

Speak2DB NLP-Driven Multi-Database Query Assistant

**Mrs V. Vanaja¹, Kothapeta Maha Lakshmi², Gaddam Sai Suhani³, Chepuri Madhulika⁴,
Chippabathini Vikas⁵**

Assistant Professor, Dept. of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India¹
IV B. Tech Students, Department of CSE (Data Science), ACE Engineering College, Hyderabad,
Telangana, India^{2,3,4,5}

ABSTRACT: Accessing information stored in databases typically requires knowledge of structured query languages such as SQL, which can be challenging for users without technical expertise. The SPEAK2DB system proposes an NLP-driven query assistant that enables users to retrieve information from database files using natural language queries. Instead of manually writing SQL statements, users can enter queries in simple conversational language, and the system interprets the request and converts it into an executable SQL command. SPEAK2DB processes the user input, identifies the relevant database file, and retrieves the requested data while enforcing role-based access control to ensure secure data access. By allowing users to interact with multiple SQLite database files through a natural language interface, the system simplifies database interaction and makes data retrieval more accessible for non-technical users.

KEYWORDS: Natural Language Processing, Database Querying, SQL Generation, Multi-Database Systems, Role- Based Access Control, Query Assistant, Data Retrieval

I. INTRODUCTION

Modern organizations rely heavily on databases to store and manage large volumes of structured information. Retrieving this information typically requires users to write queries using Structured Query Language (SQL). While SQL is powerful and flexible, it can be difficult for non-technical users to learn and use effectively. As a result, many individuals depend on database administrators or technical staff to access even simple information stored in databases.

Recent developments in Natural Language Processing (NLP) have made it possible to design systems that allow users to interact with computers using everyday language. By interpreting user queries expressed in natural language, such systems can translate human requests into structured commands that machines can understand. This approach has the potential to make database systems more accessible and user-friendly.

The SPEAK2DB system addresses this challenge by providing an NLP-driven interface that enables users to retrieve information from multiple databases using simple text or speech queries. The system interprets user requests, converts them into SQL queries, and retrieves the appropriate data while maintaining secure access through role-based permissions. This paper presents the design, implementation, and evaluation of the SPEAK2DB system and demonstrates how it simplifies database interaction for users without technical expertise.

This study focuses on designing and implementing the SPEAK2DB system as an accessible interface for database interaction. The proposed approach combines natural language processing techniques with structured database querying to interpret user requests and generate appropriate SQL commands.

By integrating natural language understanding with controlled database access, the system aims to create a more efficient and user-friendly method for retrieving information from structured data sources. The work presented in this paper explores how such an interface can improve usability while maintaining accuracy and security in database operations.

II. LITERATURE SURVEY

Natural language interfaces for databases have gained significant attention as researchers attempt to simplify how users interact with structured data. One notable contribution in this area is SQLNet, proposed by Xu et al. in 2017. The system introduced a sketch-based neural model that translates natural-language queries into SQL statements. Instead of predicting the entire SQL query at once, the model separates the process into structure prediction and value filling. This approach improved the stability of query generation and reduced the complexity often associated with reinforcement learning techniques used in earlier systems.

Building on these developments, Wang et al. proposed RAT-SQL in 2019, a relation-aware transformer model designed to better understand the relationships between database schema elements. The model captures semantic dependencies between tables, columns, and query components, enabling more accurate SQL generation. By incorporating schema relationships directly into the model architecture, RAT-SQL demonstrated improved performance when handling complex queries involving multiple tables and structured database schemas.

Further improvements were introduced by Scholak et al. in 2021 through the PICARD framework, which focused on improving the syntactic correctness of generated SQL queries. The system validates partial SQL sequences during the decoding process, ensuring that the generated queries remain consistent with SQL grammar rules. When integrated with transformer-based language models such as T5, PICARD significantly reduces invalid query generation and enhances the reliability of text-to-SQL systems.

