_query("ROLLBACK");
                        exit;
                 }
        }//end of for loop
        pg_query("COMMIT");        //apply the change to database tables;
        echo("</table></center></body>All the data migrated successfully to
destination tables..");
        session_destroy();
    }//end of if
    else
        echo "Process Terminating. Please repeat the migration process.";
    //function to connect to mysql database. location specifies the source or
destination details. function returns connection.
    function db_connect_mysql($str_qNo)
    {
        $db_server = $_SESSION[$str_qNo]['db_server'];
        $db_portno = $_SESSION[$str_qNo]['db_portno'];
        $db_usr = $_SESSION[$str_qNo]['db_usr'];
        $db_pwd = $_SESSION[$str_qNo]['db_pwd'];
        $db_tech = $_SESSION[$str_qNo]['db_tech'];
```

```
		$con = mysql_connect($db_server . ":" . $db_portno,$db_usr,$db_pwd,true) or die ("..ERROR: Could not connect to MYSQL" . mysql_error());
		return $con;
	}//end of function
	//function to connect to postgresql database. location specifies the source or destination details. function returns connection.
	function db_connect_pgsql($str_qNo)
	{
		$db_server = $_SESSION[$str_qNo]['db_server'];
		$db_dbname = $_SESSION[$str_qNo]['db_name'];
		$db_portno = $_SESSION[$str_qNo]['db_portno'];
		$db_usr = $_SESSION[$str_qNo]['db_usr'];
		$db_pwd = $_SESSION[$str_qNo]['db_pwd'];
		$db_tech = $_SESSION[$str_qNo]['db_tech'];

		$con = pg_connect("host= " . $db_server . " port = " . $db_portno . " dbname= " . $db_dbname . " user= " . $db_usr . " password= " . $db_pwd) or die
		("..ERROR: Could not connect to PostgreSQL" . pg_last_error());

		return $con;
	}//end of function
?>
```

Publications

International Level

Sr. No.	Name of Journal/Organization	Year of Publication	Title of Paper
1.	International Marketing Conference, IES – MCRC, Mumbai	2009	"Applying SOA in the Retail Sector" – An Innovative Trend in Marketing
2.	Journal of Business & Retail Management Research Vol.4, Issue.2, April 2010. ISSN 1751-8202, United Kingdom	2010	Transforming the point of sale to point of service - applying SOA in the Indian retail scenario
3.	International Journal of Computer Applications Vol 3 No 2, ISBN: 97893-80746-29-6, FCS, USA	2010	From Data Handling to Presentation of Data : Encapsulating the App.config in .NET Applications through Design Patterns
4.	International Journal of Advanced Computer Science and Applications(IJACSA), Volume 2 Issue 8, U.S ISSN : 2156-5570 (Online)	2011	XML Based Representation of DFD
5.	International Journal of Computer Application (0975 – 8887), Volume 63– No.12, FCS, USA	2013	Need of Cloud Based System for Higher & Technical Education in India

National Level

Sr. No.	Name of Journal/Organization	Year of Publication	Title of Paper
1	National Conference "Information Integrity Research, Education & Technology Meet 2008" (IIRETM'2008) 8th & 9th November, 2008	2008	Application of Design Patterns in .NET Applications
2.	National Conference SEIS 09, Acropolis Institute of Technology and Research, Indore.	2009	Study of Administration and Operation of Distributed Databases and need for Applicability of SOA for the same
3.	National Conference on Software Engineering and Information Security (NCSEIS10) 23rd - 24th December 2010	2010	XML Based Representation of Data Flow Diagrams for removal of ambiguity in their diagrammatic representation.
