Chapter I

Introduction

1.1 Overview

From the inception of data processing systems through the middle of the 80's decade, integration between systems was always a custom, platform-specific affair. Early Mainframes and Minicomputers relied on proprietary data storage systems via indexed flat files, closed simplistic database structures, and so on. The situation improved as packaged relational database products were introduced and system/network interoperability improved. APIs were created and exposed for customers so that they could more easily move data into and out of various hardware and software architectures. Many of these offerings were driven initially by large players (both hardware and software) at the time such as IBM®, DEC®, etc. In addition, the rise of pure, enterprise software companies which specialized in high performance databases also contributed to the development of the market at this time. Such companies included Oracle® and Sybase®.

A continual reliance on information systems with key databases combined with an exponential growth of data.

Heterogeneous Data is data from any number of sources, largely unknown and unlimited, and in many varying formats. In essence, it is a way to refer to data that id of an unknown format and/or content[1].Accessing to the accurate information in a timely manner is a significant challenge facing organizations today.[2] For example, a police officer needs to know if a suspect is wanted in another jurisdiction; A social worker needs to ensure that a welfare applicant is not already receiving benefits elsewhere; A judge needs to see all prior convictions against an offender . These and countless other situations require rapid access to a wide range of complete and accurate information that is often scattered across numerous agencies . However, the problem is that many agencies build information "silos" which are poorly accessible within their own organizations, let alone to the related departments outside the organization. Besides, many agencies seem to have a "genetically encoded political and cultural aversion to information sharing and cooperation, operating instead as isolated fiefdoms or, at best, as grudging partners" .There has been a spectacular explosion in the quantity of data available in electronic formats in the past few decades. This huge amount of data has been gathered, organized, and stored by a small number of individuals, working for different organizations on varied problems. In light of the ever increasing volume of data, and the expected benefits of integrating the data, a framework for performing integration over multiple data sources is necessary.

1.2What is Data Integration?

Data integration is the process of the standardization of data definitions and data structures by using a common conceptual schema across a collection of data sources McLeod, 1985; .Integrated data will be consistent and logically compatible in different systems or databases, and can use across time and users .Goodhue et al. [1992, p294] defined data integration as "the use of common field definitions and codes across different parts of an organization". According to Goodhue, et al. [1992], data integration will increase along one or both of two dimensions:

(1) the number of fields with common definitions and codes, or

(2) the number of systems or databases adhering to these standards. Data integration is an example of a highly formalized language for describing the events occurring in an organization's domain. The scope of data integration is the extent to which that formal language is used across multiple organizations or sub-units of the same organization. The objective of data integration is to bring together data from multiple data sources that have relevant information contributing to the achievement of the users' goals [AFT, 1997].

The Advanced Forest Technologies in Canada identified the following factors which must be addressed to integrate data properly:

- Identification of an optimal subset of the available data sources for integration
- Estimation of the levels of noise and distortions due to sensory, processing, and environmental conditions when the data are collected the spatial resolution, the spectral resolution, and the accuracy of the data.
- The formats of the data, the archive systems, and the data storage and retrieval
- The computational efficiency of the integrated data sets to achieve the goals of the users

1.3 Benefits of integrating heterogeneous data sources

There are some obvious advantages in integrating information from multiple data sources. Such integration alleviates the burden of duplicating data gathering efforts, and enables the extraction of information that would otherwise be impossible.

Subrahmanian, et al. [1996] gives the following examples of benefits of data integration:

• "... law enforcement agencies such as Interpol benefit from the ability to access databases of various national police forces, to assist their effort in fighting international terrorism, drug trafficking, and other criminal activities. Insurance companies, using data from external sources, including other insurance company and police records, can identify possible fraudulent claims. Medical researchers and epidemiologists, with access to records across geographical and ethnic boundaries, are in a better position to predict the progression of certain diseases. In each case, the information extracted from the integrated sources is not possible when the data sources are viewed in isolation."

1.4 Integrating diverse data source paradigms

Subrahmanian, et al. [1996] established a data source paradigm. There are two important aspects to constructing the data source paradigm: domain integration and semantic integration. Domain integration is the physical linking of data sources and systems, while semantic integration is the coherent extraction and combination of the information provided by the data and reasoning sources, to support a specific purpose .It is acknowledged that data warehousing is the most effective way to provide the business decision support data. Under this concept, data is derived from operational systems and external information providers, and subsequently conditioned, integrated, and changed into a read-only database that is optimized for direct access by decision makers. The term ‘data warehousing' describes data as an enterprise asset that must be identified, cataloged, and stored to ensure that users will always be able to find the needed information. The data warehouse is generally enterprise-wide in scope, and its purpose is to provide a single, integrated view of the enterprise's data, spanning all enterprise activities.

1.5 Different kinds Data integration.

We know from our definition that data integration is a set of processes used to extract or capture, restructure, move, and load or publish data, in either operational or analytic data stores, in either real time or in batch mode. Because of the operational and analytic nature of integrating data, the frequency and throughput of the data have developed into different types of data integration architectural patterns and technologies. Therefore, this section begins with an investigation of the architectural types or "patterns" of data integration.
The application landscapes of major companies all have their won complex structure .Data has to be exchanged between or distributed to the various application here different types of data integration identified and categorized.

The classification of data access types presented here is based on the assumption that an application needs specific data at a given time, in a specific format and in a required quality. The classification points out how these requirements can be met. The classification is based on how the application gets the data(interval, kinds of communication etc.), and on whether the application accesses the original data source or a copy of the data . A goal of this approach is to model an application landscape in which temporal and technical dependencies as well as redundancy appearance can be discovered. This is especially important when implementing new applications or replacing old application by new ones Fig.1 shows the identified data integration patterns:

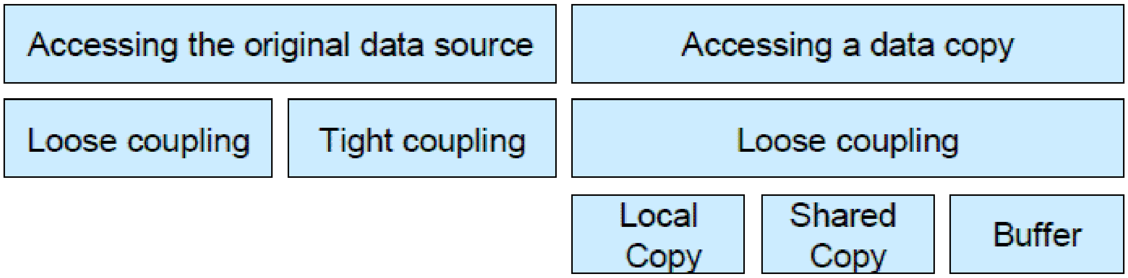


Figure 1.1 types of data integration

The types identified here can also be understood in the sense of design patterns as they are known in the object-oriented world. A patterns describes a problem which recurs regularly in our environment, and the core of the solution for this problem, so that the solution can be reused at any time.

Below redundancy-free patterns are presented (accessing the original data), differentiating between loose and tight coupled variants. subsequently, alternatives for access to data copies are presented , differentiating between three different types of data copies : local data copies, shared data copies and buffers.

1.5.1 Redundancy-free solutions

In order to avoid redundancies, the original data sources must be accessed. This is unproblematic and makes sense in some scenarios. Two different variants of how the original data can be accessed are shown here. The application can be accessed are shown here. The application can be accessed directly (tight coupling) or via a mediator(loose coupling).

The different concepts and usages scenarios are presented below, with their respective advantage and disadvantage.

1.5.1.1 Direct Access

Accessing a data source directly in only possible under certain condition .The direct call in usually made either by one of the data base management system (DBMS) involved, or by means of an API (Application Programming Interface) call. A call initiated by the database management system can only be implemented if the system is accessible and the application is not a “Black Box”. Direct access to database system is often impossible. Packaged applications usually provide an API for accessing their database systems. However, an API call only makes sense if it meets the exact requirements of the potential initiator, and the initiator application can be manipulated. An extension of APIs is not possible in most cases because the application code is not accessible. Usually this is only possible, if the software has been developed in-house. Table 1 shows usage scenarios and gives an overview of advantages and disadvantages of the direct access pattern.

Table1.1 Usage scenarios, advantages and disadvantages of the direct data integration pattern

Direct Data integration Pattern
Usage scenarios
1. industry standards are used 2. Software was developed in house (full access to source code is given).
Advantages
1. easy to implement, lowest complexity 2. No overheads
Disadvantages
1. Software components often cannot be manipulated and accessed or difficult to implement. 2. Tight coupling (lower availability ,difficult change management) 3. Locking-problems within complex transactions 4. Cross platform communication usually not possible

1.5.1.2 Data Access via Mediator:

If it is not possible or desired to access the original data directly, an application can access a source of original data through a mediator. The integration logic is transferred to the mediator, because no access to the source on the database level. The mediator essentially takes on the following tasks:

- **Transformation**: a transformation can take place on a semantic level(mapping data contents, e.g. transformation of country codes or currency codes) and on a syntactic level (transformation of different data formats or massage formats) .

-**Routing:** The routing is responsible for the distribution of messages to the applications involved. Besides, mechanisms for buffering messages are provided by the routing component.

-Composition/Decomposition:

The composition component merges several messages into one, decomposition component divides a message into several.

-Controlling: The controlling component controls the chronology and resolves dependencies.

These basic functions also appear- partly implicitly-in other approaches- although there are various terms and delimitations.

Table 1.2. Usage scenarios, advantages and disadvantages of data integration via mediator.

Data Integration Via Mediator
Usage Scenarios
1. Complex transformation, routing, composition/decomposition or controlling is required. 2. Integration of package application.
Advantages
1. No (code-)manipulation with in the affected applications is necessary. 2. Controlling complexity by encapsulation by through the mediator .
Disadvantages
1. Mediator can become very complex. 2. Increased overheads. 3. An addition component in the application landscape must be operated/managed.

1.6 Overview of the book

In this book we can see in **chapter-1 Introduction** where we can get explanation about overview, what is data integration, kinds of data integration and overview of the book **,Chapter-2 Literature review** based on several paper about Data Integration by Using Direct Access and Data Access via Mediator**,Chapter-3 Data Integration by Using Direct Access** where workflow data integration with "USHR system" developed by using PHP MySQL language,**Chapter-4 Data integration via mediator** here also developed an algorithm which is used as a mediator for integrating data and **Chapter-5 Conclusion is** about the conclusion future plan and required software which is used here data for integration **.** At the end there are some related **references**.

Chapter II

Literature review

Includes a list with brief description of documents, articles, publications, and other reference materials on data integration. The literature is obtained mostly from published documents and web site materials, and is summarized in terms of which aspects of data integration are being addressed. The World Wide Web provides a convenient medium by which to access and share information regarding data integration activities of transportation agencies.

The literature review is divided into two parts. The first part includes general literature on data integration techniques by using workflow and direct access and the second part describe data integration literature pertaining on data integration via mediator.

2.1 Data integration by using workflow

A Uniform Approach to Workflow and Data Integration[11] is present results from the application of our approach for the integration of data resources and for the reconciliation of services within the ISPIDER and the BioMap bioinformatics

projects. The BioMap integration setting demonstrates the materialized integration of structured and semi-structured functional genomics resources using XML as a unifying data model. The service reconciliation scenario discusses the interoperation of the AutoMed system with a scientific workflow tool. The work presented here is part of the ISPIDER project, which aims to develop a platform using grid and e-science technologies for supporting in silico analyses of proteomics data.

OGSA-DAI 3.0 extension packs have been ingested.[6] This includes:

- Extension pack one's document-based workflow client, data source client and advanced SQL join activities.
- Examples of dynamic resource creation activities.
- Examples of remote resources and activities that run queries on remote OGSA-DAI servers.

- The OGSA-DAI data source servlet.

I have been followed several architectural design style from OGSA-DAI.
Distributed database technology has long recognized that it is often expedient to move process to data rather than data to process, and this is also part of the rationale for the Grid[10]. With current technology, a distributed process or query involves only a small number of sites[10]. However the trend is towards increasingly distributed data; and when our queries require data from thousands or millions of sites, ubiquitous computing and ubiquitous data will become one and the same.[10]

2.2 Data integration via mediator

A mediator system allows users to pose queries against a global schema and returns answers from multiple data sources[7]. The rewriting of the user query in terms of the local sources uses mappings, which in the Local-As-View (LAV) approach, describe the source relations as views over the global schema. Among the existing algorithms that perform query rewriting in LAV, the Extended Inverse Rules Algorithm (EIRA) provides the most general approach[7]. Given a set of mappings and database facts, EIRA provides a logic program, which specifies a class of legal instances of the global system[7]. The specification of the legal instances can be used to compute certain answers for user queries that are monotone[7].

The paper "**A Flexible Model For Data Integration**" [12] is saying that there are many challenges in systems integration for architects and developers, and the industry has focused on XML, Web services, and SOA for solving integration problems by concentrating on communication protocols, particularly in regard to adding advanced features that support message flow in complex network topologies. However, this concentration on communication protocols has taken the focus away from the problem of integrating data. Flexible models for combining data across disparate systems are essential for successful integration. These models are expressed in XML schema (XSD) in Web service–based systems, and instances of the model are represented as XML transmitted in SOAP messages. In our work on the architecture of the MSDN TechNet Publishing System (MTPS), they addressed three pitfalls. We'll look at what those pitfalls are and our solutions to them, in the context of a more general problem—that of integrating customer information.

