DIGITAL SECURITY, CYBERSECURITY AND RESILIENT INFORMATION SYSTEMS

DIGITAL RISK MANAGEMENT IN RELATIONAL AND NON-SQL DATABASES

CZU: [004.65:004.056]:005.334
DOI: https://doi.org/10.53486/tids2026.26

SERGHEI OPREA
Academy of Economic Studies of Moldova,
Chisinau, Republic of Moldova,
e-mail: opreaserghei@ase.md
ORCID ID: 0000-0003-3759-8598

AURELIU ZGUREANU
Academy of Economic Studies of Moldova,
e-mail: zgureanu.aureliu@ase.md
ORCID ID: 0000-0003-3301-2457

Abstract: This comparative study analyses digital risk management strategies applicable to relational databases (SQL) and non-relational databases (NoSQL), in the context of persistent cyber threats and increasingly stringent compliance requirements imposed by information security standards. The paper examines the specific vulnerabilities associated with each data organization model, the relevant legislative framework (GDPR, ISO/IEC 27001), and risk mitigation techniques such as data encryption, role-based access control (RBAC), transaction auditing, and business continuity planning within the database environment. The comparative analysis demonstrates that, although NoSQL databases offer superior scalability, they are exposed to a considerably higher attack risk owing to flexible schemas and the traditional absence of ACID transaction guarantees. The study's findings suggest that a hybrid approach — combining the mature security mechanisms of relational systems with the flexibility inherent in the NoSQL paradigm — constitutes an optimal strategy for modern information systems.

Keywords: digital risk management, relational databases, NoSQL, cybersecurity, access control, data auditing, security standards.

JEL Classification: M15, L15

1. INTRODUCTION

Digital risk management has become a strategic priority for modern information systems amid the accelerating digital transformation of society and the exponential growth in the volumes of data stored and processed. Databases constitute one of the most valuable and, simultaneously, most vulnerable information assets of any system. Cyber-attacks targeting databases have increased significantly over the past decade, and the average cost of a data breach reached USD 4.45 million in 2023 (IBM Security, 2023), underscoring the imperative for rigorous protection strategies.

Traditional architectures based on relational database management systems (RDBMS) — such as PostgreSQL, MySQL, Oracle, and Microsoft SQL Server — have benefited from decades of security research and have incorporated mature protection mechanisms. Concurrently, growing demand for scalable systems has driven the widespread development of non-relational (NoSQL) databases, including MongoDB, Cassandra, Redis, and Neo4j — systems that, whilst offering

performance and flexibility, raise new challenges from the perspective of information security (Tudorica & Bucur, 2011).

Recent studies (Mehmood *et al.*, 2016) have found that NoSQL databases are frequently misconfigured and accessible without authentication across public networks. This reality necessitates a rethinking of risk management paradigms tailored to each type of database.

The objectives of the present paper are: (1) to identify and classify the main categories of digital risks specific to relational and NoSQL databases; (2) to conduct a comparative analysis of the security mechanisms available within the two paradigms; (3) to evaluate the applicable normative framework; and (4) to formulate a set of recommendations for the implementation of integrated data risk management.

2. CONCEPTUAL FRAMEWORK OF DIGITAL RISK MANAGEMENT

2.1 Defining Digital Risk in the Database Context

Digital risk refers to the probability and potential impact of events that may compromise the confidentiality, integrity, or availability of data stored in information systems — principles collectively known under the CIA triad (Confidentiality, Integrity, Availability) (Whitman & Mattord, 2021). Digital risk management (DRM) constitutes the systematic process of identifying, assessing, treating, and monitoring these risks through the application of standardized methodological frameworks.

The ISO/IEC 27005:2022 standard, dedicated to information security risk management, defines risk as a combination of the likelihood that a threat will materialize and the consequences thereof for the organization's assets (ISO/IEC, 2022). Applied specifically to databases, this framework requires asset inventory, identification of relevant threats, assessment of existing vulnerabilities, and calculation of the residual risk level.

Curtmola *et al.* (2006) introduced the concept of "Proof of Retrievability" (PoR) as a mechanism for verifying data integrity in distributed systems, a concept subsequently extended to cloud database environments where data custody verification becomes essential for risk management. This approach highlights that risks associated with databases extend beyond the traditional sphere of perimeter security and encompass aspects of data provenance, tamper-resistance, and recoverability.

2.2 Taxonomy of Database-Specific Risks

Digital risks associated with databases may be classified into several principal categories. The first category comprises unauthorized access risks, which include SQL Injection attacks — one of the most thoroughly documented vulnerabilities, present in the OWASP Top 10 for over two decades (OWASP Foundation, 2021) — privilege escalation, access via compromised accounts, and exploitation of insecure default configurations.