4.	Journal of Computer Science and Application (JCSA)., Volume 4, Number 2, (May, 2012) ISSN 2231-1270	2012	Study of Administration and Operation of Distributed Databases and need for Applicability of SOA for the same
5.	Indian Journal of Computer Science and Engineering (IJCSE), ISSN : 0976-5166 Vol. 4 No.1	2013	Applicability Of Cloud Computing In Academia

Bibliography

[1] “SODDA – A Service-Oriented Distributed Database Architecture” by Breno Mansur Rabelo and Clodoveu Augusto Davis Jr. published in ICEIS 2008 - Proceedings of the Tenth International Conference on Enterprise Information Systems, Volume DISI, Barcelona, Spain, June 12-16, 2008

[2] "AutoGlobe: An Automatic Administration Concept for Service-Oriented Database Applications" published by Stefan Seltzsam, Daniel Gmach, Stefan Krompass, Alfons Kemper in Proceedings of the 22nd International Conference on Data Engineering (ICDE’06) IEEE Computer Society

[3] “System Development As A Wicked Problem" by Raymond T. Yeh in International Journal of Software Engineering and Knowledge Engineering (IJSEKE) Volume: 1, Issue: 2 (1991) pp. 117-130

[4] Service-Oriented Architecture and Microsoft .NET, posted by John Charles Olamendy on January 21, 2006 in 'Strategies for Managing Computer Software Upgrades' and edited by Neal G. Shaw

[5] “Global Viewing of Heterogeneous Data Sources” by Silvana Castano, Valeria De Antonellis and Sabrina De Capitani di Vimercati in IEEE Transactions On Knowledge And Data Engineering, Vol. 13, No. 2, March/April 2001

[6] "ReST-Based Web Access to Learning Design Services" by Juan Manuel Dodero and Ernie Ghiglione published in IEEE Transactions On Learning Technologies,Vol. 1, No. 3, July-September 2008

[7] Distributed database From Wikipedia, the free encyclopedia and "Oracle Database Concepts, 11g Release 2 (11.2)" by Ashdown, Lance; Kyte, Tom with Oracle Corporation and retrieved 2013-07-17.

[8] Service-oriented architecture From Wikipedia and "SOA Reference Architecture Technical Standard : Basic Concepts" by opengroup and retrieved 2014-10-10

[9] An introduction to SOA By Raghu R. Kodali in JavaWorld on Jun 13, 2005 http://www.javaworld.com/article/2071889/soa/what-is-service-oriented-architecture.htm

[10] "Software-as-a-Service Revenue Models" by Arto Ojala Published by the IEEE Computer Society May/June 2013

[11] "On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software" by Muhammad Umair Ahmed Khan and Mohammed Zulkernine in 2009 33rd Annual IEEE International Computer Software and Applications Conference

[12] “Understanding Cloud Computing Vulnerabilities” by Bernd G robauer, Tobias Walloschek and Elmar Stöcker Copublished By The Ieee Computer And Reliability Societies March/April 2011

[13] "Research on Software Evolution Processes in Asia" in IEEE ICSM 2008 published by Tong Li

[14] "The Case for Cloud Computing" in IEEE Computer Society 2009 published by Robert L. Grossman

[15] Oracle® Database Administrator's Guide 11g Release 1 (11.1) http://docs.oracle.com/cd/B28359_01/server.111/b28310/dba002.htm#ADMIN11020

[16] IBM Knowledge Center on Distributed Database Concepts http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_53/rzahf/rzahfmultipledb.htm

[17] Various BI / MIS Reports published by All India Council For Technical Education, New

Delhi http://www.aicte-india.org/downloads

[18] “Methodology for Web Services Adoption Based on Technology Adoption Theory and Business Process Analyses” by An Liping, Yan Jianyuan, Tong Lingyun Tsinghua Science And Technology ISSN 1007-0214 20/26 pp383-389 Volume 13, Number 3, June 2008

[19] “Realizing Service-Centric Software Systems” by Olivier Nano and Andrea Zisman, City University, LondonIEEE Computer Society 0740-7459/07/2007 IEEE.

[20] "Performance and cost evaluation of an adaptive encryption architecture for cloud databases" published in IEEE Transactions on Cloud Computing by Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and Mirco Marchetti.

