Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Improve Enrollment Systems in Unversites:Zarqa University as Case Study

Essam S. Hanandeh
Zarqa University

Abstract:- The utilization of huge databases, which use well-established network technology, is a growing trend in the computing industry. A proper and effective data distribution plan is needed to combat this trend. This paper discusses the design and implementation of distributed database features from a use-case perspective. The proposed approach has enabled the efficient, reliable, scalable, and standardizedusage of information. The suggested plan also makes a significant contribution to solving the issue of course equivalence while minimizing the significant and ongoingwork needed to create a customized system for every university. The actual findings of this study show that in terms of average response and waiting times, distributed systems outperform centralized ones. Due to data distribution and duplication, thesystem throughput surpasses that of the centralized system. The centralized system crashes at a burden of 61% or more, while the distributed system crashes at 81%, according to the data analysis.

Keywords*:* Distributed Database,Reliability,Availability,Fragmentation,CloudComputing, Registration

1. Introduction

The urgent necessity to standardize and securely save the registration information of students is one of the difficulties Jordanian universities must overcome. The question of course equivalency also arises, focusing on the similarities and differences between courses offered by different universities. The difficulty highlights the necessity to comprehend different viewpoints in order to develop a consistent, scalable, and trustworthy use of information and to lessen the significant work required for each particular registration system's design and implementation. The system will be essential to resolving the course equivalency issue because it will provide common and pertinent information about the courses offered by various universities, such as the course's number, name, prerequisites; in addition, a detailed description. Because a distributed database uses data communication technologies, it could be regarded as a significant advancement in the computing industry. Thanks to this technology, databases are moving from centralized to decentralized models in order to meet the demands of computing features that may be challenging to provide through centralized computing. Users can store data as whole files or as file fragments in a distributed database. However, due to their dispersion among sites, pieces may enhance communications and data processing [1] [2]. The distributed database system, sometimes referred to as the essential building block of cloud services that employ DBMS, gives customers the capacity to access information at any time and from any location.

Additionally, clouds can be included into these systems to improve system performance.

[3] [4]. Understanding the effective goals of a distributed database design is central to mostacademic effort. Such goals include the system's modularity, which enables simple process modification, addition, and removal, and improves availability by ensuring that a defect in one database system only affects a small portion of the database rather than the entire one. Jordan has about 25 universities and academic institutions. Based on Jordanian higher education rules, each of them uses the credit hour as a framework for organizing their studytime. As a result, the processes for adding or dropping courses, registering for new courses,seeking mark transcripts, and so forth, are nearly identical across all registration systems. Numerous aspects that may allow a proper atmosphere for carrying out this procedure mustbe studied in order to effectively build the system. These include Jordan's tiny geographicarea, the extent to which all universities are connected

to the computer network, and the present database management systems. Therefore, creating the system from the bottom up may be appropriate and cost less. The purpose of this study is to provide Jordanian institutions with an illustration of the planning and execution stages of a spread enrollment database structure by:

- Assembling the specifications required to design the suggested system.
- Examining the specifications and key procedures by analyzing the requirements.
- Create a system that meets the requirements for higher education.
- Due to a lack of the necessary hardware and software design environment, a virtual implementation to assess the system's anticipated benefits will be used.

1. Related work

A software program called a student registration system controls all daily activities at a university. It offers a range of initiatives to make it easier for students to enroll, to help institutions better manage a growing number of services, and to cut down on the time and money spent using manual procedures. An explanation and thorough analysis of the student's enrollment database have been attempted in a number of studies (e.g. Sana Alyaseri, 2010; Sean Motta, 2010) [5] [6]. The practical measures required for putting these systems into place to boost the effectiveness of record-keeping in colleges and departments are studied and devised. The deployed solutions could cut down on the amount of time needed to access student records, allowing staff to better serve kids.

Djam Xaveria Youh (2010) [7] discusses the shortcomings of centralized database systems in terms of providing trustworthy, scalable, and accessible information. To handle exam results and student records, he suggested utilizing a database with a relational structure approach based on a client-server shared system. According to Little MC et al. (1999) [8], a high-availability shared software for a university's student registration system has been constructed. Their solution ensures that atomic transactions are consistent and result in a successful registration process. While Ala'a M. Al-2010 Shaikh's [9] solution addresses the issue of managing exam activities while minimizing errors, which leads to the awarding of the corresponding degree in Jordan. A distributed database-based remote registration system was introduced by Mohamed B., 2000 [10] to improve and expedite system performance and throughput, as well as to address issues that arise during the registration period for staff and students. Problems with the old registration system are fixed with his model. By simplifying the registration process, cutting costs, and making the most of the Internet, it also saves time and effort. The issue of overlapped and incompatible schedules is likewise resolved by the model. By taking into account the distributed database fragmentation, Noraziah A. et al. [11] proposed a strategy for administering the student information system termed the binary vote assignment model. Due to the fragmentation of a transaction into multiple sub-queries that operate on the fragments, the model is able to maintain data consistency while increasing parallelism.

Database management is one of the responsibilities of cloud services, as discussed by Shende and Chapke, 2015 [3]. They also stress the distributed database-based cloud architecture's capacity to manage large amounts of data while preserving scalability, availability, elasticity, and dependability. Cloud databases are a new trend that are being used for data mining, data warehousing, and data analysis, according to Mathur et al. (2011) [4]. Additionally, they looked into the benefits and drawbacks of using the database as a service.

Zhu et al. (2018) [12] discuss the shortcomings of centralized databases and evaluate how well a distributed database meets the needs of modern applications. They provide clarification on the distributed database concept's dimensions to assist designers and developers in selecting the best option to finish their task. Alameri, 2017 [13] employed a platform for cloud computing to offer dependable and simple local or remote access to the student information system.

