

# An Enhanced University Registration Model Using Distributed Database Schema

**Khaled Saleh Maabreh**

Faculty of Information Technology  
Zarqa University  
Jordan  
[e-mail: kmaabreh@zu.edu.jo]

*Received April 16, 2018; revised August 22, 2018; accepted December 9, 2018; published July 31, 2019*

---

## **Abstract**

A big database utilizes the establishing network technology, and it became an emerging trend in the computing field. Therefore, there is a necessity for an optimal and effective data distribution approach to deal with this trend. This research presents the practical perspective of designing and implementing distributed database features. The proposed system has been establishing the satisfying, reliable, scalable, and standardized use of information. Furthermore, the proposed scheme reduces the vast and recurring efforts for designing an individual system for each university, as well as it is effectively participating in solving the course equivalence problem. The empirical finding in this study shows the superiority of the distributed system performance based on the average response time and the average waiting time than the centralized system. The system throughput also overcomes the centralized system because of data distribution and replication. Therefore, the analyzed data shows that the centralized system thrashes when the workload exceeds 60%, while the distributed system becomes thrashes after 81% workload.

---

**Keywords:** Availability, Cloud Computing, Distributed Database, Fragmentation, Registration, Reliability

## 1. Introduction

One of the challenges faced by the universities in Jordan is the urgent need to standardize their student's registration information safely. Moreover, to solve the course equivalence problem; this problem involves how a course offered by one university relates to a course offered by another. The challenge indicates a need to understand the various perceptions to establish a reliable, scalable and the standard use of information, as well as to reduce the enormous efforts exerted for designing and implementing registration system individually. Therefore, the system will play a critical role in the solving of the course equivalence problem by containing standard and associated information about the courses offered by all universities, such as a course number, course name, pre-requisites and full description.

A distributed database may consider as a significant development in the computing field by using the data communication technology. Through this technology, databases are shifting from centralized to decentralized models to satisfy the requirements of computing aspects that may be difficult to obtain through centralized computing. Data in a distributed database can be saved as whole files or as fragments. Whereas, fragments could improve communications and data processing because of their distribution among sites [1] [2]. The distributed database system is also considered as an initial core to build cloud services using DBMS; this will give the user-freedom to access the information from anywhere at any time.

Furthermore, clouds can be used within these systems for enhancing the system performance [3] [4]. A great deal of academic work involves understanding the successful aims of a distributed database design. Such aims are the modularity of the system in which, processes can be easily modified, added and removed, improving the availability, where a fault in one database system will only affect one fragment instead of the entire database.

Regarding Jordan, it has more than 29 universities and institutions. All of them are using the credit hour as a studying system based on the Jordanian higher education regulations. Therefore, all registration systems almost have the same procedures, such as registering in a new course, adding or dropping a course, requesting a marks transcript, and so on. To design the system efficiently, many factors that may facilitate a suitable environment for carrying out this process have to be studied. Such factors are the small area of Jordan, the coverage of the computer network among all universities, and the database management systems currently used. Therefore, the bottom-up approach may be suitable for designing the system and could be less expensive.

This paper attempts to show the design and the implementation phases of a distributed registration database system for Jordanian universities as follows:

- Gathering the requirements needed to design the proposed system.
- Analyzing the requirements for identifying the specifications and the main procedures.
- Design the system concerning the higher education requirements.
- Virtual implementation to measure the expected benefits of the system will implement because of the lack of the required environment to design such a system regarding hardware and software.

## 2. Literature Review

A student registration system is a software that manages all day-to-day operations of a university. It provides a collection of programs to facilitate the student's enrolment and to better enabling the universities to supervise a growing number of services, as well as to reduce the work and costs involved in manual systems. Numerous studies have attempted to explain and present a comprehensive review regarding the student registration information system (for example, Sana Alyaseri, 2010 and Sean Motta, 2010) [5] [6]. They are analyzed and designed practical steps necessary for implementing such these systems to increase the efficiency of the colleges' and departments' records management. The implemented systems could reduce the work hours required to access student records and to enable the staff to provide better service to students.

Inadequacies of centralized database systems regarding reliable, scalable, and accessible information are discussed by Djam Xaveria Youh, 2010 [7]. He proposed a relational database system based on a client-server distributed database for processing student records and exams. Building a high-availability distributed application for a university student registration system is presented by Little MC et al., 1999 [8]. Their system guarantees the consistency of atomic transactions with a positive outcome regarding the completed registration process. Whereas, Ala'a M. Al-Shaikh, 2010 [9] solves the problem related to the management of the exam activities with reduction of errors resulted in the associated degree in Jordan. Mohamed B., 2000 [10] presented a remote registration system based on the distributed database to solve problems that both the staff and the students face during the registration period as well as to speed up and improve system performance and throughput. His model overcomes problems that are encountered when using the old registration system. It also saves the time and the effort by speeding up the registration process, reducing the costs, and maximizing the utilization of the Internet. The model also solves the problem of overlapped and conflicting schedules. Noraziah A. et al., 2011 [11] proposed a model called the binary vote assignment model for managing the student information system by considering the distributed database fragmentation. The model can preserve data consistency and increase the degree of parallelism because a transaction divides into several sub-queries that operate on the fragments.

