JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 8, No. 1, June 2025, pp. 35~43
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: 10.36378/jtos.v8i1.4022

Database Consistency Improvement In Job Offering System Using Normalization Method

Fahrir Rijal Fahmi Maulana[1], Febriliyan Samopa[2]
[1]Department of Technology Management, Sepuluh Nopember Institute of Technology, Indonesia
[2]Department of information systems, Sepuluh Nopember Institute of Technology, Indonesia

Article Info

Article history:

Received 01 07, 2025
Revised 04 14, 2025
Accepted 05 12, 2025

Keywords:

Database
Database Improvement
Normalization
Data Modelling

ABSTRACT

The role of information systems in modern life today is very important, especially in a company to make strategic decisions or actions. Information systems must be free from data anomalies. Data anomalies can occur due to poor database structure design. This happens in job offering systems that still have data anomalies. This study uses the normalization method to verify the quality of the database in the job offering system. Normalization is a technique that focuses on grouping several attributes into an entity based on certain criteria to achieve normal form. Normalization aims to maintain data consistency and integrity. The job offering information system has two databases to carry out all its functions. The first database is for the needs of processing job offering information system data. The second database is for the needs of processing company organizational structure data. This study has identified both databases of the job offering system. There are 2 violations of the normalization provisions found. Violations occur in the employee and schedule_wwc tables which do not follow the 3NF normalization provisions. Improvements need to be made to the database to improve data consistency in the job offering system. This study has made improvements by modeling data according to the data needs of the job vacancy information system. The modeling produces a conceptual data model (CDM), a logical data model (LDM) and a physical data model (PDM). The modeling results have been reverified using the normalization method and are free from any violations.

Corresponding Author:

Fahrir Rijal Fahmi Maulana
Department of Technology Management
Sepuluh Nopember Institute of Technology
Surabaya, Indonesia
Email: frijalfm@gmail.com

1. Introduction

Databases play a very important role in modern life today. Databases are needed in information systems to store, manage, and provide information [1]. The database used in the job offering system is a relational database. Relational databases are a common and popular type of database used today [2]. A relational database is a collection of interrelated tables that have data categories and constraints [3]. Relational databases must be well designed, otherwise they will produce abnormal data resulting in redundant data and inconsistent data. [4]. Anomaly is an event in a data set that is unusual and does not conform to the general pattern or expected results [5]. Anomalies can occur due to transactional operations such as additions, changes, and deletions [6].

Normalization is one of the database quality verification methods, this technique is used to ensure that the relational database has met the principles of normal form. Normalization can reduce redundant data and maintain atomicity and consistency criteria in the database [7]. Although the normalization process is very important, many developers tend to ignore the normalization process because it takes time and expertise [8].

The job offering information system is used to accommodate job vacancies in a work unit to internal company employees. So with the job offering information system, the company can find a replacement for vacant workers with other employees. In general, the job offering information system business process starts from: opening of job vacancies by the Human Resources (HR) unit submitted by the work unit, employees who meet the requirements will apply for the job vacancies, the incoming vacancies will be verified, and the final stage is an interview. However, in the operation of the job offering information system, obstacles were found caused by data anomalies, this resulted in the system not running according to procedure and reducing the productivity of HRD unit staff. Anomalies occur in organizational structure information, resulting in a mismatch between the position and the position and work unit of an employee.

Based on the problems that have been explained, there is a great opportunity to conduct research that aims to improve the database structure in the job offering system. The changes made are expected to improve consistency and overcome the problems that have been explained.

2. Research Method

2.1. Normalization

Database normalization is a technique used to manage relational databases in a structured and good manner, so that it can minimize data redundancy and prevent data anomalies [9]. Normalization is suitable for applications that perform intensive data manipulation, such as creating, changing, and retrieving data [10]. According to [8], there are 6 stages of normalization, namely 1NF, 2NF, 3NF, BCNF, 4NF and 5NF.