To make the design and implementation of the system architecture easier, they first collected the necessities for a student's database before analyzing and modelling the system components. Through the use of a web browser, their recommended cloud solution will makeit easier for everyone to access the system resources.

To sum up, information is essential to the expansion and advancement of any university. Student data is currently managed by various universities in Jordan independently and according to their own procedures. There are no developed standard procedures or software applications for retrieving student data. From the standpoint of higher education, this circumstance has retained outdated student data in numerous universities. This makes gathering comprehensive data from multiple colleges challenging and time-consuming. For instance, the ministry of higher education would have to manually send requests to each university to obtain the necessary data if it needed information on graduate students acrossall universities. This would take a lot of time and could result in different types of information.

Furthermore, the existing system does not allow for the online delivery of transcripts, thus students who require them must visit their university in person. Because every university defines its courses differently, the course equivalency problem is challenging to solve using the present approach. Furthermore, the majority of courses are taught consistently across colleges and may utilize shared resources. As far as I am aware and based on the literature that is currently accessible, no comprehensive study has been conducted on the development of a distributed registration database system for all Jordanian universities. Consequently, this study offers a suggested paradigm.

2. Research Methods

A preliminary investigation into the current systems was conducted. The shortcomings of the current student registration information system were also examined and reported. In the sections that followed, a distributed database system prototype was proposed, developed, and partially implemented.

Either a small number of universities, as in [6], or the use of the SQL server as a software tool to address this issue were the focus of previous research tools. Or by using Oracle 9i to explore the problem in a single institution with several branches as suggested in [5]. While the strategy that is being offered focuses on numerous colleges in Jordan. Every university has unique rules, software, and databases. Consequently, the proposed model establishes the information and its use as a standard and facilitates the solution of problems that might arise in a course. It also offers the opportunity to reuse functions and forms.

In a distributed database system, non-local data can be accessed via either a distributed transaction that accesses the data at multiple locations or a remote transaction that accesses a single site. Oracle, the DBMS used in this proposed system, uses a two-phase commit processing protocol with a prepare phase. All parties send an acknowledgment to ensure that all sites are ready for the committing phase after they have confirmed they have the resources required to commit the transaction. Every node receives a message from the global coordinator telling them to commit during the commit phase.

The node commits the local portion of the distributed transaction and releases the locks. At the end of this stage, all clones become uniform [14].

3.1 Distributed Registration System Designing

Fig. 1 depicts the proposed system's ER section. It consists of the essential characteristics of each entity to be considered as a sample for assessment, along with the connections between the various entities. The relevant partial conceptual schema is shown in Figure 2, along with a list of the key characteristics of each entity type, the foreign keys that are used for joining, and an explanation of the structural restrictions placed on each type of relationship.

As an example of a database instance, Table 1 lists the public and private higher education institutions in Jordan [15]. The information system that is utilized during the student registration process is described by each of these educational establishments using the relational database model. Al-Bayt University uses an Ingres database, which is also relationally based, for that purpose, but the majority of universities use Oracle databases. The distributed database system, which is based on the relational database as a widely used model, can be designed more simply by employing a bottom-up approach. All universities group the required general electives courses

---

for all majors under code 99 in order to simplify the fragmentation processes (university election). A suggested list of university colleges is shown in Table 2. For instance, all universities with this college's Faculty of Information Technology utilize code number 6. In order to identify the course group that is required for all students in a specific college around all universities, code 999 will be used as an identifying number for all colleges (college election). The same institutions' consistent use ofthe proposed department's name (code 99) is shown in Table 3. The students will arrange their classes for the upcoming term by going over the schedule after choosing their courses from the system.

