

***University of Ibadan Journal of
Science and Logics in ICT
Research (UIJSLICTR)***
ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 17 No. 1, June 2026

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com



A Database Model for an Optimised Relevance Ranking System

¹✉ Adebayo, E. O., ²Oguntunde, T., and ³Amoo, J. O.

¹Federal College of Education (Sp.), Oyo; ²University of Ibadan; ³Federal College of Education (Sp.), Oyo
Emails of authors: adebayoenitan2018@gmail.com, tantos557@yahoo.com, amoo.jelili1247@fcesoyo.edu.ng

Abstract

The need for efficient and intelligent search models has never been more pressing than it is in this era of digital information explosion, as timely data availability is a long-standing challenge in policy-making and analysis for low-income and developing countries. Conventional relevance ranking techniques often struggle with scalability, limited adaptation to user intent, and heterogeneous data sources. An adequately modelled database spurs an efficient relevance ranking algorithm for efficient document retrieval. An inappropriately modelled database creates errors related to broken links, which hinder users' queries from retrieving appropriate documents in the cloud. Hence, this paper is aimed at developing a model for an Optimised Relevance Ranking system for efficient document retrieval. The defended PhD theses sampled from the University of Ibadan repository between 1999 and 2021 were used for the database model development. A careful selection of database entities was done from a document retrieval scenario based on four entity selection rules. An Entity-Relationship (E-R) model was developed with respect to THESES, AUTHOR, CITATION, INSTITUTION, ASBT-KEYWORD (Abstract Keyword), THESES-KEYWORD, USER, SEARCH-RESULT, SEARCH-QUERY, RANKING-FEATURE, and FEEDBACK entities. The E-R model was migrated to a Relational Model (R-M), which is the third normal form of the database, using five conversion rules. The E-R and the R-M models were developed using MySQL Workbench. This approach aims to improve retrieval accuracy and guarantees effective query processing in contemporary information systems by focusing on structured document storage, modularity, scalability, and performance.

Keywords: BM25, Database, Five conversion rules, Optimised relevance ranking system, Four entities selection rules.

1. Introduction

An organised collection of structured, semi-structured, or unstructured data intended for effective storage, retrieval, updating, and analysis is referred to as a database [19, 21]. Databases can be customised to accommodate many workloads, including graph-based operations, time-series data, analytics, and transactional processing. A Database Management System (DBMS) is the software used to construct, manage, and maintain databases while guaranteeing data integrity and controlled access [2, 21]. Data storage and indexing, query processing and optimisation, transaction management, concurrency control, recovery mechanisms, and security enforcement are crucial tasks performed by database management systems [7, 21]. ACID (Atomicity,

Consistency, Isolation, and Durability) qualities, which guarantee dependability, consistency, and fault tolerance in database operations, frequently regulate transaction management [5,9]. These essential features are present in all DBMS architectures, despite differences in database types and infrastructure.

A database system comprises the software and applications that access it, as well as an organised collection of structured (or semi-structured) data that is electronically stored for easy access, management, and updating [6]. The system software that builds, maintains, safeguards, and grants restricted access to databases is known as a DBMS. It offers languages for defining and manipulating data, upholds security and integrity standards, facilitates concurrency and recovery, and offers administration tools for monitoring, tuning, and backup. According to Target [18], the DBMS is the layer that enables users and various

Adebayo, E. O., Oguntunde, T. and Amoo, J. O. (2026). A Database Model for an Optimised Relevance Ranking System. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 17 No. 1, pp. 202 – 212.

applications to properly execute CRUD (create, read, update, delete) activities on shared data.

Data model and schema, such as relational (tables/rows/columns), document, key-value, columnar, and graph; defines structure and query semantics [4]; query language, such as SQL for relational systems; specialized APIs or query languages for NoSQL/graph systems [4]; transactions & ACID, such as DBMSs support transactional semantics (Atomicity, Consistency, Isolation, and durability) or relax some of these properties for scalability in certain designs [4]; concurrency control and recovery, such as locking, MVCC, logging, and checkpointing to keep data correct under concurrent access and failures [4]; and storage and indexing, such as on-disk/remote storage formats, in-memory techniques, and indexes (B-trees, LSM trees, columnar encodings) that drive performance [4].