The first step of normalization, 1NF normalization only allows attribute domains to consist of one value, so 1NF does not allow multi-valued attributes, composite attributes, combination attributes, and relations within relations [11]. 2NF normalization must meet the requirements of 1NF and non-key attributes must be fully functionally dependent on all key attributes (candidate key) [1]. 3NF normalization must meet the requirements of 2NF and no non-prime attribute of R is transitively dependent on the primary key, functional dependency $X \rightarrow Y$ is a transitive dependency, if there is a set of attributes Z in R that is not a candidate key or part of any key in R, and both $X \rightarrow Z$ and $Z \rightarrow Y$ [11]. In other words, it meets 3NF normalization if no non-prime column determines another non-prime column [12]. Boyce-Codd Normal Form (BCNF) normalization, if every determinant in the relation is a candidate key [13]. 4NF normalization must meet the BCNF requirements and there is no multivalued dependency, multivalued dependency occurs if there are at least three attributes in the relation and each value of A has a set of predetermined B values and there is also a predetermined C value [13]. 5NF normalization or commonly called projection-join normal form (PJNF) if a relation does not have nontrivial join dependency, JD(R1, R2, …, Rn) with each Ri being a superkey of R [11].

2.2. Data Modelling

Data modeling is the process of mapping data requirements into a data model to ensure data quality and integrity [14]. There are 3 types of data modeling models, namely conceptual data models (CDM), logical data models (LDM), and physical data models (PDM) [15]. Each model has its own level of abstraction [16]. Data modeling requires several stages, including: exploration of needs, conceptual model design, selection of database management system (DBMS), logical model design, physical model design, and database system implementation [11].

2.2.1. Conceptual Data Model (CDM)

CDM represents the entire data structure required for business needs independently of any software system and data storage structure [17]. CDM starts from mining and analyzing needs and then produces a high-level data model that describes data needs consisting of entities, attributes, and relationships [11]. CDM is not related to the technical side (specifically for DBMS), so it can be used as a reference by non-technical users [18].

2.2.2. Logical Data Model (LDM)

LDM is the result of mapping CDM into the form of a data model implementation of a database (relational database) [18]. LDM is a further mapping into a database form. LDM contains a normalized model, in addition to containing the rules needed in the application process [19]. LDM helps bridge communication between software developers, business analysts, and database administrators [20].

2.2.3. Physical Data Model (PDM)

PDM is the result of mapping from LDM to an internal schema that produces a low-level data model. PDM describes a detailed concept of how data is stored in computer storage by DBMS [11]. In a relational database, PDM represents the format of tables, columns, and relationships between tables. In addition, some DBMS have options to improve performance quality [21].

3. Methodology

There are 7 stages in this research, including: identifying anomalies in the database structure used, conducting mining and needs analysis, designing a conceptual data model, designing a logical data model, designing a physical data model, conducting a gap analysis model, and conducting a database migration strategy. The stages of this research can be seen in Figure 1 below.

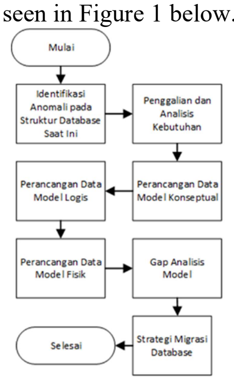

Figure 1 Research Methodology

3.1. Anomaly Identification

This research begins by identifying the database structure used. The identification process is conducted using the Physical Data Model (PDM), which is generated through the PostgreSQL DBMS tool, pgAdmin. The PDM provides a detailed blueprint of the existing database schema, including tables, relationships, keys, and constraints. To ensure that the database structure is optimized and adheres to best practices in data modeling, this model is then analyzed using the normalization method. Normalization is a critical process in database design aimed at reducing redundancy and ensuring data integrity by organizing the data into well-structured tables that comply with normal forms, from the first normal form (1NF) through to higher levels where applicable. This stage is crucial because it allows researchers to pinpoint structural weaknesses or inefficiencies in the current design, such as repeating groups, partial dependencies, or transitive dependencies that could lead to data anomalies. By identifying and addressing only those tables or relationships that do not meet normal form requirements, unnecessary modifications to already well-structured and optimized components are avoided. This targeted approach helps maintain the integrity of the database while enhancing performance, scalability, and reliability. Ultimately, a well-normalized database structure forms a solid foundation for accurate data analysis, efficient query processing, and effective implementation of data mining algorithms later in the study.

3.2. Elicitation and Analysis Requirements

This process is carried out by thoroughly exploring the information available within the local clone position offering system. The exploration aims to gain a comprehensive understanding of how the system operates and what data it manages. To support this exploration, several supporting documents are utilized. These include system usage guidelines, which offer insights into user interactions and functionalities, as well as technical development documents, which provide in-depth details on system architecture, data structures, and backend processes.