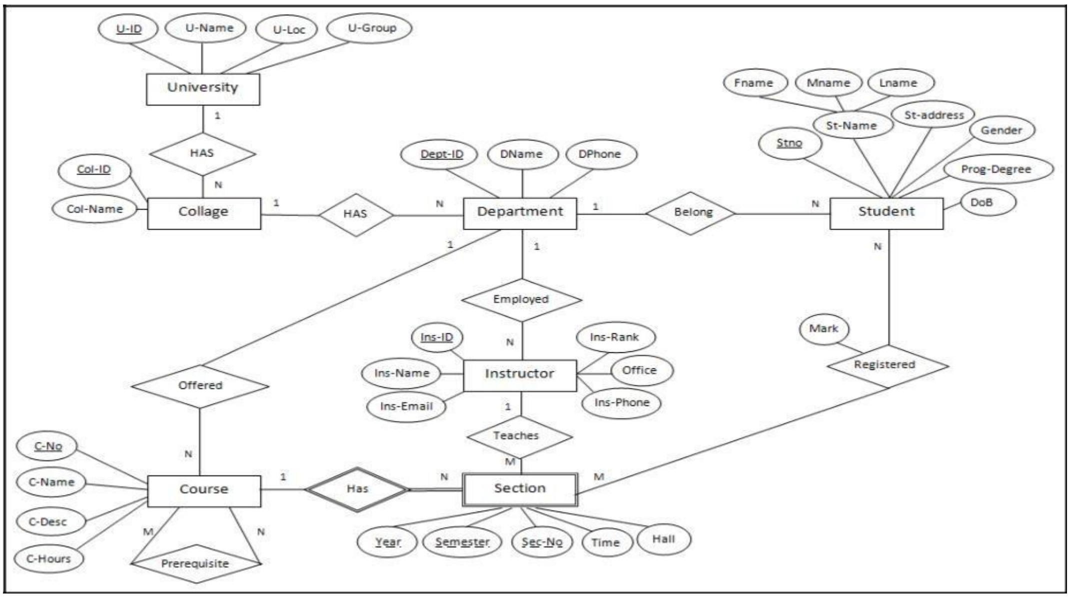


Fig. 1. ER Segment of the proposed system

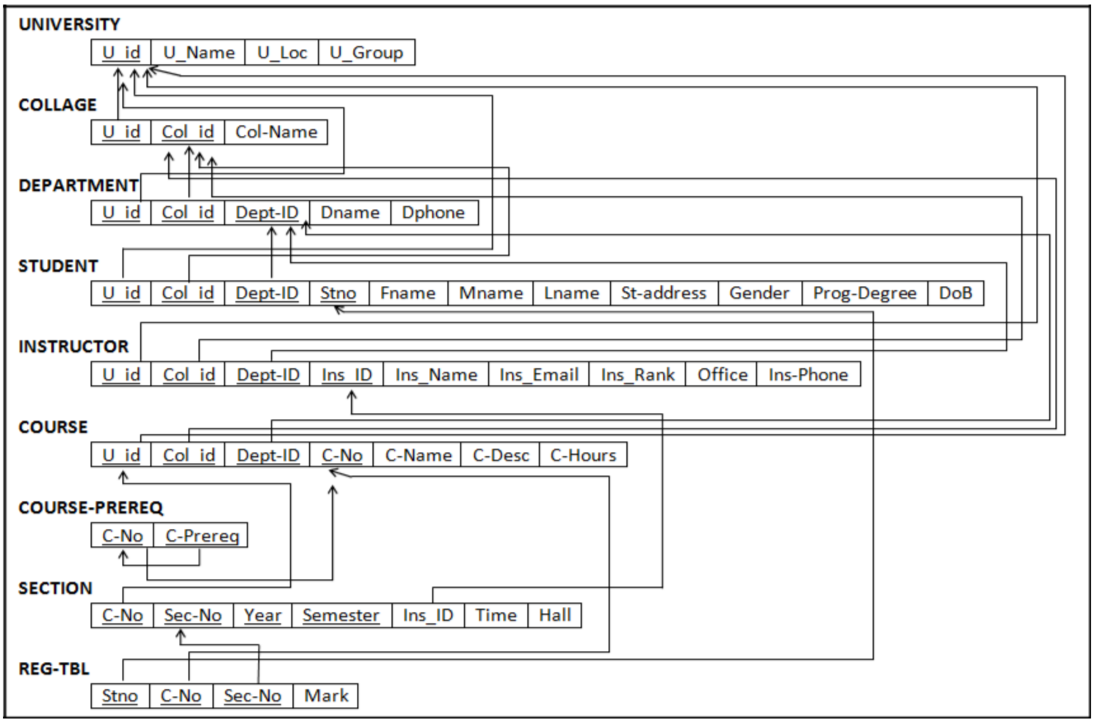


Fig. 2. The partial conceptual database schema for the proposed syste

Table 1. Jordanian Universities

ID	Name	Location	Group
1	The University of Jordan	Amman	Public
2	Yarmouk University	Irbid	Public
3	Jordan University of Science and Technology	Irbid	Public
4	The Hashemite University	Zarqa	Public
5	AL-Balqa Applied University	Balqa	Public
6	Mutah University	Karak	Public
7	Al al-Bayt University	Mafraq	Public
8	AL-Hussein Bin Talal University	Ma'an	Public
9	Tafila Technical University	Tafila	Public
10	German Jordanian University	Amman	Public
11	Zarqa University	Zarqa	Private
12	Princess Sumaya University for Technology	Amman	Private
13	Al - Ahliyya Amman University	Amman	Private
14	Applied Science University	Amman	Private
15	Philadelphia University	Amman	Private
16	Petra University	Amman	Private
17	Al-Zaytoonah University of Jordan	Amman	Private
18	Isra University	Amman	Private
10	Middle East University	Amman	Private
20	Amman Arab University	Amman	Private
21	American University of Madaba	Madaba	Private
22	Jerash University	Jerash	Private
23	Jadara University	Irbid	Private
24	Irbid National University	Irbid	Private
25	Aqaba university of technology	Aqaba	Private
26	Ajloun National University	Ajloun	Private

Table 2. Some College names within the universities

Col-ID	Col-name
1	Faculty of Medicine
2	Faculty of Engineering
3	Faculty of Science
	Faculty of Pharmacy
	Faculty of Admister and Business
4	Faculty of Information Technology
5	Faculty of Dentistry
6	Faculty of Nursing
7	Faculty of Arts

Table 3. Some Department names within college

Dep-ID	Dep-name
1	Department of Anatomy and Histology
2	Department of Pharmaceutical Sciences
3	Department of Civil Engineering
4	Department of Computer Science
5	Department of Computer Information Systems
6	Department of Software Engineering

3.2 Nonfunctional Requirements

A set of circumstances that may be measured as having an impact on a system's operational behavior and are described in the system design is known as a non-functional requirement [16]. Among the requirements that must be taken into account in the suggested system are:

- Performance: Despite having many users, system operations should proceed quickly.
- Reliability: The system must continue to function properly even when there are manyusers.
- Ease of Use: In order for students who are not computer specialists to effectively use the system, the user interaction must be made easy and obvious.
- Load and Concurrency: The enormous volumes of data that the system must manage, along with multiple users requesting these data at the same time, include information about timetables and data that is simultaneously downloaded from the database by multiple users.
- Familiarity of the interface: Users who are accustomed to the previous system's behaviorswill find it easy to adapt to the new system thanks to the new system's interface, which shares some of those behaviors.
- Real-time Feedback: Just like central processing, the new registration system ought to display the data and any updates instantly.
- Web Accessibility: The system needs to be built so that it can work with websites.
- Availability: Duplicate data will make this feature easier to implement. The data must be made available by the system with the necessary level of performance. As a result, data are accessible in multiple copies at various locations.
- Security: A set of authentication and authorization guidelines, along with a selection ofencryption methods, are required to keep the system secure and confidential.