[21] “Architected to Last: The Expanding Relevance of Service-Oriented Architecture” sponsored by IBM, authored by Stephen D. Hendrick in April 2011

[22] "Successful Software Product Line Practices" by John D. McGregor, Dirk Muthig, Kentaro Yoshimura, Paul Jensen published in IEEE Computer Society 0740-7459/10/2010

[23] "Need of Cloud Based System for Higher & Technical Education in India" in International Journal of Computer Application (0975 – 8887), Volume 63– No.12, FCS, USA by Swapna Kolhaktar, A K Ajay, Atul Naik

[24] "Applicability Of Cloud Computing In Academia" by Amarendra Kumar Ajay, Swapna S Kolhatkar and Atul B Naik in Indian Journal of Computer Science and Engineering (IJCSE) Volume 4 Issue 1 February-March 2013 e-ISSN:0976-5166 p-ISSN:2231-3850 Vol. 4, No. 1

[25] “XML Representation of DFD - Removal of Diagramming Ambiguity” in International Journal of Advanced Computer Science and Applications(IJACSA), Volume 2 Issue 8, U.S ISSN : 2156-5570 (Online) by Swapna Kolhatkar

[26] “Study of Administration and Operation of Distributed Databases and need for Applicability of SOA for the same” by Swapna Kolhatkar and Prof Harshad J Patel in Journal of Computer Science and Application (JCSA)., Volume 4, Number 2, (May, 2012) ISSN 2231-1270

[27] “The intelligent organisation” by Passionned Group in Business Intelligence book http://www.passionned.com/data-integration/etl-extract-transform-and-load

[28] A white paper on Microsoft Azure StorSimple : The Next Step for Hybrid Storage By Mark Peters, Senior Analyst July 2014

[29] Paper Trends 2006 : Retail IT highlights how different aging POS platforms need up-gradation and architecture investments will pay off in the end http://whitepapers.techrepublic.com.com/abstract.aspx ?docid=166761

[30] “An Overview of Service Oriented Architecture in Retail” a white paper published by TechRepublic looks at the challenges faced by retailers and benefits of SOA to the retail enterprise

[31] “Creating Sustainable Retail Solutions with SOA” by Annika Granebring , Péter Révay , Ph D C

[32] “SOA –the Architecture of choice for demand driven retail” published by TechRepublic

[33] “Driving Innovation in retail with SOA” by Eqos Limited in March_2008

[34] Design Patterns in C# and VB.NET - Gang of Four (GOF) www.dofactory.com/Patterns/Patterns.aspx.

[35] Design Patterns for .Net published by Designpatternsfor.net

[36] White paper on “Structural Design Patterns and .NET Framework 2.0” published by TechRepublic

[37] “Enterprise Solution Patterns Using Microsoft .NET” published by MSDN, MicroSoft library

[38] Article on “Storing database connection settings in .NET” published by TechRepublic

[39] “.Net Architecture and Design Pattern” published by theserverside.net

[40] “Design Patterns: From Analysis to Implementation” published by NetObjectives

[41] “Design pattern” Resources published by TechRepublic search.techrepublic.com.com/search/design+pattern.html

[42] A Beginner’s Guide to Design Patterns by Nikko Bautista and Singleton Design Pattern – An Introspection w/ Best Practices by Mainak Goswami

[43] “An approach to teaching design patterns using musical composition” by John Hamer

[44] "C# 3.0 Design Patterns: Use the Power of C# 3.0 to Solve Real-World Problems" by Judith Bishop

[45] “Create a Microsoft Access Database Using ADOX and Visual Studio” published by CodeGuru. wwww.codeguru.com/vb/gen/vb_database/microsoftaccess/article.php/c5149

[46] "A Multidimensional Weighted-Attributes Framework (MWAF) For Evaluating Agent-Oriented Software Engineering Methodologies" by Abdel-Halim Elamy and Behrouz Far in IEEE CCECE/CCGEI, Ottawa, May 2006

[47] “Towards a Better Understanding of Software Evolution: An Empirical Study on Open Source Software” by Guowu Xie, Jianbo Chen and Iulian Neamtiu in 2009 IEEE Proc. ICSM 2009, Edmonton, Canada

[48] "Securing XML Documents with Author-X" published in IEEE Internet Computing In May June 2001 by Elisa Bertino, Silvana Castano and Elena Ferrari

[49] "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center" by Guohui Wang and T. S. Eugene Ng published in IEEE Infocom 2010 proceedings

