Building Distributed Database System for Development of Students' Exams

Hind Khalid Hameed
Information Engineering College, University of Al-Nahrain /Baghdad.

ABSTRACT

Distributed databases are becoming very traditional now a days. Today's company environment has an increasing need for distributed database applications as the need for reliable, scalable and accessible information all the time more getting higher. In addition, that system provides an advance on communication and data processing suitable for its data distribution throughout different network sites. Not only is data access faster, but the error rate is low when they happen, and it provides local control of data for users. In general, the definition of a Distributed database is a collection of databases that can be stored at different computer network sites. The inadequacies of centralized database systems and manual systems in managing students’ exams require the use of distributed database systems in this paper. The proposed system is a relational database designed in a technique that each academic department in the university has its individual database. The master database is hosted at Center processing unit exams. In this paper, we build a network between parts of the system in order to achieve sharing of information with each other. Three-Tier architecture is the methodology that is used to design and implement the system. The proposed system architecture includes three tier architecture and the Database Management System (DBMS) which combine in a web application.

INTRODUCTION

In today’s world of common dependence on information systems, all kinds people need access to companies’ databases. In addition to a company’s own employees, these include the company’s customers, potential customers, suppliers, and vendors of all types. It is possible for a company to have all of its databases concentrated at one mainframe computer site with worldwide access to this site provided by telecommunications networks, including the Internet. Although the management of such a centralized system and its databases can be controlled in a well-contained manner and this can be advantageous, it poses some problems as well. For example, if the single site goes down, then everyone is blocked from accessing the databases until the site comes back up again. Also the communications costs from the many far PCs and terminals to the central site can be expensive. One solution to such problems, and an alternative design to the centralized database concept, is known as distributed database [1].

As well as, a distributed database system offers many advantages compared to centralized database systems and manual systems of record keeping. A distributed database system has been proposed for different reasons ranging from organizational decentralization and economic treatment for more autonomy. Some of the advantages of distributed database system are location transparency, performance transparency, copy transparency, transaction transparency, fragment transparency, schema change transparency, and local DBMS transparency [2].

Distributed System [3] in Figure 1. Data, Process, and Interface components of an information system are distributed to multiple locations in a computer network. Accordingly, as the task load could be distributed across the network distributed systems are required for functional distribution, inherent distribution in the
application domain, Economics, better performance, and increased reliability.

Figure 1: Distributed Database System [3]

**Three-Tier Distributed Database system.**

The most recent versions of DDBMSs and middleware build a developer's task of implementing a three-tier client-server architecture much simpler. Many of the components that required a heavy programming effort are now available in off-the-shelf versions strong and flexible enough to handle most tasks. Organizations that combine such products into their information architecture will obtain the benefits of three-tier client-server. These architectures will allow for more flexibility and the ability to rapidly take advantage of business and technology opportunities that will arise in the future. However, three-tier client server systems will always require skilled planning and implementation to ensure the present and future needs of the firm can be met. This is another area where DDBMS features will never replace talented people [4].

One way to improve performance is to add server to handle process management. The middlewere usually consist of transaction-processing (TP) monitor or message-oriented middleware. Clients access a TP monitor in the middle tier that, in turn, accesses a database server on the backend [5]. A transaction-processing monitors that support more simultaneous connections than message-oriented middleware. However, message-oriented middleware provides more flexibility in the kinds of messages supported.A second way to improve performance is to add an application server for specific kinks of processing such as writing, neither approach, the additional server software can reside on a separate computer. Alternatively, the additional server software can be distributed between the database server and PC clients [6]. In this paper, designed the system in order to achieve the advantages of a Distributed Database system, also because of the manual processing of students’ result, the process of retrieving missing results takes longer, especially when it has to do with courses from another department. Sometimes, complementary and amended results are not updated immediately to reflect the true status of students’ record, thus resulting in inconsistent records and much time is taken to process students’ transcript and sometimes, there are discrepancies in the results. In this paper, an attempt is made to design and implement distributed database system for performance of exams processing , thus eliminating most of the problems associated with the centralized system and the manual system of student record keeping. In this paper, the system has been use three-tier architecture

**Related Work**

Research has followed that development of application in field distributed database system.