Shende and Chapke, 2015 [3] address the recent trend in cloud service based on a database management system and offering it as one of the functions in the cloud. They also highlight the architecture of the cloud based on a distributed database for handling a large volume of data to improve the reliability, elasticity, availability, and scalability. Mathur et al., 2011 [4] discuss the cloud databases as a new trend to be used for data analysis, data warehousing and data mining purposes. They also investigated the advantages and shortcomings of using the database as a service.

Zhu et al., 2018 [12] Study the effectiveness of a distributed database on today's application with the explanation of centralized database shortcomings. They clarify the concept's dimensions of a distributed database to help the designers and the developers in making the suitable choice for completing their work. For providing convenient and reliable access to student information system locally or remotely, a cloud computing platform was used by Alameri, 2017 [13]. They gathered the requirements of a student information management system, analyzed and modeled the system components to facilitate the design and the

implementation of system architecture. Their proposed cloud system will promote the access to the system resources by using a web browser for all system users.

As a summary, the information plays a critical role in the development and growth of every university. Currently, various universities in Jordan independently manage student information in their own ways. No common processes and programs for retrieving student information are developed. From the higher education perspective, this situation has kept disintegrated student information in different universities. Thus, the collection of integrated information across universities is difficult and time-consuming. For example, if the higher education ministry needs information about the graduate students around all universities, it must manually send a request to all universities and then collect the required information, which requires a considerable amount of time and may result in different forms of data. Furthermore, if a student needs a transcript, then he must go to his university to receive it because the current system is unable to provide this document online. The course equivalence problem is also difficult to overcome in the current system because each university has its own way of describing its courses. Moreover, most courses are taught in most universities in the same way and may use the same references. To the best of my knowledge and concerning the literature, there has been no investigative research conducted regarding the design of a distributed registration database system for all universities in Jordan. Therefore, a proposed model is suggested in this research.

### 3. Research Methods

Preliminary investigations about the existing systems were carried out. A study of the current student registration information system, as well as their inadequacies, were also reported. In the following sub-sections, a prototype of a distributed database system was proposed, designed and virtually implemented.

Previous research tools that handle this issue are concentrated either on some colleges as in [6] by using the SQL server as a software tool. Or by studying the issue in a single university with various branches as proposed in [5] by using Oracle 9i. While the presented approach focuses on multiple universities in Jordan. Each university has its own regulations, applications, and database. Therefore, the proposed model facilitates solving the course equivalent problem, as well as making the information and their application use as a standard. Moreover, it is offering the re-usability for functions and forms.

In a distributed database system, an access to non-local data could be either by a remote transaction which accesses a single site, or by a distributed transaction that accessing the data at multiple locations. Oracle (the DBMS used in this proposed system) uses the two-phase commit processing protocol with a prepare phase. In this phase, the participating nodes are asked to commit or roll back the transaction. If all sites are prepared to commit, then the global transaction commits, else it will be rolled back. In a committing phase, and when all participants are guaranteed that they have the necessary resources to commit the transaction, they send an acknowledgment ensuring that all sites are prepared. In the commit phase, the global coordinator sends a message to all nodes telling them to commit. Furthermore, each

node commits the local portion of the distributed transaction and releasing locks. At the end of this phase, all replicas become consistent. [14].

### 3.1 Design of the Distributed Registration System

An ER segment of the proposed system is shown in Fig. 1. It contains the primary entities and the relationships among them; each entity consists of the fundamental attributes for considering as a sample that is to be used for the evaluation purposes. The corresponding partial conceptual schema is presented in Fig. 2, which specifies the key attributes of each entity type and the structural constraints on each relationship type, and illustrates the foreign keys used for joining purposes.

As a sample of the database instance, Table 1 shows the Jordanian higher education institutions for both the public and private sectors [15]. All of these institutions use the relational database model to represent the information system used in the students' registration process. Most of the universities are using an Oracle database for that purpose, whereas an Al al-Bayt University uses an Ingres database, which is also based on a relational model. Adopting a bottom-up approach will simplify the design of the distributed database system based on the relational database as a common model used. For simplifying the fragmentation techniques, the code 99 is used as an identification number for all universities to group the courses that are required as a general elective for all majors in all universities (university election). A proposed sample of the university colleges is shown in Table 2. For example, code number 6 for the Faculty of Information Technology is used around all universities that have this college. Code 999 will be used as an identification number for all colleges to identify the course group that is required of all students in a certain college around all universities (college election). The same thing (code 99) is used regarding the proposed department's name within the same colleges, as shown in Table 3. After organizing the courses offered by the system, the students will then review the scheduled timetable to plan their classes for the upcoming term.