By referring to these resources, the study is able to map out the core components of the system, including its input and output mechanisms, user roles, and data flow. This stage is crucial in identifying the specific data requirements necessary for the system to function effectively. The collected information helps to determine what data must be captured, how it should be structured, and how it integrates with other system modules. In turn, this allows for the alignment of data preparation and modeling processes with the actual needs and limitations of the system, ensuring relevance and consistency throughout the research. Ultimately, this step lays a solid foundation for the subsequent phases of analysis, design improvement, and potential system optimization.

3.3. Conceptual Data Model Design

The process of designing a Conceptual Data Model (CDM) involves converting the outcomes of data requirements analysis into a structured format such as an Entity Relationship (ER) or Enhanced Entity Relationship (EER) model. As explained in prior studies [22], this modeling process is conducted through a series of systematic steps. It begins with the identification of all relevant entities, along with their associated attributes and interrelationships. Following this, the process involves determining key attributes that serve as unique identifiers for each entity. The model then defines the cardinality of relationships—clarifying how many instances of one entity relate to instances of another—and specifies any participation constraints. Additionally, the modeling phase includes identifying weak entity sets, if any, which rely on other entities for their identification. It may also extend to exploring more advanced structural features, such as specialization and generalization, as well as hierarchical or lattice-like relationships between entities. These steps ensure that the conceptual model accurately captures both the static structure and inherent logic of the domain being represented.

3.4. Logical Data Model (LDM) Design

LDM design transforms the CDM results into a relational database. This transformation process is formed in several stages. According to [22], if the CDM result is ER then there are 7 stages, including: regular entity type mapping, weak entity type mapping, 1:1 binary relationship type mapping, 1:N binary relationship type mapping, M:N relationship type mapping, multivalued attribute mapping, and N-ary relationship type mapping, and if the CDM result is EER then there are 2 additional stages, namely: specialization and generalization mapping and shared subclass mapping (multiple inheritance).

3.5. Physical Data Model (PDM) Design

PDM is implemented using PostgreSQL DBMS tool, PGAdmin. In PDM, tables, columns, primary keys, and foreign keys need to be defined. In addition, for performance purposes, indexes also need to be defined.

3.6. Gap Analysis Model

At this stage, the database structure before and after the repair has been obtained, then a comparative analysis or gap analysis will be carried out. This stage will carry out a performance evaluation using load testing with the JMeter tool.

3.7. Database Migration Strategy

In the final stage of this research, it is necessary to determine the database migration strategy. Data migration is the movement of data from one environment to another, which can be interpreted as a new database or another database [23]. Data migration must be done properly, otherwise it will result in system failure, data loss, and prolonged system downtime [24]. There are several migration strategies, namely the Big Bang strategy, Gradual Strategy, Parallel Strategy, and Always Up Strategy [25].

4. Result and Discussion

4.1. Anomaly Identification

There are provisions within the current database design that do not meet normalization requirements, specifically concerning the third normal form (3NF). Upon detailed analysis, it was found that the functional dependencies in the m_jabatan table fail to satisfy the 3NF criteria. This is due to the presence of non-prime attributes—attributes that are not part of any candidate key—that are transitively dependent on the table's primary key. Specifically, attributes such as work unit, position, and grade are not directly dependent on the primary key but instead rely on the postcode attribute, which itself is dependent on the primary key. This indirect dependency structure violates 3NF, which requires that every non-prime attribute must depend only on the primary key and not on other non-prime attributes.

In addition to this, the jadwal_wwc table was also found to be in violation of 3NF due to similar issues of transitive dependency. In this table, attributes such as id_jadwal, nik_pejian, nik_unit_kerja, and jadwal_kar are all transitively dependent on the id_jadwal_wwc primary key through the id_kuota attribute, which itself is a non-prime attribute. This dependency chain suggests that the structural integrity of the table could lead to redundancy and anomalies during data updates, deletions, or insertions.

To address these normalization issues, the database structure must be refined by decomposing the affected tables into new relations that eliminate transitive dependencies while preserving data integrity and ensuring lossless joins. This step is essential not only for achieving compliance with normalization standards but also for optimizing the performance of data operations and ensuring the reliability of downstream processes such as data mining, reporting, and analytics..