The second category encompasses data integrity risks, manifested as data corruption resulting from hardware failures, concurrency errors in simultaneous transactions, or deliberate attacks. Mohan (1992) describes write-ahead logging and transaction-based recovery mechanisms in relational systems as the cornerstone of data integrity, a foundational contribution to understanding the differences relative to NoSQL systems, which relax ACID properties in favor of the BASE model (Basically Available, Soft state, Eventual consistency).

The third category addresses availability risks, generated by Denial-of-Service attacks (DoS/DDoS) targeting the database layer, storage infrastructure failures, or the absence of adequate disaster recovery plans. The final major category consists of compliance risks, arising from non-

compliance with legal requirements concerning the protection of personal data, particularly the General Data Protection Regulation (GDPR) within the European context.

3. SPECIFIC VULNERABILITIES IN RELATIONAL AND NoSQL DATABASES

3.1 Vulnerabilities in Relational Databases

Relational databases present a well-documented vulnerability profile. SQL Injection remains the most prevalent threat, exploiting deficiencies in application-level input validation to execute arbitrary commands against the database. Halfond *et al.* (2006) produced an exhaustive taxonomy of SQL Injection attacks, classifying six principal techniques: tautologies, illegal/erroneous queries, union-based attacks, piggyback queries, inference attacks (blind SQL injection), and stored procedure exploitation.

Privilege escalation represents another critical vulnerability in relational systems, amplified by non-compliant administration practices such as excessive privilege grants or the use of service accounts with administrative rights. Clarke (2009) documents that over 80% of database-level security incidents involve legitimate users abusing improperly granted privileges, underscoring the importance of the Principle of Least Privilege (PoLP).

Vulnerabilities related to unsecured communication channels, default server configurations, and inadequate patch management complete the picture of principal risks for relational systems. Litchfield's (2005) study on Oracle Database security reveals the existence of severe vulnerabilities in authentication components and PL/SQL mechanisms, indicating that delayed application of security patches creates significant windows of exposure.

3.2 Vulnerabilities in NoSQL Databases

NoSQL databases, originally conceptualized for performance and scalability, lacked robust security mechanisms in their early versions. Okman et al. (2011), analyzing security issues specific to NoSQL databases, identified five principal vulnerability categories: absence of default authentication, lack of granular access control, injection vulnerabilities (NoSQL Injection), cryptographic weaknesses, and risks associated with data distribution across clusters.

NoSQL Injection represents the equivalent of SQL Injection for non-relational systems and exploits the flexible nature of JSON document-based queries or system-specific operators. Studies by Sinha & Mall (2019) demonstrate that applications using MongoDB are vulnerable to injections via operators such as $where or $regex when user inputs are not properly validated. The impact can be equivalent to that of classical SQL Injection, including exfiltration of the entire database.

The eventual consistency model adopted by many NoSQL systems introduces a specific risk: temporary windows of inconsistency during which different nodes of the cluster may return different versions of the same data. Pritchett (2008) describes the CAP theorem (Consistency, Availability, Partition tolerance), demonstrating that distributed systems cannot simultaneously guarantee all three properties. This fundamental limitation has direct implications for transactional integrity in financial or medical applications.

An emerging risk specific to cloud-native NoSQL environments is the prevalence of insecure default configurations. The study by Haber & Rolls (2020) on identity management in cloud environments documents that a significant proportion of cloud database instances — including MongoDB Atlas, Amazon DynamoDB, and Google Firestore — are exposed to the Internet without authentication enabled, owing to permissive default settings or administrative errors.

4. SECURITY MECHANISMS AND RISK MITIGATION STRATEGIES

4.1 Access Control and Authentication

Role-Based Access Control (RBAC) constitutes the foundation of any database security strategy. Ferraiolo et al. (2001) formalised the RBAC model, demonstrating that it enables efficient implementation of the Principle of Least Privilege by associating permissions with organizational roles rather than directly with individual users. Mature relational systems such as Oracle Database and PostgreSQL implement RBAC natively, with support for role hierarchies and Separation of Duties (SoD).

NoSQL databases have progressively adopted RBAC mechanisms in more recent versions. MongoDB introduced full RBAC support in version 2.6 (2014), including predefined roles and the ability to define custom roles. Nevertheless, comparative studies by Grolinger et al. (2013) reveal that the granularity of access control in NoSQL systems remains inferior to that available in mature RDBMS environments, limiting the capacity to implement complex security policies.

