Chapter 1 – Introduction to Database Integration Tool

1.1 Project Title and Interpretation

The goal of this project is to research, analyze, design and implement a software tool which can interconnect different databases on different Database Management Systems (DBMS). Product is expected to provide a single access point for data, collected by different organizational information systems and to integrate that data to obtain meaningful information without disturbing local autonomy.

1.2 Project Motive

To meet the growing business demands, and to operate flexibly and efficiently, and also to comply with regulatory and other requirements, enterprises need access to data from a wide variety of sources, including heterogeneous databases. Heterogynous Database Integration Systems involve combining data dwell in different data sources and providing users with a unified view of these data. Use of this data is often made more challenging by requirements such as restrictions on making copies or extracts of the data, ongoing access to real-time updates and providing centralized data views that combine real-time data from operational systems with historical information from data warehouses.

Data Integration is ideal for companies seeking a flexible, scalable and efficient way to integrate and deliver data from diverse distributed sources to key enterprise applications that require fresh data consolidated from multiple operational systems. Examples of these applications include co-operate performance management, integrated customer service, risk management, regulatory and financial reporting and a variety of government and military applications.

1.3 Project Objectives and Goals

1.3.1 Objective

Objective of the project is to discover a mechanism to integrate databases easily and efficiently, and to provide more user friendly environment to increase organizational data availability.

1.3.2 Goals

Based on above objectives project goals can be stated as follows.

- Develop a database environment independent platform for data integration.
- Develop a Schema integration Mechanism based on above platform.
- Open an easy channel for data migration between Heterogeneous databases.
- Develop a Data archiving Mechanism.
- Develop an open source database management tool with the above major functionalities.

1.4 Product Features

This project is mainly focusing on database integration and utilizes the derived advantages.

- Federated query processing
- Data migration between databases
- More reliable and manageable data archiving method

Developed Product will be a tool that runs on a server or a single node. And it can connect with several databases through plug-ins as shown in figure 1.1.

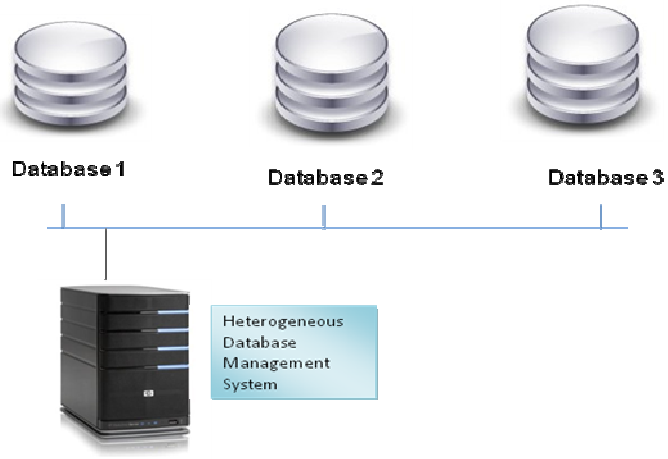

Figure 1.1 - Database Integration Tool connecting to Heterogeneous databases

Database integration system adheres to adapter design pattern which general interfaces to databases are converted to Database Management system dependent interface, using plug-ins (Figure 1.2).

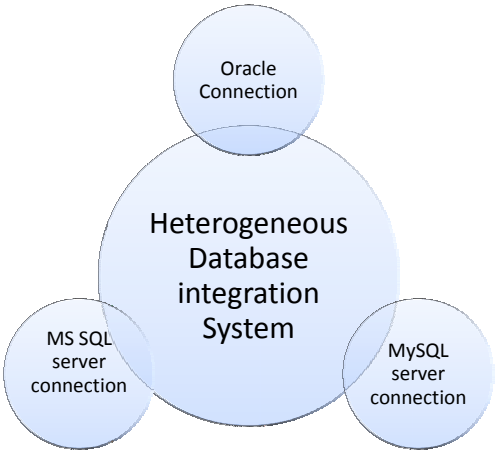

Figure 1.2 - Connection plug-ins

Each adapter has to implement a defined interface using database management system specific JDBC (Java Database Connectivity) driver. This interface is used to grab the database specific information.

This tool will be made open-source software. So any interested party can develop plug-ins (Figure 1.2) that complies with the tool. This will allow them use this tool to connect to any other Database Management system.

1.4.1 Database integration

Most important and the groundwork utility of this tool is the database integration. Tool provides a user friendly graphical user interface to create federated schemas and data retrieval.