Relational DBMS (RDBMS) include ACID, SQL, and mature tooling (e.g., PostgreSQL, Oracle, SQL Server) (Oracle, 2020); NoSQL systems, such as document, key-value, wide-column, and graph databases, which sacrifice some relational features for scale and flexibility; and NewSQL or converged systems, which seek to combine relational semantics (or SQL compatibility) with distributed scalability and high throughput. Diverse models and architectures are trending toward convergence, according to recent research and system development [17].

A database system is an organised setting that uses a database management system (DBMS) to effectively store, manage, and retrieve data. It is the foundation of contemporary information systems, allowing for data protection, consistency, and accessibility in a wide range of applications, including government, business analytics, healthcare, education, and finance [4, 6]. Database systems have developed from straightforward record-keeping mechanisms to sophisticated, distributed, and cloud-based infrastructures in tandem with the growth of data-driven decision-making [17]. Database systems are primarily designed to offer a dependable and methodical way to handle massive amounts of data.

In today's digital world, database systems are essential for organising, protecting, and utilising

data. They guarantee the accuracy, accessibility, and actionability of data, which is one of an organisation's most significant assets. They provide more than just storage; they facilitate communication, strengthen analytics, and foster strategic expansion. Database systems continue to adapt through hybrid architectures, cloud integration, and AI optimisation as technology demands change.

However, relevance ranking is crucial to information retrieval systems' ability to provide users with useful results. Conventional ranking techniques struggle to handle dynamic datasets and contextual queries as data volume and diversity rise. An optimised relevance ranking system is built around a well-designed database model that facilitates effective data management, rapid retrieval, and seamless integration with the ranking algorithm.

This study focuses on the development of a database model specifically for an optimised relevance-ranking system, with an emphasis on data structure, indexing techniques, and interaction with ranking processes.

2. Related Works

Oguntunde [13] modelled a secure multifactor authentication database for e-procurement in a production environment. The paper argued that traditional procurement is manual and prone to manipulation, forgery, missing documents, tattered documentation, or outright fraud. A case study was built around the procurement data gathered. The paper selected entities from the case study using the four tests for entity validity. Entity-Relationship (E-R) and Relational model (R-M) were developed, and problematic relationships were resolved. Security for the e-procurement database was achieved by a many-to-many (M:1) relationship of the AUTHENTICATION table.

Paiva *et al.* [14] examined DBMS usage trends in 362 open-source software projects on GitHub. While MySQL and PostgreSQL remain the most extensively used relational database systems, the authors discovered that MongoDB and Redis are the most popular NoSQL substitutes. According to the poll, polyglot persistence architectures in which different DBMS systems coexist inside the same application environment to meet different data management needs are also

growing in popularity. Dritsas and Trigka [3] looked into the integration of machine learning methods with modern DBMS design and investigated the use of machine learning for anomaly detection, query optimisation, indexing, and self-tuning databases. The authors found and reported that AI-driven database systems significantly improve query planning accuracy, workload adaptation, and overall system efficiency. The study's conclusions indicate that machine learning integration is a groundbreaking development in the architecture of next-generation database management systems.

Zambrano et al. [22] developed a high-performance in-memory database architecture for intelligent video monitoring in healthcare systems. The researchers compared the suggested architecture with MySQL, PostgreSQL, MongoDB, Redis, and Microsoft SQL Server. The results showed that for real-time data processing applications, the in-memory design greatly reduced latency and improved read and write performance. The study highlighted the growing importance of specialised database architectures for high-speed and mission-critical systems.

Napoli et al. [11] looked at NoSQL database models for knowledge graph processing. The authors compared document-oriented, graph-based, and multi-model DBMS architectures using graph-centric workloads. The findings showed that graph-oriented designs outperform conventional relational systems for handling highly linked and semantically rich datasets. The study's recommendations state that organisations should align DBMS architectural choices with data complexity and query requirements. The long-term effects of schema redesign and migration in evolving database environments were evaluated in the SkyServer NoSQL refactoring study (2026). According to the study, optimised NoSQL schemas may initially improve performance but may gradually lose their effectiveness when workloads shift. The researchers concluded that flexible and adaptable database structures are necessary to ensure long-term performance in changing circumstances.