Transformation of data is considered as one of the important tasks in data warehousing and data integration[13]. With the massive use of XML as data representation and exchange format over the web in recent years, transformation of data in XML for integration purposes becomes necessary[13]. In XML data transformation, a source schema and its conforming data is transformed to a target schema[13]. Often, source schema is designed with constraints and the target schema also has constraints for data semantics and consistency[13]. Thus, there is a need to see whether the target constraints are implied from the source constraints in data transformation[13]. Towards this problem, we define two important XML constraints namely XML key and XML functional dependency(XFD)[13]. The writers then use important transformation operations to see if the source constraints are satisfied by the source document, then the target constraints are also satisfied by the target document, their study is towards the utilization of constraints data integration and data warehousing in XML[13].

The paper "**Normalization In Xml And Data Exchange**[14]" is saying that their aim in this study was to define an exchange model providing a common structure of shared Medicare data to allow a better, easier and structured communication within and between hospital information systems. Results: they realized an XML-based model that we detailed the content and the structure. Thus, seen the confidential character of Medicare data, they described an approach to secure the data transfer. they were situated regarding existing models and standards such as HL7, DICOM and the PMSI and we took into account critics made for them. Conclusion/Recommendations: The model they proposed provide a practical solution allowing a secure and structured Medicare data exchange and will serve as a summary version of the common Computerized patient record.

In the paper, "**Design of Integration Security System using XML Security**[15]"they design an integration security system that provides authentication service, authorization service, and

management service of security data and a unified interface for the management service. The interface is originated from XKMS protocol and is used to manage security data such as

XACML policies, SAML assertions and other authentication security data including public keys.

The system includes security services such as authentication, authorization and delegation of authentication by employing SAML and XACML based on security data such as authentication data, attributes information, assertions and polices managed with the interface in the system. It also has SAML producer that issues assertions related on the result of the authentication and the authorization services.

The paper "**On Views And Xml**[16]" is telling that they believe that a declarative specification of XML views should encompass aspects that are typically not found in relational or object database views. This comes from Web applications that are by nature distributed. So, for instance, a view should specify aspects such as replication and provide active features such as change notifications.

The paper" **Data Integration: The Teenage Years**[17]" offers a perspective on the contributions of the Information Manifold and its peers, describes some of the important bodies of work in the data integration field in the last ten years, and out-lines some challenges to data integration research today.

They note in advance that this is not intended to be a comprehensive survey of data integration, and even though the reference list is long, it is by no means complete.

In the paper "**Integrating Heterogeneous Data Sources With Xml And Xquery**[18]",They are telling that XML has emerged as the leading language for representing and exchanging data not only on the Web, but also in general in the enterprise. XQuery is emerging as the standard query language for XML. Thus, tools are required to mediate between XML queries and heterogeneous data sources to integrate data in XML. This paper presents the e-XMLMedia mediator, a unique tool for integrating and querying disparate heterogeneous information as unified XML views. It describes the mediator architecture and focuses on the unique distributed query processing technology implemented in this component. Further, they evoke the various applications that are currently being experimented with the e-XMLMedia Mediator.

“**A Semantic Approach To Xml-Based Data Integration**[19]” is The paper that describes a prototype tool, named DIXSE, which supports the integration of XML Document Type Definitions (DTDs) into a common conceptual schema. The mapping from each individual DTD into the common schema is used to automatically generate wrappers for XML documents, which conform to a given DTD. These wrappers are used to populate the common conceptual schema thereby achieving data integration for XML documents.

In the paper “**An Advanced Xml Mediator For Heterogeneous Information Systems Based On Application Domain Specification**” they present and illustrate the various approaches which they followed for the design and the development of an Advanced Xml Mediator (AXMed). The goal of this mediator is to ensure an integration of several resources of non-materialized heterogeneous data. The AXMed design which they propose is based on application domains specification. This system of mediation represents the fruit of a thorough research of internal architecture and the existing operating mode of several mediators, namely Médience, LeSelect, Tsimmis, Agora, etc. The AXMed architecture system provides uniform access to the mentioned kind of data sources. The advantage of this system is very easy to configure and maintain by writing simple XML files, describing the structure and mapping of a new source. Indeed, adding additional data sources does not need restarting and redefinition of the system core.

Chapter III

Data Integration by Using Direct Access

3.1Workflow data integration

As I explained in first chapter that data integration can be categorize in two ways.

1. By Using Direct Access and

2. Data Access via Mediator.

Workflows are used for data integration. In this chapter I tried to show how workflows use for data integration i.e how data can be integrated by using direct access and work flow. For that reason here I have been implemented a university system(db1) and HR department database (db2)application software by using the PHP and My SQL language. After implemented two different database I was run those in the local server individually.

Then we select two different key for integrate these two different database. In this chapter I tried to show up the implementation of db1 and db2 integration with architectural diagram UML model and forms that I was designed.

The given name for the database application software is "University System Integrated with HR System(USHR system)".

"USHR system", vision is to enable the sharing of data resources to enable collaboration, to support:

- Data access - access to structured data in distributed heterogeneous data resources.
- Data transformation e.g. expose data in schema X to users as data in schema Y.
- Data integration e.g. expose multiple databases to users as a single virtual database as for example in "USHR system " we integrate and data access from db1 and db2 with the condition that if faculty taken courses more than four then their salary will increase 6000 more than other .
- Data delivery - delivering data to where it's needed by the most appropriate means e.g. web service, e-mail, HTTP, FTP, GridFTP.

To achieve this is a “USHR system”, framework that executes workflows. These are equivalent to programs or scripts and contain what we call activities. Each activity is a well-defined functional unit - data goes in, something is done, data comes out - and can be viewed as equivalent to programming language methods.

Workflows are submitted by clients to “USHR system” web services. It is designed to act as a toolkit for building higher-level application-specific data services as a sample.

3.2 Architectural diagram of data integration for USHR system

Important points to note are In “University System Integrated with HR System(USHR system)”- as follows-

- *A workflow* consists of a number of units called *activities*. An *activity* is an individual unit of work in the workflow and performs a well-defined data-related task such as running an SQL query, performing a data transformation or delivering data.
- Activities are connected. The outputs of activities connect to the inputs of other activities.
- Data flows from activities to other activities and this is in one direction only.
- Different activities may output data in different formats and may expect their input data in different formats. Transformation activities can transform data between these formats.
- Workflows, and therefore query itself, do not just carry out data access, but also data updates, transformations and delivery.

Here in figure. 1 showing the “University System Integrated with HR System(USHR system)” where two SQL queries are executed on two separate databases. The results of the first query are then transformed in some way. This transformed data is then joined in some way with the results from the second query. The joined data is then delivered by some means.

Before the integration we need to fixed that which data we need to access so that we can integrate them. In this “USHR system” one condition is applied that is also shown in figure 1 and in figure 2. The condition is like if course is more than 4 then an extra 6000 salary will add to each faculty. Where db1 has the information about the faculty how many course they are taken in each semester and db2 has the information that as a employee what is his general salary. In that case first access to db1 and select all from the *fac_info* table then count how many courses taken each faculty in each semester then access to db2 and the activity is make the quary on db2 . In db2 create an SQL query i.e select all from the table *mas_emp_sal_info* and where employe_id=fac_id then course count if the course is more than four then the salary will add more 6000 otherwise the salary will be as usual . after adjusting the salary the result is deliver to URL in address http:localhost/stdproject/teacher_salary .

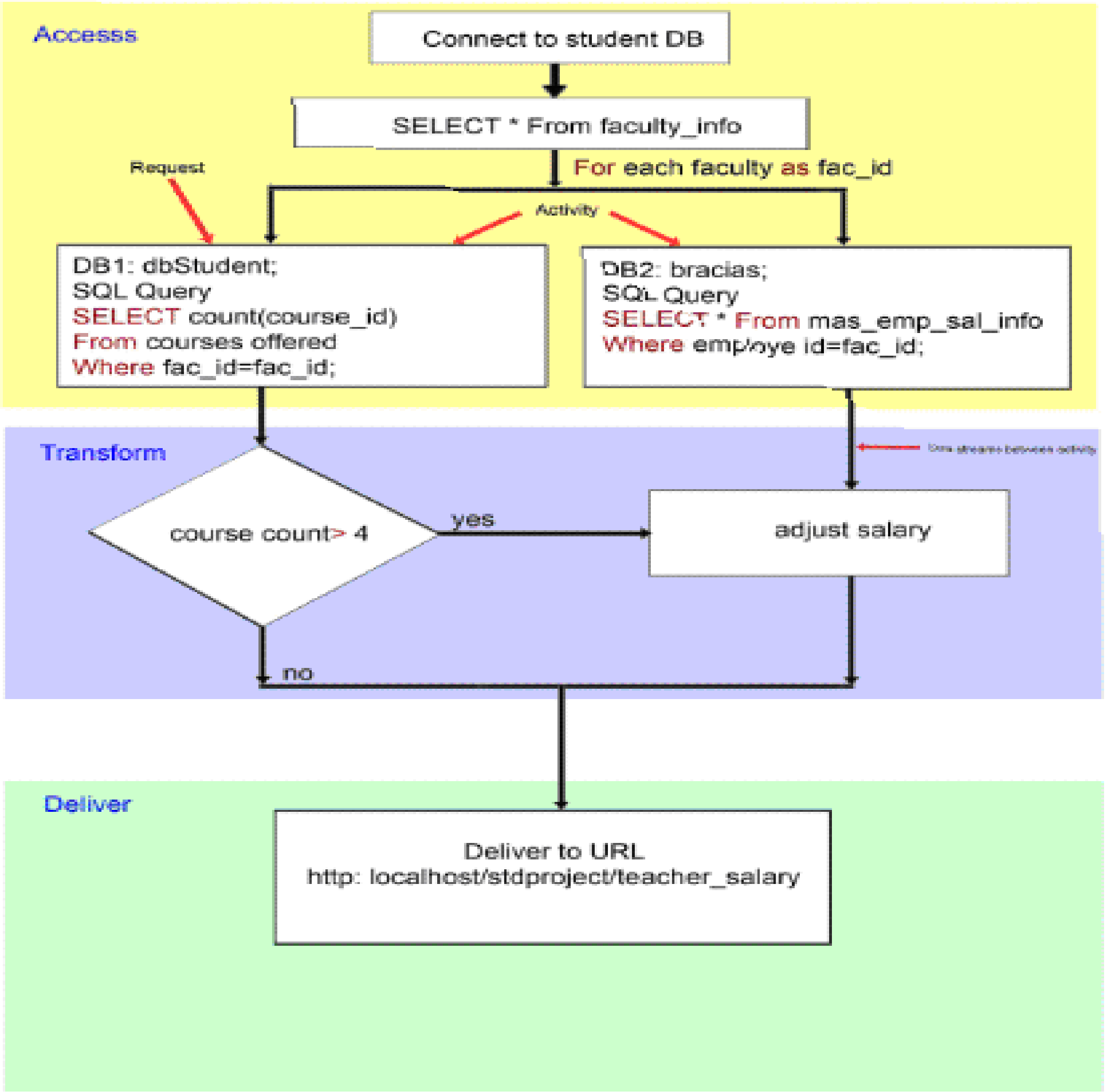


Figure . 3. 1 Data integration using workflow in USHR system

3.3 Flowchart for "USHR system"

From the flowchart of figure. 2 (i) we can see that user can access data from the db1 and db2 . After accessing both database we can integrate the data but it will never change the previous database. As showing the figure .2 db1 is sending data and bd2 is also sending the data after that they are integrating in teacher's salary and giving information to the client. Figure.2 is showing the following steps.

- SQL query in dbstudent (db1) database
- Get faculty_ids from *course_offered* and *fac_course_details* table
- SQL query on bracias(db2) database for salary
- Count the course if the course is greater than four then
- Add extra 6000 salary
- Joint salary table with the faculty table and adjust salary otherwise
- Salary will be as it is in *mas_emp_sal_info* table
- Show the result of integrated database
- Here the workflow is done only one way i.e no data will be return to its previous data base and previous database record will never changed.