3.3 Data Fragmentation, Replication and Allocation

Data replication is essential to data architectures in order to manage enormous volumes of resources. Multiple data copies are kept at all or some locations to provide consumers with local, fast access to data along with improved availability and dependability. Furthermore, it enhances the efficiency and efficacy of data resource utilization. However, because a transaction can be split up into multiple sub-transactions or agents that work on the pieces, data fragmentation in a distributed database promotes parallelism, dependability, and efficient data consumption. Data fragmentation and replication will be covered in more detail in this section so that the recommended system can be utilized to its fullest potential.

Determining certain predicates according to one or more properties that limit the database relation tuples results in the creation of a horizontal fragment. Each student attends a single university under the proposed arrangement. We came to the conclusion that the data related to aspecific student should be kept on the DDBMS server of the school where the student is enrolled in order to enable simpler access. The university identification number (U-ID) column's value must thus be used to divide the STUDENT table horizontally into 29 sections. Below is an illustration of fragmentation using a relational algebraic operation.

STUDENT_Frag (i) = σ <U_ID=i > (STUDENT); where (i) is the university ID and 1≤ i

≤29.

The INSTRUCTOR table is also horizontally fragmented because each instructor is working at only one university, as shown below:

INSTRUCTOR_Frag (i) = σ <U_ID=i > (INSTRUCTOR);

where (i) is the university ID and 1≤ i ≤29.

With these directives, every relation or table yields 29 distinct fragments. Every piece for both tables can be found at the designated location (the university, for example).

A database's data accessibility can be determined by the quantity of data that can flow simultaneously and the frequency of data access within a reasonable timeframe. It can be accomplished in a distributed database via redundancy or replication. Additionally, fragmentation boosts availability by enabling a user query to focus on particular data on a fragment and ignore other data. By using this technique, the average wait time for transactions that require additional parts for the same data object will be lowered. Making it more likely that the system is using the available data also improves reliability [12] [17]. A distributed database's expanding number of concurrent transactions likewise results in a growing amount of data. To handle the increasing user base, scalability must be properly taken into account. Oracle simply manages scaling, whether it is vertically (scaling up) or horizontally (scaling out) by adding or removing nodes or increasing the amount of data. [18][19].

3.4 Development and Implementation

The network infrastructure, the environment in which Jordanian universities operate, and the rules and regulations that are in place at each university are some of the factors that restrict the proposed system. As a result, the suggested system will be implemented across three virtual sites using real data samples from previous semesters from various colleges. To define global entities, a single site will serve as the central location. Any site may add colleges, departments, teachers, students, and courses, but each site will be in charge of creating its own calendar (timetable). No matter where they are studying, students can be added, removed, or registered on any website. Because the data is fragmented, each site retains the student information that is pertinent to it.[24] The tables for colleges, departments, and courses are completely duplicated because they are updated almost daily, but the tables for students, instructors, schedules, and registration are dispersed. The system's performance is unaffected by the update propagation issue.

All tables requiring frequent writes are kept locally to reduce overhead that might impact system performance in the event of an update propagation. For instance, a university-specific schedule is available at each location. This implies that any changes resulting from the deletion of a class, addition of a new course, or alteration of a course section will be noted locally in this table. However, since students lack the necessary authorization to write on the course schedule table, it must be read-only.

Additionally, the following presumptions were taken into account when creating the proposedsystem:

1. Although the students can register for their courses from any site, they are all enrolled in asingle location

2. Every campus (university) has its own student administration.
3. It is important to build a necessary fragmentation that accounts for the optimal trade-off between resource costs and user performance (in terms of reaction time and throughput).
4. Efforts are made to assign the pieces to the ideal places as efficiently as possible.
5. To satisfy local processing and prevent update propagation issues, efforts are undertaken to optimally reproduce the required data objects. Using the Oracle program SQL*Net (or Net8 for Oracle 8i and higher), which is designed for remote data access, three sites are suggested and connected to one another. [20] [21]. It permits server-server and client-server communication across any network. As a result, the databases and the applications that use them can exist on many machines and interact with one another. It also gives its applications protocol independence. As a result, a program can be distributed among machines running different protocols and run across any network protocol.[23] Transparency options offered by the Oracle network include location, replication, and fragmentation transparency. The suggested system formulates the user questions provided by the interface through the presentation layer, utilizing a three-tier client-server architecture. SQL queries are then also answered by the database server at that point. To provide an illustration of the suggested system tables, a student table will be made, divided into three sections, and positioned at three different universities:

Create Table Student (

U_id Number (3) Not Null,Col_id Number (3) Not Null,Dept_id Number (3) Not Null,Stno Number (10) Not Null, Fname Varchar2 (30),

Mname Varchar2 (30),

Lname Varchar2 (30),

St_Address Varchar2 (60),

Gender Char (1),

Prog_Fegree Number (2),DoB Date)

PARTITION BY HASH(U-id)

(PARTITION ZU_Student TABLESPACE tbl_spc1, PARTITION UiVn2_Student TABLESPACE tbl_spc2, PARTITION UiVn3_Student TABLESPACE tbl_spc3);

The Create statement above will result in 29 fragments, according to the database displayed in Fig. 2. But since we don't have enough access to all of Jordan's universities, we did suggest three for our study: Zarqa University and two more online universities (UiVn2 and UiVn3). Table spaces tbl spc1 for Zarqa University, tbl spc2 for UiVn2, and tbl spc3 for UiVn3 are where each piece will be placed. The same process will be used to build the additional tables. The area designated for storage will be used to hold the broken tables that need to be moved. The replicated tables will be copied to some or all of the sites, depending on what is needed. For example, a schedule table may be unique to each site.