Yan et al. [20] looked at DBMS and Large Language Model (LLM) integration strategies in industrial applications. Their empirical research

revealed a number of architectural approaches that facilitate automated querying, intelligent analytics, and AI-assisted decision-making. The study focused on the growing integration of AI infrastructures with DBMS technologies in modern business systems. Empirical data generally indicates that current DBMS architectures are rapidly evolving toward intelligent, distributed, scalable, and cloud-native systems. These architectures are rapidly integrating distributed processing methods, hybrid storage models, and AI-driven optimisation to meet the needs of contemporary data-intensive applications.

Deep learning applications in database query execution and schema optimisation were examined by Milicevic and Babovic [8]. According to their empirical review, machine learning techniques greatly enhance query optimisation, indexing strategies, and schema selection in contemporary database systems. The researchers found that in dynamic computing environments, AI-assisted modelling strategies boost adaptive database performance and decrease the complexity of manual schema design. A thorough, comprehensive literature study of self-tuning database systems and autonomous schema creation was carried out by Mozaffari et al. [10]. In both SQL and NoSQL databases, their study assessed automated schema optimisation methods. The results showed that by dynamically modifying conceptual, logical, and physical schemas in accordance with workload patterns, self-tuning database systems enhance overall performance. However, the researchers found important issues with adaptive indexing in distributed systems, optimisation cost, and schema evolution.

AI-assisted data modelling and design frameworks were examined in a more recent empirical investigation by Stanek and Novotny [15]. The researchers examined 28 studies on AI-supported database modelling between 2023 and 2025 using a PRISMA-guided review methodology. According to their research, artificial intelligence greatly improves metadata management, schema development, validation, and optimisation. The study also found that in AI-assisted database design environments, human-in-the-loop (HITL) techniques are crucial for enhancing model quality, governance, and semantic accuracy.

The function of database design in flood early warning systems was examined empirically by Chinguwo and Deus [1]. The impact of database modelling on the efficiency and usability of real-time information systems was examined in their study. The findings showed that in disaster management applications, well-structured database models greatly enhance data integration, retrieval effectiveness, and system responsiveness. To ensure data consistency, reliability, and scalability in critical information systems, the researchers concluded that efficient database design is crucial.

An extensive empirical review of bitemporal database modelling methods was carried out by Neelamegam et al. [12]. Their research looked at how contemporary database systems incorporate two temporal dimensions: transaction time and valid time. The results showed that bitemporal database architectures enhance traceability, temporal consistency, and historical data tracking in enterprise applications. However, the study found shortcomings in cloud-native optimisation, AI-assisted temporal modelling, and industrial-scale implementation. Data interoperability and distributed data modelling have also been highlighted in recent research conducted in 2026. Modern database models increasingly prioritise interoperability, data sovereignty, and secure data sharing across distributed platforms, according to research on data spaces architecture. The results of these investigations demonstrated that cloud-native and cross-organisational data environments are increasingly dependent on flexible and modular modelling frameworks.

Database usage trends among 362 open-source software projects were empirically analysed by Paiva et al. [14]. According to their research, MongoDB and Redis are extensively used for semi-structured and real-time data applications, while relational databases like MySQL and PostgreSQL continue to be the most popular for structured data storage. The survey also found that polyglot persistence architectures, where various storage technologies coexist to accommodate a range of application requirements, are becoming more and more popular.

According to Dritsas and Trigka [3], retrieval accuracy and relevance ranking improve significantly when conventional retrieval

techniques are combined with machine-learning-based ranking algorithms. By dynamically adjusting to user behaviour and query patterns, their empirical evaluation showed that AI-assisted retrieval systems outperform traditional retrieval models in complex search contexts.

Furthermore, Milicevic and Babovic [8] examined the use of deep learning techniques in query execution and optimisation. According to their review, machine learning techniques enhance indexing strategies, workload forecasting, and query plan selection. The study showed that, especially in dynamic database systems, AI-assisted query processing frameworks outperform conventional rule-based optimisation techniques in terms of execution efficiency. In 2025, Stewart and colleagues [16] conducted an empirical evaluation of query optimisation techniques in large-scale distributed databases. Their results demonstrated that, as compared to traditional approaches, AI-enhanced optimisation strategies decreased query execution latency by about 45%. The report emphasised how artificial intelligence is increasingly being used in query processing frameworks to enhance responsiveness and resource efficiency. The analysed empirical investigations show that between 2020 and 2026, there was a significant evolution in data storage, retrieval, and query processing technologies. With their increased scalability and fault tolerance, distributed and cloud-native architectures are gradually replacing or supplementing traditional storage systems. Similarly, retrieval techniques have advanced from straightforward keyword-based matching to machine learning-powered intelligent and context-aware retrieval frameworks. The results consistently show that BM25 and hybrid retrieval models perform better in relevance ranking and retrieval accuracy than conventional TF-IDF methods. Additionally, AI-driven retrieval systems have become strong substitutes that can adjust to shifting user requirements and data properties.