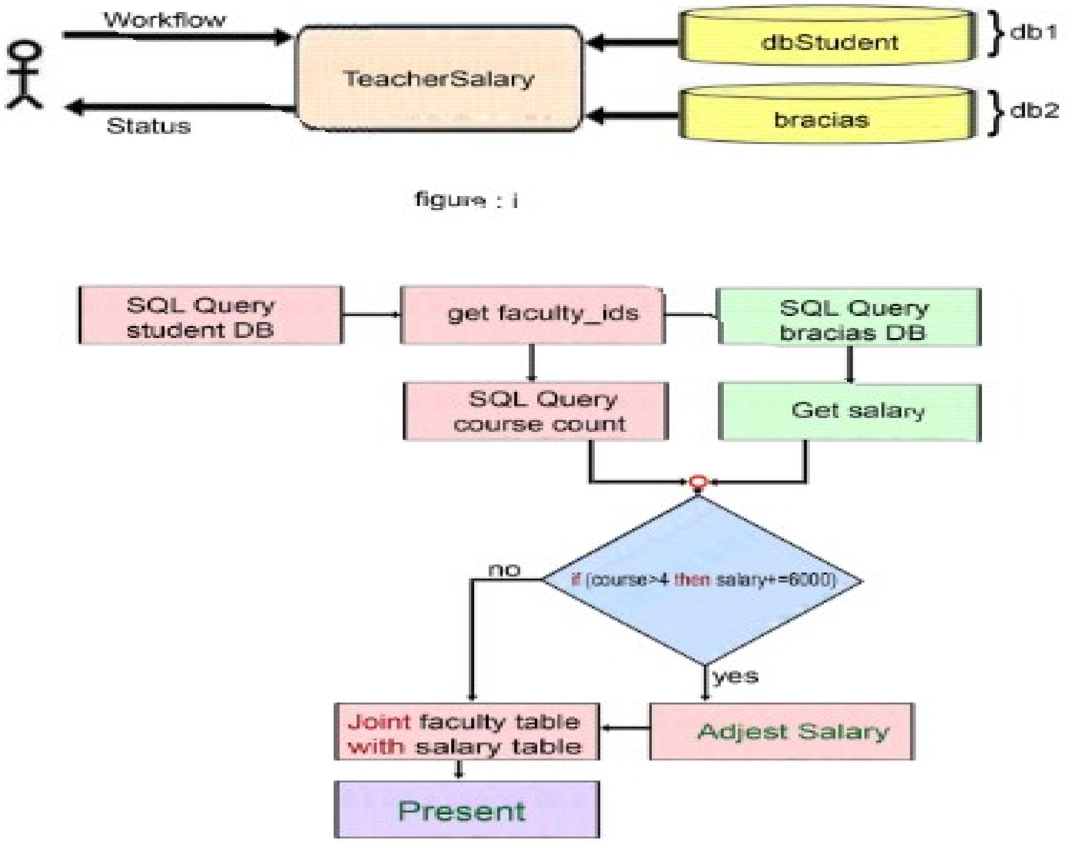


figure : i

figure : ii

Figure 3. 2 (i) and (ii) Data integration using single workflow and server in USHR system

3.4 Activities

An *activity* is a well-defined workflow unit with a specific name. They can be dropped into a deployed server without requiring any recompilation or recoding of server.

Example activities include:

- SQLQuery - Execute an SQL query on a relational database.
- ListDirectory - List the files in a directory.
- XSLTransform - Execute an XSL transform on an XML document.
- DeliverToFTP - Deliver data to an FTP server.

3.5 Activity inputs, outputs and blocks

An activity can have 0 or more named inputs and 0 or more named outputs. Blocks of data flow from an activity's output into another activity's input. Activity inputs may be optional or required. If optional then a default value may be adopted.

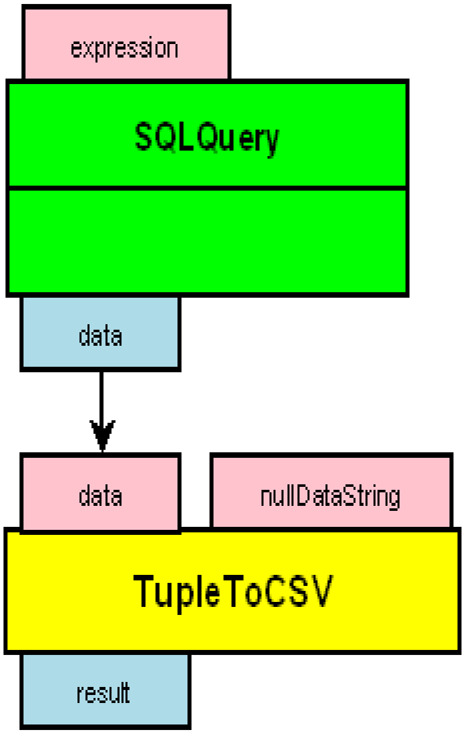


Figure 3.3 Activity inputs and outputs

In an activity, there is no distinction between inputs and parameters. Workflows use the notion of *input literals* which are used by clients to provide a parameter. It is up to the client whether an input value to an activity is provided by them or is provided via the output of another activity in the workflow.

All the required inputs of an activity must be connected to the output of another activity or have an associated input literal and all the outputs of an activity. TupleToCSV means tuple to coma separated values as showing in following figure.

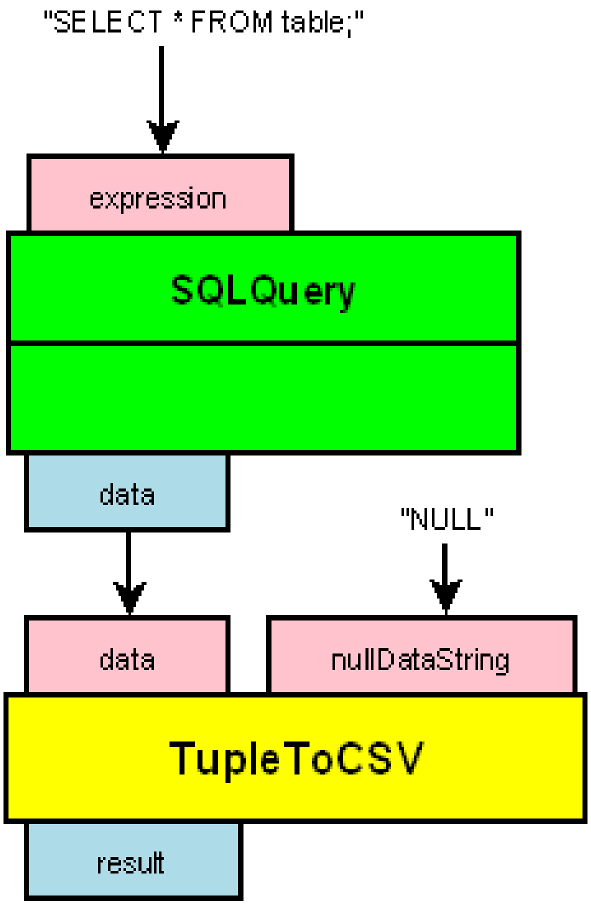


Figure 3.4 Activity inputs, input literals and outputs

3.6. Activities and resources

Some activities can be targeted at “USHR system”, resources. These are termed *resource-specific* activities. The activity interacts with the resource. The most common type of resource with which activities interact are types of data resources. “USHR system”, data resources are components which abstract actual databases (or other data resources) into an “USHR system”, compliant form.

Activities can be defined to interact with any type of “USHR system”, resource, e.g. there are activities for populating “USHR system”, data sources (WriteToDataSource) or dumping state to or retrieving state from “USHR system”, sessions (e.g. ObtainFromSession and DeliverToSession).

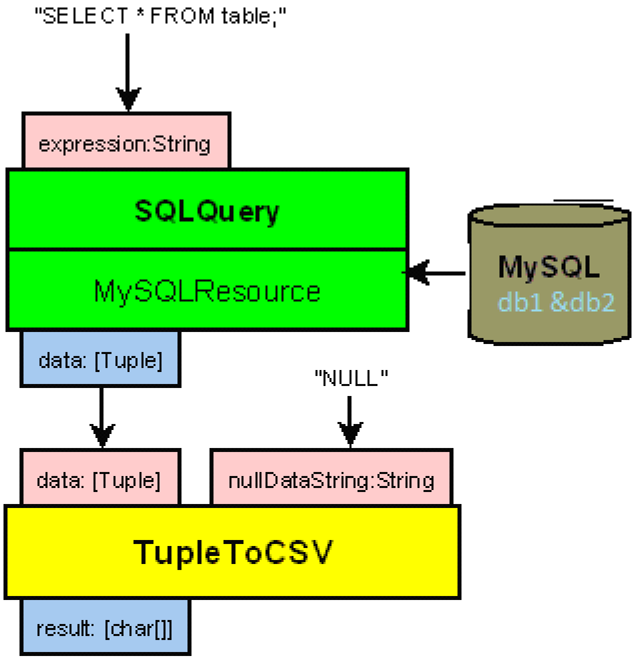


Figure 3.5 Activities and resources

3.7 Workflow execution

Clients execute workflows using "USHR system" as follows: a client submits their workflow (or request) to a *data request execution service (DRES)*. This is a web service which provides access to a *data request execution resource (DRER)*.

The data request execution resource USHR system's workflow execution component. It:

- Parses the workflow.
- Instantiates the activities specified in the workflow.
- Provides activities with their target resources (if any).
- Executes the workflow.
- Builds a *request status*.
- Returns the request status to the client (via the data request execution service).
- The DRER also contains a handler for handling session creation if the client wants to execute related workflows and share state between these.
- The DRER executes a number of workflows concurrently and can also queue a number more.

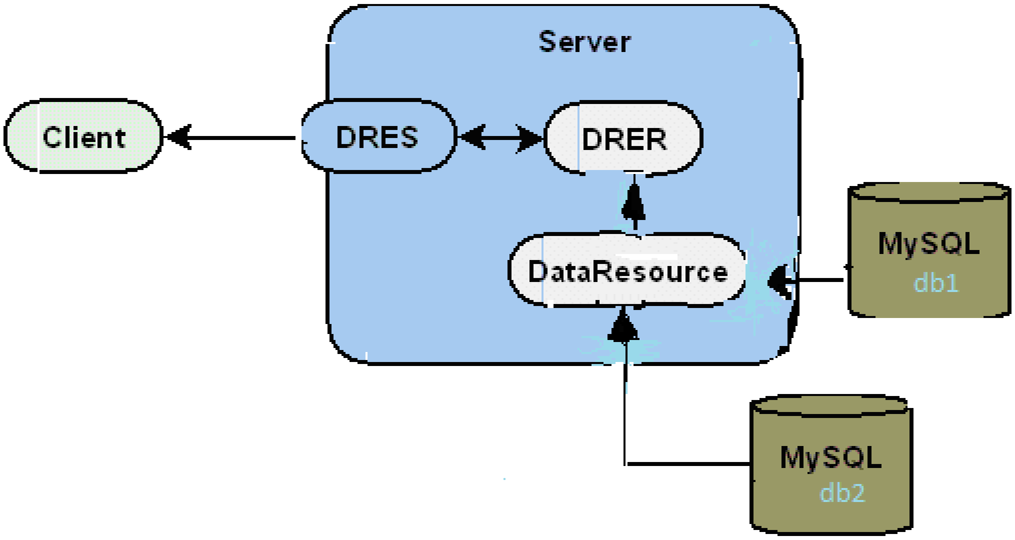


Figure 3.6 Executing a workflow

3.8Workflow execution types

When a client submits a workflow they can specify one of two modes of execution:

- *Synchronous execution* - the data request execution service returns a request status to the client only when the workflow has completed execution.
- *Asynchronous execution* - the data request execution service returns a request status to the client as soon as the workflow starts executing. Along with this, will be the ID of a *request resource* which the client can use to monitor the request status - here return to this in the discussion of request resources and request management services shortly.

Asynchronous execution is the recommended mode of operation as this gives a client more control over the execution of the workflow as we describe shortly. Synchronous execution can be useful for workflows that are very simple and quick to execute.

3.9 Key features of workflow

The key features of workflow execution are:

- All activities in a workflow all execute in parallel
- Data streams through activities in a pipeline-like way
- Each activity operates on a different portion of a data stream at the same time (if the activities are well defined!).

This allows efficient processing of large data volumes as well as reduced memory footprints.

3.10 UML model for "USHR system"

In this section I tried to show all UML model of "USHR system", which was implantation by using the PHP and MySQL. MySQL is used for creating the database. The following UML model will explain clearly which key is connected with whom.

The UML model for University database i.e dbstudent shows in figure.6 is as follows which is about all information of student and about all faculties. Table *Std-info* has a primary key std_id: bigint(20) is used as foreing key in table *std_cource_details* as sc_id ; Table *courced_offered* has a primary key cource_id which is used as a foreign key in table *std_course_details* as sc_course_id which is also used as a foreign key in the table *fac_course_details* as fc_id . On the other hand table *faculty_info* has a primary key fac-id which is used as a foreign key at table fac_cource_details.

Table *faculty_info* is used later for data integration by using the key fac_id.

The UML model for Human Resource division i.e bracias is shown in figure 7 where is almost 88 database table but we use only two table for our desired data integration as shown in figure 8. Here two important table from where we access data for integration is table *mas_employees* and *mas_emp_sal_info* . Table *mas_employees* has a primary key employeobjectid which is using as a foreign key in table *mas_emp_sal_info* as a employee_id.

