

<Conference abbreviation>

<Conference Series name>

<Volume number and Year> <DOI Number>

Bridging the Semantic Gap: Natural Language to Relational Algebra and SQL Query Generation

Kirti H. Wanjale, Pragati A. Bandagar, Manasvi S. Bandkar, Aadi A. Bhangdiya,
Yogini P. Bhamare

*Vishwakarma Institute of Technology Pune, Maharashtra, India, kirti.wanjale@vit.edu,
pragati.bandagar24@vit.edu, manasvi.bandkar24@vit.edu, yogini.bhamare24@vit.edu*

Abstract.

There are Natural Language to SQL systems that have an immense contribution towards easing database access to users in the form of using their ordinary language for queries without needing a complicated syntax. On the other hand, conventional database management systems require the use of Structured Query Language which poses an obstacle to the use of these systems by non-programmers. In order to solve this problem, this paper presents an implementation of a system that will provide a connection between natural language, relational algebra, and SQL. The suggested method will follow a process-oriented approach in which the user's query is processed to obtain its relational algebra equivalent after which it will be translated to a suitable SQL command. Such a connection will help to make more clear the link between the intention of the user and the operation to be performed in the database. The experiment results show that there is improvement in the accuracy and clarity of the translation process from natural language to SQL.

Keywords. Natural Language, Structured Query Language (SQL), Relational Algebra (RA), Semantic Parsing, Query Generation

1. INTRODUCTION

The growing volume of data has increased the need for accessible database querying. Traditional SQL requires technical expertise, limiting database access for non-technical users. Natural Language-to-SQL (NL2SQL) systems address this challenge by enabling queries through natural language, although translation remains complex. Existing machine learning and semantic parsing approaches [2], [6] often lack transparency and interpretability. This study proposes a systematic mapping between Natural Language Queries (NLQ), Relational Algebra (RA), and SQL to improve traceability and understanding. The model also supports teaching NLP, RA, and SQL concepts [7].

2. RELATED WORK

Many attempts have been made to create NLTOSQL approaches to simplify the database access process. The early attempts were through rule-based approaches [6], which were

difficult to use when processing complex queries. Later attempts included neural and semantic parsing approaches [4], which offered more accurate results, albeit lacking explainability [1-10]. The current state-of-the-art NLTOSQL approaches include approaches like QuerySpeak [2] and the ones involving chatbots [3]. Techniques based on LLMs [1], [10] perform better, whereas some recent studies also use relational algebra [7], [8] for better explainability.

Table 1. Comparison of existing work with proposed work

Reference	NLP	RA	Transparency
[1]	Yes	No	No
[2]	Yes	No	No
[3]	Yes	No	No
[4]	Yes	No	No
[7]	No	Yes	Yes
[8]	Yes	Yes	Yes
This work	Yes	Yes	Yes

3. METHOD, EXPERIMENTS AND RESULTS

The proposed system uses a Java-based multi-stage framework to convert natural language questions into SQL queries. First, the user query is normalized, and relevant entities and keywords are identified and semantically tagged. The system then checks the query for ambiguity and consistency. Valid queries are converted into relational algebra and corresponding SQL statements. The Application Layer uses an NLP parsing engine to interpret user intent and extract relevant information, which is passed to the Relational Algebra/SQL Generator for query construction. The generated SQL is executed through Java Database Connectivity (JDBC), which connects the application to the MySQL database. Retrieved results are processed and presented through the Presentation Layer, providing users with an accessible interface for submitting natural language questions and receiving database results.



Figure 4.1. System Flowchart

4.1 Functional Illustration

The Design system produces many complete NLQ to SQL query conversions through relational algebra. Some of those examples are included in Table 2 below.

Table 2. Sample Outputs

Component	Example Output
NLQ	(1) show doctors specialization cardiology (2) show recent 10 appointments
SQL Query	(1) <code>SELECT * FROM doctor WHERE LOWER(specialization) LIKE 'cardiology'</code> (2) <code>SELECT * FROM appointment ORDER BY visit_day DESC LIMIT 10</code>
RA Expression	(1) σ condition ($\pi * (\text{doctor})$) (2) τ (appointment)

The system improves traceability and transparency by generating Relational Algebra (RA) and SQL simultaneously, allowing users to follow the translation from logic to executable code [14][19]. MySQL compatibility through JDBC supports reliable query execution, while NLP processing handles linguistic variations effectively [12]. Its modular Java architecture further supports reliability and easier debugging [14]. The system also serves as a teaching instrument by demonstrating the relationship between NLQ, RA, and SQL [19][14]. Its modular design enables future extensions, including semantic relationships and query optimization [1]. This combination improves usability, learning, maintainability, and functional extensibility.