Osofisan A. O. and Oguntunde T. [23] asserted that an object is an abstraction of a real -world entity and in general terms, an object may be considered equivalent to an Entity Relation (E-R)'s model. Database designs start with an abstract view of the overall data environment and add details as the design comes closer to

implementation. It was said that the basic building blocks of all data models are entities, attributes, relationships and constraints. Tables are related to each other through the sharing of a common attribute. There are three levels of abstractions: external, conceptual and internal. Entity Relation Diagram (ERD) is used to represent the external view; conceptual model integrates all external views together, while the internal model maps the conceptual model to the Database Management Systems (DBMS). The Relational Database model has three well defined components: a logical data structure, a set of integrity rules and a set of operation that determines how data are manipulated.

3. Methodology

3.1 Research Design

This study uses an experimental, quantitative research design to develop a database model for an optimised relevance ranking system.

3.2 Dataset

The University of Ibadan Postgraduate College provided the dataset, comprising one thousand and sixty (1060) defended PhD theses from various departments between 1999 and 2021. The thesis metadata comprises research titles, abstracts, authors, keywords, year, and institution.

3.3 The Entity Relationship Diagram (ERD) for the Relevance Ranking Database System

Figure 1 represents the entity relationship diagram (ERD) developed. The ERD represents a document management and retrieval system that stores research documents (theses), their metadata, associated authors, keywords, research keywords, authors' affiliated institutions, authors' citations, users, search queries, ranking features, search results, and feedback. The THESES table holds the record for each document. Each document is identified by a unique Th_id in the THESES table and contains attributes like Th_abstract, Th_title, Th_pub_date, Th_doc_type, Th_full_text, Th_URL, and Th_file_path (presumably the storage location of the document). The Th_pub_date field uses the system's datetime to automatically record when the document (thesis) was added to the system. This table acts as the primary point of reference for other tables in the ERD.

THESES table was associated with the author's table. This design allows one-to-many relationships, meaning an author can submit a minimum of one THESIS and a maximum of many (1:m), and a THESIS can belong to a minimum of one author and a maximum of one (1:1). Each author is uniquely identified in the authors table by Auth_id and has Auth_fname, Auth_mname, Auth_lname, Auth_ORCID, and Auth_Affiliation as attributes.

THESES table was linked to the THESIS-KEYWORD table, and there exists a many-to-many relationship, which implies that a keyword can appear in a minimum of one THESIS and a maximum of many (1:m), while at the same time, a THESIS can have a minimum of one KEYWORD and a maximum of many (1:n), which is a problem that must be resolved. Similarly, the THESES table is associated with the ABST_KEYWORD (Abstract keyword) table with a many-to-many relationship (n:m), which implies that a THESIS can have a minimum of one abstract keyword and a maximum of many (1:m), while an abstract keyword (ABST_KEYWORD) can appear in a minimum of one THESIS and a maximum of many (1:n), which is a problem and must be resolved. This relationship supports indexing and searching, as multiple keywords can describe a single document and vice versa. The keywords table itself stores each keyword with a keyword_id and the keyword text.

The schema also includes the RANKING_FEATURE table linked to the THESES, SEARCH_RESULT, and SEARCH_QUERY tables with a relationship of many-to-one (n:m), respectively. This implies that a thesis, search query, and search result can have a minimum of one ranking score and a maximum of many (1:m), whereas a RANKING_FEATURE belongs to a minimum relevance of a document (THESIS) to the search query that produces the search result. Each of these tables has its unique ID and attributes as shown in Figure 1. The RANKING_FEATURE table stores the results of retrieval models applied to documents for specific query_text inputs. of one THESIS, SEARCH QUERY and SEARCH RESULT and a maximum of one of each of them (1:1). A many-to-one relationship is a normal relationship.