4.2. Elicitation and Analysis Requirements

The elicitation and analysis requirements resulted in attributes and entities needed to run the job offering business process. There are 10 entities obtained, namely employee entities, positions, work units, positions, job offers, interview schedules, interview sessions, applications, verifications, and announcements. Related entities have their own attributes.

4.3. Conceptual Data Model Design

The data needs that have been obtained will be transformed into EER form. After being transformed, there are a total of 14 entities, namely announcements, employees, positions, positions, work units, interview schedules, interview sessions, job offers, applications, verification, non-management, management, HR staff, and head of work units. There are 5 weak entities, namely announcements, verification, applications, interview schedules, and interview sessions. There are 2 subclasses, namely employees into management and non-management, and management into HR staff and head of work units. The results of the CDM transformation can be seen in Figure 2.

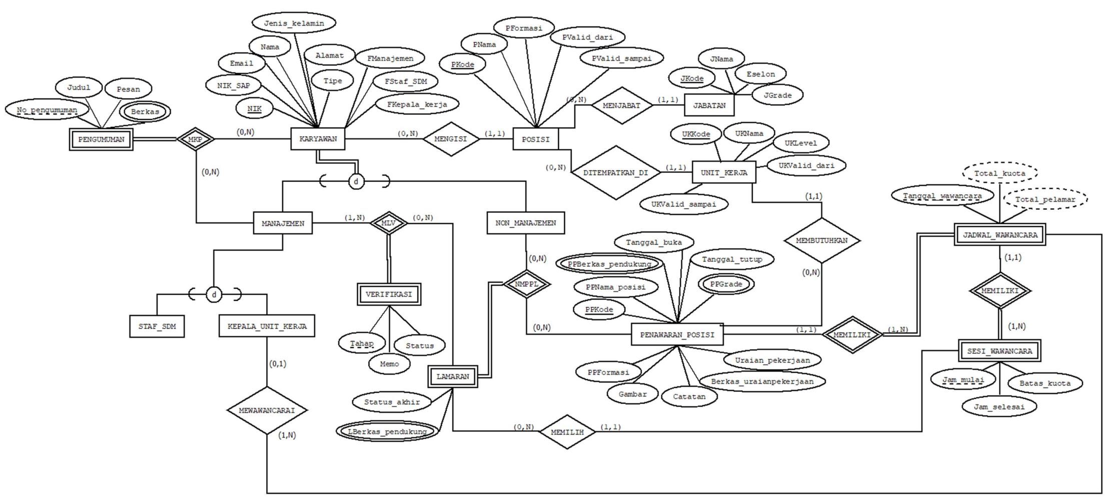

Figure 2 The Result of Conceptual Data Model Mapping

4.4. Logical Data Model Design

After constructing the conceptual data model, the next step involves translating it into a relational schema suitable for implementation within a database management system. This logical transformation results in a set of normalized tables, each representing entities and their relationships in the system. In total, the process yields 17 relational entities, which include core components such as application_schedule, interview_session, interview_schedule, application_file, application, verification, and employee.

Additionally, supporting entities such as nouncing, announce_file, position_file_offer, position_work_unit_offer, position, position_grade_offer, and sub_employee are also derived from the conceptual design. These relations are structured to maintain data integrity and support efficient query operations, reflecting the data requirements and interactions identified in the earlier design stages. The resulting Logical Data Model (LDM) is visually represented in Figure 3, illustrating how each entity connects and how the overall system structure supports the application's functionality. This transformation step ensures that the design is ready for physical implementation while preserving the consistency and completeness of the original data model.