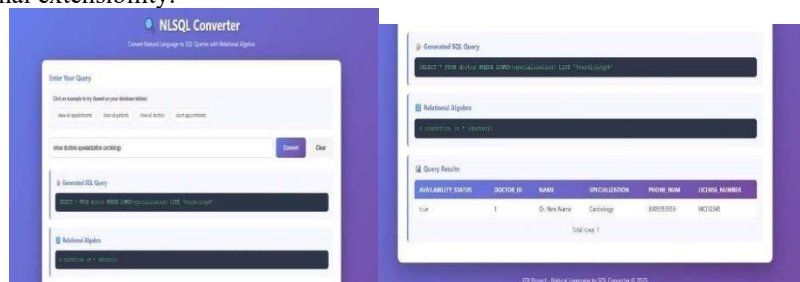


Figure 4.2. Results of the system

The developed NL2SQL system effectively bridges natural language queries and database processing through relational algebra and SQL. Relational algebra improves transparency by allowing users to trace query processing. The system performs reliably for straightforward to moderately complex queries, although complex or ambiguous queries require improved semantic analysis. Overall, the system supports efficient database access while providing educational value for understanding query processing.

4. CONCLUSIONS

This work addresses database access challenges for non-technical users by translating natural language into relational algebra and SQL queries executable through MySQL using JDBC. The system generates precise, understandable, and traceable queries while retrieving

relevant database results. However, it currently supports simple queries. Future work will focus on complex queries, improved semantic accuracy, and query optimization.

5. REFERENCES

1. A. K. Sharma, S. C. Kanumuri, P. AR, and S. S. Sharath, "Advancing Natural Language to SQL: A Comparative Study of Open Source LLMs on Benchmark Datasets," *2025 IEEE Symposium on Computational Intelligence in Natural Language Processing and Social Media (CI-NLPSoMe Companion)*, 2025.
2. T. S. V. Reddy, T. Charithat, T. Sathwik, S. V. Samitha, and N. Panda, "QuerySpeak: From Human Language to Database Queries," *2024 5th IEEE Global Conference for Advancement in Technology (GCAT)*, 2024.
3. M. S. Mahagave et al., "AI Enabled Chat-Driven Database Management for NLP to SQL Conversion," *2025 Seventh International Conference on Computational Intelligence and Communication*
4. R. Xu and A. Singh, "Neural Semantic Parsing Natural Language into SQL," *arXiv:2106.13858v1 [cs.CL]*, 2017.
5. D. M. L. Nethravathi et al., "Structuring Natural Language to Query Language: A Review," *Engineering, Technology & Applied Science Research*, 2020.
6. G. Singh and A. Solanki, "An algorithm to transform natural language into SQL queries for relational databases," *Selforganizology*, 2016.
7. R. Suryawanshi et al., "Mapping of Relational Algebra Into SQL Queries," *Project Report, Vishwakarma Institute of Technology*, 2025.
8. F. S. C. Tseng, A. L. P. Chen, and W. P. Yang, "On mapping natural language constructs into relational algebra through E-R representation," *Information Sciences*, 1993.
9. A. Giordani and A. Moschitti, "Corpora for Automatically Learning to Map Natural Language Questions into SQL Queries," *ACL Anthology*, 2010.
10. X. Liu et al., "A Survey of Text-to-SQL in the Era of LLMs: Where are we, and where are we going?," *arXiv:2408.05109v5 [cs.DB]*, 2025.

Biographies



Dr. Kirti H. Wanjale is a Professor in the Department of Computer Engineering at Vishwakarma Institute of Technology, Pune. Her research interests include Artificial Intelligence, IoT, Data Science, and intelligent computing systems. She has published research papers, authored books, and holds several patents in AI, IoT, automation, and smart systems.



Pragati A. Bandagar is a Computer Engineering student at Vishwakarma Institute of Technology, Pune. Her areas of interest include Web Development, Data Science, Artificial Intelligence, and Machine Learning. She has published research paper in field of smart management. She is keen to explore emerging technologies in the field of Computer Engineering.



Manasvi S. Bandkar is a student in the Department of Computer Engineering at Vishwakarma Institute of Technology, Pune, Maharashtra, India. Her research interests include artificial intelligence, smart monitoring systems, and intelligent technology applications.



Aadi A. Bhangdiya is a student in the Department of Computer Engineering at Vishwakarma Institute of Technology, Pune, Maharashtra, India. His academic interests include Artificial Intelligence, Machine Learning, Data Analytics, Web Technologies, and Software Development. He has published research in international conference proceedings.



Yogini P. Bhamare is a student in the Department of Computer Engineering at Vishwakarma Institute of Technology, Pune, Maharashtra, India. Her research interests include artificial intelligence, intelligent systems, and emerging computer technologies.