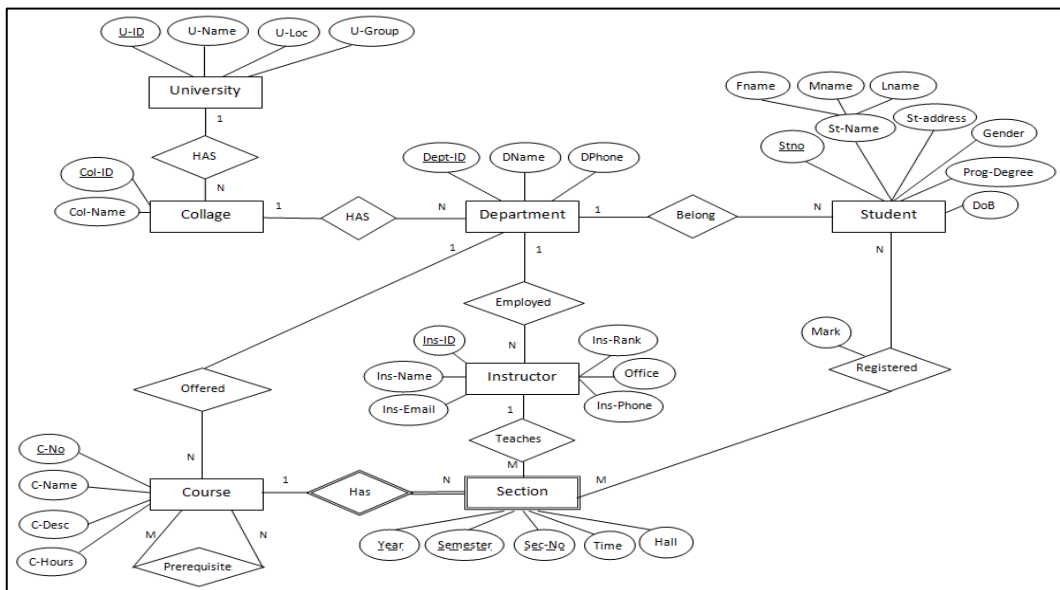


Fig. 1. ER Segment of the proposed system

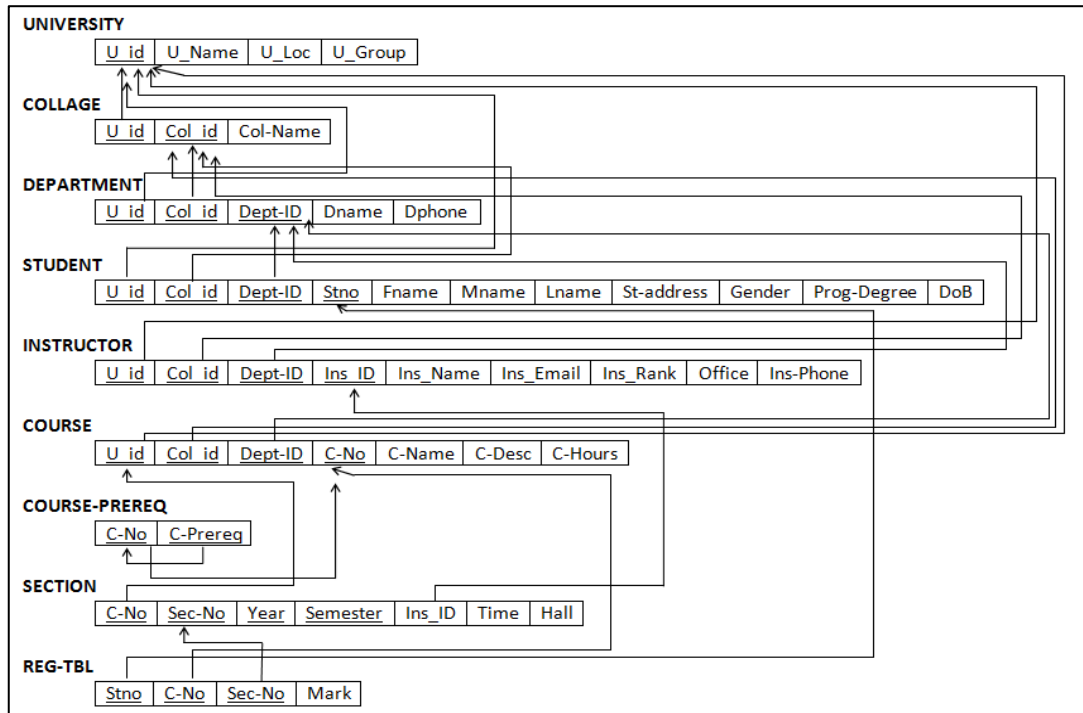


Fig. 2. The partial conceptual database schema for the proposed system

Table 1. Jordanian higher education institutions

U-ID	U-name	U-Location	U-Group
1	The University of Jordan	Amman	Public
2	Yarmouk University	Irbid	Public
3	Jordan University of Science and Technology	Irbid	Public
4	Mutah University	Karak	Public
5	The Hashemite University	Zarqa	Public
6	Al al-Bayt University	Mafraq	Public
7	AL-Balqa Applied University	Salt	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	Jadara University	Irbid	Private
13	Amman Arab University	Amman	Private
14	Middle East University	Amman	Private
15	Al - Ahliyya Amman University	Amman	Private
16	Applied Science University	Amman	Private
17	Philadelphia University	Jerash	Private
18	Isra University	Amman	Private
10	Petra University	Amman	Private
20	Al-Zaytoonah University of Jordan	Amman	Private
21	Irbid National University	Irbid	Private
22	Jerash University	Jerash	Private
23	Princess Sumaya University for Technology	Amman	Private