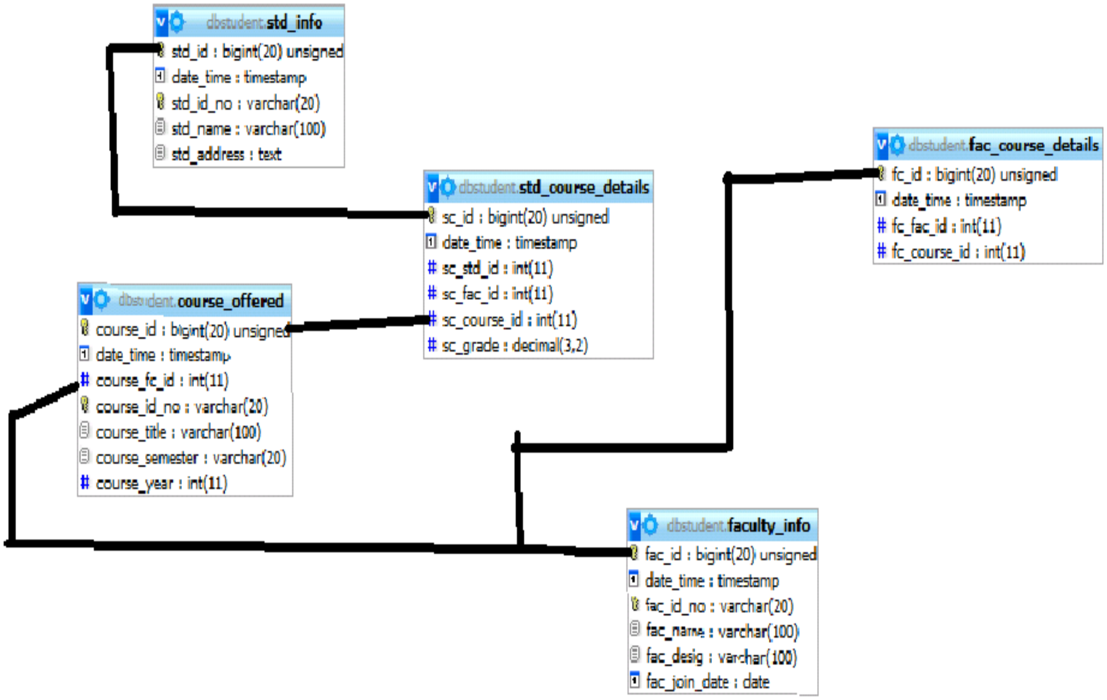
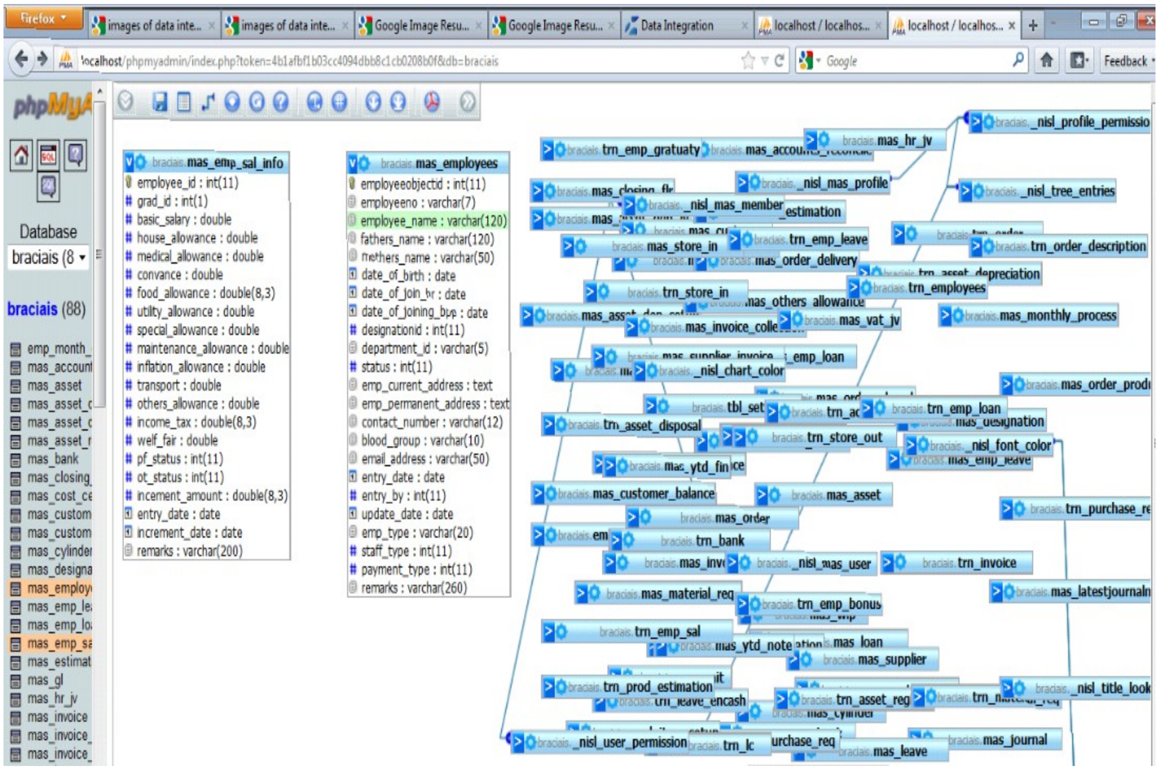


Figure 3.7 UML model for dbstudent (db1)



localhost/phpmyadmin/index.php?token=4b1afbf1b03cc4094dbb8c1cb0208b0f&db=braciais

braciais.mas_emp_sal_info
employee_id : int(11)
grad_id : int(1)
basic_salary : double
house_allowance : double
medical_allowance : double
convance : double
food_allowance : double(8,3)
utility_allowance : double
special_allowance : double
maintenance_allowance : double
inflation_allowance : double
transport : double
others_allowance : double
income_tax : double(8,3)
welf_fair : double
pf_status : int(11)
ot_status : int(11)
incement_amount : double(8,3)
entry_date : date
increment_date : date
remarks : varchar(200)

braciais.mas_employees
employeeobjectid : int(11)
employeeno : varchar(7)
employee_name : varchar(120)
fathers_name : varchar(120)
mothers_name : varchar(50)
date_of_birth : date
date_of_join_hr : date
date_of_joining_bup : date
designationid : int(11)
department_id : varchar(5)
status : int(11)
emp_current_address : text
emp_permanent_address : text
contact_number : varchar(12)
blood_group : varchar(10)
email_address : varchar(50)
entry_date : date
entry_by : int(11)
update_date : date
emp_type : varchar(20)
staff_type : int(11)
payment_type : int(11)
remarks : varchar(260)

Figure. 3.8 UML model for HR department(bracias database)

Figure 8 showing the database for **"USHR system" and** Implementation result for the database by using MySQL . In figure.8 we can see fac_id from *table facultu_info* of dbstudent(db1) is connected with the table of bracias(db2) database's table *mas_emp_sal_info* 's employee_id field name . In I was explained before employee_id is the foreign key which formed from the table *mas_employees* whose a primary key employeeobjectid.

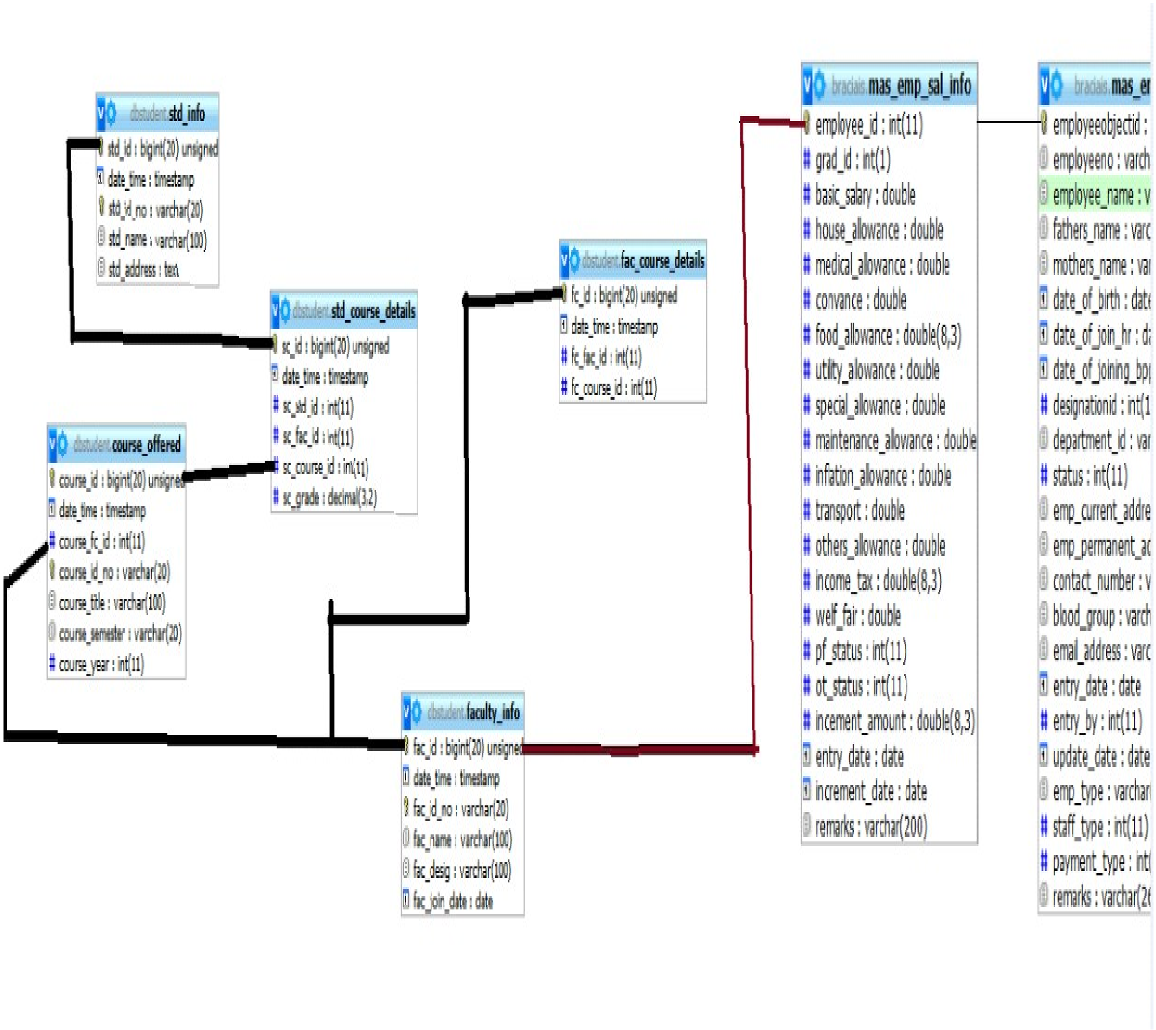
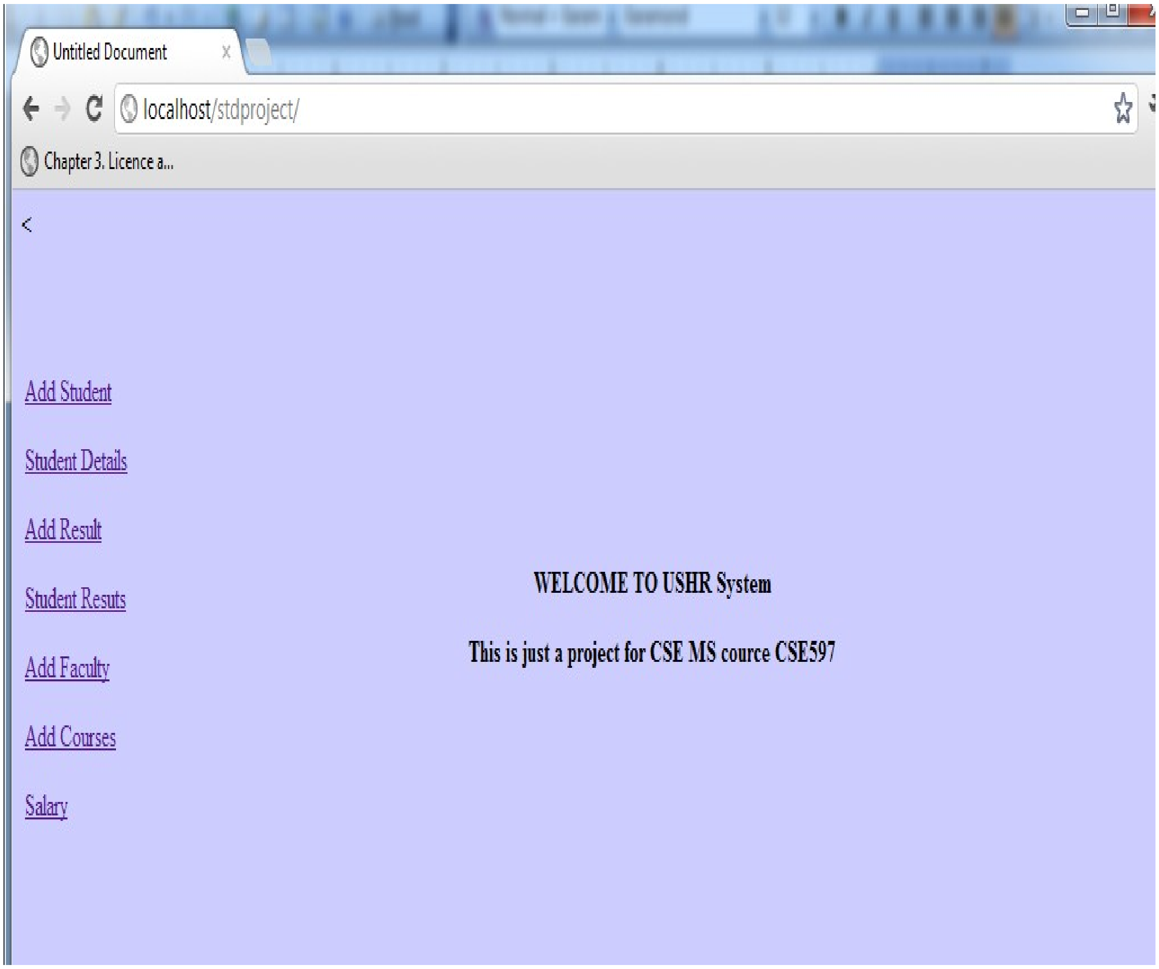


Figure 3.9 The database for "USHR system" with integrating db1 and db2

3.11.1 "USHR system" UI forms from dbstudent(db1) database:

This is the index form of "USHR system" from the dbstudent(db1) where we can see the link form of Add salary which is used for add a new student figure 10 is showing that. When we click on student detain then figure.11 will be shown where if we give the student name and id the information of that student's will be shown. Figure .12 is showing Student's course add details form .

