Database Administration Practical Aspects in Providing Digitalization of Educational Services

https://doi.org/10.3991/ijet.v17i20.32785

Pavel Petrov[1](✉), Ivan Kuyumdzhiev[1], Rami Malkawi[2], Georgi Dimitrov[3], Oleksii Bychkov[4]
[1]University of Economics – Varna, Varna, Bulgaria
[2]Yarmouk University, Irbid, Jordan
[3]University of Library Studies and Information Technologies, Sofia, Bulgaria
[4]Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
petrov@ue-varna.bg

Abstract—When delivering digitalization of educational services, a specific administration plan should be established, designed for maximum access, quick recovery from failure, and lowering the length of time it is unavailable, all while guaranteeing good security performance. It should include responsibilities such as providing dedicated servers for digitizing educational services, increasing security by using encryption and security-compliant network protocols, and database administration. To speed up the digitization of educational services, it is recommended that all databases be placed on a single server that will serve as the publisher, and that separate databases be distributed to three different servers that will act as the subscribers.

Keywords—system administration, educational services, digitalization, database administration

1 Introduction

Faced with widespread digitalization in educational services, every organization must strive to meet modern-day quality and security standards for information and other assets. Simultaneously, improving the utilization of available resources such as data, application systems, technologies, employees, and material resources is one of management's primary responsibilities. The growing number of educational services available via the Internet, as well as the digitalization of educational services, increases the role and importance of information systems. Educational institutions must provide and seek out opportunities to improve the efficiency of their systems on a regular basis.

To increase accessibility and productivity, the digitalization of educational services requires special attention. Online applications can serve many users, and it is difficult to predict the frequency and intensity of requests. In this regard, a special strategy for the administration of this system should be developed, which should allow maximum access, provide opportunities for rapid recovery from failure, and reduce the time it is unavailable, all while maintaining good security performance.

The system administration strategy should include activities such as providing dedicated servers for the digitalization of educational services, increasing security by adding encryption, additional operating system settings, and network protocols that adhere to security standards, and database administration.

The following main components (concepts) can be considered the primary directions in database system administration:

1. Identifying the content, structure, and integrity control of databases.
2. Managing database security through authentication modes, login accounts, roles, and permissions, as well as their management.
3. Database backup and restore – manage database backup and restore. Virus failure, unauthorized access, electrical failure, and database damage are all possibilities. To avoid fatal consequences from these adverse situations, strategies for archiving and restoring databases, as well as replacing damaged devices, are implemented.
4. Database performance observations and the development of visions for changes to improve productivity. Many routine database administration tasks, the majority of which are maintenance activities, can be automated. Automation necessitates a special server configuration that allows for the monitoring of user processes and actions, error checking, and warning when certain events occur. The tasks that are performed can be set to be performed only once or on a regular basis.

2 System administration strategy for availability and speed

To achieve good performance in many cases and to increase database accessibility, we propose combining two concepts in MS SQL Server that have previously been used separately – replication and database mirroring [1, 2].

Replication is a type of data transfer in which multiple copies of the same data are created and maintained. The data is distributed among the various countries, giving them greater autonomy, and providing necessary updates over time. There are several advantages to replication:

– Allows for the operation of stand-alone servers that do not need to be connected all the time.
– Allows for the maintenance of separate servers and systems, as working on the same device causes problems.

It is necessary to choose the roles to be performed by the servers, as well as the replication topology, in order to create an effective replication scheme in the digitalization of educational services. Different replication topologies can be developed because each SQL Server installation can be a publisher, distributor, subscriber, or any combination of the three – publishers create and modify data, resellers store data for one or more publishers, and subscribers receive the data.

To increase the speed of digitalization of educational services, we recommended to install all databases on one central master server which take the role of "publisher", and different databases replication to be sent on three different servers which taking the

role of "subscribers". For the type of replication, it is advisable to choose the one with the best speed – transactional replication. In this way, only changes to the publisher's databases will be sent, which will reduce network traffic and system load.

The data packets received from subscribers are referred to as publications, and their transmission is initiated whenever the publisher server changes. Publications include one or more articles, which can be part or all in one table or another database object. The replication mechanism also allows for horizontal (restriction of the rows it contains) or vertical (restriction of the columns it contains) filtering of the articles.