[50] "Understanding Cloud Computing Vulnerabilities" by Bernd Grobauer, Tobias Walloschek, and Elmar Stöcker Copublished By The IEEE Computer And Reliability Societies March / April 2011

[51] “Microsoft in Education, Teacher Guides” published by Microsoft in Education Ref - http://www.microsoft.com/education/en-us/teachers/guides/Pages/webapps.aspx

[52] "There is a greater need for automation & process management in Indian educational institutes" by Sharon Lobo on Feb 28, 2011 published in Cxotoday.com

[53] “Microsoft in Education, Education Apps” published by Microsoft in Education http://www.microsoft.com/education/en-us/solutions/Pages/cloud_computing.aspx

[54] "Automating Cloud Services Life Cycle through Semantic Technologies" published in IEEE Transactions On Sercies Computing, Vol. 7, No. 1, January-March 2014 by Karuna P. Joshi, Yelena Yesha, and Tim Finin

[55] Microsoft LiveEdu http://www.microsoft.com/india/education/solutions/liveedu.aspx

[56] "SOA in Variability-Intensive Environments: Pitfalls and Best Practices" by Matthias Galster, Laurens Lapre and Paris Avgeriou

[57] "Heterogeneity in Mobile Cloud Computing:Taxonomy and Open Challenges" by Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani and Rajkumar Buyya published in IEEE Communications Surveys & Tutorials, VOL. 16, NO. 1, First Quarter 2014

[58] India Education Review - “We have discussed a lot, now it's time to act” by Sam Pitroda, Advisor to Prime Minister of India

[59] “Introduction to Cloud Computing” by William Voorsluys, James Broberg, and Rajkumar Buyya Ref - http://media.johnwiley.com.au/product_data/ excerpt/90/04708879/0470887990- 180 .pdf

[60] “Gartner Says Cloud Computing Will Be As Influential As E-business” STAMFORD, Conn., June 26, 2008 http://www.gartner.com/newsroom/id/707508

[61] “A Reference Model for Developing Cloud Applications” byMohammad Hamdaqa,

Tassos Livogiannis and Ladan Tahvildari of Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada

[62] Michael Keen's Business Technology Review : Viewpoints on Business, Innovation, Strategy, and of course, Technology posted on March 09, 2009 at DABCC Ref : http://www.dabcc.com/article.aspx?id=10100

[63] "Building a Service-Oriented eBanking Platform Services Computing" published in 2004 IEEE International Conference on (SCC'04) Shanghai, China dated, September 15-September 18, ISBN: 0-7695-2225-4 by Tony Chao Shan, Wachovia Corporation, Charlotte, NC

[64] Statistical reports - Department of Higher Education, Ministry of Human Resource Development, Government of India. Reports like CollegeDCF_2.pdf, EduStatGlance_2011_0.pdf, India_CountrySummary.pdf, UniversityDCF_2.pdf, StateProfile0506.pdf

[65] Bell, Michael (2008). "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. p.3.ISBN 978-0-470-14111-3.

[66] "Incorporation of ERP in Educational Institutions: An Empirical Study" by Karminder Ghuman, Sonika Chaudhary, Chandigarh Business School, Mohali in International Conference on Technology and Business Management dated March 26-28. 2012

[67] "A Novel Executable Modeling Approach for System-of-Systems Architecture" by Bingfeng Ge, Keith W. Hipel, Kewei Yang and Yingwu Chen published in IEEE Systems Journal, Vol. 8, No. 1, March 2014

[68] "System of Systems for Quality-of-Service Observation and Response in Cloud Computing Environments" published in IEEE Systems Journal by Paul C. Hershey, Shrisha Rao, Charles B. Silio and Akshay Narayan

[69] Wiki answers.com : The_advantages and disadvantages of data flow diagrams

[70] Chris Gane and Trish Sarson. Structured Systems Analysis: Tools and Techniques. McDonnell Douglas Systems Integration Company, 1977

[71] "The Relevance of the XML Data Format for Access to Historical Datasets and a Strategy for Digital Preservation" published in D-Lib Magazine, February 2005, Volume 11 Number 2, ISSN 1082-9873, The eXtensible Past

[72] "Thinking XML: A glimpse into XML" in the financial services industry Interesting lessons from a no-nonsense territory for XML by Uche Ogbuji (uche@ogbuji.net), Principal Consultant, Fourthought, Inc.