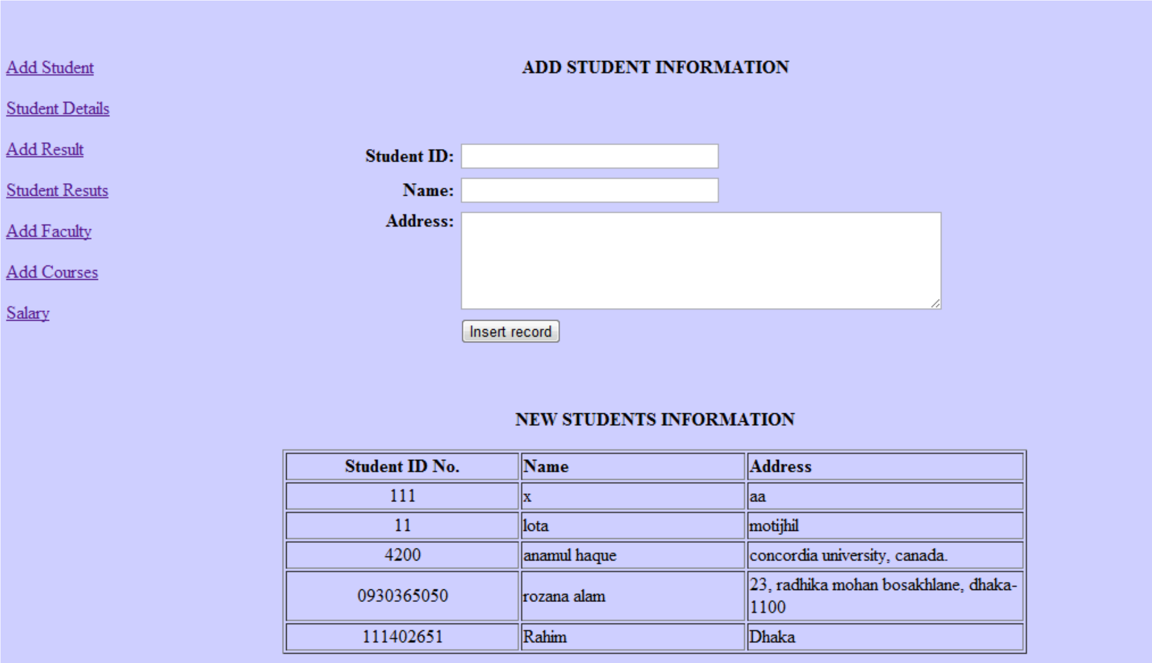
If we click on add faculty then figure .13 will be shown where we can add information about a faculty i.e name, fac_id , faculty designation and joint date . Figure .14 is used for Student's result add . If we want to see the "USHR system" then we need to write the address http://localhost/stdproject/ into the address bar.



WELCOME TO USHR System

This is just a project for CSE MS cource CSE597

Figure. 3.10 University Database form(db1)



ADD STUDENT INFORMATION

Student ID:

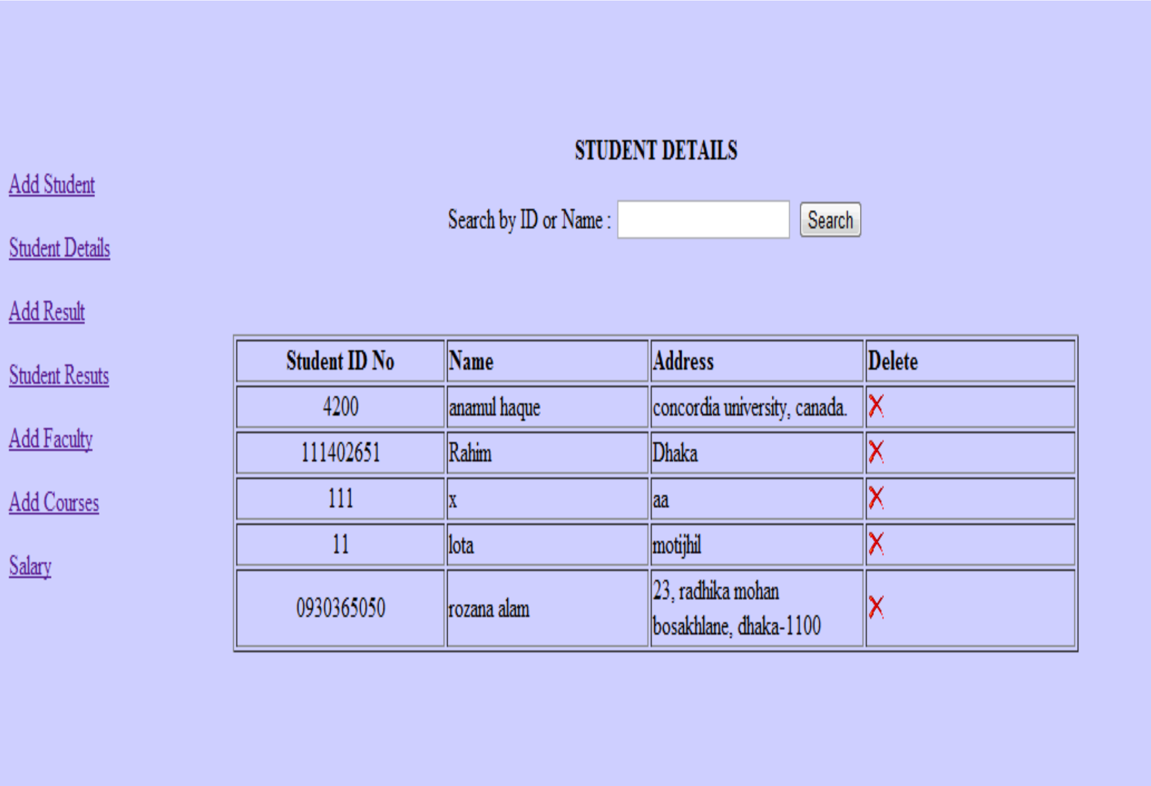
Name:

Address:

NEW STUDENTS INFORMATION

Student ID No.	Name	Address
111	x	aa
11	lota	motijhil
4200	anamul haque	concordia university, canada.
0930365050	rozana alam	23, radhika mohan bosakhlane, dhaka-1100
111402651	Rahim	Dhaka

Figure. 3.11 Add Student information form(db1)

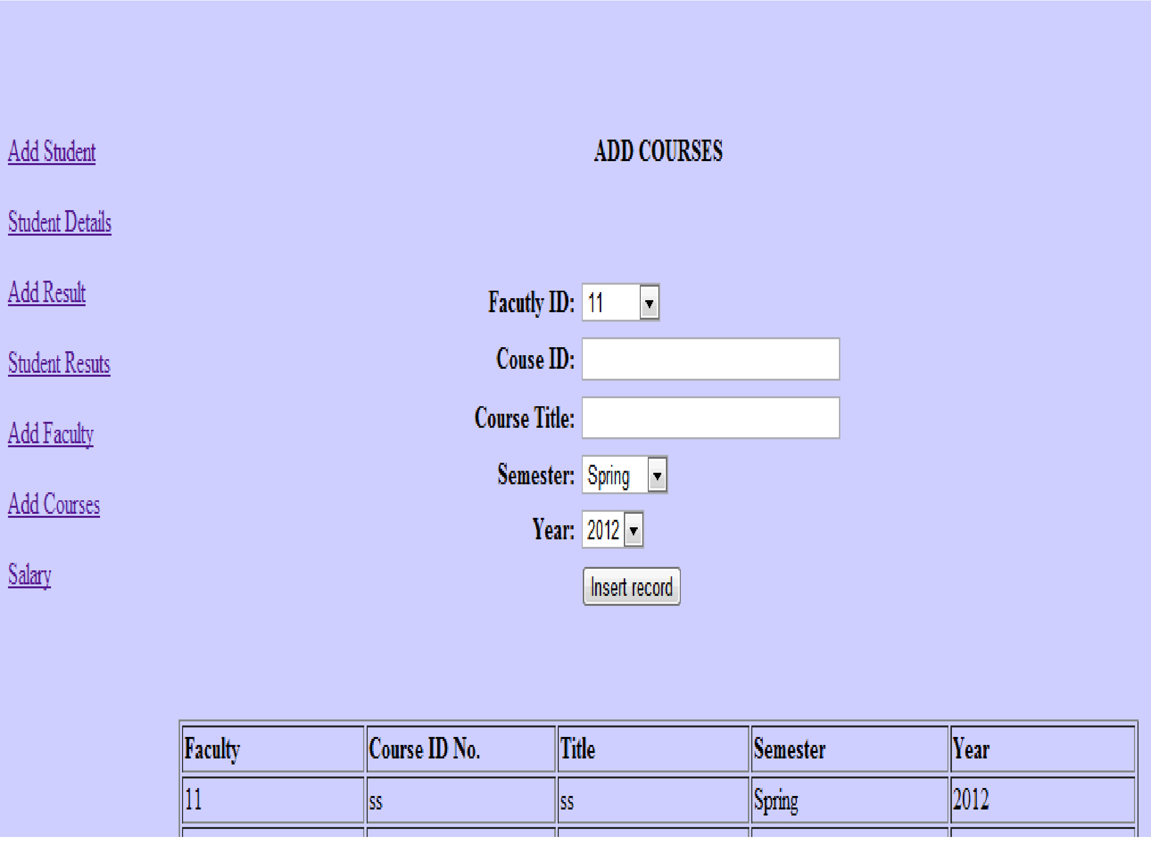


STUDENT DETAILS

Search by ID or Name :

Student ID No	Name	Address	Delete
4200	anamul haque	concordia university, canada.	✗
111402651	Rahim	Dhaka	✗
111	x	aa	✗
11	lota	motijhil	✗
0930365050	rozana alam	23, radhika mohan bosakhlane, dhaka-1100	✗

Figure. 3.12 Student information search form(db1)



ADD COURSES

Facutly ID: 11

Couse ID:

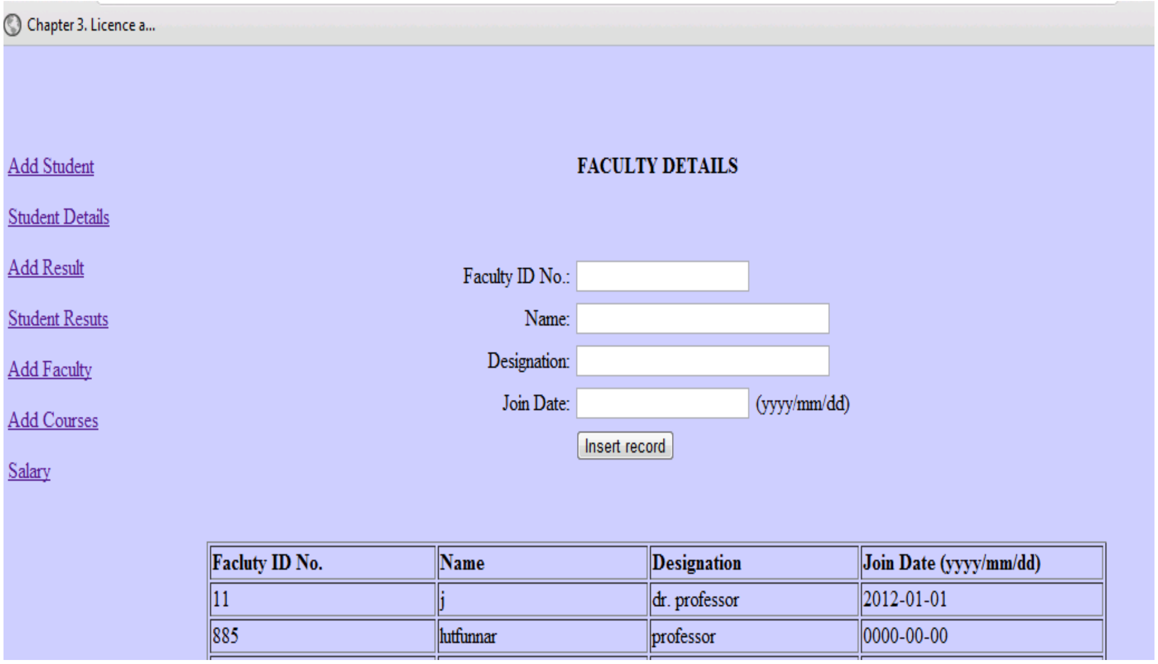
Course Title:

Semester: Spring

Year: 2012

Faculty	Course ID No.	Title	Semester	Year
11	ss	ss	Spring	2012

Figure. 3.13 Student's course add form(db1)



FACULTY DETAILS

Faculty ID No.:

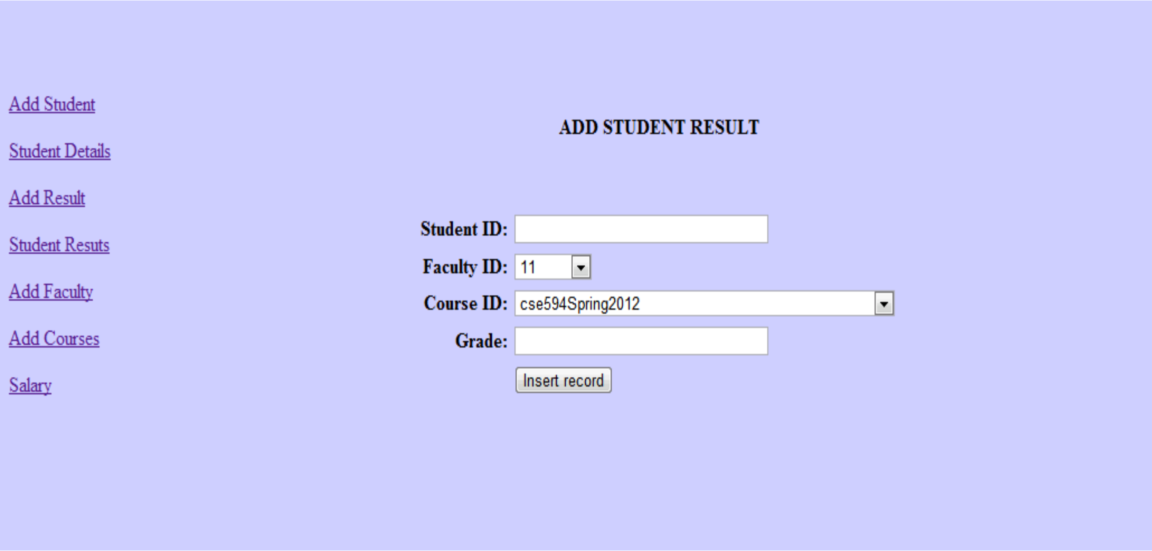
Name:

Designation:

Join Date: (yyyy/mm/dd)

Facluty ID No.	Name	Designation	Join Date (yyyy/mm/dd)
11	j	dr. professor	2012-01-01
885	lutfunnar	professor	0000-00-00

Figure. 3 .14 Faculty details form add form(db1)



ADD STUDENT RESULT

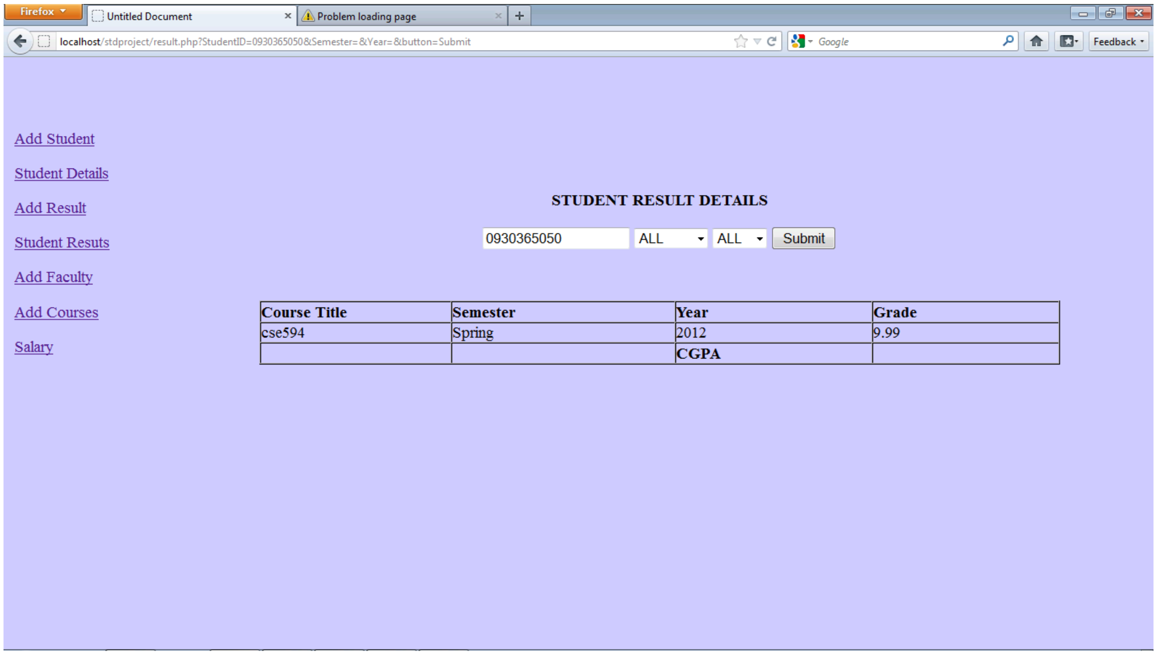
Student ID:

Faculty ID: 11

Course ID: cse594Spring2012

Grade:

Figure. 3.15 Student's result add form(db1)



localhost/stdproject/result.php?StudentID=0930365050&Semester=&Year=&button=Submit

STUDENT RESULT DETAILS

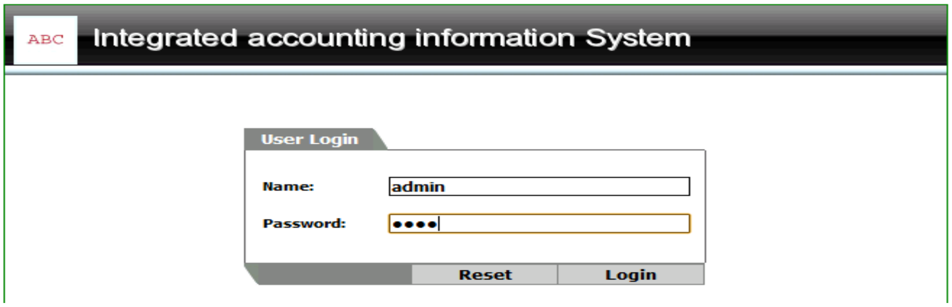
0930365050 ALL ALL

Course Title	Semester	Year	Grade
cse594	Spring	2012	9.99
		CGPA	

Figure: 3.16 Student's result (db1)

3.11.2 "USHR system", forms from HR database form(db2) :

Figure 3.17 is showing the login form of human recourse database .where we can use admin as a user name and password is abcd. Figure 3.18 is showing the ABC integrated accounting information system from db2 which has two branches one is maintenance another is HR and if we click on HR branch then several forms and report will be shown. Figure 3.19 is showing the Employee Entry Form from db2 where we can add a new information about new employee. Figure 3.20 is about Salary Detail Entry Form from db2 where we can add salary of an employee. HR department has many information about load and about many other forms but I am showing here only four forms that are used for integrating with db1 . To see the HR database we need to write the address http://localhost/ias-rozana/login.php in to the address bar.



Name: admin

Password: ●●●●

Figure. 3.17 login form from db2

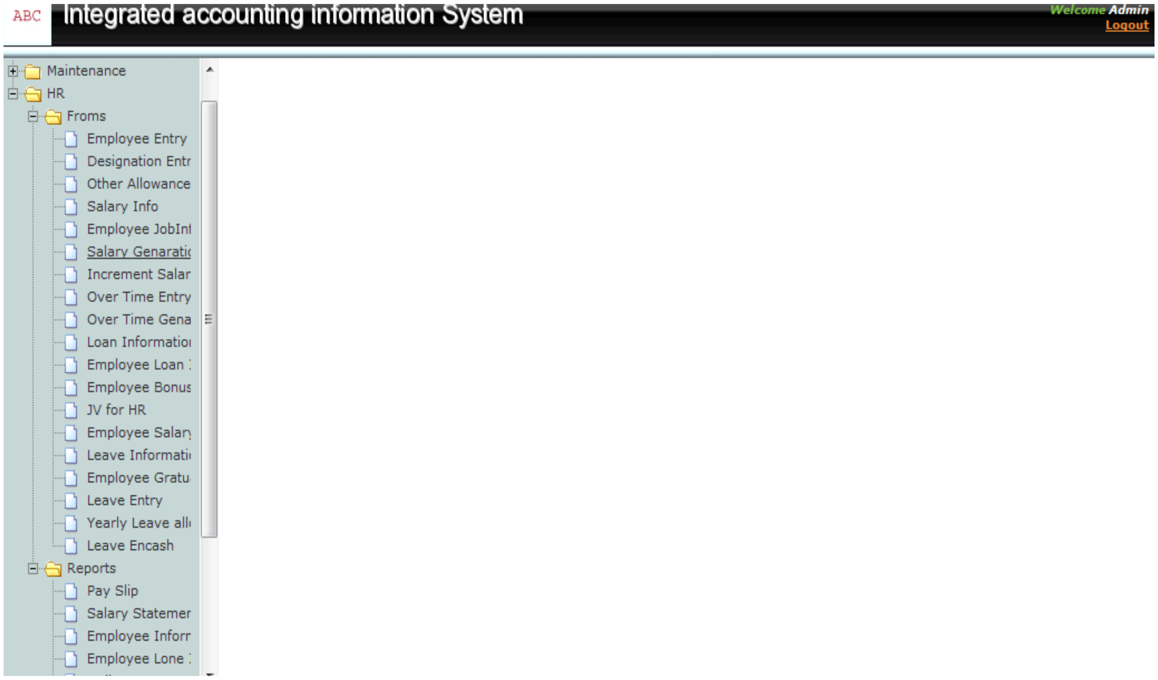
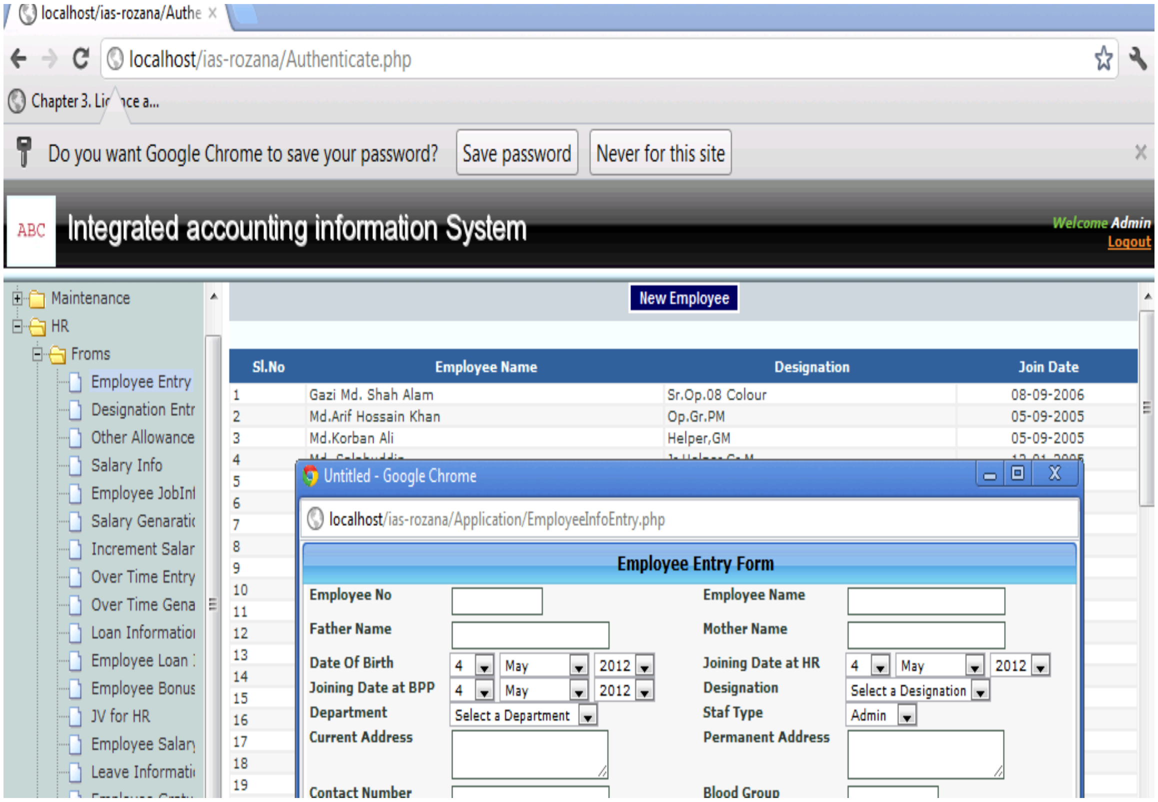