We did, however, recommend three of Jordan's universities for our research because we do not have sufficient access to all of them: Zarqa University and two additional online universities (UiVn2 and UiVn3). Table spaces tbl spc1 for Zarqa University, tbl spc2 for UiVn2, and tbl spc3 for UiVn3 are where each piece will be placed. The same process will be used to build the additional tables. The area designated for storage will be used to hold the broken tables that need to be moved. The replicated tables will be copied to some or all of the sites, depending on what is needed. For example, a schedule table may be unique to each site. The homogeneously distributed database description of the suggested system is shown in Fig. 3. For upcoming description updates, it also displays the geographic dispersion of a few Jordanian universities.

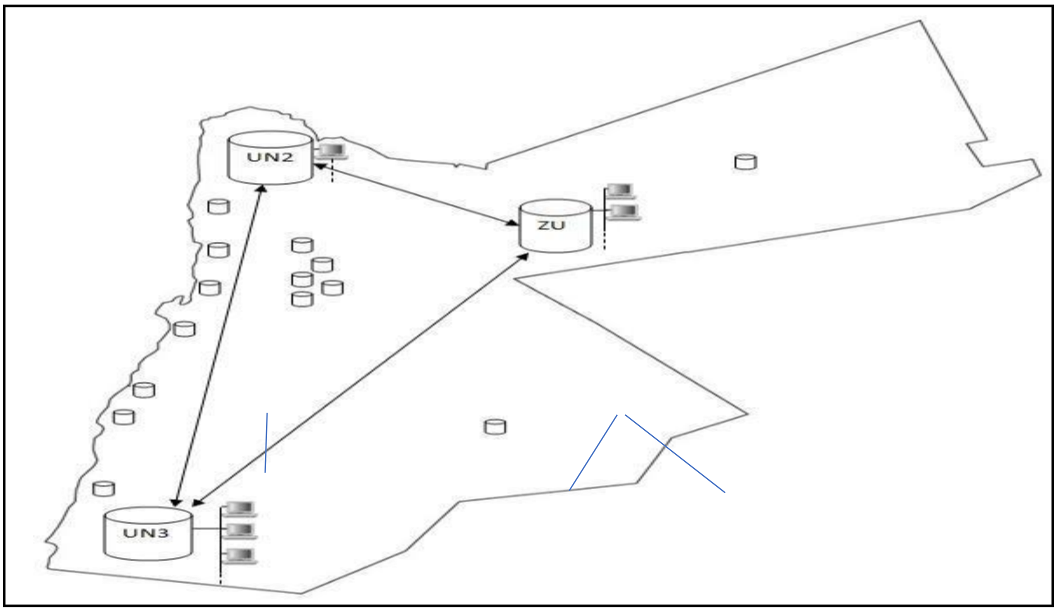


Fig. 3. Distributed database description for three universities

The three websites will be connected via a database. By prefixing the database network domain in front of each database's global name, the system allows for unique database identification. To illustrate global database names, Fig. 4 displays a representative hierarchical layout of databases throughout a network. The root domain of the system is higher education, which is separated into the public and private university sectors, respectively. While UiVn3 and ZU-DB for ZU University are the subordinates to the private domain, UiVn2 is the subordinate to the public domain and has the UiVn2-DB database.Here are a few instances of database link creation.

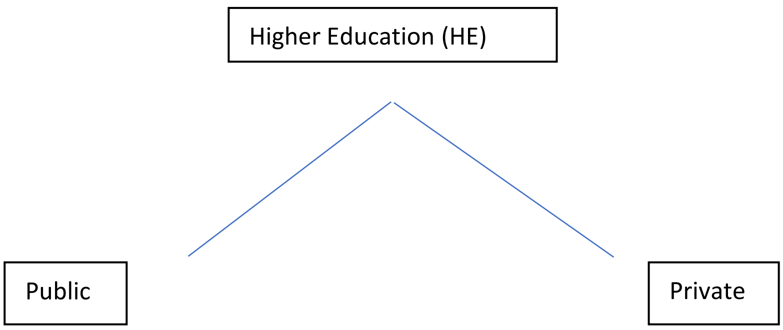
CREATE PUBLIC DATABASE LINK ZU_DB.ZU.Private.HE USING 'ZU';

CREATE PUBLIC DATABASE LINK

UiVn3_DB. UiVn3.Private.HE USING ' UiVn3';

CREATE PUBLIC DATABASE LINK

UiVn2_DB. UiVn2.Public.HE USING ' UiVn2';



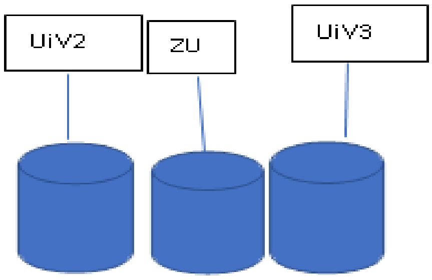


Fig. 4. The hierarchical arrangement of distributed databases

The object owner's privilege set allows all users and PL/SQL subprograms in the database to access another user's objects in a remote database by establishing public database links between sites. Hence, a local user does not need to be a user on the distant database to access a remote database over a database link.

Discussion and Results

The application of the system in real life faces numerous challenges, as was mentioned in the introduction to this study. Some of these challenges include the significant costs associated with building the infrastructure needed for system design as well as the absence of actual data that must be utilized. The student registration system prototype will be implemented in two stages: a distributed database consisting of three sites will be used for the second phase, while a centralized database (one site) will serve as the foundation for the first. This will assist in overcoming these obstacles and putting the proposed model to the test. Moreover, and without a comparable system in place for purposes of comparison, the same data set that. Additionally, a set of random transactions are produced by the simulation program in accordance with the specifications shown in Table 4. Data is gathered and analyzed after the same transaction set is run against both systems to compare system throughput and performance. The simulation programmed makes the assumption that each virtual user accurately represents each real user in order to simulate expected application activity performance. Using variable transaction weights also accounts for multiple concurrent loads. For instance, transaction B, a heavyweight, might read a section of a course while transaction A completes the registration process, which requires calling a stored procedure based on the query results. Provide a sample of transactions during test configuration that closely resembles the features of a population workload at peak testing. Forinstance, 30% of queries are read-only, 10% are for teachers, and 60% are for student registration. Virtual use of the dynamic data is required in these exchanges. A few examples of dynamic transaction flows are adding or deleting courses, requesting a transcript of grades, and scheduling courses. The variety of activity that real users engage in is dynamically modelled by these activities. In the event of writing activities, all parties involved in a distributed transaction involving two or more sites must consent to carrying out the main transaction, even though reading operations obtain their data locally or from the closest site [14]. By taking into account the assumptions of comparable models that can be found in the literature, the parameters listed in Table 4 are gathered using measurements in a suggested system [22].