These replication characteristics allow precise load distribution and, accordingly, the required information on different servers, possibly located in different physical locations. Given the fact that the application can be accessed by geographically remote users, we recommend grouping data for audited organizations with close coordinates and placing them on the same server subscriber. This would further reduce the time required to retrieve information. This architecture provides good performance but is extremely difficult to maintain and in the event of a failure to recover information and return to normal operation is difficult.

We offer the creation of mirror copies of databases to improve the system's resilience. You'll need two SQL Server installations to implement a mirror strategy, and it's highly recommended that they be on different physical servers [3, 4]. The databases are accessed on one server, and all completed transactions are received on the other. If the master server fails, the roles can be reversed, with the mirror server taking over as master until the problem is resolved. As a result of this technology, the amount of time the system is unavailable is reduced.

To combine the above two technologies, consider the agents, serving the replication process. The replication process used by SQL Server includes other agents [5, 6] that are not relevant to the statement and will thus be omitted. Each one runs on the server, which serves as a distributor and performs a specific function:

- Snapshot Agent synchronizes the databases of the various servers for the first time.
- The Log Reader Agent connects to the publisher server and retrieves transactions marked for replication in the reseller.
- Distribution Agent – establishes contact with subscribers and transmits the necessary transactions.

The success of database mirroring and replication is solely dependent on selecting the appropriate database and, more specifically, the role played by the server on which it is installed. The information stored on subscriber servers can be retrieved at any time from the publisher server via the distributor. The distributor server, on the other hand, does not store data that cannot be retrieved at a loss, so creating a mirror is pointless. As a result, the server publisher remains the architecture's weak point.

To successfully implement the proposed topology, the Log Reader Agent must be properly configured. It reads information from the main publisher server during normal server operation. Simultaneously, the process of creating a mirror copy of the databases ensures that all changes made to the main server publisher are replicated in its mirror. If the main server crashes, the Log Reader Agent continues to retrieve information about the reseller, but this time from a mirror image of the server. The process can be repeated

until the main server is fully restored, at which point the publishing servers can resume their original roles.

This scheme enables different types of data to be sent to subscriber servers to serve individual audited companies. The load is reduced by distributing the database across multiple servers and creating a mirror image of the publisher server allows for automatic response in the event of a failure. Companies will have constant access to various forms and reports containing audit information, and changes to the database will be sent to the publisher server.

3 System administration strategy for backup and restoration

A unique database architecture necessitates a unique backup and recovery strategy. The database's operation is critical to the organization's survival; in the event of a system failure, some of the data may be damaged or lost. To address issues related to database corruption, an archiving strategy must be developed.

A successful archiving strategy ensures the following [7, 8, 9]:

- minimizing data loss;
- recovering lost data;
- recovering data with minimal production time costs.

The topology of the system has a significant impact on the archiving strategy in this case. It's redundant to create backup databases on subscriber servers and the reseller, and the primary publisher server keeps a mirror image of its databases on another server ready to take over the role. To avoid a scenario in which both publisher servers fail, archives of the databases on one of them must be created. The archives must be run to the server that receives the database mirrors and then stored in a different location because the load on the server that receives the database mirrors is lower. In most cases we consider it will be sufficient to run periodic full archives and transaction logs because both servers are less likely to crash when they are in different physical locations.

When running full archives, which are usually time consuming, the database will still be available, and transaction log archives will have no effect on server performance.

The proposed administration strategy for the digitalization of educational services creates preconditions for ensuring the quickest possible access and offers opportunities for quick recovery from failure and reducing the time it is unavailable. Adherence to the above security management principles, as well as regular inspections, will improve the resilience of educational services to malicious external factors.

The loading speed of websites has always been a significant factor in determining their quality [10, 11], alongside multilingual support and the use of standards. It is dependent on several factors, including the server's hardware and software, the caching of static information, the server's Internet connection, and the availability of multimedia content [12, 13]. Even with strict monitoring of each of these factors, a situation may arise in which server resources are insufficient to service the requests in appropriate time frame in applications with too many users. Unwanted delays may occur at higher loads or when the maximum number of turns permitted is reached at the same time. For example, if there are 100 active database calls at the same time and the limit is only 30,

the remaining 70 will be queued while the server processes the previous ones. Dynamic caching is one of the mechanisms used to address this issue.