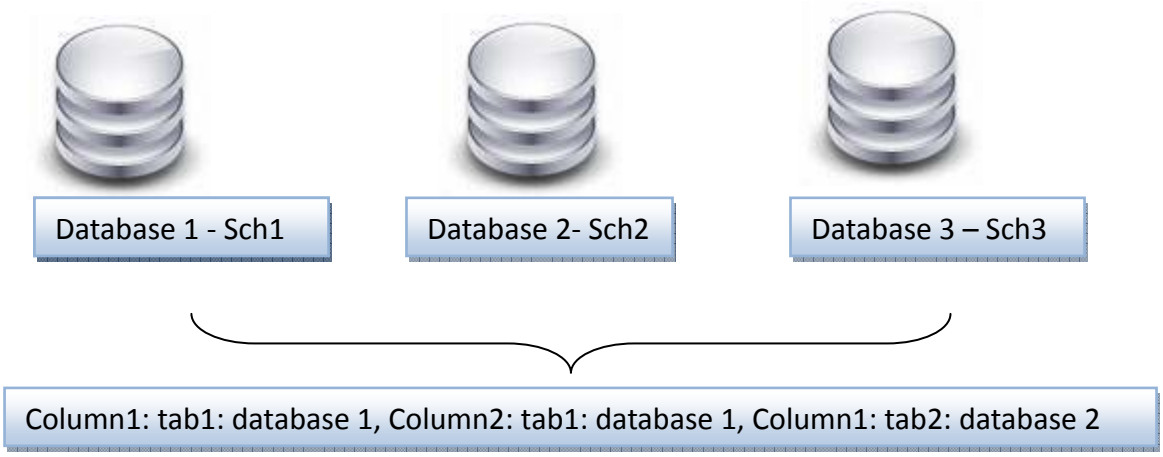

Figure 1.3 – Federated Schema

User has to connect to database schemas to the system using unique connection names. Each connection is treated as a separate schema with own collection of tables. Each column in a connection is treated as a separate object. In federated schema each selected column is identified as DATABASE: TABLE: COLUMN (Figure 1.3). When integration process is running required data is extracted from each database and tool itself join that data to make the federated view .Our Database integration-platform separates the required data set from each connected database.

Federated schema is stored in XML (Extendible Markup Language) format. Stored XML file describes the required data from each database. And the tool provides functionality to customize data retrieval from each database. It also provides an external interface that can export integrated result set.

Federated data set retrieves real time data from the connected database. User can store retrieved data set in the intermediate storage or export into an XML file. Stored XML file can be compressed using XMILL.

XMILL provides a high compression level compared to the other existing compression techniques. As an extended feature XML files can be encrypted using a defined key.

System embeds a database to provide rich set of functions to manipulate intermediary stored data.

1.4.2 Extract Transform Load (ETL)

Integrating distributed data among organizational information systems cannot always accomplish by schema integration. Therefore Database Integration Tool enhanced by amalgamating ETL capabilities.

Computerworld - **ETL** stands for **extract, transform and load**, the processes that enable companies to move data from multiple sources reformat and cleanse it, and load it into another database, a data mart or a data warehouse for analysis, or on another operational system to support a business process.

- Extracting data from outside sources
- Transforming it to fit operational needs (which can include quality levels)
- Loading it into the end target (database or data warehouse)

Connected databases can be query and results can be saved internally or transmit to another connected database to make more meaningful information.

1.4.3 Data exchange between databases by schema mapping.

One of the main utility of a tool that connects several data repositories should have data transmission. Integrated database management system provides a rich function to perform this task.

Results of a federated query or XML export of data can be transmitted to a connected database. In this process destination data structure has to be mapped with the source data structure. If destination data structure is not available then the system provides the facility to create required destination structure.

1.4.4 Database archiving method.

Data will be stored in self discretionary manner and compressed for later use. And those exports are compressed and encrypted. Exported data can be restored to any compatible destination.