X. Liu, 2005 [7] proposed an analytical model of the 3-tiered Web services architecture. A test bed is built to measure, model parameters based on industry standard server components and benchmark. Validation results show the proposed model predicts performance measures such as response time and throughput accurately.

S. J. Alyaseri, 2010 [8] this paper presents a distributed registration database system using Oracle9i distributed features such as replication and fragmentation. The analysis and design steps of the distributed registration database system are shown as well as the practical steps required for implementing the distributed database system using Oracle9i.

Dr. C. Sunil Kumarn, et al 2012 [9] security features must be addressed when escalating a distributed database. The choice between the object oriented and the relational data model, several factors should be considered. The most important of these factors are single and multilevel access controls (MAC), protection and integrity maintenance. While determining which distributed database replica will be more secure for a particular function, the choice should not be made exclusively on the basis of available security features. One should also the effectiveness and efficiency of the delivery of these characteristics. In this paper, the security strengths and weaknesses of both database models and the thorough problems, initiate in the distributed environment are conversing.

Prof. C. M. Jadhav, 2014 [10] the definition of DDBMS states that the system should make the distribution transparent to the end user and application programmer. Purpose of transparency in DDBMS is to hide implementation details of distributed database system. The main objective of implementing distribution transparency in a heterogeneous distributed database system is to offer transparent access to data no matter where in the network the data is located by providing a single system image to end user. Heterogeneity in distributed database system refers to a system containing multiple databases of different kinds. It is necessary to implement distribution transparency because applications are written depending on how much distribution transparency is provided by the DDBMS.
A Practical Application

Components of distributed database systems.

The distributed database system must contain (at least) the following components:

- Computer Workstations (sites or nodes) that form the network system. The distributed database system must be independent of the computer system hardware.
- Network Hardware and Software Components that be located in each workstation. The network components allow all sites to interact and exchange data. Network system independence is an advantageous distributed database system quality.
- Communication Media that carry the data from one workstation to another. The distributed database management system must be communications-media-independent; that is, it must be able to support several types of communication media.
- The Transaction Processor (TP), which is the software components, found in each computer that requests data. The transaction processor receives and processes the application’s data requests (remote or local). The TP is also known as the application processor (AP) or the transaction manager (TM).
- The Data Processor (DP), which is the software component residing on each computer that stores and retrieves data located at the site. The DP is known also known as the data manager (DM).

Environment Preparation

Servers in each system had been connected as is shown in Figure 2 which describes the implement topology of the servers. The installation of each server is described in separated sections with figures that show the installation procedure. Where this design has been proposed for the implementation of the proposed system.

![Figure 2: Implemented proposed topology.](image)

Design Overview

Preliminary investigations about the existing systems (centralized system and manual record system) were carried out. A detailed study of the student information handbook of Bagdad University was also carried out. The inadequacies of the existing systems were identified and a prototype of a distributed database system was proposed, designed and implemented using a database (PHP, mysql), a web browser with Java Script compatible with Internet Explorer 5.0 or later versions made available at every client workstation.

Proposed system

The proposed system used distributed database system that consists of multiple Database Management Systems management of multiple servers (sites or nodes) connected by a network. The distributed databases are made up of homogeneous (similar) nodes. Each site or node may have some measure of independence, that is, it can allow connections locally while still participating as a node in the distributed database system. The general workflow of the proposed system can be described as follows as shown in Figure 3.

![Figure 3: proposed system](image)

The focus of mechanization is to improve the efficiency and good organization that can easily reached. In view of this, the proposed system is fully mechanical with the aim of eliminating the deficiencies problems in manual and centralized systems. The system is designed in such a way that each academic department has its own database, including all Dean’s Offices, CPU_EXAM, Exams and Records, Student Affair Division, and Senate. The master database is expected to be hosted by CPU_EXAM. The system is designed with some security, such as data, unauthorized modification is not allowed.