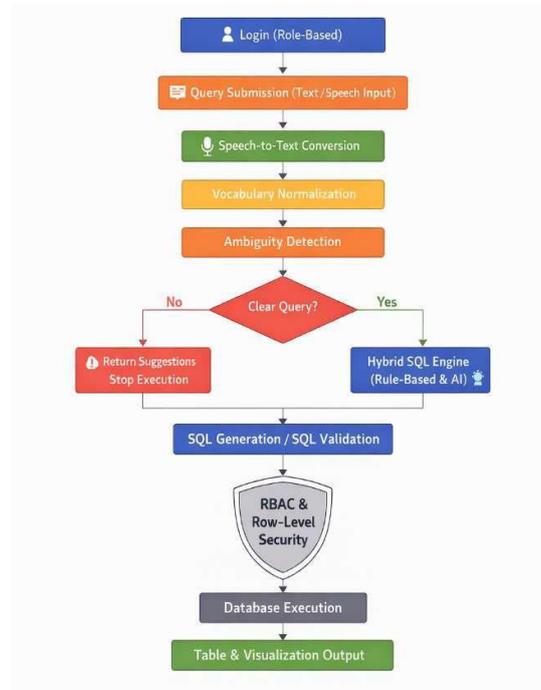
More recent research has explored advanced structural modelling techniques to improve query understanding. Chen et al. proposed LGESQL in 2022, which uses graph neural networks to represent SQL grammar and database schema structures. This approach allows the system to better capture relationships between schema components and generate more accurate complex queries.

Similarly, Zhang et al. introduced RESDSQL in 2023, which incorporates a structurally decoupled framework along with a re-evaluation mechanism to improve schema linking and query accuracy. These advancements highlight the ongoing progress in natural language query systems and demonstrate how modern architectures continue to improve the reliability and performance of database query assistants.

In addition to model-based approaches, researchers have also explored conversational systems that allow users to interact with databases in a more natural way. Instead of requiring users to write complete queries, these systems support dialogue-based interaction where the system can ask follow-up questions if a request is unclear or incomplete. This makes the interaction feel more intuitive and reduces the difficulty for users who are not familiar with technical query languages.

By combining natural language understanding with simple conversational mechanisms, such systems help improve usability and make database access more practical for everyday users. These developments highlight the growing interest in designing database interfaces that focus not only on accuracy, but also on user experience and accessibility.

III. METHODOLOGY



The SPEAK2DB system follows a structured process that allows users to interact with database systems using natural language. The workflow begins with a role-based login, where users authenticate themselves before accessing the system. This step ensures that each user interacts with the system according to their assigned permissions. After logging in, users can submit their queries either as text input or through speech. Allowing both input methods provides flexibility and makes the system easier to use for individuals with different preferences.

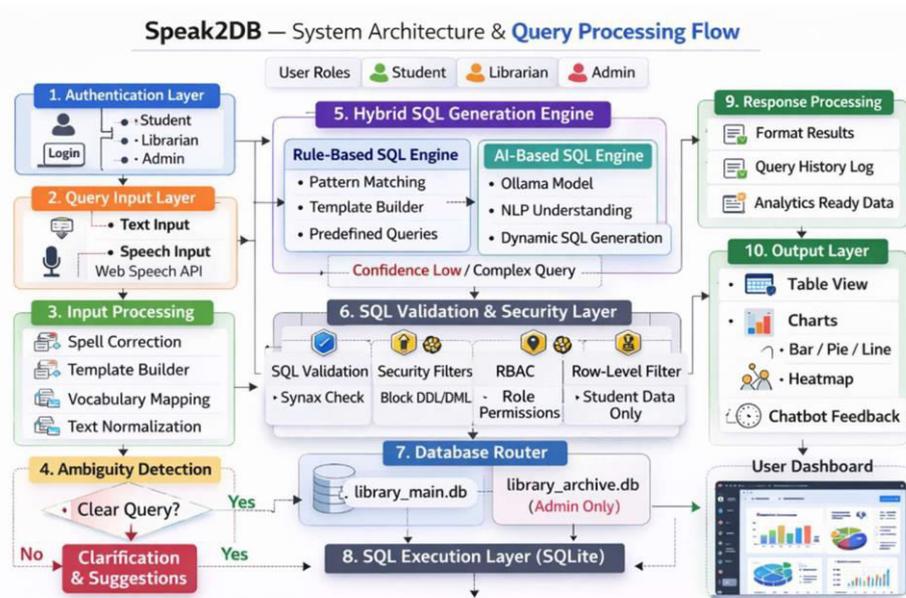
When a user submits a spoken query, the system first performs speech-to-text conversion to transform the audio input into textual form. Once the query is available in text format, the system applies vocabulary normalization, which helps standardize words and phrases used in the query. This step improves the system's ability to interpret different ways of expressing the same request.