24	Jordan Academy of Music	Amman	Private
25	American University of Madaba	Madaba	Private
26	Ajloun National University	Ajloun	Private
27	Faculty of Educational Sciences and Arts /UNRWA	Amman	Private
28	Aqaba university of technology	Aqaba	Private
29	Jordan Applied University College	Amman	Private
99	All Universities	Jordan	Both

**Table 2.** A sample of college names within the universities

Col-ID	Col-name
1	Faculty of Medicine
2	Faculty of Dentistry
3	Faculty of Nursing
4	Faculty of Pharmacy
5	Faculty of Engineering
6	Faculty of Information Technology
7	Faculty of Science
8	Faculty of Business
9	Faculty of Arts
10	Faculty of Shari'a
...	.....
999	All Faculties

**Table 3.** Sample of department names

Dept-ID	Dname
1	Department of Anatomy and Histology
2	Department of Conservative Dentistry
3	Department of Clinical Nursing
4	Department of Pharmaceutical Sciences
5	Department of Civil Engineering
6	Department of Computer Science
7	Department of Computer Information Systems
8	Department of Software Engineering
9	Department of Arabic Language and Literature
10	Department of Geography
..	..
..	..
999	All Departments

### 3.2 Nonfunctional Requirements

A non-functional requirement is a set of conditions that have a measurable effect on a system operations behavior, which is detailed in the system architecture [16]. Some of the requirements to be accounted in the proposed system are:

- **Performance:** The system activities should be fast despite a large number of users.
- **Reliability:** The system must still be reliable even with a large number of users.
- **Ease of Use:** The system must make the user interaction simple and clear so that students who are not specialized in computers can efficiently use the system.

- **Load and Concurrency:** The system must deal with large quantities of data and a large number of users accessing these data, including timetable information and data simultaneously retrieved from the database by many users.
- **The familiarity of the interface:** The new system will have an interface that shares some of the old system behaviors so that users who are familiar with the old system can quickly adjust to the new system.
- **Real-time Feedback:** The new registration system should display the information and show the changes made to it in real time as in central processing.
- **Web Accessibility:** The system must be designed in such a way that is compatible with websites.
- **Availability:** The system must ensure that data continues to be available at a required level of performance; replicated data will facilitate this feature. Therefore, data exist in multiple copies at different sites.
- **Security:** The system must be secure and confidential through a set of authentication and authorization rules, as well as a set of encryption tools.

### 3.3 Data Fragmentation, Replication and Allocation

Data replication is an essential component used for managing huge amounts of resources in data architectures. It provides users with fast and local access to data, as well as increased availability and reliability because multiple data copies exist at all or some sites. It also enhances the efficiency and the effective use of data resources. On the other hand, data fragmentation in a distributed database improves the reliability and the effectiveness of data usage, and increases the degree of parallelism, because a transaction can be divided into several sub-transactions or agents that operate on the fragments. In this section, data fragmentation and replication will be described in further detail to obtain the maximum utilization of the proposed system.

A horizontal fragment is produced by specifying some predicates based on one or more attributes that restrict the database relation tuples. In the proposed system, each student studies at only one university. For ease of use, we decided that information for a given student should be stored in the DDBMS server at the university where the student studies. Therefore, the STUDENT table needs to be fragmented horizontally into 29 fragments based on the value of the university identification number (U-ID) column. An example of fragmentation using a relational algebra operation is shown below:

$$\text{STUDENT\_Frag}(i) = \sigma_{\langle \text{U\_ID}=i \rangle}(\text{STUDENT});$$

where (i) is the university ID and  $1 \leq i \leq 29$ .

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

$$\text{INSTRUCTOR\_Frag}(i) = \sigma_{\langle \text{U\_ID}=i \rangle}(\text{INSTRUCTOR});$$

where (i) is the university ID and  $1 \leq i \leq 29$ .

These commands produce 29 different fragments for each relation or table. Each fragment for both tables is located at the specified site (i.e., university).



Regarding the data availability in a database, it can be measured by the frequency of accessing the data in an acceptable time, and by the data volume that can flow at a time. It can be achieved through redundancy or replication in a distributed database. Fragmentation also enhances the availability by allowing the user query to access specific data on the fragment and leave the others. This method will reduce the average waiting time for other transactions that need other fragments for the same data object. Furthermore, reliability is also improved by increasing the probability that the system is running through available data [12] [17]. The number of concurrent transactions in a distributed database is growing which yields the volume of data also to increase. Therefore, scalability must be considered carefully to handle the growing number of users. Oracle easily handles the scalability either horizontally (scaling out) by adding new nodes or removing a node or by increasing the amounts of data, or vertically (scaling up) by upgrading server resources [18] [19].