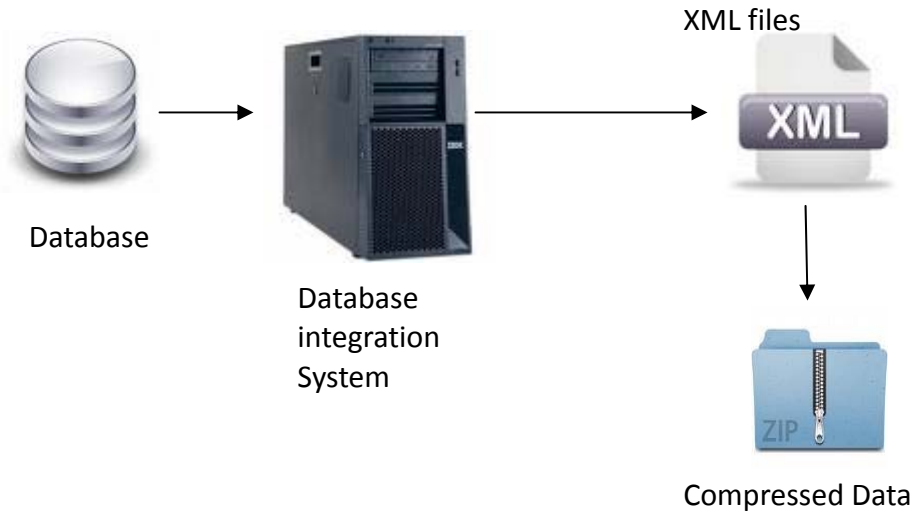

Figure 1.4 – Data export and archiving method

1.5 User Classes and Characteristics

System will anticipate following users

A. Database administrators
B. Database Developers who implement interface specification for a particular database management system.

1.5.1 Database administrators

Database administrators are the technical users of the system. They will add databases to the system through plug-ins. Database administrators would export or import data among connected databases. Or integrate database sachems to export data

In case of export or import schema mapping can be done either prior to the export or before importing data.

1.5.2 Developers

Proposed system is developed as open-source extendable tool. So community developers are expected to change, improve and extend the product. Plug-in development is

primarily expected from developers. Therefore interface between proposed system and the database management systems should be well defined.

1.6 Open-Source Development Environment

Database Integration System is published as an open-source product under GNU GPL [1](GNU General Public License). Project is initially started (2008) with www.sourceforge.net. "www.sourceforge.net" is the world's largest open-source software development website.

Database Integration System is a significant project in the Sourceforge community. This project ranked to second place among Database Integration Projects.

Project web – http://ihdbm.sourceforge.net

1.7 What next

This document is intended to provide a comprehensive and broad view of Database Integration Tool.

Second chapter of this document provides a literature review about the related methodologies of the product. Especially it concentrates on the database integration. It contains comprehensive survey on historical approaches and current trends of the database integration. Also second chapter stated an analysis report about similar products available in the software industry.

---

[1] The GNU General Public License is a Free Software license. Like any Free Software license, it grants to you the four following freedoms:

1. The freedom to run the program for any purpose.
2. The freedom to study how the program works and adapt it to your needs.
3. The freedom to redistribute copies so you can help your neighbor.
4. The freedom to improve the program and release your improvements to the public, so that the whole community benefits.

Third chapter, Requirements and Analysis chapter is about the detail and specific requirements of the product. It is a complete description of the behavior of the developed system. It stated functional, non functional and constraints of the database integration tool in detail. Also requirements are break down in to more manageable and understandable units.

In the forth chapter we have explained the design techniques used in Database Integration System. System detailed design is done using Unified Modeling Language. Database Integration system is and Open-Source product. Therefore in this chapter have stated architectural concepts in detail for better understanding of the product.

In the implementation chapter which is fifth chapter explains all major code and module structures.

Chapter 2 – Database Integration Background

2.1 Literature Review

Integration of Heterogeneous databases is the main research area in this project.
By literature review on Integrating of Heterogeneous databases we can find that, there have been various attempts for database integration from different perspectives. The work broadly classifies these attempts into three categories [2]: *Structural approaches, Semantic approaches and intelligent approach.*

The project described here follows an approach more close to the Structural data integration [14] and uses special techniques to overcome inherent drawbacks.
Latter section of this chapter reviews about available tools which become more realistic in database integration systems. And interestingly all are commercial products.

2.1.1 Structural approaches

This Approach also called the common data model approach, as participating databases are mapped to a common data model [51]. Defining a global schema was the earliest structural approach to integrate heterogeneous databases. This process was very labor intensive and requires a database administrator to play beyond his role and understand all under laying database systems and their architectures. He has to understand what is being integrated and how to integrate it, and was the major drawback of the approach. Following this approach, multi database languages such as MSQL[2], IDL, and DIRECT was introduced to allow users to define user views to solve the conflicts that cannot be resolved through algorithmic approach.