The following subsections describe the design of the proposed system through two stages:

- The first stage deals with the design of Three-Tier system.
- The second stage describes the Database Tables’ Relationships, System Testing and Flowchar.

Three-Tier Architecture Design

Three-tier architecture comes into existence to improve management of code and contents and to improve the performance of the web based applications. The proposed system consists of three layers which can be defined as follows; Figure 4 shows the structure of three-Tier system:

1. Presentation or client layer.
   Presentation Layer contains interface code, which is dependable in providing portable presentation logic.
2. Logical or application server layer. Which consists of two logical layers:
   - Business Logic layer.
   The business layer functions between the presentation layer and data access logic layers, sending the client's data requests to the database layer through the data access layer after converting the request(s) to an SQL Query.
   - Data Access layer.
   This layer contains a code to remove data, answer queries and modify Database. Data access code consists of SQL statement translation code. That is usually part of the DBMS. It provides access to the database in Execute a set of SQL statements or stored procedures.

3. Database layer.
   This layer made up SQL database management system component that provides the mechanism to store and retrieve data and grant the permissions to each user as it is planned by the system administrator.

   **Database Tables’ Relationships**

   The system consists of a relational database of student records which could be accessed by all the departments. The proposed system is an available distributed database information system. The database contains CREATE TABLE statements for students’ records, including all courses, clearance procedures, registration procedures; add/drop course procedures, entering of scores procedures, and great point average (GPA) procedures. Each table in the database is designed with a specific format for each field. Input is process next to the file to create the necessary output. For security reasons, accessing mode for each file in the database fragments is exactingly departmental-based (i.e., records in other departments can be viewed (read) but cannot be modified by other departments). A relational database model is based on the model that data is organized and stored in two dimensional table called relations.

   A relational database consists of a collection of tables, each of which is assigned a unique name and a row in a table represents a relationship among a set of values. A table consists of a heading defining the table name, column names and a body containing rows of data. In this system that used relation database (create database, create table, insert, select query) and determinate the primary key and Foreign Key in the tables, and join among tables.

   **STUDENT** (StdMatNo1, StdFirstName, StdLastName, StdAge, StdCity, StdState, StdZip, StdFaculty, StdDept, StdLevel, StdScore, StdGPA)
   **COURSE** (CourseCode, CourseTitle, CreditUnit, Score)
   **STAFF** (StaffID, StaffFirstName, StaffLastName, StaffAge, StaffCity, StaffState, StaffZip, StaffPhoneNumber)
   **DEPARTMENT** (DeptName, DeptHead, DeptLoc)

   Form the above Course Table, the attribute StdMatNo1 in Student Table is a foreign key in the Course Table because it saves as a primary key in Student Table and as none key attribute in Course Table.

   SQL CREATE TABLE statement for the above course database structure is shown below:

   **CREATE TABLE Course**
   (CourseCode CHAR(7) NOT NULL, StdMatNo CHAR(6) NOT NULL, CourseTitle CHAR(20) NOT NULL, CreditUnit CHAR(1) NOT NULL, Score DECIMAL(3,2) NULL, Primary Key(CourseCode))

   **Insert into table**
   Course(CourseCode,StdMatNO,CourseTitle,CreditUnit,Score)
   Values(COSC203,SCNO01,Database1,3,20) ;

   SQL SELECT to QUERY information from table is shown in Table 1, Table2

   **SELECT * FROM Course**;

   **Table 1: Course Table**

<table>
<thead>
<tr>
<th>CourseCode</th>
<th>StdMatNo</th>
<th>CourseTitle</th>
<th>CreditUnit</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSC203</td>
<td>SCN001</td>
<td>Database</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>COSC201</td>
<td>SCN002</td>
<td>C++</td>
<td>3</td>
<td>40</td>
</tr>
</tbody>
</table>

   **Table 2: Course Database Structure.**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Type</th>
<th>Field Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseCode</td>
<td>CHAR</td>
<td>7</td>
</tr>
<tr>
<td>StdMatNo</td>
<td>CHAR</td>
<td>6</td>
</tr>
<tr>
<td>CourseTitle</td>
<td>CHAR</td>
<td>20</td>
</tr>
<tr>
<td>CreditUnit</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>Score</td>
<td>DECIMAL</td>
<td>3.2</td>
</tr>
</tbody>
</table>