### 3.4 Development and Implementation

Many limitations obstruct the actual implementation of the proposed system, such as network infrastructure limitations, the environment used among Jordanian universities, and the current laws and regulations of each university. Therefore, three virtual sites will be used for implementing the proposed system with actual data samples used in past semesters from different universities. One site will be used as a central site that will be responsible for defining global entities. The colleges, departments, instructors, students, and courses can be added from any site, and each site will be responsible for producing its own schedule (timetable). Students can be added, deleted, or registered from any site, regardless of their study location. The information about the students in each site is stored within its own site because of the fragmented data. The timetables of students, instructors, schedules, and the registration tables are fragmented, while the tables of colleges, departments, and courses are fully replicated because these tables are almost stable regarding updating. Therefore, update propagation problem will not affect the system performance.

To reduce the overhead that could affect the system performance in case of update propagation, all tables that required to be repeatedly written are placed locally. For example, every site has its own timetable which is specifically designed for each university, meaning that all entries that come from adding a new course, modifying a course section, or dropping a class will be written in this table locally, while the course schedule table has to be read-only because students do not have sufficient permissions to write on this table.

Moreover, the following assumptions are considered in developing the proposed system:

1. The students are enrolled in a single site, but they can register their courses from any site.
2. Each site (University) manages its own students.
3. A necessary fragmentation that takes into account an optimized manner through the balance between the resource costs and the performance achieved by the user (regarding response time and throughput) is created.
4. Attempts to optimally allocate the fragments to the best possible sites are made.
5. Attempts to optimally replicate the necessary data objects are made to satisfy local processing and to avoid update propagation problems.

Three sites are proposed and connected to each other by using the SQL\*Net (or Net8 for Oracle 8i and higher), which is an Oracle software presented for remote data accessing [20] [21]. It enables both client-server and server-server communications across any network. Therefore, the databases and their applications can reside on different computers and communicate as a peer-to-peer application. It also provides protocol independence to its applications. Therefore, an application can run over any network protocol and can be distributed without changing to other computers running other protocols. Oracle network provides a series of transparencies, such as location, replication, and fragmentation transparencies. Three-tier client-server architecture is used for the proposed system to formulate the user queries provided by the interface through the presentation layer. In addition, the database server then handles the queries by SQL. As an example of the proposed system tables, a student table will be created and fragmented (partitioned) into three fragments and located into three different universities as follows:

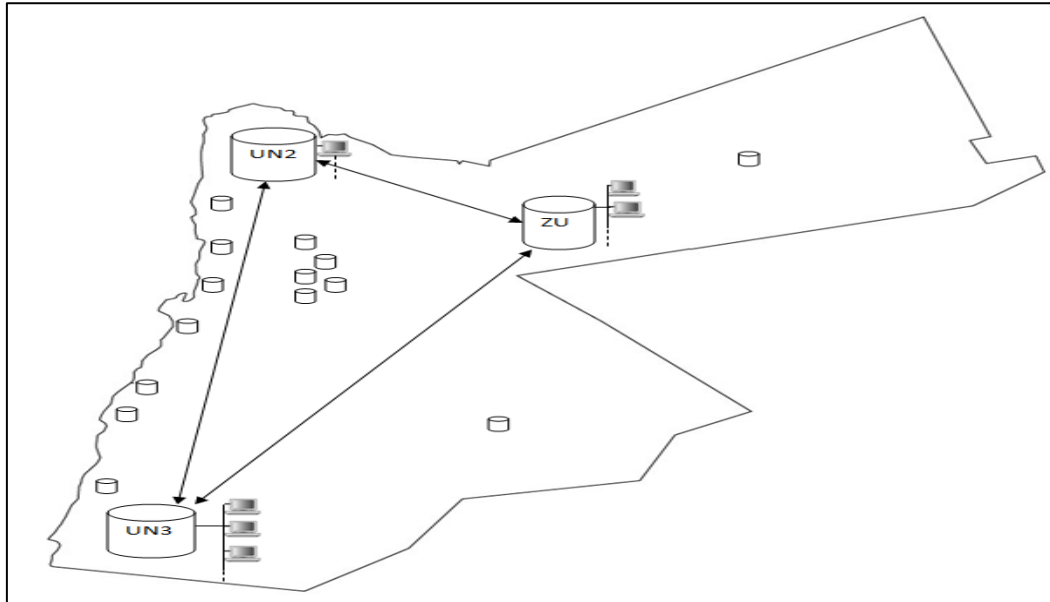
```

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 UN2_Student TABLESPACE tbl_spc2,
 PARTITION UN3_Student TABLESPACE tbl_spc3);

```

According to the database shown in Fig. 2, the above Create statement will produce 29 fragments. However, given that we do not have sufficient privileges to access all of the universities in Jordan, we proposed three universities (sites) in our study: Zarqa University and two other virtual universities (UN2 and UN3). Each fragment will be located at the specified university table space, tbl\_spc1 for Zarqa University, tbl\_spc2 for UN2, and tbl\_spc3 for UN3. The other tables will be created in the same way. The tables that need to be fragmented will be fragmented and located at the specified location. The replicated tables will be replicated over to some or all sites, as required. Each site may have its own tables, such as the schedule table (timetable).