[73] "An introduction to Model Driven Architecture Part I: MDA and Today's Systems" published by Alan Brown with IBM http://www.ibm.com/developerworks/rational/library/3100.html

[74] "The Importance of Data Modeling as a Foundation for Business Insight", an article published by Larissa Moss and Steve Hoberman in Search SSWUG.ORG

[75] "Comparison of Approaches to Service Deployment" by Vanish Talwar, Qinyi Wu, Calton Pu, Wenchang Yan, Gueyoung Jung and Dejan Milojicic; published in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS'05)

[76] "An Enhanced Extract-Transform-Load System for Migrating Data in Telecom Billing" published in IEEE ICDE 2008 by Himanshu Agrawal, Girish Chafle, Sunil Goyal, Sumit Mittal, Sougata Mukherjea

[77] "Expanding the Criteria for Evaluating Socio-Technical Software" in IEEE Transactions On Systems, Man, And Cybernetics—PART A: Systems And Humans, Vol. 38, NO. 4, JULY 2008 published by Brian Whitworth, Victor Bañuls, Cheickna Sylla, and Edward Mahinda

[78] "Cost Analysis of the Service Migration Problem between Communication Platforms" in 2008 IEEE published by Dayyan Shayani, Carmen Mas Machuca, Monika Jager and

Andreas Gladisch

[79] SQLMaestro Database Management Solutions by SQL Maestro Group
Ref : http://www.sqlmaestro.com/products/anysql/maestro

[80] Data Migration Through Toad software
http://data.software.informer.com/download-data-migration-through-toad

[81] Aqua Data Studio Documentation by AquaFold

[82] Redgate SQL Monitor Documentation by Redgate

[83] "From Data Handling to Presentation of Data : Encapsulating the App.config in .NET Applications through Design Patterns" in International Journal of Computer Applications Vol 3 No 2, ISBN: 97893-80746-29-6, FCS, USA by Swapna Kolhatkar

[84] "Transforming the point of sale to point of service - applying SOA in the Indian retail scenario" in Journal of Business & Retail Management Research Vol.4, Issue.2, April 2010. ISSN 1751-8202, United Kingdom by Shruti Joshi and Swapna Kolhatkar

[85] TIBCO services and Documentation

[86] The Open Group Documentation

[87] DbVisualizer Ref : http://www.dbvis.com

[88] Squirrel SQL Universal SQL Client Version 3.6
Ref : http://squirrel-sql.sourceforge.net

[89] TOAD Productivity Software for Database Developers, DBAs, and Analysts
Ref : https://www.toadworld.com

[90] “Cloud computing” from Wikipedia and "The NIST Definition of Cloud Computing" by National Institute of Standards and Technology and retrieved 24 July 2011

[91] w3schools for XML, DTD, XSD, "XML 1.0 Specification" by World Wide Web Consortium & retrieved 2010-08-22; Watt, Andrew H's “Sams teach yourself XML in 10 minutes” with Sams Publishing

[92] “An E-learning tool for Database Administration” by Paul Douglas and Steve Barker in Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’05)

[93] “SOA Monitoring for Enterprise Computing Systems” by Paul Hershey and Donald Runyon in 11th IEEE International Enterprise Distributed Object Computing Conference

[94] An Overview of Service-Oriented Architecture in Retail by Moin Moinuddin, Microsoft Corporation, January 2007

[95] Case Study: SOA Retail Business Pattern – Martin Keen, Kulvir Singh Bhogal, Sunil Dube, Rashmi Kaushik, Geert Van De Putte, Albert Wong, Sravan Yallapragada

[96] Published in *the Vision* 2000. 4(1), 35-40. Retail Management In India : Some Global Issues By Mrinmoy K Sarma School Of Management Sciences Tezpur University, Assam

[97] Challenges facing the Indian Organized Retail sector - Maps of India.com's India Business Directory

Books

[1] Service-Oriented Architecture by Thomas Erl

[2] Principles of Distributed Database System by Ozsu M Tamer and Patrick Valduriez

[3] Professional Ajax, 2nd Edition by Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett

[4] PHP: The Complete Reference by Steven Holzner

[5] MySQL: The Complete Reference by Vikram Vaswani

[6] Beginning Databases with PostgreSQL, 2nd Edition by Neil Matthew & Richard Stone

[7] Fundamentals of database systems (3rd edition) by Elmasri and Navathe

[8] Professional ASP.NET 4 in C# and VB by Scott Hanselman, Devin Rader, Bill Evje

[9] Executing SOA: A Practical Guide for the Service-Oriented Architect Norbert Bieberstein, Robert Laird, Keith Jones and Tilak Mitra