In essence, caching is a record of the outcome of an operation (database query or formatted output), typically in the form of a file, and its visualization in a subsequent request to retrieve the same result. In the presence of a large number of users, information caching is primarily used. The required information is downloaded and saved only once with dynamic caching. At each new execution of the script that outputs the data, a check is made for the presence of cached information to be visualized; if there is none, a request to the database is made and a new cache copy of the result is saved.

Not only is there a significant advantage in data retrieval, but also in the speed with which the output is visualized. The reason for this is that it is not necessary to re-run all the development tools in order to generate the output of a given page; instead, it is only necessary to display its last saved copy. This takes a fraction of the time it takes for the system to regenerate everything.

We propose to use one of the most common approaches for dynamic data caching in web application development systems, which is to save the results of SELECT queries as serialized arrays in files. It is much faster to read a file and deserialize arrays than to run a complex SELECT query with many tables. This cuts server access and data retrieval time in half.

As each system's functionality grows, so do the chances of unexpected flaws. The error events generated by the described development tools must be closely monitored. In most development environments, the module is referred to as Error Handler. This component is essential for system maintenance and monitoring. It enables rapid response and the elimination of errors in its operation. In general, this component provides all necessary information in the form of an e-mail, which is sent to the developer who created the functionality with each error. The error generated by the server is included, as well as the name and full path to the file where the error occurred, the line number in the code, and all subsequent files and libraries affected by the error files and libraries with the corresponding lines and paths, all the way to the operation's end. A list is used to organize and grade all the data.

In the event of an error, an additional benefit of using this approach is the provision of specific information that can be seen by ordinary users of the system. As a result, in addition to the purely aesthetic effect, valuable system information is protected from leakage. This includes file names, databases, and tables, which improves security and lowers the risk of hacking.

4 System administration strategy for security

When developing a security strategy, we must consider the system's topology, which is hidden from the end users. On several levels, the security system can be implemented:

– Physical security as well as network protocol security.
– The safety of the operating system.
– Database protection.

Given the sensitive nature of the data, all connections in the system must be encrypted, both between users and the database and between individual servers. The analysis of possible solutions reveals that the widely used SSL/TSL protocol provides a high level of security [14]. It ensures that the information sent remains unchanged and sends it only to the intended server, encrypting it with an asymmetric algorithm and a public key. It is supported by almost all modern browsers, client-server applications, and a wide range of communication devices [15]. In case of VPNs the possible use and impact of other well-established protocols such as IPSec which can provide strong authentication should be considered.

Physical access to the servers must be restricted, and they must be housed in facilities that meet current security standards. After ensuring physical security, special attention must be paid to operating system security. The servers on which the systems for digitalization of educational services will be installed must have the following technical control mechanisms:

- limited access rights, created and maintained in accordance with educational service needs.
- risk assessment, vulnerability, prevention, and detection, as well as continuous monitoring.
- conducting breakthrough tests on a regular basis.
- the existence of a change management process to ensure that changes and refinements to system software, educational services software, network components, and data are made in a controlled manner.

The database management system's security is limited to creating accounts to restrict access to the database server. Each of these accounts must have no more than the bare minimum of rights. To keep their number manageable, we propose the establishment of a common account for all employees in audited companies and separate accounts for members of audited companies' management as well as all employees in auditing companies.

The administrator should only allow authorized updates based on the database connections. If users are permitted to use web forms, the administrator must include mechanisms in the databases to validate all updates to ensure their authorization and security.

Controlling table access is difficult to design and implement, so it is frequently overlooked. Effective collaboration between system users, including the developer and database administrator, is required for success. It is necessary to conduct regular database security audits to achieve a good result. They're usually related to the transactions that have been recorded in the transaction log. Its use in most organizations is limited to a tool that provides information after a security issue is reported.

We recommend recording and analyzing the following activities for successful administration:

- Successful and unsuccessful login attempts.
- Successful and failed attempts to gain access to security-critical sites, such as creating, opening, closing, modifying, and deleting.
- Modifications to user authentication data.
- Blocking users based on which ports are blocked and why.
- Access denial because of too many failed login attempts.

Procedures related to the analysis of potential problems can be performed using both specially written scripts and commonly available finished products, such as: Nmap (https://nmap.org/), Nessus (https://www.tenable.com/products/nessus), LanGuard (https://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard), AIDA64/Everest (http://www.lavalys.com/) and others.

Nmap is a free tool that lets you create a complete network map, including computers, servers, services provided by each machine, and open ports [16, 17].

