Comprehensive Technical Report: The Empirical Impact of Database Normalization in Modern Computing

Armita Thapa

Patan College for Professional Studies, Nepal

armita.magar@patancollege.edu.np

Abstract*-recent academic literature (2025–2026) has shifted from theoretical modeling of database normalization to empirical quantification of its effects on performance, storage, energy efficiency, and Artificial Intelligence (AI) integration. This report synthesizes data from five primary studies. The evidence suggests that the transition from First Normal Form (1NF) to Second Normal Form (2NF) is the most critical design phase, yielding a 311% increase in throughput and a 74% reduction in energy consumption per transaction [5]. In industrial Big Data, normalization combined with advanced analytical techniques reduced dataset rows from 56 million to 283 daily, cutting query latency by over 99.9% [4]. However, while higher normalization (3NF+) is essential for the accuracy of AI-driven aggregation queries, it introduces structural complexities that challenge Large Language Models (LLMs) in simple data retrieval tasks [1].*

1. Introduction

Database normalization is a formal process used to organize relational tables to minimize data redundancy and eliminate anomalies (Insertion, Update, and Deletion). While normalization theory has been the bedrock of database design for decades, its application in modern Decision Support Systems (DSS), Big Data management, and Natural Language to SQL (NL2SQL) systems is currently being re-evaluated [2]. The central conflict remains the "join penalty"—the computational overhead of reassembling data from multiple tables—versus the benefits of integrity and storage reduction. This report examines how contemporary hardware and software handle these trade-offs.

2. Performance and Energy Consumption Analysis

The environmental and performance impact of normalization has become a focal point of recent "Green Computing" research.

2.1. Throughput Gains

Empirical testing on PostgreSQL using the Internet Movie Database (IMDb) dataset reveals that **1NF is highly inefficient** due to the storage of large, redundant strings and multi-valued attributes [5]. Transitioning to **2NF** isolates these attributes, allowing the database engine to process transactions significantly faster. Research indicates that this move increases throughput by a factor of 4 [5].

2.2. Energy Efficiency

The energy required to process a single transaction is directly proportional to CPU and I/O cycles. Because 1NF requires the system to scan and process redundant data, it is energy intensive. Moving to **2NF reduces energy consumption by 74%** [5]. Interestingly, normalizing further (from 2NF to 4NF) provides minimal additional energy gains (less than 1%), suggesting that 2NF/3NF is the "efficiency sweet spot" [5].

3. Storage Efficiency and Big Data Management

Storage management is critical for organizations handling petabyte-scale data or industrial sensor feeds.

3.1. Redundancy and Volume

Normalization effectively reduces the physical footprint of data. Studies in inventory management and general relational systems indicate that 3NF compliance can reduce total database size by approximately **30%** [3]. However, some researchers caution that over-normalization (2NF to 4NF) can actually **increase** storage requirements by roughly **7%** due to the added overhead of table headers, metadata, and row identifiers [5].

3.2. Case Study: Industrial Big Data

In cable manufacturing, the "4Vs" of Big Data (Volume, Velocity, Variety, and Veracity) present massive challenges. One study documented a system collecting data from 45 TAGs at one-minute intervals, generating **56 million records daily** [4]. By implementing a strategic normalization process and MSSQL optimization, the researchers reduced the daily dataset to just **283 rows** [4]. This allowed complex analytical queries that previously took **40 minutes** to execute in under **0.1 seconds** [4].

4. Normalization in the Era of AI (NL2SQL)

As organizations integrate Large Language Models (LLMs) to allow users to query databases in natural language, schema design has become a critical performance factor.

4.1. The Complexity Challenge

LLMs, even leading models, struggle with highly normalized schemas in "zero-shot" settings. Normalized tables (2NF/3NF) require the AI to accurately predict join types and select correct base tables, leading to a drop in accuracy for simple retrieval tasks [1].

4.2. The Aggregation Advantage

Conversely, for **aggregation queries** (e.g., "What is the average sensor reading?"), normalized schemas are far superior. Denormalized (flat) tables often contain duplicate data entries for a single entity. When an AI attempts to aggregate this data, it produces mathematically incorrect results due to these duplicates. Normalized schemas provide a "cleaner" mathematical structure, ensuring higher accuracy for complex data analysis [1].

5. Conclusion

The collective research confirms that database normalization remains an indispensable tool for modern data architecture. While the "join penalty" is a valid concern, the empirical benefits of normalization—specifically the **74% reduction in energy** and **99.9% improvement in Big Data query latency**—far outweigh the costs in most production environments [4, 5]. For AI applications, a balanced approach is required: while 3NF provides the necessary integrity for aggregation, developers should use "few-shot" examples to help AI models navigate the increased schema complexity [1].

6. References

[1] Kohita, R. (2025). *Exploring Database Normalization Effects on SQL Generation*. arXiv:2510.01989v1 [cs.CL]. https://arxiv.org/abs/2510.01989

[2] Fotache, M., Cluci, M. I., Taipalus, T., & Talaba, G. (2026). *The effects of database normalization on decision support system performance*. Information Systems, 136, 102636. https://doi.org/10.1016/j.is.2025.102636

[3] Hardini, M., Agarwal, V., Apriani, D., Widjaya, I. A., & Setiawaty, E. (2025). *Application of Database Normalization in Increasing Data Storage Efficiency*. International Transactions on Artificial Intelligence (ITALIC), 3(2), 201-211. https://journal.pandawan.id/italic/article/view/799

[4] Altınışık, S. B., & Bilgin, T. T. (2025). *Optimizing Big Data Management on Microsoft SQL Server: Enhancing Performance through Normalization and Advanced Analytical Techniques*. International Journal of Innovative Engineering Applications, 9(1). https://dergipark.org.tr/tr/download/article-file/4273819

[5] Taipalus, T. (2025). *On the effects of logical database design on database size, query complexity, query performance, and energy consumption*. arXiv:2501.07449v1 [cs.DB]. https://arxiv.org/abs/2501.07449