[2] Multi database extension for SQL

Carlo Batini [51] was a person done a comprehensive survey on structural integration. He explained the integration process in five steps: pre integration, comparison, conformation, merging and restructuring

Another most widely used approache for integration under structural approach is, co-operate autonomous component database systems or Federated Database System (FDBMS). This was first proposed by O. *Heimburger*, A. *McLeod* [45] to refer a collection of databases in which the sharing is made more effective by allowing export of schemas which define sharable part of each local database to be integrated into a global schema. A federated database management system provides controlled and coordinated manipulation of the component database systems. FDBMS[3] represents a compromise between no integration and total integration.

Sheth and Larson [44] define a classification for federated database systems into two categories: loosely coupled and tightly coupled. In loosely coupled systems users are responsible to creating and maintaining the federation and in tightly coupled systems Administrators have total responsibility to maintain the federation. Also they have defined a reference architecture that can support integration. Data, Database, Commands, Processors, Schema and Information mapping are the main components of the database federation architecture. According to the FDBMS there are five schema levels imposed on structural approach: Local schema of a database, a component schema, export schema that each local database can export and a federated schema that represents entire schema, and external schema that represents the view of each user.

Spaccapietra [46] view integration approach to solve most integration problems. This methodology allows users to define their own real world objects using ERC+[4] which is an

[3] The term 'federated database system' (FDBS) was first proposed by Heimburger and McLeod to refer to a collection of databases in which the sharing is made more explicit by allowing export schemas which define a sharable part of each local database to be integrated into a global schema.

[4] ERC+ meets database application requirements by merging traditional semantic data models features with object oriented capabilities such as structural object orientation, inheritance, and object identity.

enriched relationship model to include generalization. ERC+ allows users to build an attribute tree whose root is a real world object. Also this approach assumes that user has required knowledge to identify and build views.

Completely different approach was taken by *Geller* [47]. In his approach he moves away from using semantics to identify objects. In this approach structural integration is achieved based on the structural similarities of the objects even if they differ semantically. This approach uses Object oriented – dual[5] model. However this was not suited for legacy systems since they systems are not object oriented.

All the above presented approaches have a common underlying principle which is, all uses a form of a common model to represent user views and performed some translation of local schema to global schema.

Structural approach is difficult and cumbersome due to its inherent embedded semantics within each local or global schema.

2.1.2 Semantic approaches

Semantic approaches use a higher order language that can express information ranging over individual databases. Ontology based integration approaches belong to this category. Many research projects (SHOE, ONTO Broker, OBSERVER) [45] and others use ontologies to create a global schema.

Many authors reintroduced semantic integration as the hardest to understand and resolve context .Same as the structural approach there are several other models that have been proposed in semantic model. *Sheth and KashYep* [48] presented a semantic approach that demonstrated semantic similarities between two objects and relate this to structural classification.

[5] Structure and semantics represented separately

In literature about semantic approach, various types of relations have been discussed. e.g. semantic equivalence, semantic resemblance, semantic compatibility, semantic discrepancy, semantic reconciliation and semantic relativism [52].

There are several approaches that addressed semantic integration. And one major project that addressed major issues in building federated schema was the FEMUS project [53] of the Swiss institute of technology. The FEMUS project has experimented with two different canonical models: ERC+ and COCOON. A repository is used to help the federation.

Another important project is the Pegasus project at HP laboratories. A number of semantic integration techniques are proposed as part of the project. The approach suggested by *Kent* [49] is to perceive a database not just as a model of real world entities, but as a repository of knowledge. In this context, a multi-database system is modeled as a collection of knowledge repositories.

Also *M.W. Bright* [54] [55] has developed a summary schema model as an extension to multi-database systems to provide linguistic support to automatically identify semantically similar entities with different access terms.

The approach used by Weigand [50] was deviated from other approaches as the focus change from data semantics to the semantics of communication. Proposed systems identify communication structure as the essential components for interoperability and propose integrated semantics for information and communication semantics. Defined framework address the problem of semantic in more realistic way by including the semantics of massages in addition to the data.

Edward Sciore [56] presented the theory of semantic values as a unit of exchange to facilitate semantic inter-operability. He provides a theory of semantic values as a unit of exchange that facilitates semantic interoperability between heterogeneous information systems and uses that semantic value to convert from one context to another.

One of the major drawbacks of this approach is semantically similar piece of data may have deferent representation; may be differ on names, data types and relations.

Common characteristic that all semantic approaches adhere is, semantics are extracted using knowledge of entire application domain

2.1.3 Intelligent Integration approach

Two major ideas or concepts are proposed in the literature to achieve intelligent integration. The first is based on establishing some form of intelligent co-operation between heterogeneous systems by transforming passive information systems into intelligent information processing agents. The second approach is based on the concept of mediators that intermediary service by linking data resources and application programs.