After preprocessing, the system performs ambiguity detection to determine whether the user's request is clear enough to generate a valid database query. If the system identifies that the query is incomplete or ambiguous, it stops the execution process and returns suggestions to the user for clarification. This mechanism helps prevent incorrect or misleading database results. If the query is clear, it proceeds to the hybrid SQL engine.

The next stage involves SQL generation and validation, where the system constructs a structured SQL query corresponding to the user's request. Before execution, the system enforces role-based access control and row-level security to ensure that users can only retrieve data they are authorized to view. This step is particularly important in environments where different users have different access privileges.

Finally, once the query passes validation and security checks, it is executed against the database. The retrieved results are then processed and displayed to the user in a structured format. The system presents the output in the form of tables and visualized results, allowing users to easily understand and analyze the retrieved data. This complete workflow ensures that natural language queries are accurately interpreted, securely processed, and presented in a user-friendly format.

IV. SYSTEM ARCHITECTURE



The architecture of the SPEAK2DB system is designed as a layered framework that manages user interaction, query processing, database access, and result presentation. The process begins with the authentication layer, where users log into the system based on their assigned roles such as student, librarian, or administrator. This role-based authentication ensures that every user interacts with the system according to their permitted level of access. By establishing user identity at the beginning of the workflow, the system creates a secure foundation for all subsequent operations.

Once authentication is completed, the user submits a query through the query input layer, which supports both text and speech-based inputs. Speech queries are captured through the web speech interface and converted into text before further processing. The system then performs several preprocessing steps in the input processing layer, including spell correction, vocabulary mapping, and text normalization.

After preprocessing, the system performs ambiguity detection to determine whether the query is clear enough to be processed. If the request is incomplete or unclear, the system provides clarification suggestions to the user. For clear queries, the request is forwarded to the hybrid SQL generation engine, which combines rule-based query templates with AI-driven natural language understanding. The rule-based component handles common query patterns using predefined templates, while the AI-based component supports more complex or flexible queries by dynamically generating SQL statements.

Before the generated query is executed, it passes through the SQL validation and security layer, where the system checks the syntax of the query and applies security restrictions. This layer enforces role-based access control and row-level filtering to ensure that users only retrieve data they are authorized to access. After validation, the database router determines which SQLite database file should be used for the query and directs the request to the appropriate database. The query is then executed within the SQL execution layer.

Finally, the results obtained from the database are processed in the response processing and output layers. The system formats the retrieved data and prepares it for visualization on the user dashboard. Results may be displayed as structured tables or visual representations such as charts and graphs, allowing users to easily interpret the information.

V. RESULTS

NLP-Driven Multi-Database Query Assistant

You have been logged out.

USERNAME
admin

PASSWORD

Available Accounts

- ADMIN / PASS
- LIBRARIAN / PASS
- FACULTY_EMAIL / PASS

Student Accounts (100 Roll Numbers)