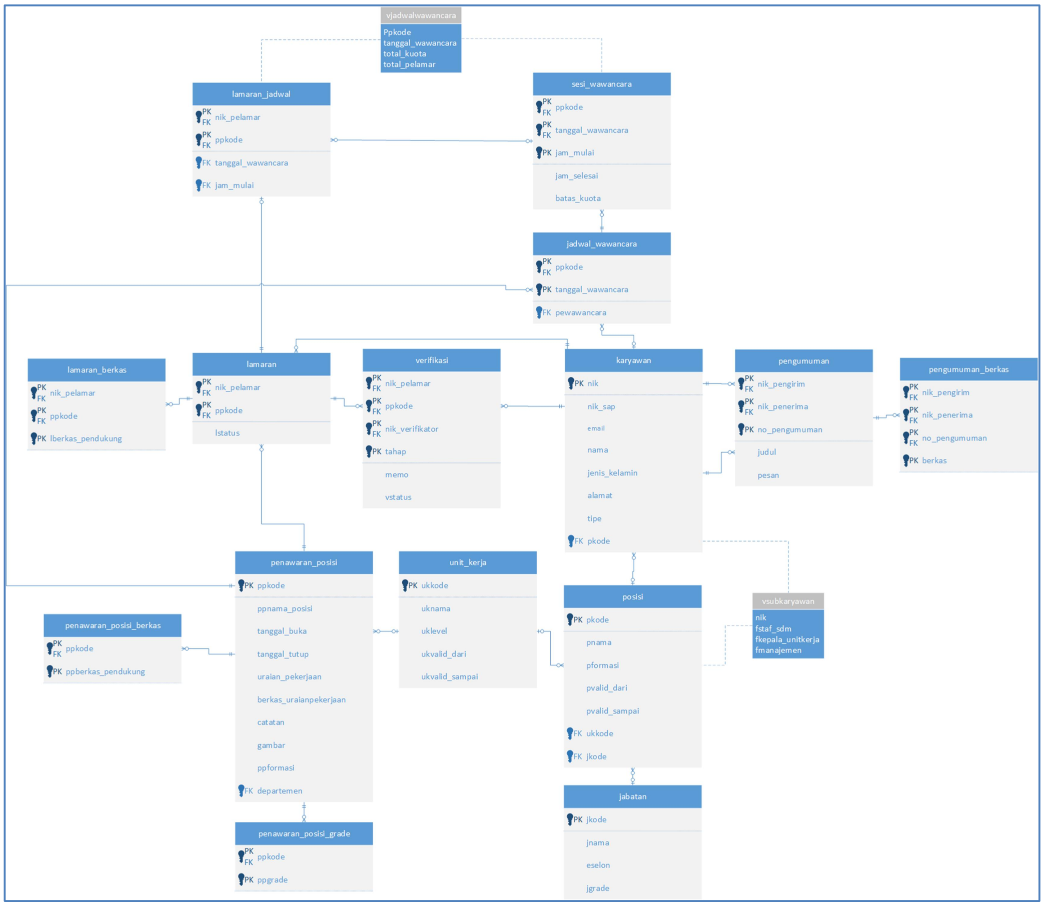

Figure 3. The Result of Logical Data Model Mapping

4.5. Physical Data Model Design

The Physical Data Model (PDM) serves as the final stage in the database design process, where the logical structures defined in the Logical Data Model (LDM) are translated into actual database implementations. This implementation is carried out using the PostgreSQL Database Management System (DBMS), specifically through the pgAdmin interface, which provides a graphical environment for managing database objects. Each relation identified in the LDM is realized as a table in the PDM, with careful attention given to defining the appropriate data types, constraints, and keys for each column to ensure data integrity and enforce business rules. This includes specifying primary and foreign keys, not-null constraints, unique constraints, and any necessary default values. The goal of this stage is to produce a fully operational schema that is optimized for performance, scalability, and consistency within the PostgreSQL environment. Figure 4 presents a visual representation of the PDM implementation, highlighting how the abstract components of the previous stages are concretely realized in a structured and functional database system. This step is crucial for bridging the gap between conceptual design and practical deployment, ensuring that the system is not only theoretically sound but also technically executable.