Table 4. System parameters and value

Parameter	**Description**	**Values**
Objects	Number of database objects	5000
Rep_degree	Degree of replication	0.40
Tables	Number of tables in a database	10
Transactions	Number of transactions in the system	10000
Min -Tsize	Minimum number of operation	1
Max -Tsize	Maximum number of operation	10
Queuelen	Maximum queue length	30

Both of the suggested approaches were regularly applied in order to ensure the accuracy of the data collected. The study's conclusions indicate that the distributed registration system outperforms the centralized one. These results align with those obtained from previous research conducted by Djam XaveriaYouh (2010), Sean Motta [6], and Sana Alyaseri [5].(7). The expected throughput, average waiting time, and average response time are used to compare the two suggested systems' respective performances. The proposed distributed system outperforms the centralized system in terms of average reaction time, as shown in Fig. 5. Moreover, data replication leads to this behaviour by making data more accessible, which in turn encourages local access. Both curves, however, rise as workloads increase because the centralized system's load increases while the distributed system's update procedures spread. Furthermore, as Fig. 6 shows, a distributed application has a lower average waiting time than a centralized application. This outcome may be explained by the fact that there are numerous choices (sites) for the data, making it as difficult to get as possible, especially when the majority of searches are read. The read mode and the presence of numerous copies of each data item do, however, shorten the total access wait time.

The throughput of the suggested systems is shown in Fig. 7. Due to dispersed workload and data replication, distributed systems outperform centralized systems. The number of sites, however, might place some restrictions on these results. Therefore, it may be possible to increase both the number of locations and the level of data replication in order to increase system throughput in subsequent research and when the system is actually implemented.