Nessus is a free program for personal use only; it cannot be used for commercial purposes. Allows you to scan network devices and set a specific policy to comply with during the scanning process [18]. When the results are received, the program identifies potential security breaches, influencing factors, and appropriate solutions. The application can also be used for continuous network monitoring to detect improper software use, monitor the use of USB devices, and detect changes in server configuration.

Network scanning, checking for availability and installing the latest updates, monitoring the use of USB devices, remote access, and verification of compliance with the password management policy are all services offered by Languard [19, 20], which are like Nessus. The program can be used for no more than two computers for free.

AIDA64/Everest (the new product name is AIDA64) provides detailed information on each installed hardware component of the computer (processor, hard disk, video card, and so on), performs comparative tests, and includes diagnostic and productivity reporting tools [21].

5 Conclusion

An archiving strategy must be established to handle database corruption issues. The operation of the database is important to the organization's sustainability. Backup databases are created redundantly on subscriber and reseller servers, and the principal publisher server maintains a mirror image of its databases on another server. To eliminate the possibility of both publisher servers failing, archives of the databases on one of them must be generated. The archives must be run to the server that receives the database mirrors and then saved somewhere else. One of the strategies used to alleviate this issue is dynamic caching. At each new execution of the script that outputs the data, the presence of cached information is checked.

When establishing a security plan, we must examine the topology of the system. All system connections, both between users and the database and between specific servers, must be secured. The servers' physical access must be controlled, and they must be kept in facilities that fulfill contemporary security standards. The security of the database management system is limited to creating accounts to restrict access to the database server. Each of these accounts should just have the bare minimum of rights. Procedures for analyzing possible problems can be carried out utilizing both specially generated scripts and commercially available final products.

6 Acknowledgment

Financed by NPI-45/2020 from University of Economics – Varna Science Fund.

7 References

[1] Katta, R., Ali, A. A., Chramcov, B., Krayem, S., & Jasek, R. (2021). Formal development of fault tolerance by replication of distributed database systems. In *Computer Science On-line Conference*, 293–306. https://doi.org/10.1007/978-3-030-77442-4_25

[2] Pohanka, T., & Pechanec, V. (2020). Evaluation of replication mechanisms on selected database systems. *ISPRS International Journal of Geo-Information, 9*(4), 249. https://doi.org/10.3390/ijgi9040249

[3] Ma, Y. (2019). Design and implementation of a college teacher training system based on client-server structure. *International Journal of Emerging Technologies in Learning (iJET), 14*(12), pp. 121–132. https://doi.org/10.3991/ijet.v14i12.10716

[4] Khan, F., & Alotaibi, S. R. (2020). Design and implementation of a computerized user authentication system for E-learning. *International Journal of Emerging Technologies in Learning (iJET), 15*(09), pp. 4–18. https://doi.org/10.3991/ijet.v15i09.12387

[5] Butterstein, D., Martin, D., Stolze, K., Beier, F., Zhong, J., & Wang, L. (2020). Replication at the speed of change: A fast, scalable replication solution for near real-time HTAP processing. *Proceedings of the VLDB Endowment, 13*(12), 3245–3257. https://doi.org/10.14778/3415478.3415548

[6] Carter, P. A. (2016). SQL server agent multi-server environments. In *Expert Scripting and Automation for SQL Server DBAs*, 43–75. https://doi.org/10.1007/978-1-4842-1943-0_3

[7] Thomas, J., & Galligher, G. (2018). Improving backup system evaluations in information security risk assessments to combat ransomware. *Computer and Information Science, 11*(1). https://doi.org/10.5539/cis.v11n1p14

[8] Mao, L. (2018). Application of browser/server architecture in college English online learning system design. *International Journal of Emerging Technologies in Learning (iJET), 13*(03), pp. 129–140. https://doi.org/10.3991/ijet.v13i03.8395

[9] Kuyumdzhiev, I. (2019). Comparing backup and restore eficiency in MySQL, MS SQL server and MongoDB. *International Multidisciplinary Scientific GeoConference: SGEM*, 19(2.1), 167–174. https://doi.org/10.5593/sgem2019/2.1/S07.022

[10] Bartuskova, A., & Krejcar, O. (2015). Loading speed of modern websites and reliability of online speed test services. In *Computational Collective Intelligence*, 65–74. https://doi.org/10.1007/978-3-319-24306-1_7