Multi-factor authentication (MFA) and certificate-based authentication protocols (mTLS) represent additional layers of security recommended for access to critical databases. Liu et al. (2015) propose an adaptive authentication model for cloud databases that adjusts authentication requirements according to access context (location, device, behavior), significantly reducing the risk associated with credential compromise.

4.2 Data Encryption

Encryption constitutes an essential protection mechanism for both data at rest and data in transit. Boneh & Franklin (2001) introduced Identity-Based Encryption (IBE) as an advanced paradigm, subsequently applied in cryptographic key management scenarios for distributed databases. Transparent Data Encryption (TDE), available in SQL Server, Oracle, and MySQL Enterprise, protects the physical database files without requiring application-level modifications.

In the NoSQL context, encryption presents additional challenges due to the distributed nature of storage. Data fragmented (sharded) across multiple clusters requires end-to-end encryption at the field level (field-level encryption), a technique introduced by MongoDB in version 4.2. Popa et al. (2011) propose CryptDB, a relational database system capable of executing SQL queries on encrypted data, opening the prospect of secure query processing in cloud environments without exposing plaintext data to platform operators — a concept equally relevant to modern NoSQL implementations.

4.3 Auditing and Monitoring

Database Activity Monitoring (DAM) constitutes a fundamental instrument for anomaly detection and security incident investigation. Sallam *et al.* (2015) propose an auditing framework based on behavioral analysis of SQL queries, employing machine learning techniques to identify deviations from normal access patterns. Mature RDBMS environments offer native auditing facilities (Oracle Audit Vault, SQL Server Audit), whereas the NoSQL ecosystem relies more frequently on third-party solutions or integration with Security Information and Event Management (SIEM) platforms.

Real-time monitoring of performance and access patterns enables the detection of brute-force attacks, slow data exfiltration attempts, and suspicious activities by privileged users. Wang et al. (2010) demonstrate the efficacy of anomaly detection methods based on time-series analysis of database queries, achieving detection rates exceeding 95% for SQL Injection attacks with fewer than 1% false positives in their experimental settings.

5. NORMATIVE AND COMPLIANCE FRAMEWORK

5.1 GDPR and the Protection of Personal Data

The General Data Protection Regulation (GDPR), in force since May 2018 across the European Union, imposes stringent requirements regarding the security of personal data processing, including within storage systems. Article 32 of the GDPR obliges data controllers to implement appropriate technical and organizational measures, including pseudonymization and encryption of data, the ability to ensure the ongoing confidentiality, integrity, availability, and resilience of processing systems, and the ability to restore data availability in a timely manner in the event of an incident (Regulation (EU) 2016/679, 2016).

Implementing GDPR requirements in NoSQL systems presents specific technical challenges, particularly with regard to the right to erasure (the right to be forgotten, Art. 17 GDPR). The distributed and redundant nature of NoSQL storage makes it difficult to guarantee the complete deletion of all copies of a dataset. Politou et al. (2018) analyze these techno-legal tensions and propose solutions based on Crypto-Shredding — the destruction of cryptographic keys rendering data irrecoverable without its physical deletion — as a complementary compliance mechanism.

5.2 ISO/IEC Standards and the NIST Framework

The ISO/IEC 27001:2022 standard defines the requirements for an Information Security Management System (ISMS), providing a holistic framework for managing security risks, including those associated with databases (ISO/IEC, 2022). ISO 27001 implementation entails adherence to a continuous Plan-Do-Check-Act (PDCA) cycle, encompassing risk assessment, risk treatment via Annex A security controls, and periodic auditing of the effectiveness of implemented measures.

The NIST Cybersecurity Framework (CSF), initially published in 2014 and revised in 2024, structures cyber risk management around six principal functions: Govern, Identify, Protect, Detect, Respond, and Recover (NIST, 2024). Applied at the database level, this framework guides organisations in data asset inventory (Identify), implementation of security controls (Protect), active monitoring (Detect), incident response (Respond), and business continuity assurance (Recover).

6. COMPARATIVE ANALYSIS AND DISCUSSION

A systematic comparison of the two database paradigms from a risk management perspective reveals both notable advantages and significant limitations for each. Relational systems benefit from security maturity: ACID mechanisms guarantee transactional integrity, native granular access control facilities enable the implementation of PoLP, and a rich ecosystem of auditing and monitoring tools facilitates regulatory compliance. Disadvantages include limited horizontal scalability and higher licensing costs for commercial enterprise variants.