The AUTHOR table in the schema is linked to the INSTITUTION table with a many-to-one (normal) relationship, which implies that an author can be affiliated with one and only one (1:1) institution, whereas an institution can have a minimum of one AUTHOR affiliate and a maximum of many (1:m). AUTHOR table is also linked to the CITATION table with a one-to-many (1:m) relationship that implies that an AUTHOR can be cited a minimum of one time and a maximum of many (1:m), whereas a citation belongs to a minimum of one AUTHOR and a maximum of one (1:1).

The schema also contains the USER table with user attributes such as User_id, User_fname, User_mname, User_lname, User_email, and User_role and is linked to the SEARCH_QUERY table with a one-to-many (1:m) relationship. This implies that a USER can submit a minimum of one query and a maximum of many (1:m), whereas a SEARCH_QUERY belongs to a minimum of one USER and a maximum of one (1:1). The query passes through the RANKING_FEATURE to the SEARCH_RESULT table, which was linked to the FEEDBACK table with a many-to-many (m:n) relationship, which is a problem that must be resolved.

4. Steps in Migrating Enhanced Entity Relationship Diagram (EERD) for the Relevance Ranking System to Entity Relational Model (ERM)

The Enhanced Entity Relationship Diagram for the relevance ranking system were migrated to a Relational Model (Enhanced Relational Model) using MySQL Workbench following the five (5) migration rules below:

1. Every entity name translates to a relation name
2. Every primary key in an entity translates to a primary key in a relation
3. Every attribute of an entity becomes an attribute of a relation
4. For max cardinality m:1, the primary key of the many-side becomes a foreign key on the 1:side
5. For max cardinality m:n, a new relation will be born, and the primary keys of the two participating entities become attributes of the new relation, plus other attributes.

5. Results and Discussion Section

5.1 Results

Figure 2 represents the ERM developed in this study. After migrating the ERD (Figure 1) to ERM (Figure 2), the eleven (11) entity tables with ten (10) relations in the ERD become fourteen (14) relations in ERM.

In Figure 2, the relationship between CITATION and the AUTHOR tables (Figure 1) makes the Auth_id to become a foreign key in the CITATION relation (Figure 2); between AUTHOR and INSTITUTION tables, inst_id went from INSTITUTION table (Figure 1) to become a foreign key in the AUTHOR relation (Figure 2); AUTHOR and THESES tables (Figure 1) gave rise to Auth_id becoming a foreign key in the THESES relation (Figure 2); THESES gave its Th_id as foreign key in RANKING_FEATURE (Figure 2); USER and SEARCH_QUERY enabled user_id to become a foreign key in SEARCH_QUERY relation (Figure 2); the sq_id in SEARCH_QUERY become a foreign key in RANKING_FEATURE relation (Figure 2) and sr_id from SEARCH_RESULT went to become a foreign key in the RANKING_FEATURE relation (Figure 2). All these in Figure 1 were many-to-one (m:1 or 1:m) relationships.

The relationships among ABST_KEYWORD and THESES; THESIS_KEYWORD and THESES; and SEARCH_RESULT and FEEDBACK were all many-to-many (m:n) problems which needed resolution. The resolution between ABST_KEYWORD and THESES tables in Figure 1 was done by creating a bridge relation in-between, named THESES_has_ABST_KEYWORD, through which the many-to-many (m:n) relationship was broken into two many-to-many relationships (Figure 2). Between THESIS_KEYWORD and THESES, a new relation called THESES_has_KEYWORD that broke the many-to-many (m:n) relationship (Figure 1) into two many-to-one (m:1) relations (Figure 2).