   Some of the relations and CREATE TABLE and QUERY statements in the database are presented below:

   SQL CREATE TABLE statement for the student table structure in Table 4 is shown below:

   **CREATE TABLE Student**
   (StdMatNo1 CHAR(6) NOT NULL, StdFirstName VARCHAR(15) NOT NULL, StdLastName VARCHAR(15) NOT NULL, StaffAge INTEGER(3) NOT NULL, StdCity VARCHAR(13) NOT NULL,
SQL SELECT statement for student table in Table3 is shown below:

```sql
SELECT * FROM Student;
```

**Table 3: Student Table.**

<table>
<thead>
<tr>
<th>StdMatNo1</th>
<th>StdFirstName</th>
<th>StdLastName</th>
<th>StaAge</th>
<th>StdCity</th>
<th>StdFaculty</th>
<th>StdDept</th>
<th>StdLevel</th>
<th>StdScore</th>
<th>StdGPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC001</td>
<td>Ali</td>
<td>Abd</td>
<td>22</td>
<td>Bagdad</td>
<td>computer</td>
<td></td>
<td></td>
<td>200</td>
<td>33</td>
</tr>
<tr>
<td>SC002</td>
<td>ahmed</td>
<td>Simer</td>
<td>23</td>
<td>bagdad</td>
<td>science</td>
<td>computer</td>
<td></td>
<td>200</td>
<td>25</td>
</tr>
</tbody>
</table>

**Table 4: Structure table**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Type</th>
<th>Field Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>StdMatNo1</td>
<td>CHAR</td>
<td>6</td>
</tr>
<tr>
<td>StdFirstName</td>
<td>VARCHAR</td>
<td>15</td>
</tr>
<tr>
<td>StdLastName</td>
<td>VARCHAR</td>
<td>15</td>
</tr>
<tr>
<td>StaAge</td>
<td>INTEGER</td>
<td>3</td>
</tr>
<tr>
<td>StdCity</td>
<td>VARCHAR</td>
<td>13</td>
</tr>
<tr>
<td>StdFaculty</td>
<td>CHAR</td>
<td>25</td>
</tr>
<tr>
<td>StdDept</td>
<td>CHAR</td>
<td>15</td>
</tr>
<tr>
<td>StdLevel</td>
<td>VARCHAR</td>
<td>3</td>
</tr>
<tr>
<td>StdScore</td>
<td>DECIMAL</td>
<td>3.2</td>
</tr>
<tr>
<td>StdGPA</td>
<td>DECIMAL</td>
<td>3.2</td>
</tr>
</tbody>
</table>

SQL VIEW statement for student table and course table is shown in Table 5 below:

```sql
create view d1 as select concat(a.StdFirstName, a.StdLastName) name, a.StdDept, a.StdScore, a.StdGPA, b.CourseTitle, b.CreditUnit from student a, course b where a.StdMatNo1 = b.StdMatNo
select * from d1
```

**Table 5: view d1**

<table>
<thead>
<tr>
<th>name</th>
<th>Dept</th>
<th>StdScore</th>
<th>CourseTitle</th>
<th>CreditUnit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahmedsimer</td>
<td>computer</td>
<td>60</td>
<td>C++</td>
<td>3</td>
</tr>
<tr>
<td>AliAbd</td>
<td>computer</td>
<td>50</td>
<td>Database1</td>
<td>3</td>
</tr>
</tbody>
</table>

SQL VIEW statement for STAFF table and DEPARTMENT table is shown in Table 6 below:

```sql
create view d2 as select a.DeptHead, a.DeptName, a.StaffCity, a.StaffZip from STAFF a, DEPARTMENT b where a.StaffMatNo1 = b.depMatNo
select * from d2
```