NoSQL systems excel in handling massive volumes of unstructured data and in ensuring high availability through distributed replication. However, from a security perspective, schema flexibility — considered a functional advantage — becomes a liability, as it makes the uniform application of data validation and sanitization policies considerably more difficult. Leavitt's (2010) study offers an early warning regarding the risk of sacrificing security in favor of scalability, a finding subsequently confirmed by numerous documented security incidents.

From a risk management perspective, the fundamental difference lies in the maturity of security ecosystems. Grolinger *et al.* (2013) demonstrate, through analysis of six representative NoSQL systems (MongoDB, CouchDB, BigTable, HBase, Cassandra, Redis), that none of them satisfied all the security criteria required for sensitive data processing at the time of the study — a situation that

has partially improved in subsequent versions, but which maintains NoSQL in a position of higher risk relative to mature relational solutions.

A modern approach recommended in the literature is the "polyglot persistence" architecture — the simultaneous use of multiple database types, each for the use case for which it is best optimized (Sadalage & Fowler, 2012). From a risk management perspective, this approach amplifies the complexity of security governance but also enables the optimization of protection mechanisms specific to each system, maximizing the efficiency of risk control.

7. RECOMMENDATIONS FOR INTEGRATED DATA RISK MANAGEMENT

On the basis of the literature review and the applicable normative framework, the present study formulates a set of recommendations for the effective management of digital risks in database systems, irrespective of the paradigm employed.

The first recommendation concerns the implementation of a comprehensive and up-to-date data asset inventory, which is a fundamental prerequisite for any risk assessment. Identifying critical data, classifying it according to sensitivity, and documenting data flows are preconditions for the application of appropriate security controls (NIST, 2024).

The second recommendation concerns the adoption of the "security by design" principle in database schema design. This entails integrating security and privacy requirements from the data modelling phase onwards, rather than as an added layer at a later stage. Encrypting sensitive fields at the application level prior to storage in the database offers additional protection against scenarios involving server-level compromise (Popa *et al.*, 2011).

The third recommendation relates to the implementation of a continuous vulnerability assessment and database-specific penetration testing program. Whitman & Mattord (2021) emphasize that periodic risk assessments — at least annually or following significant infrastructure changes — are essential for maintaining an acceptable risk profile. The use of specialized tools such as SQL map for relational databases or NoSQL map for NoSQL systems enables the proactive identification of vulnerabilities before they can be exploited.

The fourth recommendation is addressed specifically to organizations operating NoSQL databases and consists of explicitly configuring all security settings, avoiding reliance on default configurations. The security checklist should include: enabling mandatory authentication, configuring TLS/SSL for all connections, restricting access to administrative interfaces, enabling auditing, and disabling unused features.

The fifth recommendation concerns the development and periodic testing of Disaster Recovery Plans (DRP) and Business Continuity Plans (BCP). Studies by Whitman & Mattord (2021) show that organizations that regularly test these plans significantly reduce recovery time (Recovery Time Objective — RTO) and data loss (Recovery Point Objective — RPO) in the event of actual incidents.

8. CONCLUSIONS

Digital risk management in relational and non-SQL databases represents a complex domain situated at the intersection of information security, data governance, and regulatory compliance. The analysis conducted in the present study demonstrates that, although the two database paradigms exhibit distinct risk profiles, the fundamental principles of risk management — identification, assessment, treatment, and continuous monitoring — remain applicable in both contexts.

Relational databases benefit from a more mature security ecosystem, supported by decades of research and deployment in critical environments. NoSQL systems, while having substantially

narrowed the security gap relative to their relational counterparts, continue to present specific challenges related to eventual consistency, key management in distributed environments, and the granularity of access control.

The continuously evolving normative framework — GDPR, ISO/IEC 27001, NIST CSF — obliges organizations to adopt a proactive and systematic approach to data risk management. Current technological trends, including the mass migration to multi-tenant cloud environments and the proliferation of microservice architectures with heterogeneous databases, will amplify the complexity of security governance and will necessitate the continuous adaptation of risk management strategies.

Future research should focus on the development of integrated risk management frameworks adapted to polyglot persistence architectures, the evaluation of the effectiveness of deep query inspection techniques for NoSQL systems, and the impact of emerging technologies — such as quantum databases and blockchain-based storage systems — on the digital risk landscape.

REFERENCES

Boneh, D. & Franklin, M. (2001) *'Identity-based encryption from the Weil pairing'*, in Kilian, J. (ed.) Advances in Cryptology – CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139. Berlin: Springer, pp. 213–229.

Clarke, J. (2009) *SQL Injection Attacks and Defense*. Burlington, MA: Syngress.