Figure . 3 .18 ABC integrated accounting information system from db2

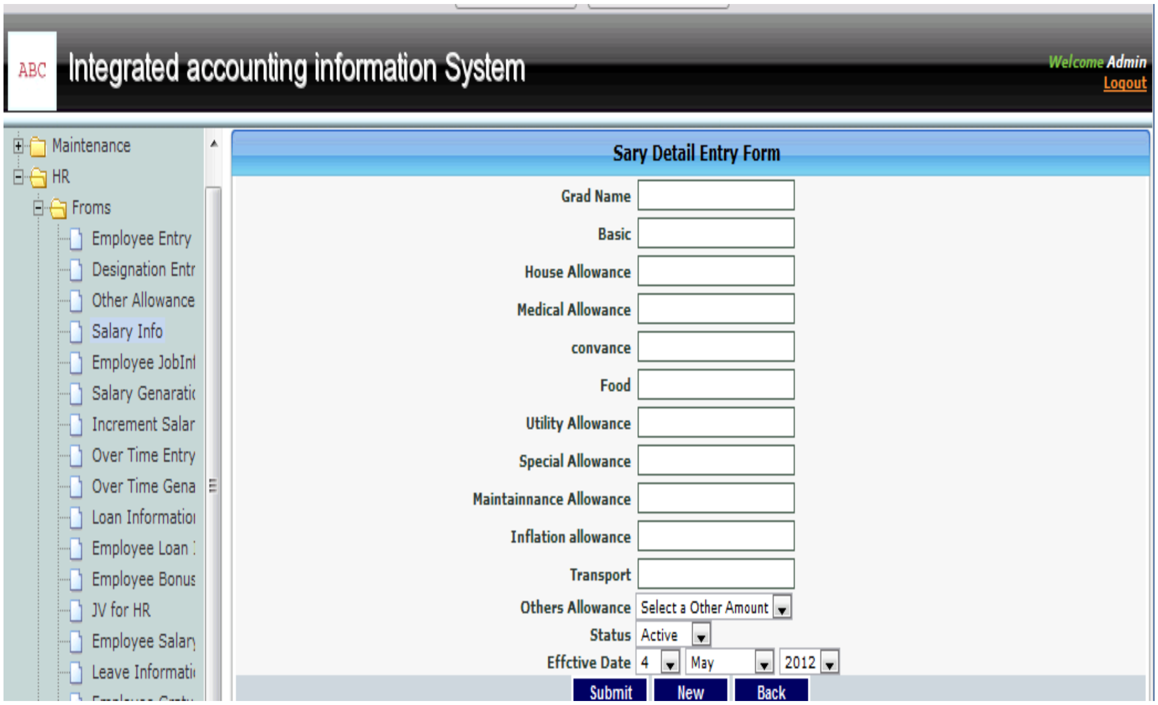


Sl.No	Employee Name	Designation	Join Date
1	Gazi Md. Shah Alam	Sr.Op.08 Colour	08-09-2006
2	Md.Arif Hossain Khan	Op.Gr.PM	05-09-2005
3	Md.Korban Ali	Helper,GM	05-09-2005
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

Employee Entry Form

Employee No		Employee Name	
Father Name		Mother Name	
Date Of Birth	4 May 2012	Joining Date at HR	4 May 2012
Joining Date at BPP	4 May 2012	Designation	Select a Designation
Department	Select a Department	Staf Type	Admin
Current Address		Permanent Address	
Contact Number		Blood Group	

Figure .3.19 Employee Entry Form from db2



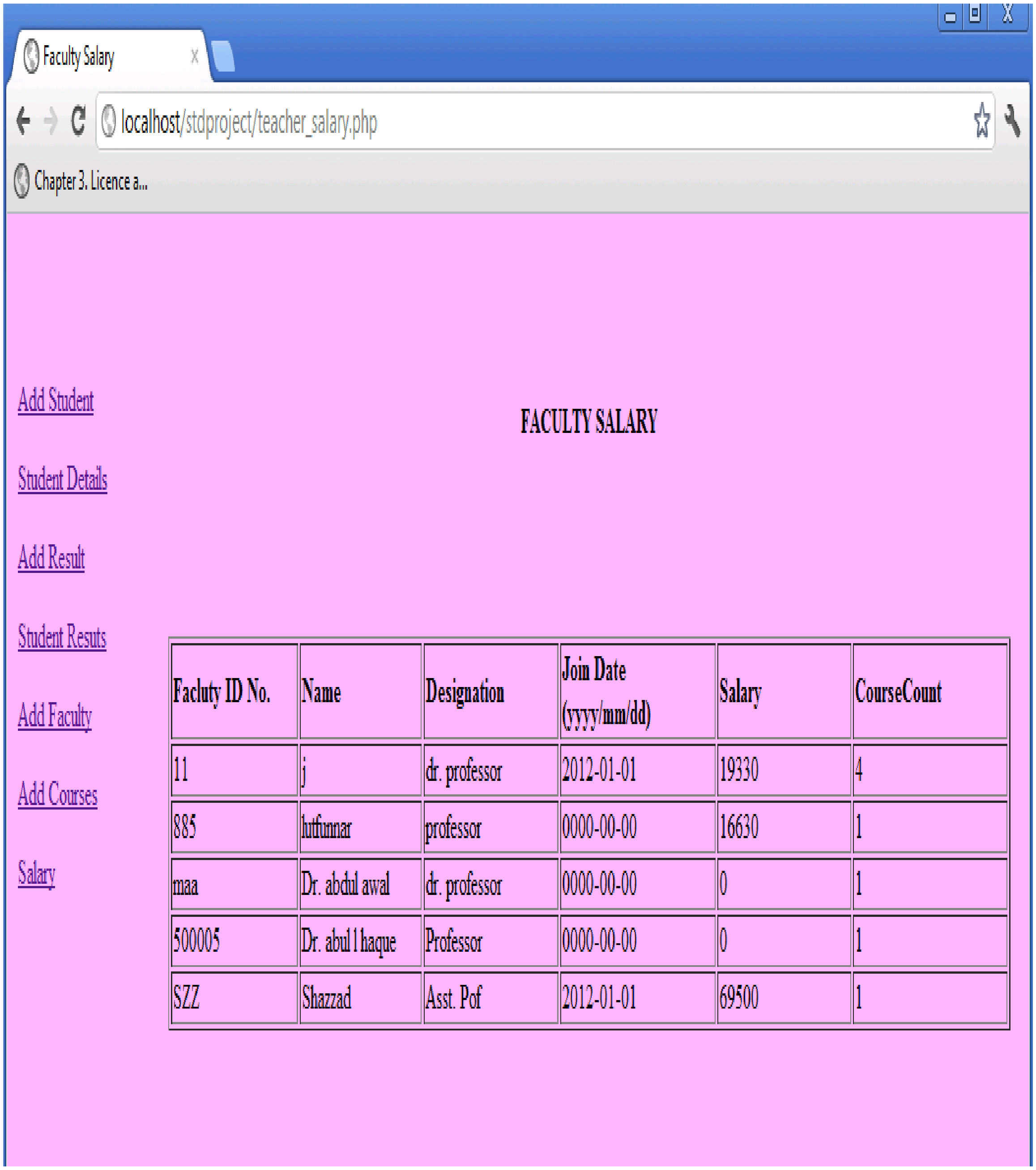
Sary Detail Entry Form

Grad Name	
Basic	
House Allowance	
Medical Allowance	
convance	
Food	
Utility Allowance	
Special Allowance	
Maintainnance Allowance	
Inflation allowance	
Transport	
Others Allowance	Select a Other Amount
Status	Active
Effctive Date	4 May 2012

Figure .3.20 Salary Detail Entry Form from db2

3.11.3 Data integration form for db1 and db2

As explained in section 3.1 and 3.2 with the flowchart and architectural diagram the resulted form is as shown in figure.18 where showing Faculty Salary Form after integrating db1 and db2 .If we write the address http://localhost/stdproject/teacher_salary.php into address bar we can see the form as shown in figure .3.21 .



FACULTY SALARY

Facluty ID No.	Name	Designation	Join Date (yyyy/mm/dd)	Salary	CourseCount
11	j	dr. professor	2012-01-01	19330	4
885	lutfunnar	professor	0000-00-00	16630	1
maa	Dr. abdul awal	dr. professor	0000-00-00	0	1
500005	Dr. abul l haque	Professor	0000-00-00	0	1
SZZ	Shazzad	Asst. Pof	2012-01-01	69500	1

Figure . 3.21 Faculty Salary Form after integrating db1 and db2

Chapter IV

Data Integration via Mediator

4.1 Mediator System

A mediator system allows users to pose queries against a global schema and returns answers from multiple data sources. We have two data sources one is dbStudent9(db1) and another is bracias(db2). The rewriting of the user query in terms of the local sources uses mappings, which in the Local-As-View (LAV) approach, describe the source relations as views over the global schema. Among the existing algorithms that perform query rewriting in LAV, the Extended Inverse Rules Algorithm (EIRA) provides the most general approach. Given a set of mappings and database facts, EIRA provides a logic program, which specifies a class of legal instances of the global system. .The legal instances can be used to compute certain answers for user queries that are monotone.

4.2 Source Query Condition

We showed how we obtain a reduced list of sources based on the conditions in the query and view definitions. We use the sources from the two of our databases to retrieve data used as facts in the logic program for computing certain answers. However, retrieving all the data, when we require only a subset of data is inefficient. Hence, the next step in our optimization is to apply the built-in conditions from the query to the sources to retrieve the appropriate data.

Algorithm 1: Source Query

```
Require: $Q(\bar{X})$, RelSrc
procedure SourceQuery(RelSrc, Q)
    ar = List of built − in variables in the body of $Q(\bar{X})$
    for each predicate Vi in RelSrc do
        for each field Xk in predicate Vi do
            if Xk is not in DB2 then add Xk
                Concatenate data retrieval query for Vi with condition for Xk
            end if
        end for
    end for
end procedure
```

First, we look at source relations that contain the built-in variables from the query. For these source relations, we apply built-in conditions from the query for the appropriate variables. The source relations that do not contain the built-in variables, are queried for all the data. In this step, we consider all built-in operators. We show the steps in Algorithm 1.

A source relation V is defined in terms of global relations in LAV as:

V (X)   P1(X1), P2(X2), .., Pk(Xk),Cv. …………………………..(4.1)

Here, the body is a conjunction of global relations Pi(Xi), i = 1, .., k and built-in conditions Cv. We assume that, the rules are safe, i.e. all variables occurring in the head of the rule are also present in the body of the rule. We also assume that, if a variable is part of a built-in condition, it also appears as part of a predicate in that rule. A join variable is one that occurs in more than one predicate in the body of the rule.

A query, Q(¯X) is given as:

Q(¯X)   P1(X1), P2(X2), .., Pn(Xn),Cq………………………….…... (4.2)

Here, Pi, i = 1, .., n, are global relations and Xi, i = 1, .., n, are variables in the body of the query. The body of the query is a conjunction of atoms on global relations.

Cq is a conjunction of built-in conditions, namely, any of =,_,_ and 6=.

Definition 1 A source predicate V is relevant to obtain certain answers to a Datalog query Q given by Equation (4.2), if there exists a mapping given by Equation (4.1)such that, Pi is in the antecedent of the mapping and V appears in the consequent of the mapping and Pi appears in Q.

4.3 The Domain Predicate

The dom atoms are obtained from rules that define dom predicates using the relevant source predicates. To do this, the existential variables are identified from the view definitions of the relevant source relations. The source predicates whose extensions contain values for the existential variables are used for defining the dom predicate. The source predicates are loaded with values from the relevant data sources using import commands.

The import command for a source relation loads data into the corresponding source predicate when running the logic program in . The import command for db1 is as follows:

#import (bracias, ”test”, ”test”, ”Select Distinct _ From db1
where faculty = fac_id”,db1).

The import command loads the result of the query statement against the source relation db1 in the data-source student database into the predicate in the logic program. We use facts from the relevant source relations in the active domain and this reduces the number of ground atoms in the stable models. The ” ” in the body of the rules is used to mask the other attributes, which do not appear anywhere else in the same rule. We combine the rules for the dom predicates with the logic program.

4.4 Architectural diagram for the Mediator System for db1 and db2

The architectural diagram of figure 4.1 is saying that first the table *faculty_info* collect the record set1 from the database dbstudent(db1) then it will add with the record set2 *mas_sal_info* table which is from the database of bacias(db2). Then the mediator algorithm will compare two record set. The field which are not in record set2 but in record set1 those field will add into record set2. After adding the record set db2 will need update.