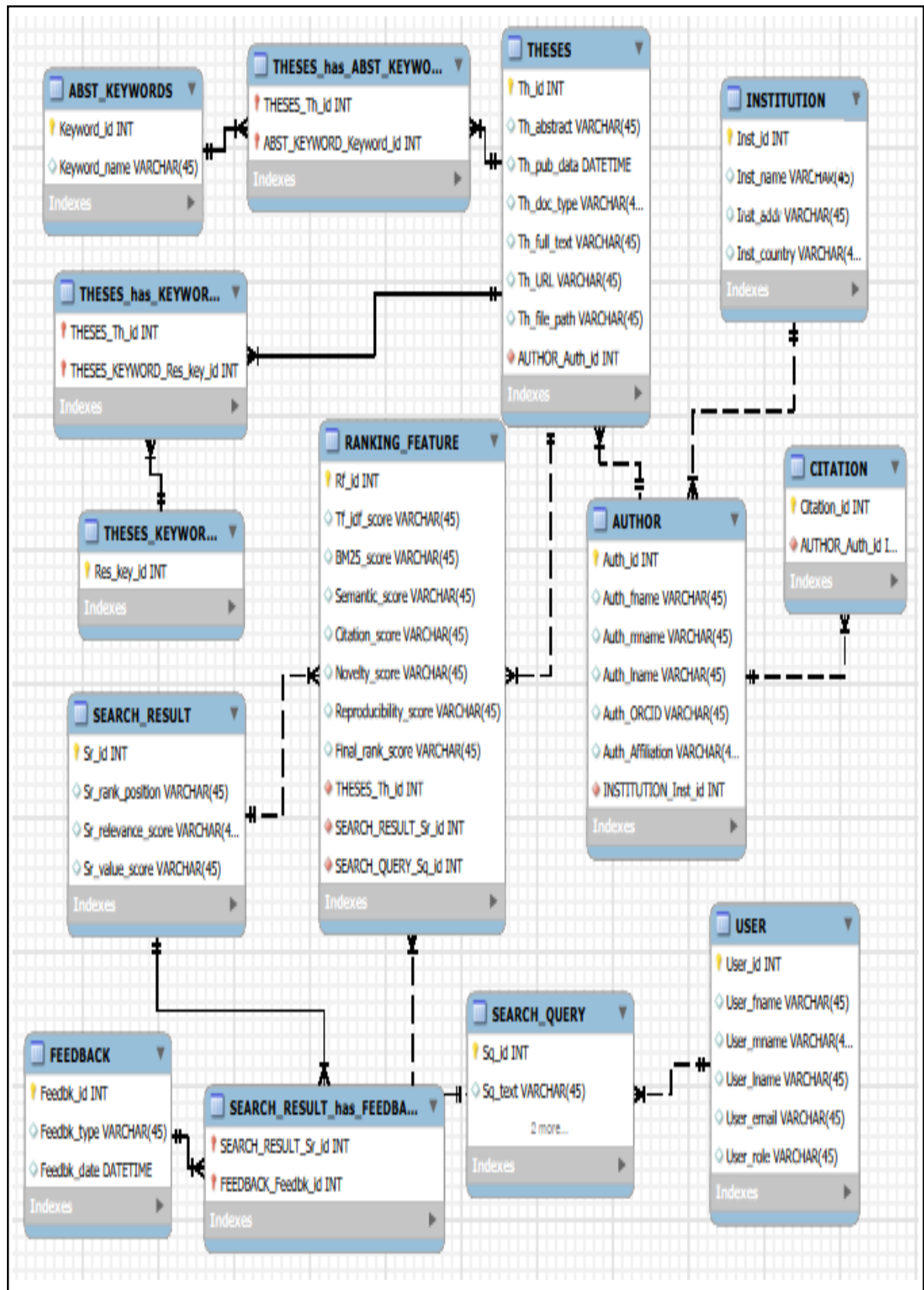


Figure 2: The Entity Relational Model (ERM) for the Relevance Ranking Database Design

Figure 1 was resolved in Figure 2 by creating a bridge relation named `SEARCH_RESULT_has_FEEDBACK`, which resolved the many-to-many relationship into two many-to-one (m:n) relations (Figure 2). Likewise, the relationship (m:n) between the `SEARCH_RESULT` and `FEEDBACK` tables in

5.2 Discussion

The study developed an Entity-Relationship Model (ERM) for the relevance ranking database design for effective document retrieval. Identifying the key entities and relationships needed in a thesis-document retrieval environment, creating an Entity-Relationship (E-R) model, and then converting the E-R model into a relational database model were the key tasks. The study identified `THESES`, `AUTHOR`, `CITATION`, `INSTITUTION`, `ASBT-KEYWORD`, `THESES-KEYWORD`, `USER`, `SEARCH-RESULT`, `SEARCH-QUERY`, `RANKING-FEATURE`, and `FEEDBACK` as the key entities using defended PhD theses from the University of Ibadan repository (1999–2021). MySQL Workbench was used to implement and model the database, and the Third Normal Form (3NF) was used to normalise the relational model.