Curtmola, R., Garay, J., Kamara, S. & Ostrovsky, R. (2006) *'Searchable symmetric encryption: Improved definitions and efficient constructions'*, in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06). New York: ACM, pp. 79–88.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R. & Chandramouli, R. (2001) *'Proposed NIST standard for role-based access control*', ACM Transactions on Information and System Security, 4(3), pp. 224–274.

Grolinger, K., Higashino, W. A., Tiwari, A. & Capretz, M. A. M. (2013*) 'Data management in cloud environments: NoSQL and NewSQL data stores'*, Journal of Cloud Computing: Advances, Systems and Applications, 2(1), pp. 1–24.

Haber, M. J. & Rolls, B. (2020) *Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution*. New York: Apress.

Halfond, W. G. J., Viegas, J. & Orso, A. (2006) *'A classification of SQL-injection attacks and countermeasures'*, in Proceedings of the IEEE International Symposium on Secure Software Engineering (ISSSE). Washington: IEEE, pp. 13–15.

IBM Security (2023) Cost of a Data Breach Report 2023. Armonk, NY: IBM Corporation.

ISO/IEC (2022) ISO/IEC 27001:2022: Information Security, Cybersecurity and Privacy Protection – Information Security Management Systems – Requirements. Geneva: International Organization for Standardization.

ISO/IEC (2022) ISO/IEC 27005:2022: Information Security, Cybersecurity and Privacy Protection – Guidance on Managing Information Security Risks. Geneva: International Organization for Standardization.

Leavitt, N. (2010) '*Will NoSQL databases live up to their promise*?', Computer, 43(2), pp. 12–14.

Litchfield, D. (2005) The Database Hacker's Handbook: Defending Database Servers. 1st edn. Indianapolis: Wiley.

Liu, Q., Wang, G. & Wu, J. (2015) *'Time-based proxy re-encryption scheme for secure data sharing in a cloud environment*', Information Sciences, 258(1), pp. 355–370.

Mehmood, A., Natgunanathan, I., Xiang, Y., Hua, G. & Guo, S. (2016) '*Protection of big data privacy'*, IEEE Access, 4, pp. 1821–1834.

Mohan, C. (1992) 'ARIES: *A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging'*, ACM Transactions on Database Systems, 17(1), pp. 94–162.

NIST (2024) The NIST Cybersecurity Framework (CSF) 2.0. Gaithersburg, MD: National Institute of Standards and Technology. Available at: https://doi.org/10.6028/NIST.CSWP.29 (Accessed: 15 January 2025).

Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E. & Abramov, J. (2011) '*Security issues in NoSQL databases'*, in Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). Washington: IEEE, pp. 541–547.

OWASP Foundation (2021) OWASP Top Ten 2021. Available at: https://owasp.org/www-project-top-ten/ (Accessed: 10 February 2025).

Politou, E., Casino, F., Alepis, E. & Patsakis, C. (2018) '*Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions'*, Journal of Cybersecurity, 4(1), pp. 1–20.

Popa, R. A., Redfield, C. M. S., Zeldovich, N. & Balakrishnan, H. (2011) *'CryptDB: Protecting confidentiality with encrypted query processing*', in Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP'11). New York: ACM, pp. 85–100.

Pritchett, D. (2008) '*BASE: An acid alternative'*, ACM Queue, 6(3), pp. 48–55.

Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L 119, pp. 1–88.

Sadalage, P. J. & Fowler, M. (2012) *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence.* Upper Saddle River, NJ: Addison-Wesley.

Sallam, A., Bertino, E., Hussain, S. R., Landers, D., Lefebure, R. M. & Railkar, R. (2015) *'DBSAFE – An anomaly detection system to protect databases from exfiltration attempts'*, IEEE Systems Journal, 9(4), pp. 1168–1180.

Sinha, S. & Mall, R. (2019) '*An efficient approach for preventing injection attacks in web applications'*, International Journal of Information Security, 18(6), pp. 777–791.

Tudorica, B. G. & Bucur, C. (2011) '*A comparison between several NoSQL databases with comments and notes'*, in Proceedings of the RoEduNet International Conference 10th Edition: Networking in Education and Research. Washington: IEEE, pp. 1–5.

Wang, L., Liu, A. & Bhatt, D. (2010) '*Specification-based intrusion detection for database systems'*, in Jonker, W. & Petković, M. (eds.) *Secure Data Management*. Lecture Notes in Computer Science, vol. 6358. Berlin: Springer, pp. 116–133.

Whitman, M. E. & Mattord, H. J. (2021) *Principles of Information Security*. 7th edn. Boston: Cengage Learning.