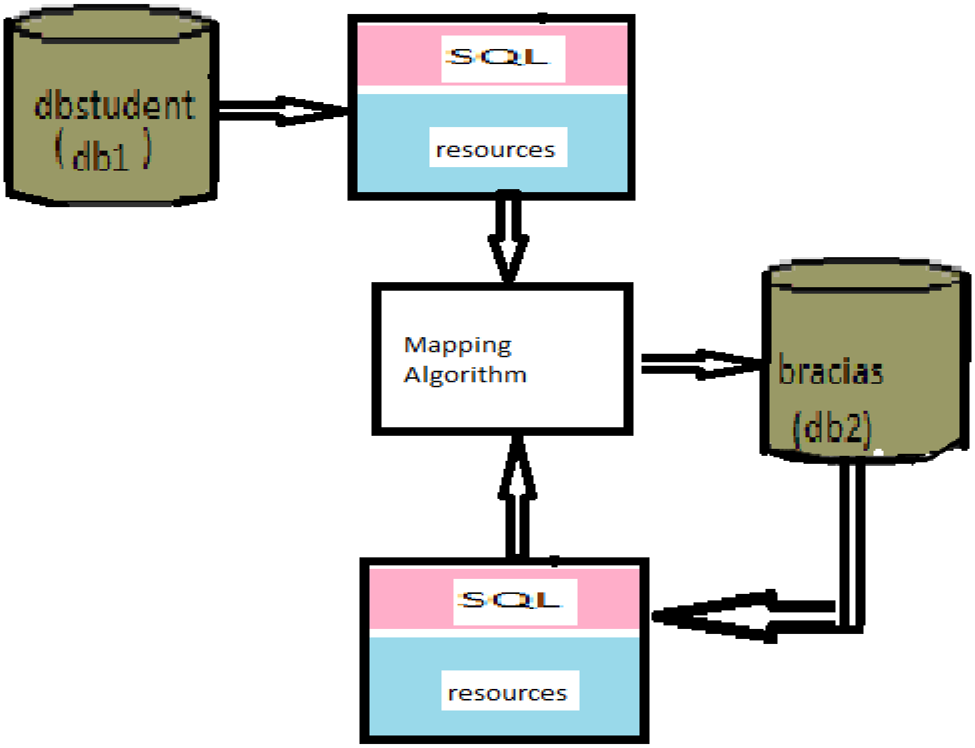


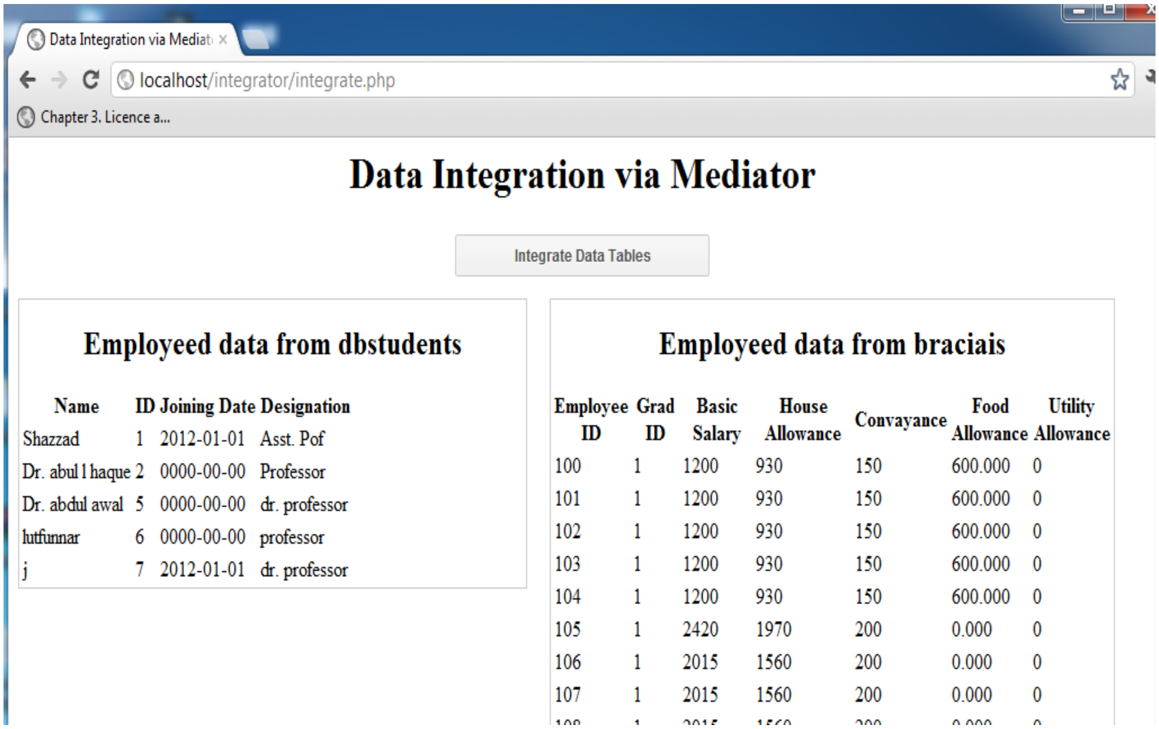
Figure .4.1 Architectural diagram for Data integration via mediator

4.5 Result for the source query mapping algorithm

By using the PHP language I have run the source query mapping algorithm which is used as a mediator to integrate two different database db1 and db2 as explained in section 4.3,4.4, and in the architectural diagram of section 4.5. In which we can see the steps of algorithm 1 is work as follows –

1. Start function.
2. Target the record set.
3. For each record (start loop).
4. For each index in a record (start loop).
5. If this index is not in DB2 then add it in db2.
6. Transfer data in the added fields.
7. End Inner loop.
8. End outer loop.
9. End the function.

To get the answer of this algorithm for db1 and bd2 need to have a comparison of keys or fields they are fac_id_no and employee_id. i.e both field have the same type of values as like fac_id_no=1 then need to employee_id=1; employee_id=2 then fac_id_no=2 and so on. On the other hand fistly we need to run the db1(dbstudent) and db2(bracias). Then we need to write the address http://localhost/integrator/integrate.php into address bar . In this page we will see list of document with their field name as shown in figure 4.2 . After clicking the button on the given page I .e Integrate Data tables we can see the figure 4.5 where will get the integrated data as explained in before 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 section .


Data Integration via Mediat ×
localhost/integrator/integrate.php
Chapter 3. Licence a...

Data Integration via Mediator

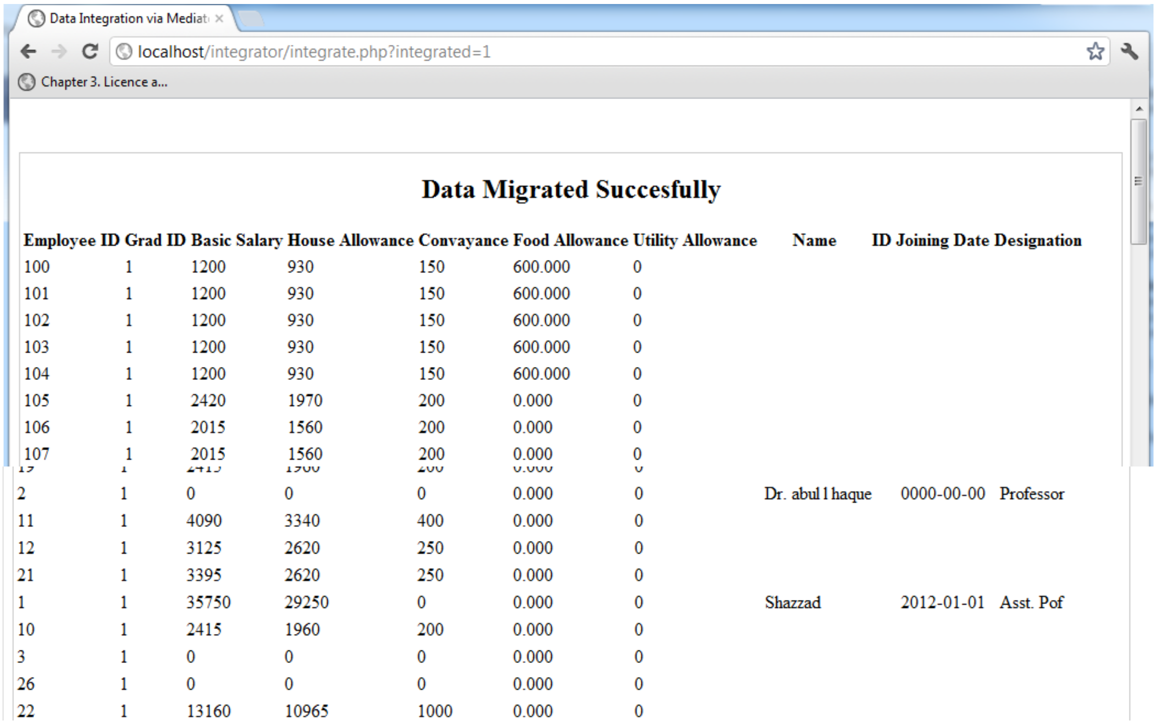
Employeed data from dbstudents

Name	ID	Joining Date	Designation
Shazzad	1	2012-01-01	Asst. Pof
Dr. abul l haque	2	0000-00-00	Professor
Dr. abdul awal	5	0000-00-00	dr. professor
lutfunnar	6	0000-00-00	professor
j	7	2012-01-01	dr. professor

Employeed data from braciais

Employee ID	Grad ID	Basic Salary	House Allowance	Convayance	Food Allowance	Utility Allowance
100	1	1200	930	150	600.000	0
101	1	1200	930	150	600.000	0
102	1	1200	930	150	600.000	0
103	1	1200	930	150	600.000	0
104	1	1200	930	150	600.000	0
105	1	2420	1970	200	0.000	0
106	1	2015	1560	200	0.000	0
107	1	2015	1560	200	0.000	0
108	1	2015	1560	200	0.000	0

Figure 4.2 Data integration via Mediator form


Data Integration via Mediat ×
localhost/integrator/integrate.php?integrated=1
Chapter 3. Licence a...

Data Migrated Succesfully

Employee ID	Grad ID	Basic Salary	House Allowance	Convayance	Food Allowance	Utility Allowance	Name	ID	Joining Date	Designation
100	1	1200	930	150	600.000	0				
101	1	1200	930	150	600.000	0				
102	1	1200	930	150	600.000	0				
103	1	1200	930	150	600.000	0				
104	1	1200	930	150	600.000	0				
105	1	2420	1970	200	0.000	0				
106	1	2015	1560	200	0.000	0				
107	1	2015	1560	200	0.000	0				
19	1	2415	1960	200	0.000	0				
2	1	0	0	0	0.000	0	Dr. abul l haque		0000-00-00	Professor
11	1	4090	3340	400	0.000	0				
12	1	3125	2620	250	0.000	0				
21	1	3395	2620	250	0.000	0				
1	1	35750	29250	0	0.000	0	Shazzad		2012-01-01	Asst. Pof
10	1	2415	1960	200	0.000	0				
3	1	0	0	0	0.000	0				
26	1	0	0	0	0.000	0				
22	1	13160	10965	1000	0.000	0				

Figure 4.3 Data migrated form

Chapter V

Conclusion

5.1 Required software

1. PHP5
2. Apache2.0
3. MySQL database
4. Windows XP

5.2 Future work

In this paper I tried to show different kinds of data integration where we use "USHR system" for show up the both kinds of data integration i.e direct access and mediator based data integration. In my future work will be based on total educational system of Bangladesh where all universities different database will be integrated .So that the people of any country can access the information of any student easily anytime from anywhere with that web sites.

5.3 Conclusion

Data management system traditionally focused on systems in which data is logically and physically centralized. The world wide web, ubiquitous computing ,Personal Data management Present challenges to data management where data needs to be created everywhere and by anyone, and accessed anytime and through various media, Data integration presents one point in the exciting journey from central data management to the vision of ubiquitous data[10] . We have made significant progress over the last few years, and we must continue to focus towards the ultimate goal.

BIBLOGRAPHY

[1] http://wiki.answers.com/Q/What_is_heterogeneous_data#ixzz1td1ADdm8

[2] Research and Practical Experiences in the Use of Multiple Data Sources 2
© 2003 Center for Technology in Government www.ctg.albany.edu .

[3] Data integration Patterns ; Alexander Schwinn,Joachim Schelp ; Institue of Information Management,University Of St.G {Alexander . Schwinn,Joachim.scheil}@unisg.ch www.alexandria.unisg.ch/export/DL/205051.pdf

[4] [RieVog96 c]Riehm, R;Vogler,P.:middleware-.

[5] [AISJ77] Alexander,C.;Ishikawa, S;Silver,M;Jacobson,M.;Fiksdah1.KingI.;Angel,S.: A Pattern Language,Oxford University Press,New York 1977.

[6] http://ogsa-dai.sourceforge.net/documentation/ogsadai3.1/ogsadai_ 3.1- axis/OverviewUsingOGSADAIExecutingWorkflows.html

[7] A Mediator-based Data Integration System for Query Answering using an Optimized Extended Inverse Rules Algorithm- By Gayathri Jayaraman ; http://people.scs.carleton.ca/~bertossi/papers/thesisGJ.pdf

[8] Review of Data Integration Practices and their Applications to Transportation Asset Management prepared for Federal Highway Administration ,U.S. Department of Transportation ;prepared by Cambridge Systematics, Inc.; 100 CambridgePark Drive, Suite 400 ;Cambridge, Massachusetts 02140
http://www.camsys.com/pubs/data_integration.pdf

[9] www.google.com

[10] Ubiquitous Data
http://research.microsoft.com/en-us/um/people/gmb/papers/ubinet03.pdf

[11] A Uniform Approach to Workflow and Data Integration Lucas Zamboulis 1, 2, Nigel Martin;, Alexandra Poulovassilis ,School of Computer Science and Information Systems, Birkbeck, Univ. of London ;Department of Biochemistry and Molecular Biology, University College London
http://www.dcs.bbk.ac.uk/~lucas/pubs/papers/ZMP07b.pdf

[12] A Flexible Model For Data Integration msdn.microsoft.com/en-us/library/bb245674.aspx

[13] Towards Evolving Constraints In Data Transformation For Xml Data Warehousing
http://www.springerlink.com/content/22qk430757257556/

[14] Normalization In Xml And Data Exchange
http://www.mendeley.com/research/normalization-xml-data-exchange/

[15] Design of Integration Security System Using Xml Security
http://.waset.org/journals/waset/v8/v8-26.pdf

[16] On Views And Xml ; http://www.cs.toronto.edu/~libkin/dbtheory/serge.pdf

[17] Data Integration: The Teenage Years
http://www.cs.washington.edu/homes/alon/files/halevyVldb06.pdf

[18] Integrating Heterogeneous Data Sources With Xml And Xquery
http://dl.acm.org/citation.cfm?id=679818

[19] A Semantic Approach To Xml-Based Data Integration
http://dblab.cs.toronto.edu/~prg/docs/ER2001-DIXSE.pdf

[20] An Advanced Xml Mediator For Heterogeneous Information Systems Based On Application Domain Specification
http://www.tmrfindia.org/ijcsa/V3I26.pdf

[21] Supporting Xml Security Models Using Relational Databases: A Vision
http://pike.psu.edu/publications/xsym03.pdf

[22] W3.schools.com