**Table 6: view d2**

<table>
<thead>
<tr>
<th>DeptHead</th>
<th>DeptName</th>
<th>StaffCity</th>
<th>StaffZip</th>
</tr>
</thead>
<tbody>
<tr>
<td>phd.abd alkream</td>
<td>computer</td>
<td>Bagdad</td>
<td><a href="mailto:abd_alkream@yahoo.co">abd_alkream@yahoo.co</a></td>
</tr>
<tr>
<td>phd.simer ali</td>
<td>information</td>
<td>Bagdad</td>
<td><a href="mailto:semir_al@yahoo.com">semir_al@yahoo.com</a></td>
</tr>
</tbody>
</table>
**System Testing**

**Unit Testing:** Every independent unit (register courses, show registered courses, update score, etc.) in this system was tested.

**Integration Testing:** Using the relationship between the units, the interface each unit presents, was tested with another unit for its validity.

**System Test:** Having been able to validate the interactions between the different units, the system was tested as a whole.

**Flowcharts for the system**

There are different types of tools that can be used to achieve the design of the present systems. The top down approach has been used to achieve the design the system present, three flowchart the first name Login Flowchart for Staff, that can check names authorized to enter system is can shown in figure 5, the second name Flowchart for System Information Processing is shown in figure 6 and the third name Flowchart for Student Records shown in figure 7.

---

**Figure 5:** Login Flowchart for Staff.

**Figure 6:** Flowchart for Student.
Conclusion

As distributed applications are becoming a certainty, distribution, design is becoming a new and appropriate area within database design. Distribution requires its have theory, problem definitions, solution methods, and methodologies. Recent information has shown that distributed database systems turn out to perform better than their manual/centralized counterparts. The inadequacies of the manual/centralized systems in management our students' results necessitated the used of 3-tier form of the Client/Server model distributed database system in this paper, used in the proposed system has the benefit of allowing the division of the database server and all clients' requests for the database, by this means creating more secure environment for the database and distribute the processing of transactions. The proposed 3-Tier architecture leads to reduce the bandwidth required for the hosting server, in both terms of download per day limitation and the general data bit rate to the whole system and for each site to improve the transactional processing of database, in this paper using Three-Tire provides a successful distributed database link that achieved between the sites. Since all programs in a distributed database written using language (mysql language) where we found how the process added and quickly retrieve the data is through this paper conclude that can basic principles applied distributed database appropriate language with any application we want.

References

[1]. Haroun Rababaah "Distributed Databases Fundamentals and Research", Advanced Database – B561, Dr. H. Hakimzadeh, Department of Computer and Information Sciences Indiana University South Bend, Spring 2005.

[2] Djam Xaveria Youh " Design and Implementation of a Client Server Distributed Database for Student Results Processing.", M.Sc.,Department of Mathematics, Gombe State University, PMB 127, Gombe State, Nigeria., E-mail: dxaveria@yahoo.com ,Telephone: +235-8068993095 , November 2010.

[3] Swati Gupta "A Study on Distributed Database System", Assistant Professor, Department of Computer Science, Amity University Haryana, Gurgaon, Haryana, INDIA. IJETCAS 15-367; © 2015.

[4] Prof. (Dr.) Anand K. Tripathi," A Framework of Distributed Database Management Systems in the Modern Enterprise and the Uncertainties removal", Head of Deptt. CSE & IT, SR Group of Institutions,CSE,Jhansi , India College,Jhansi ,India dr.aktripathi@gmail.com monikatripathi.d@gmail.com, 2012.


[9] 1* Dr.C.Sunil Kumar, 2J.Seetha, 3S.R.Vinothaa"Security Implications of Distributed Database Management System Models", I, 2,3Anna University, Chennai, India, E-mail: Sunil1977.dce@gmail.com,2seetha.venkat80@gmail.com,3vinotharamaraj@gmail.com,2012.
[10] Prof. C. M. Jadhav* Bhaskar R Nadargi”
Implementation of Distribution Transparency in Heterogeneous Distributed Database System Using Aglet*, CSE & Solapur University CES & Solapur University Maharashtra, India Maharashtra, India, 2014.