The results demonstrated that a solid basis for an optimised relevance-ranking system is provided by careful entity selection and appropriate relationship definition. By converting the E-R model into a 3NF relational model, data redundancy is decreased, data integrity is enhanced, and anomalies related to document (thesis) insertion, deletion, and updating are less likely. Additionally, ranking algorithms may access relevant information without needlessly duplicating data because of the modular structure created by the division of search queries, search results, ranking features, and user feedback into discrete entities. This is especially crucial for an intelligent retrieval system, which may then integrate and optimise ranking features based on document relevancy.

The results have significant implications for query processing, scalability, system maintainability, and retrieval accuracy. Inconsistencies and broken relationships that could lead to relevant documents being left out of search results can be reduced with a well-

structured database. An architectural foundation for integrating machine learning or optimisation approaches into the retrieval process is also provided by the inclusion of ranking-feature and feedback entities. Furthermore, additional theses, institutions, authors, keywords, and user feedback can be supported by the relational structure without necessitating a significant database restructure. As a result, the database model functions as both an enabling infrastructure and a storage mechanism for an optimised relevance-ranking system.

The results of the study align with current database and information-retrieval research, which generally highlights efficient indexing, well-defined linkages, organised data representation, and normalisation as critical components of dependable information systems. However, the study focuses on the database architecture that facilitates the ranking process, in contrast to previous studies that primarily focus on the ranking algorithm itself, such as TF-IDF, BM25, or machine-learning-based ranking. This distinction is important because the presence of redundant, inconsistent, poorly connected, or insufficient data in the underlying database might compromise the efficacy of a ranking algorithm. As a result, the proposed model enhances current relevance-ranking techniques by offering an organised database on which these algorithms can function.

The proposed model expands the database structure to include search behaviour and ranking-related entities, including `SEARCH-QUERY`, `SEARCH-RESULT`, `RANKING-FEATURE`, and `FEEDBACK`, in contrast to traditional database models that are primarily intended for document storage. As a result, the model is better suited for modern intelligent retrieval systems where relevance is assessed using query characteristics, ranking criteria, and user interaction in addition to document content. As a result, the study shows that database modelling is not a stand-alone task apart from ranking-system development, but rather a crucial initial step in optimised information retrieval.

Thus, the main contribution is the development of an entity-relationship model (ERM) for the relevance ranking database design for efficient document retrieval.

6. Conclusion

An efficient relevance ranking system requires a well-structured database model. This study presented the design and implementation of an Entity-Relationship (ER) database model and Relational Model (RM) for an optimised relevance ranking system using MySQL Workbench. Within a cohesive and organised database framework, the proposed model effectively incorporates the essential elements of information retrieval, including document storage, term indexing, and metadata management. The system guarantees a high degree of data integrity and consistency, facilitates specialisation and generalisation, and captures complicated relationships by utilising ER modelling.

MySQL Workbench made it easier to see, validate, and implement the database architecture, guaranteeing that the conceptual ER model was successfully converted into a logical and physical schema. This offers scalability for managing big and dynamic datasets in addition to improving system maintainability. Overall, by combining ranking, indexing, and storage techniques into a unified framework, the proposed ER-based database model gets around some of the main drawbacks of traditional retrieval systems. It lowers response times, increases retrieval accuracy, and offers a versatile framework for incorporating sophisticated and adaptive ranking methods.

To further improve system performance and user satisfaction in contemporary information retrieval environments, future work may expand this model by adding semantic search capabilities, machine learning-driven ranking models, and real-time user customisation.