Both information agents and mediators require background knowledge of the application domain. This knowledge has to be “discovered” from existing applications to provide the required intelligence needed for integration.
One of the active research areas dealing with intelligent integration is based on the paradigm of intelligent and Co-Operative Information Systems. The design goal was the approach is that any computing resource should be able to transparently and efficiently utilize the resource.

In past several years, many systems have been developed in various research projects on data integration using the techniques mentioned above.

Here are some of the more prominent representative systems:

- Pegasus takes advantage of object-oriented data modeling and programming capabilities. It allows the user to access and to manipulate multiple autonomous heterogeneous distributed object-oriented relational and other information systems through a uniform interface.

- Mermaid uses a relational common data model and allows only relational schema integration.
- Clio was developed by IBM around 2000. It involves transforming legacy data into a new target schema. Clio introduces an interactive schema mapping paradigm, based on value correspondences.
- Garlic uses an ODMG-93 based object oriented model. It extends ODMG to allow modeling of data items in the case of a relational schema with weak entity.
- TSIMMIS and Med Maker were developed at Stanford around 1995. They use the Object Exchange Model (OEM) as a common data model. OEM allows irregularity in data. The main focus is to generate mediators and wrappers based on application specification.
- MIX a successor of TSIMMIS, uses XML to provide the user with an integrated view of the underlying database systems. It provides a query/browsing interface called Blended Browsing and Querying. These were the prominent techniques in the structuring approach. There are many other techniques which use ontology as a common data model or use ontologies to translate queries over component databases. Below we present some of these techniques:
- Information Manifold employs a local-as-view approach. It has an explicit notion of global schema/ontology.
- The OBSERVER system uses a different strategy for information integration. It allows individual ontologies and defines terminological relationships between them, instead of creating a global ontology to support all the underlying source schemas.

2.2 Software Market Analyses

Form software market place we can find very few sophisticated products developed with the same objective as this project. And all products are commercial and non open source.

Sybase [32] Data Integration Suite, a flexible and scalable solution that combines key data integration techniques including replication, data federation, ETL (Extract, Transform and Load), real-time events and data search with integrated development and administration.

Key features of the tool are,

- Access to multiple, diverse data sources, and the ability to create a single, integrated view of data.
- Access to a variety of heterogeneous data sources, including mainframe data sources
- Capture and propagation of real-time events in data sources to applications
- Search and query of information in structured and unstructured data using context-sensitive searches
- Development of applications using Sybase Workspace
- Management of DI Suite components using a common system administration console
- Common installer which performs interactive and silent installation using a script-driven utility

Sybase integration suite and described product of this project has common features. Both products follows ETL(Extract, Transform, Load) phenomena.

Oracle [24] also provides a solution for database integration. The Oracle Transparent Gateway for DRDA enables you to:

- Integrate heterogeneous database management systems so that they appear as a single homogeneous database system
- Read and write data from Oracle applications to data in DB2/OS390, DB2/400, DB2 Universal Database, DB2/VM, and IBM SQL/DS on VM databases in addition to any Oracle Database server data.

Oracle Transparent Gateway for DRDA gives your company the ability to develop its information systems without forfeiting its investments in current data and applications. The gateway gives you access to your Oracle and DB2 data with a single set of applications while you continue to use existing IBM applications to access your IBM data. You can also use more productive database tools and move to a distributed database technology without giving up access to your current data.

If you choose to migrate to Oracle Database technology and productivity, the gateway allows you to control the pace of your migration. As you transfer applications from your previous technology to the Oracle Database, you can use the gateway to move the DB2 data into Oracle databases.

Microsoft also comes up with a solution called distributed query. Distributed Query Allows SQL Server to access any non-SQL Server data that exposes an OLE DB interface.

Altova XML [41], Data Management tool is another commercial product which is capable to integrate database and schema mapping.

Kettle is a leading open source ETL application on the market. It is also a repository based tool. It has slightly modified ETL, Kettle as it is composed of four elements, ETTL, which stands for:

- Data extraction from source databases
- Transport of the data
- Data transformation
- loading of data into a data warehouse

Kettle is 100% java tool and supports fairly large number of databases; it provides similar ETL functions as data integration tool. E.g.

- Migrating data between applications or databases
- Exporting data from databases to flat files
- Loading data massively into databases
- Data cleansing
- Integrating applications