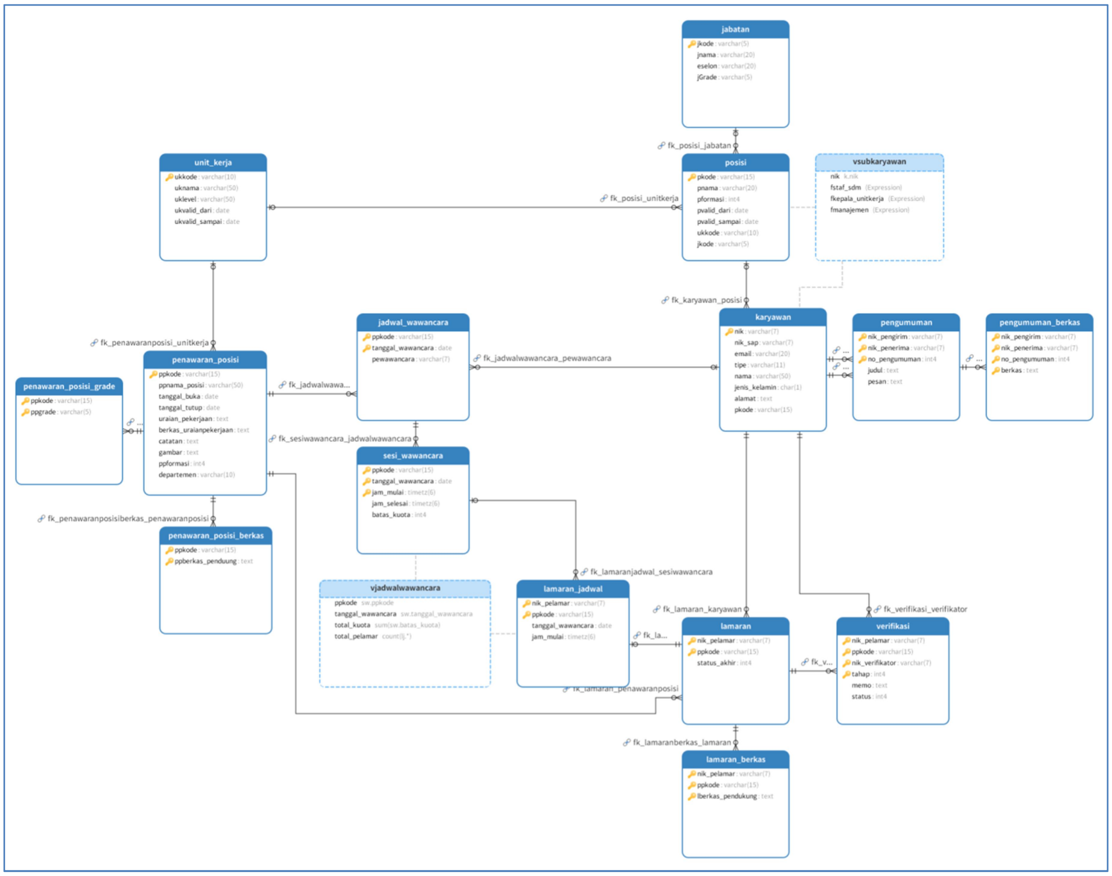

Figure 4 Result of Physical Data Model Mapping

4.6. Gap Analysis Model

The analysis is done by exporting an excel file on the verification feature that will display all job vacancies and their applicants. This will test the performance of data retrieval from the database before the repair is carried out. Performance testing is carried out as a whole and will process a total of 50,000 rows. The average results before and after the database repair are 101482.2 ms and 106070.4 ms respectively. From both results it can be concluded that database repair causes performance to be slower than before the repair, this is because the normalization method divides the table into several tables based on normalization provisions, so that if data is loaded or retrieved, the related tables must be merged back. These results are still tolerable because there are no errors in the test results.

4.7. Migration Strategy

Database migration is carried out using the Big Bang method. The Big Bang method was chosen because it is in accordance with the characteristics of the job offering system which allows for downtime, low database complexity, and a complete database overhaul. Big Bang migration must go through 4 stages, including initialization, development, testing, and cutoff [26]. In the research conducted, the migration strategy was compiled with 5 stages, namely initialization, development, testing, execution and maintenance.

5. Conclusion

A consistent relational database structure must be free from data anomalies. This research has identified the job vacancy information system database using the normalization method. The identification results found 2 violations of normalization provisions. Violations occurred in the employee and schedule_wwc tables which did not comply with the 3NF normalization provisions. This study has made improvements by modeling data according to the data needs of the job vacancy information system. The modeling produces a conceptual data model (CDM), logical data model (LDM) and physical data model (PDM). The modeling results have been verified using the normalization method and are free from any violations.

Acknowledgement

We extend our deepest gratitude to all contributors who have supported this research on improving database consistency in a job offering system using the normalization method. We appreciate the invaluable insights from our colleagues and supervisor for the constructive feedback, which greatly enhanced the quality of this study.

References

[1] Ying Bai and Satish Bhalla, “Introduction to Databases,” 2021. doi: https://doi.org/10.1002/9781119608493.ch2.

[2] Konrad Fraczek and Malgorzata Plechawska-Wojcik, *Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications*, vol. 716. in Communications in Computer and Information Science, vol. 716. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-58274-0.