Fig. 5. Average transaction response time

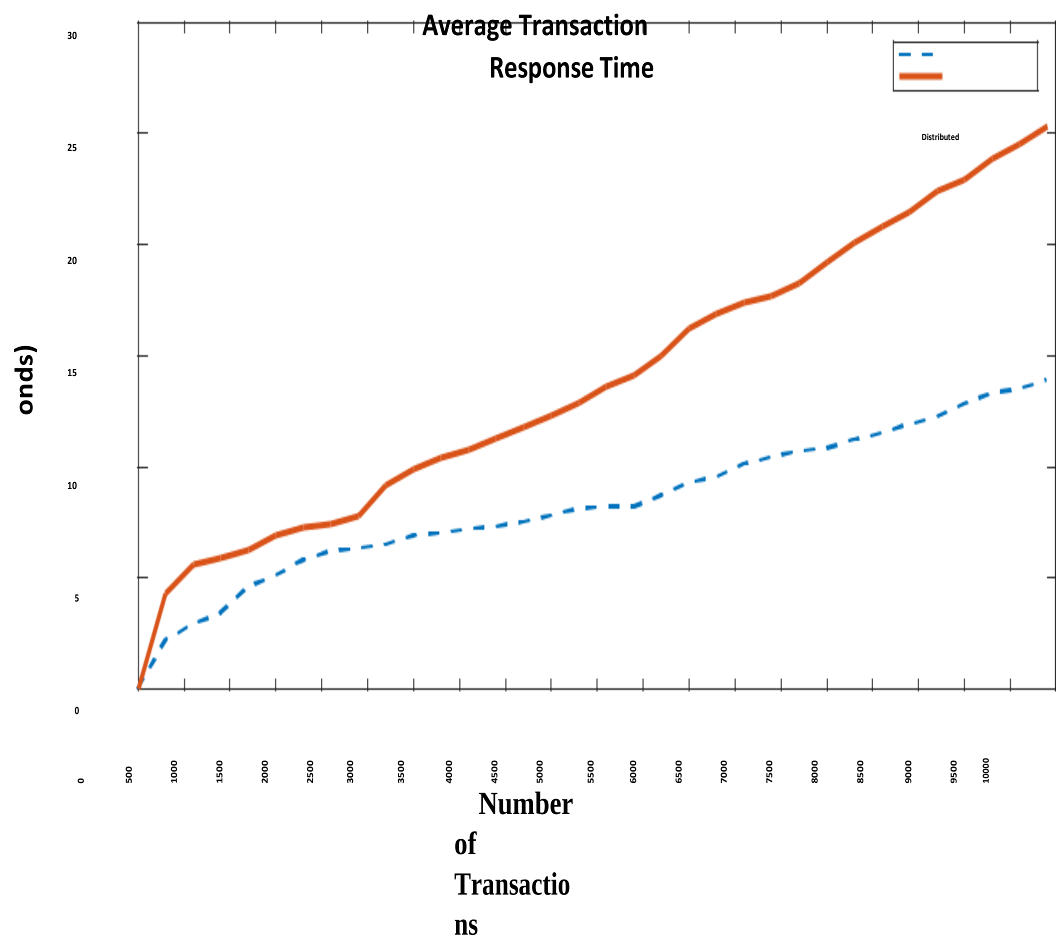


Fig. 6. Average transaction waiting time

Average Transaction Waiting Time

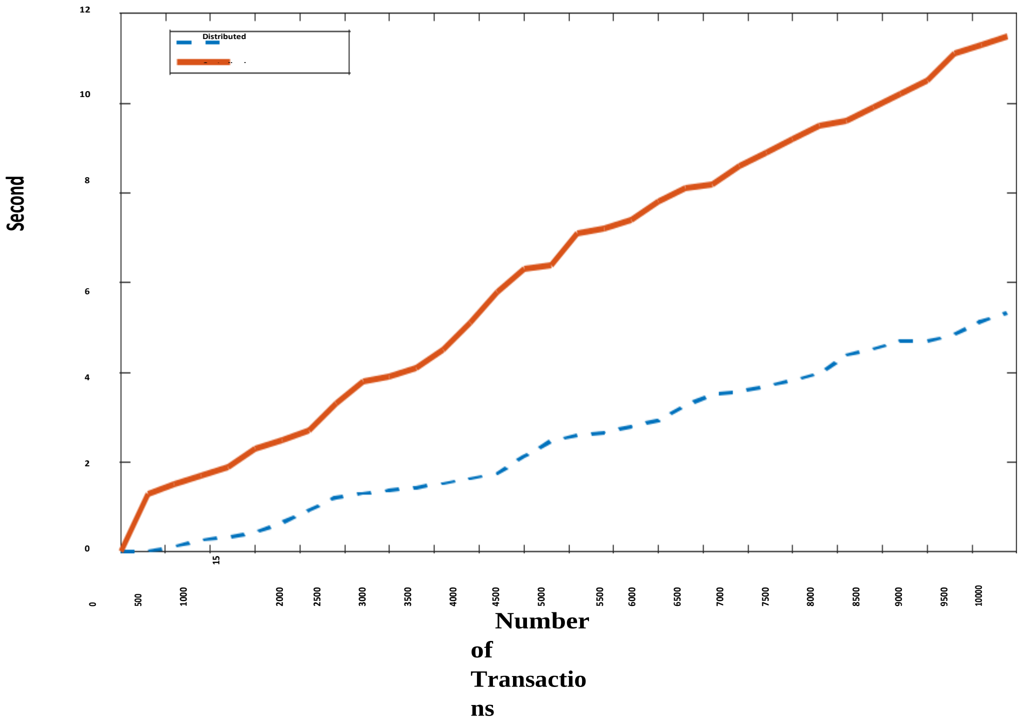
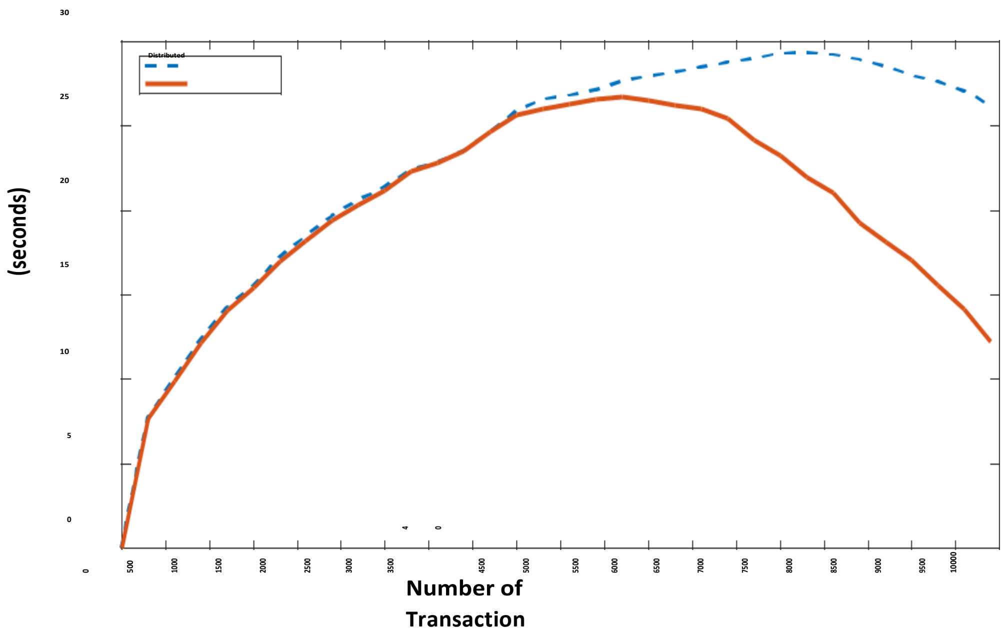


Fig. 7. System throughput

System Throughput



Conclusions

Big databases' use of network technologies in the future is attracting academic attention in the field of computing. More data dissemination is therefore required in order to make use of the current network technologies. Using a case study of the university registration system,this study demonstrated how to design and implement distributed database features like fragmentation, replication, dependability, and availability. A simulation programme wasused to generate a workload for both systems prior to data collection and analysis. The analysis's most obvious conclusion is that the decentralized registration system performs better than the centralized one. As a result, the suggested distributed system outperforms the centralized one in terms of average response time and average waiting time.

These results align with those of previous investigators. Additionally, by implementing the suggested method, the university's record-keeping becomes more effective, data availability and integrity are improved, time wasted on unnecessary tasks is reduced, and staff members are better able to serve instructors and students. Additionally, the suggested methodology offers consistent implementation of system forms and functions and tackles the issue of course equivalency. These findings have a big impact on how we interpret how to use the suggested schema in the real-world setting. A cloud system that achieves availability, reliability, and scalability while also boosting performance can be started using the suggestedapproach as its first core in the future.

These findings are consistent with those of earlier researchers. Additionally, by implementing the suggested method, the university's record-keeping becomes more effective, data availability and integrity are improved, time wasted on unnecessary tasks is reduced, and staffmembers are better able to serve instructors and students. Additionally, the suggested methodology offers consistent implementation of system forms and functions and tackles the issue of course equivalency. These findings have a big impact on how we interpret how to use the suggested schema in the real-world setting. A cloud system that achieves availability, reliability, and scalability while also boosting performance can be started using the suggested approach as its first core in the future.