References

- [1] Chinguwo, D., & Deus, D. (2025). A systematic review of database design in the development and usage of floods early warning systems. *African Journal on Land Policy and Geospatial Sciences*, 8(8). <https://doi.org/10.48346/IMIST.PRSM/ajlp-gs.v8i8.59222>
- [2] CoreDeploy. (2025). *What is a DBMS? Database management systems explained*. Retrieved from <https://coredeploy.tech/foundations/database-management-systems/articles/what-is-dbms.html>
- [3] Dritsas, E., & Trigka, M. (2026). Machine learning in modern database systems: Techniques, architectures, and deployment challenges. *Engineering Applications of Artificial Intelligence*, 170, 114225. <https://doi.org/10.1016/j.engappai.2026.114225>
- [4] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson.
- [5] GeeksforGeeks. (2026). *ACID properties in DBMS*. Retrieved from <https://www.geeksforgeeks.org/dbms/acid-properties-in-dbms/>
- [6] IBM. (2024). *Artificial intelligence and information retrieval*. <https://www.ibm.com/>
- [7] IONOS. (2025). *Database management systems (DBMS): Tasks and functions*. Retrieved from <https://www.ionos.co.uk/digitalguide/hosting/technical-matters/database-management-systems-dbms/>
- [8] Milicevic, B., & Babovic, Z. (2024). A systematic review of deep learning applications in database query execution. *Journal of Big Data*, 11(173), 1–29. <https://doi.org/10.1186/s40537-024-01025-1>
- [9] MongoDB. (2026). *ACID transactions in databases*. Retrieved from <https://www.mongodb.com/resources/basics/databases/acid-transactions>
- [10] Mozaffari, M., Dignös, A., Gamper, J., & Störl, U. (2024). Self-tuning database systems: A systematic literature review of automatic database schema design and tuning. *ACM Computing Surveys*, 56(11), 1–37. <https://doi.org/10.1145/3665323>
- [11] Napoli, R., Celesti, A., Villari, M., & Fazio, M. (2026). How to evaluate NoSQL database paradigms for knowledge graph processing. *arXiv*. <https://arxiv.org/abs/2602.07612>
- [12] Neelamegam, G., Bhogal, J., Samimi, P., & Ozturkoglu, O. (2026). Comprehensive insights into bitemporal databases: A PRISMA-guided systematic literature review. *Journal of Data, Information and Management*. <https://doi.org/10.1007/s42488-026-00162-x>
- [13] Oguntunde (2025). Towards a secure multifactor authentication database model for e-procurement in production environments.

- Proceedings of the 39th iSTEMS Multidisciplinary Bespoke Conference & 39th Extended Conference. 17th- 19th July/17th – 19th October, 2025. University of Ghana, Accra, Ghana. Pp. 405 – 415.
- [14] Paiva, C. A., Maximino, R., Paiva, F., Vieira, R. A., Espanha, N., Pimentel, J. F., Wiese, I., Gerosa, M. A., Steinmacher, I., Murta, L., & Braganholo, V. (2026). Analysing the adoption of database management systems throughout the history of open-source projects. *arXiv*. <https://arxiv.org/abs/2605.06817>
- [15] Stanek, S., & Novotny, O. (2025). AI-assisted data modelling and design: A systematic literature review. *Quality Innovation Prosperity*, 29(3), 39–61. <https://doi.org/10.12776/qip.v29i3.2253>
- [16] Stewart, N., Adams, D., & Foster, U. (2025). Empirical analysis of query optimisation strategies for reducing execution latency in large-scale databases. *Journal of Innovation in Governance and Business Practices*, 1(1), 88–117.
- [17] Stonebraker, M., & Pavlo, A. (2024). *What goes around comes around... and around...* SIGMOD Record, 53(2). Retrieved from <https://db.cs.cmu.edu/papers/2024/whatgoesa-round-sigmodrec2024.pdf>
- [18] TechTarget. (2024). *Data management and enterprise technology insights*. [TechTarget Official Website](https://www.techtarget.com/searchdatamanagement/definition/database-management-system)
- [19] Wikipedia contributors. (2026). *Database*. Retrieved from <https://en.wikipedia.org/wiki/Database>
- [20] Yan, Z., Yuan, G., Guo, Q., & Lu, J. (2025). DBMS–LLM integration strategies in industrial and business applications: Current status and future challenges. *arXiv*. <https://arxiv.org/abs/2507.19254>
- [21] Yasar, K., & Mullins, C. S. (2024). Database management system (DBMS). TechTarget. Retrieved from <https://www.techtarget.com/searchdatamanagement/definition/database-management-system>
- [22] Zambrano, A., Torres, P., & Molina, R. (2026). Development of a high-performance in-memory database architecture for intelligent video surveillance in critical patient care. *Frontiers in Digital Health*,
- [23] Osofisan, A. O. and Oguntunde, T. (2017). Dynamic Characterization of Collaborative Objects in an ICT based Environment for Collaborative Learning and Research, University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR), Vol. 1, pp. 64 - 78. ©U IJSLICTR Vol. 1, June 2017.