A database link will be created among the university (site) databases. It is considered as a pointer defined as an entry in a data dictionary table that defines a one-way communication path from an Oracle database server to another. On the replication techniques, the replicated objects are used for the replicated data within the system. It is a database object existing on multiple servers in a distributed database system. However, any updates made to a replicated object at one site will apply to the copies at all of the other sites. Fig. 3 shows the homogeneously distributed database description of the proposed system. It also shows the geographical distribution of some Jordanian universities for future description updates.



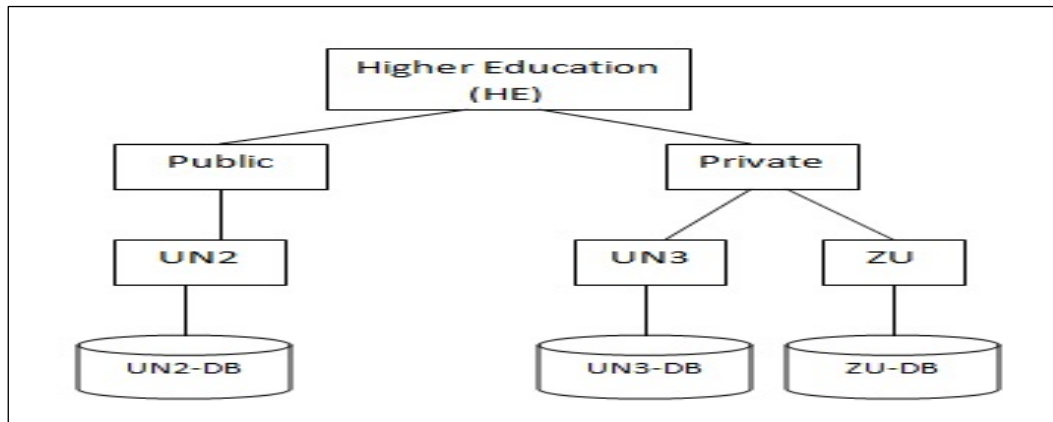
**Fig. 3.** Distributed database description for three universities

A database link will be created among the three sites. Each database in the system is uniquely distinguished by its global name by prefixing the database network domain. **Fig. 4** shows the representative hierarchical arrangement of databases throughout a network to illustrate global database names. Higher education is the root domain of the system that is subdivided into two sub-domains, that is, public and private, which represents the university sector. The subordinate to the public domain is UN2 with the UN2-DB database, whereas the subordinates to the private domain are UN3, which contains a UN3-DB database, and ZU-DB for ZU university. The following are examples of creating the database links.

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

```
CREATE PUBLIC DATABASE LINK
  UN3_DB.UN3.Private.HE USING 'UN3';
```

```
CREATE PUBLIC DATABASE LINK
  UN2_DB.UN2.Public.HE USING 'UN2';
```



**Fig. 4.** The hierarchical arrangement of distributed databases

By creating the public database links between sites, all of the users and PL/SQL subprograms in the database are allowed to access another user's objects in a remote database by the privilege set of the object owner. Therefore, a local user can access a remote database via a database link without having to be a user on the remote database.

#### 4. Results and Discussion

As was pointed out in the introduction to this paper, there are many difficulties facing the implementation of the system in real life. Some of these difficulties are the need to establish the infrastructure that is required for designing the system with the high cost involved as well as, the lack of the real data that have to be used. To overcome these difficulties and to proof the suggested model, a prototype for the student registration system will be implemented through two phases, the first one is based on a centralized database (one site), and the other one is constructed using a distributed database consisting of three sites. Furthermore, and in the absence of existing similar system for comparison purposes, the same data set that comes from discrete events simulation program is using in both mentioned approaches. The simulation program randomly fills the tables with a number of a database objects as follows: Student table has 10000 students, instructor table has 500 instructors, and the course table has 1000 different courses. These tables are fragmented and allocated across the sites when using the distributed database system. The simulation program also generates a set of random transactions according to the parameters presented in **Table 4**. To evaluate the system performance and throughput, the same transaction set is executed against both systems, and then, the data is gathered and analyzed.

To accurately mimic expected an application activity performance, the simulation program considers that each virtual user represents each real user. Multiple concurrent loads are also taken into account with varying transaction weights. For example, transaction A may check a course section (read an item) while another heavyweight transaction B can be as complicated as executing a registration process which involves stored procedure to be run on the results of a query. When setting up the test, create a sample with mixing transactions, which represents the characteristics of a population workload during peak usage in testing mode. For example, 60% registering students, 30% read-only queries, 10% instructors. These transactions need to virtually use the dynamic data. Dynamic transaction flows include

different operations such as adding or dropping courses, requesting marks transcript or submitting the course reservation. These operations are dynamically emulating the diverse activity of real users. Regarding distributed transaction, where two or more site are involved in performing a task, and in the case of writing operations, all participant must agree on committing to undertake the principal transaction, while the reading operations get their data locally or from the nearest site [14].