Refrences

[1] M. T. Ozsu, "Principles of Distributed Database Systems, 3rd ed," *Prentice Hall Press*, 2011. Article (CrossRef Link).

[2] R. Elmasri and S. Navathe, "Fundamentals of database systems," *Pearson London*, 2016.

[3] S. B. Shende and P. P. Chapke, "Cloud Database Management System (CDBMS)," *Compusoft*,vol. 4, no. 1, p. 1462, 2015. Article (CrossRef Link).

[4] A. Mathur, M. Mathur, and P. Upadhyay, "Cloud Based Distributed Databases: The Future Ahead," *International Journal on Computer Science and Engineering*, vol. 3, no. 6, pp. 2477- 2481, 2011.Article (CrossRef Link).

[5] S. Alyaseri, "Distributed University Registration Database System Using Oracle 9i," *Computer and Information Science*, vol. 3, no. 1, pp. 59-67, 2010. Article (CrossRef Link).

[6] S. M. Motta, "Design of a Comprehensive Student Information System (SIS) and User Interfacefor the Honors College at USF," *USF College of Engineering USF Honors College*, 2010. Article (CrossRef Link).

[7] Y. D. Xaveria, "Design and Implementation of a Client Server Distributed Database for Student Results Processing," *The Pacific Journal of Science and Technology*, vol. 11, no. 2, pp. 288-295, 2010. Article (CrossRef Link).

[8] M. Little, S. Wheater, D. B. Ingham, C. Richard Snow, H. Whitfield, and S. K. Shrivastava, "The University Student Registration System: A Case Study in Building a High-Availability Distributed Application Using General Purpose Components," in *Proc. of Advances in Distributed Systems, Advanced Distributed Com-puting: From Algorithms to Systems, LNCS*, vol. 1752, pp. 453-471, 1999. Article (CrossRef Link).

[9] A. a. Al-Shaikh, "Online Registration System," *International Journal of Computer Science and Security (IJCSS)*, vol. 4, no. 3, pp. 331-345, 2010. Article (CrossRef Link).

[10] A. E. B. Mohamed, "An Integrated Remote Registration Systems in a Distributed Data Base

Environment," in *Proc. of 4th International Conference: Circuits, Systems & Computers, CSCC2000, World Scientific and Engineering Society press (WSES), IEEE, Vouliagmeni, Greece Conference 2000*, 2000. Article (CrossRef Link).

[11] A. Noraziah, A. A. C. Fauzi, M. M. Deris, M. Y. M. Saman, N. M. Zain, and N. Khan, "Managing Educational Resource - Student Information Systems Using BVAGQ Fragmented Database Replication Model," *Procedia - Social and Behavioral Sciences*, vol. 28, no. Supplement C, pp. 127-132, 2011. Article (CrossRef Link).

[12] T. Zhu, J. W. Guo, H. Zhou, X. Zhou, and A. Y. Zhou, "Consistency and Availability in Distributed Database Systems," *Journal of Software*, vol. 29, no. 1, pp. 131-149, 2018. Article (CrossRef Link).

[13] I. A. ALAMERI and G. RADCHENKO, "Development of Student Information Management System based on Cloud Computing Platform," *Journal of Applied Computer Science & Mathematics*, vol. 11, no. 2, pp. 9-14, 2017. Article (CrossRef Link).

[14] "Oracle-Corporation," *Two-Phase Commit Mechanism*, 2018. Article (CrossRef Link).

[15] Jordan, "The Annual Statistical Report," *Ministry of Higher Education and Scientific Research2016*, 2017. Article (CrossRef Link)..

[16] K. E. Kendall and J. E. Kendall, "Systems analysis and design, 9th ed," *Upper Saddle River, NJ:Pearson*, p. 552, 2013. Article (CrossRef Link).

[17] A. Boicea, F. Radulescu, C. O. Truica, and L. Urse, "Improving Query Performance in Distributed Database," *Journal of Control Engineering and Applied Informatics*, vol. 18, no. 2, pp. 57-64, 2016. Article (CrossRef Link).

[18] K. Gopalakrishnan, "Oracle Database 11g Oracle Real Application Clusters Handbook," *McGraw-Hill Education Group*, 2011. Article (CrossRef Link).

[19] J. Kuhlenkamp, M. Klems, and O. Röss, "Benchmarking scalability and elasticity of distributed database systems," *VLDB Endowment*, vol. 7, no. 12, pp. 1219-1230, 2014. Article (CrossRef Link).

[20] Oracle-Corporation, "Oracle's Understanding SQL*Net," *Oracle White Paper, California, USA*, 2002. Article (CrossRef Link).

[21] S. Fogel and P. Lane, "Oracle Database Administrator's Guide. 11g Release 2 (11.2)," *OracleCorp*, 2013. Article (CrossRef Link).

[22] R. Gallersd and M. Nicola, "Improving Performance in Replicated Databases through Relaxed Coherency," in *Proc. of the 21th International Conference on Very Large Data Bases*, pp. 445- 456, 1995. Article (CrossRef Link).

[23] Sibel Özyer, "Induction of Co-existing Items Available in Distributed Version Control Systems for Software Development", The International Arab Journal of Information Technology (IAJIT) ,Volume 20, Number 06, pp. 874- 879, November 2023, doi: 10.34028/iajit/20/6/4.

[24] Mahmoud A. and Zrigui M., "Semantic Similarity Analysis for Corpus Development and Paraphrase Detection," The International Arab Journal of Information Technology, vol. 18, no. 1, pp. 1-7, 2021. https://doi.org/10.34028