[3] N. Gupta and R. Agrawal, “NoSQL Security,” in *Advances in Computers*, vol. 109, Academic Press Inc., 2018, pp. 101–132. doi: 10.1016/bs.adcom.2018.01.003.

[4] M. Albarak and R. Bahsoon, “Prioritizing Technical Debt in Database Normalization Using Portfolio Theory and Data Quality Metrics,” 2018.

[5] R. Foorthuis, “On the nature and types of anomalies: a review of deviations in data,” Oct. 01, 2021, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1007/s41060-021-00265-1.

[6] S. Saha Bagui, “Database Design Using Entity-Relationship Diagrams (3rd Edition),” 2023.

[7] Kunal Kumar and S. K. Azad, *Database Normalization Design Pattern*. 2017.

[8] M. Albarak, R. Bahsoon, I. Ozkaya, and R. Nord, “Managing Technical Debt in Database Normalization,” *IEEE Transactions on Software Engineering*, vol. 48, no. 3, pp. 755–772, Mar. 2022, doi: 10.1109/TSE.2020.3001339.

[9] N. Mendjoge, A. R. J. Meera, N. D. J. Sanghvi, and D. J. Sanghvi, “Intelligent Tutoring System for Database Normalization,” 2016.

[10] Kamal Hingorani, Dexter Gittens, and Nicholas Edwards, “REINFORCING DATABASE CONCEPTS BY USING ENTITY RELATIONSHIPS DIAGRAMS (ERD) AND NORMALIZATION TOGETHER FOR DESIGNING ROBUST DATABASES,” *Issues In Information Systems*, vol. 18, no. 1, pp. 148–155, 2017, doi: 10.48009/1_iis_2017_148-155.

[11] R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems 7th,” 2015.

[12] L. Davidson, *Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance*. Springer, 2020. doi: 10.1007/978-1-4842-6497-3.

[13] Jeffrey A. Hoffer, V. Ramesh, and Heikki Topi, “Modern Database Management 12th Edition,” 2015.

[14] Laura Sebastian and Coleman, “Meeting the Challenges of Data Quality Management,” 2022.

[15] R. T. Yarlagadda, “DATA MODELS IN INFORMATION TECHNOLOGY,” 2016.

[16] S. Al-Fedaghi, “Conceptual Data Modeling: Entity-Relationship Models as Thinging Machines,” Sep. 2021, doi: 10.22937/IJCSNS.2021.21.9.33.

[17] R. Sherman, *Business intelligence guidebook : from data integration to analytics*. 2014.

[18] Rahmawati Titik, Yulia Sari Eka, Tanjung Shakti Anjasmara, and Nanda Yomura Atthaya, “ANALISIS PERANCANGAN DATABASE MANAGEMEN SISTEM UNTUK SISTEM PENUNJANG PROSES BISNIS WEDANG UWUH INSTAN (ANALYSIS OF SYSTEM MANAGEMENT DATABASE DESIGN FOR INSTANT WEDANG UWUH BUSINESS PROCESS SUPPORT SYSTEMS),” 2023.

[19] Matthew West, “Developing High Quality Data Model,” 2011.

[20] J. V Luisi, “Pragmatic Enterprise Architecture,” 2014.

[21] S. Dwivedi, L. Chourasiya, A. Professor, and M. Bhopal, “Data Modeling: A Perspective In Changing Database Scenario,” 2022. [Online]. Available: http://www.webology.org

[22] R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems 6th,” 2010.

[23] A. Fahmi and Y. H. Putra, “Database Migration Strategies and Techniques to Minimize Unexpected Dysfunctionality,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1757-899X/662/6/062003.
[24] A. Abdou Hussein, “Data Migration Need, Strategy, Challenges, Methodology, Categories, Risks, Uses with Cloud Computing, and Improvements in Its Using with Cloud Using Suggested Proposed Model (DMig 1),” *Journal of Information Security*, vol. 12, no. 01, pp. 79–103, 2021, doi: 10.4236/jis.2021.121004.
[25] Johny Morris, “Practical Data Migration,” 2020.
[26] N. W. Trisnawaty, A. N. Hidayanto, and Y. Ruldeviyani, “Database and Application Migration in the Financial Services Industry Sector in the Acquisition Environment and Environmental Science,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1755-1315/700/1/012022.