The parameters summarized in **Table 4** are collected using measurements in a proposed system by considering the assumptions of similar models found in the literature [22].

**Table 4.** System parameters and values

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

The two proposed systems were repeatedly executed to ensure the integrity of the collected data. The results of this study indicate that the distributed registration system behaves better performance than the centralized. These results are in agreement with those obtained from previous research work such as Sean Motta [6], 2010, Sana Alyaseri, 2010 [5], and Djam XaveriaYouh, 2010 [7]. To compare the performance of the two proposed systems, the average response time, average waiting time and the throughput are estimated. The proposed distributed system has better average response time than the centralized system as shown in **Fig. 5**. Moreover, this behavior is occurred due to the data replication, which enhances the data availability and then leads to increased local access. Although, both curves are increasing whenever the workload is increased because the load becomes heavy on the centralized system while the update processes are propagated in a distributed system. Besides, the average waiting time in a distributed application is less than in a centralized one as shown in **Fig. 6**. This results may be explained by the fact that the data is available in many alternatives (sites) which makes the access to these data as low as possible, especially, when the most of the queries mode is read. However, the read mode and the existing of multiple data item copies reduce the overall waiting time for access.

**Fig. 7** Shows the throughput of both proposed systems. A distributed system overcomes the centralized because the workload is distributed and data are replicated. However, these findings may be somewhat limited by the number of sites. Therefore, in future investigations and when the system is really implemented, it might be possible to enhance the system throughput by increasing both the number of sites and the degree of data replication.