[11] Mousa, I. R. (2019). Development of a numerical index to assess the quality of websites design. *Webology, 16*(2), 72–82. https://doi.org/10.14704/WEB/V16I2/a191

[12] Shroff, P. H., & Chaudhary, S. R. (2017). Critical rendering path optimizations to reduce the web page loading time. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, 937–940. https://doi.org/10.1109/I2CT.2017.8226266

[13] Huang, J., Zhu, H., Liu, M., Zhang, T., & Wang, J. (2022). Achieving fast page load for websites across multiple domains. *Transactions on Emerging Telecommunications Technologies*, e4439. https://doi.org/10.1002/ett.4439

[14] Petrov, P., Dimitrov, G., & Ivanov, S. (2018). A comparative study on web security technologies used in Irish and Finnish banks. In *Conference Proceedings of 18 International Multidisciplinary Scientific Geoconference SGEM 2018*, 2–8. https://doi.org/10.5593/sgem2018/2.1/S07.001

[15] Petrov, P., Dimitrov, P., Stoev, S., Dimitrov, G. P., & Bulut, F. (2020). Using the universal two factor authentication method in web applications by software emulated device. *International Multidisciplinary Scientific GeoConference: SGEM*, 20(2.1), 403–410. https://doi.org/10.5593/sgem2020/2.1/s07.052

[16] Shah, M., Ahmed, S., Saeed, K., Junaid, M., & Khan, H. (2019). Penetration testing active reconnaissance phase-optimized port scanning with nmap tool. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1–6. https://doi.org/10.1109/ICOMET.2019.8673520

[17] Liao, S., Zhou, C., Zhao, Y., Zhang, Z., Zhang, C., Gao, Y., & Zhong, G. (2020). A comprehensive detection approach of nmap: Principles, rules and experiments. In *2020 International Conference on Cyber-Enabled Distributed Computing*, 64–71. https://doi.org/10.1109/CyberC49757.2020.00020

[18] Jetty, S. (2018). *Network Scanning Cookbook: Practical Network Security Using Nmap and Nessus 7*. Packt Publishing Ltd.

[19] Kumar, R., & Tlhagadikgora, K. (2018). Internal network penetration testing using free/open source tools: Network and system administration approach. In *International Conference on Advanced Informatics for Computing Research*, 257–269. https://doi.org/10.1007/978-981-13-3143-5_22

[20] Tanasache, F., Sorella, M., Bonomi, S., Rapone, R., & Meacci, D. (2019). Building an emulation environment for cyber security analyses of complex networked systems. In *Proceedings of the 20th International Conference on Distributed Computing*, 203–212. https://doi.org/10.1145/3288599.3288618

[21] Nikitin, V. I., & Dronov, Y. V. (2021). Development and analysis of a method for increasing the security of hardware and software of industrial facilities operating on the basis of lans of various hierarchical structures. In *Information Innovative Technologies*, 16–21.

8 Authors

Pavel Petrov is working as Associate Professor in the Department of Informatics at the University of Economics – Varna, Bulgaria. His research interests include distributed web systems, big data, and cloud computing. (email: petrov@ue-varna.bg).

Ivan Kuyumdzhiev is working as Associate Professor in the Department of Informatics and Computer Science at the University of Economics – Varna, Bulgaria. He is Director of the Center for Research and Application of New Information and Communication Technologies (CIPNICT). (email: ivan_ognyanov@ue-varna.bg).

Rami Malkawi is working as Assistant Professor in Information Systems Department, Faculty of Information Technology at Yarmouk University in Jordan. Before pursuing his academic career, Dr. Malkawi worked as Web designer, Computer Programmer and Systems Analyst for more than 10 years. (email: rmalkawi@yu.edu.jo).

Georgi Dimitrov is working as Professor in the University of Library Studies and Information Technologies, Sofia, Bulgaria. He is Vice Dean of Department of Information Technology (email: g.dimitrov@unibit.bg).

Oleksii Bychkov is working as Associate Professor in the Taras Shevchenko National University of Kyiv, Kyiv, Ukraine. He is Head of the Department of Program Systems and Technology (email: bos.knu@gmail.com).

Article submitted 2022-05-28. Resubmitted 2022-07-08. Final acceptance 2022-07-08. Final version published as submitted by the authors.