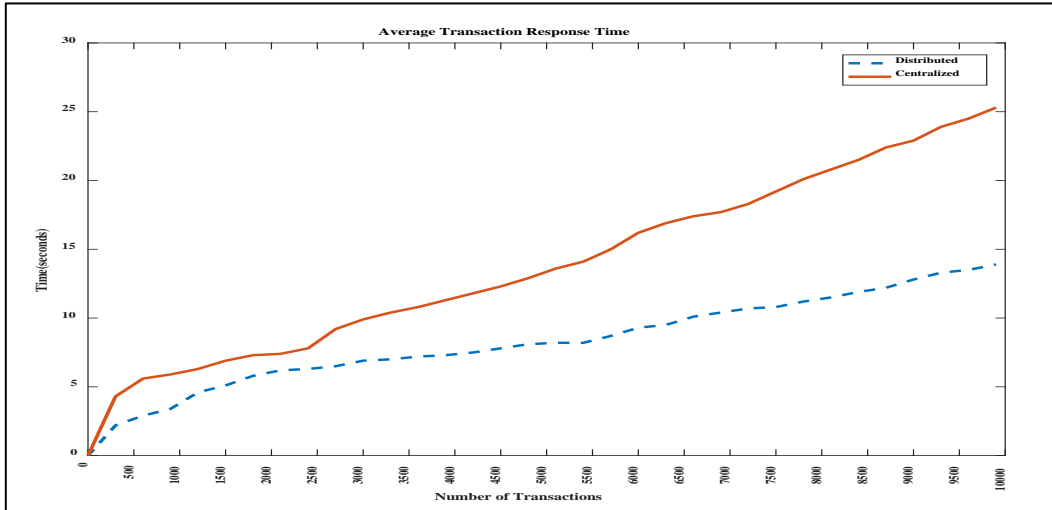


Fig. 5. Average transaction response time

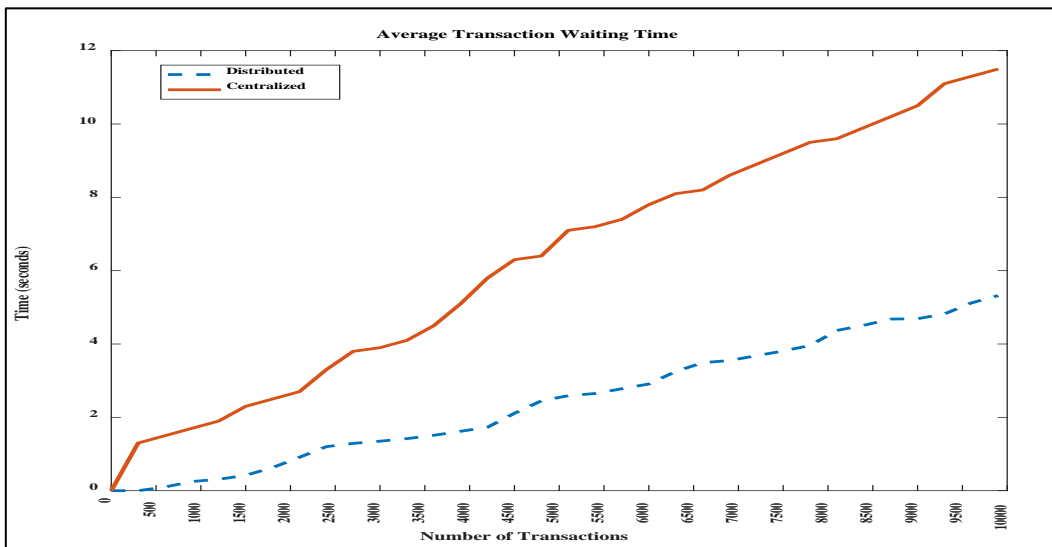


Fig. 6. Average transaction waiting time

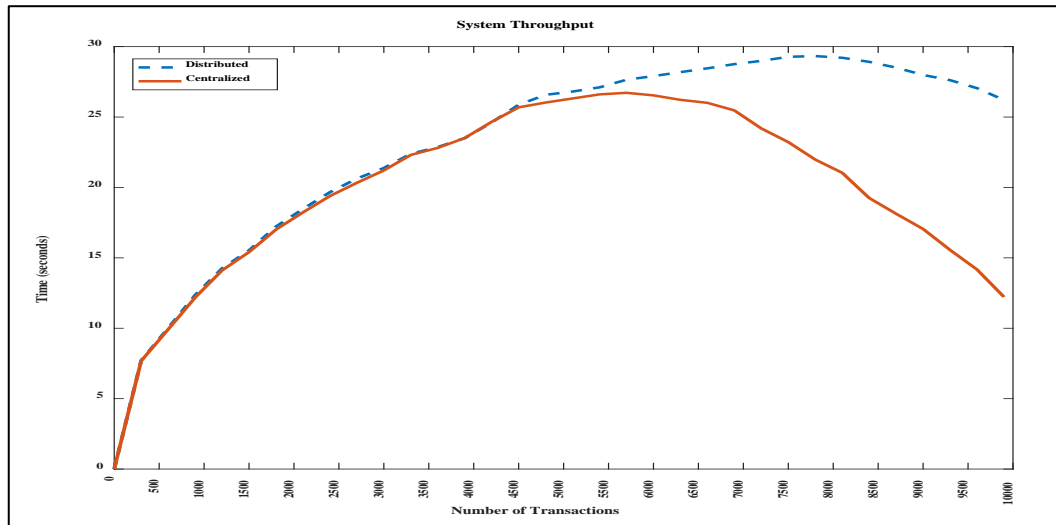


Fig. 7. System throughput

## 5. Conclusions and Recommendations

The future of big databases based on a network technology is becoming a topic of research interest in the computing field. Therefore, there are increasingly growing needs for more data distribution to utilize the established network technology. This research presented practical steps to design and implement a distributed database features such as fragmentation, replication, reliability, and availability through a university registration system case study. Furthermore, a simulation program was used to generate a workload for both systems, and then the data is collected and analyzed. The most apparent finding to emerge from the analysis is that the distributed registration system has better performance than the centralized. Hence the proposed distributed system has a better average response time and less average waiting time than the centralized system. As well as, it overcomes the centralized system throughput because the workload is distributed and data are replicated.

These results are in agreement with those obtained from previous researchers. The proposed system also increases the efficiency of the university's record management, increases data availability and integrity, minimizes the time spent on duplicated efforts, and enhances the ability of staff to provide better service to both the students and the instructors. Moreover, the proposed system solves the course equivalence problem and achieves the standardized use of system functions and forms. These findings have significant implications for the understanding of implementing the proposed schema in the real mini-world environment. Moreover, the suggested approach may be considered as an initial core for starting a cloud system in the future, which achieves the availability, reliability, and scalability, in addition to enhancing the performance.

As for future investigations, the regulations and instructions in Jordanian universities that would unify the study plans of the disciplines at all universities should be modified. Therefore, the system can be activated and operated in a realistic setting with an actual database. However, the study findings may be somewhat limited by the number of sites tested. Consequently, it might be possible to enhance the system throughput by increasing both, the number of sites, and the degree of data replication.

## Acknowledgments

This research is funded by the Deanship of the Research and Graduate Studies at Zarqa University/Jordan.

We are grateful to the Head of the Programming Division at Zarqa University/Nida Al-Shawish for the help in obtaining technical information about the admission and registration system at Zarqa University.

## References

- [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. [Article \(CrossRef Link\)](#).
- [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 Interface for 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. Wheeler, 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 Computing: 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 Research 2016*, 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)," *Oracle Corp*, 2013. [Article \(CrossRef Link\)](#).
- [22] R. Gellersd 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 \(CrossRef Link\)](#).



**Khaled Maabreh** Graduated in computer science from Yarmouk University, Jordan in 1989. He received MSc in Information Technology from Al-Neleen University, Jordan in 2003. The Ph.D. is obtained from Amman Arab University, Jordan in 2008. He has more than 24 years of experience, including developing IT-related projects in al-Albait University, Jordan from 1994-2008. His research interests are in performance modeling, distributed database, transaction processing, and E-learning. He has published several papers in various areas of computer science and information technology in International Indexed Journals and Conferences. Currently, he has a full academic post as assistant professor and the head of Computer Information System Department at Zarqa University - Jordan.

Email: kmaabreh@zu.edu.jo.