- MT3001 / PASS
- IT2002 / PASS
- EE4003 / PASS
- BM4004 / PASS

Fig. 5.1

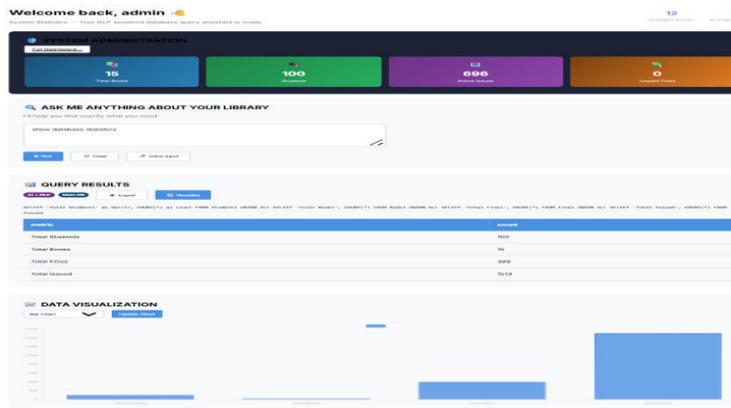


Fig. 5.2

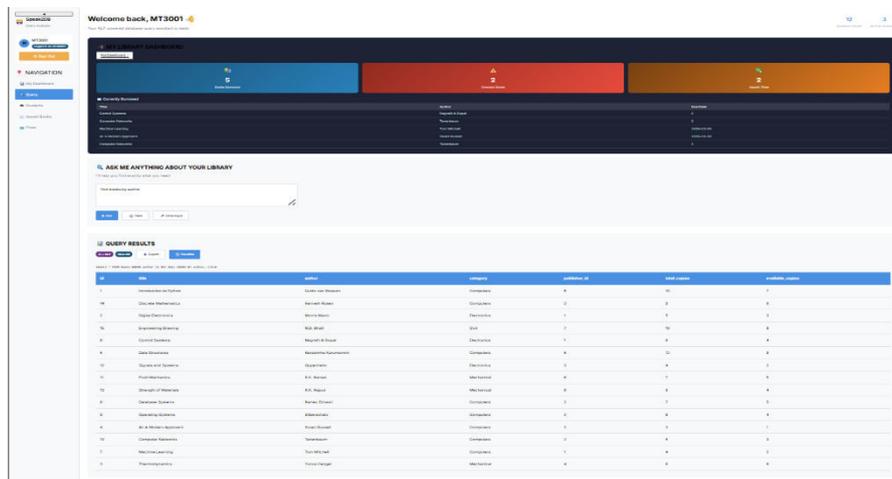


Fig. 5.3

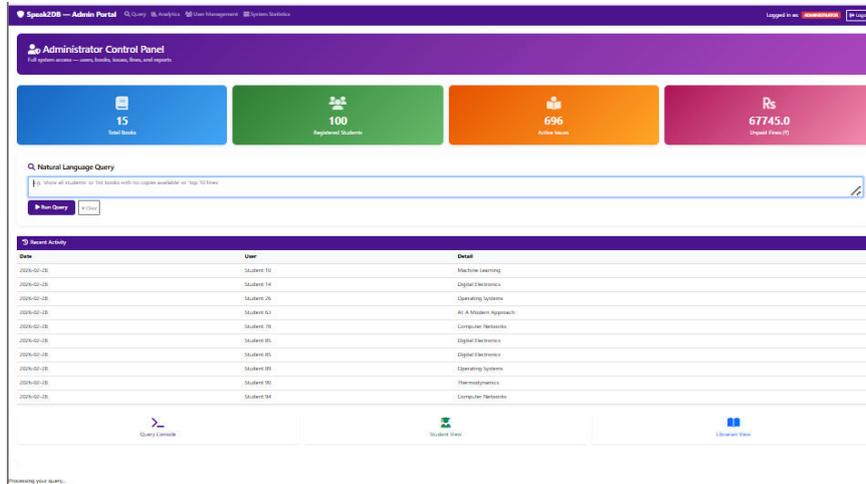


Fig 5.4

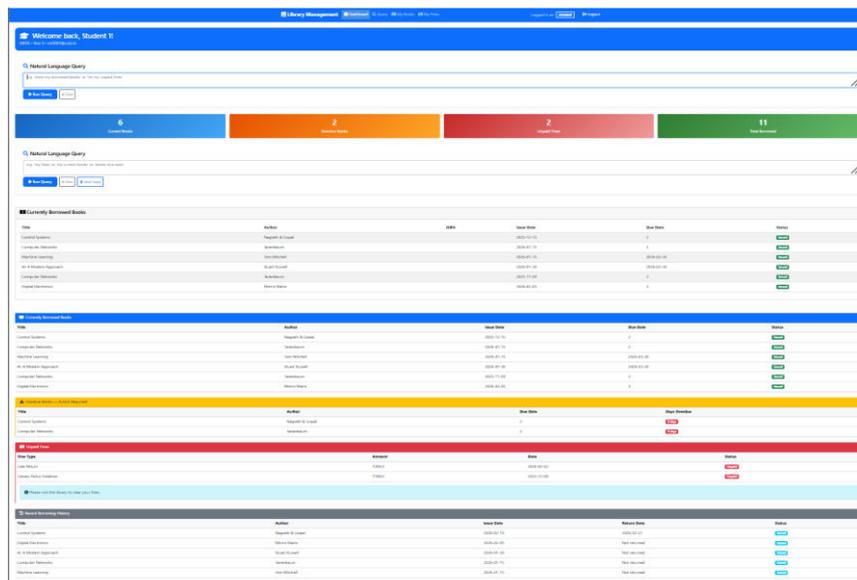


Fig 5.5

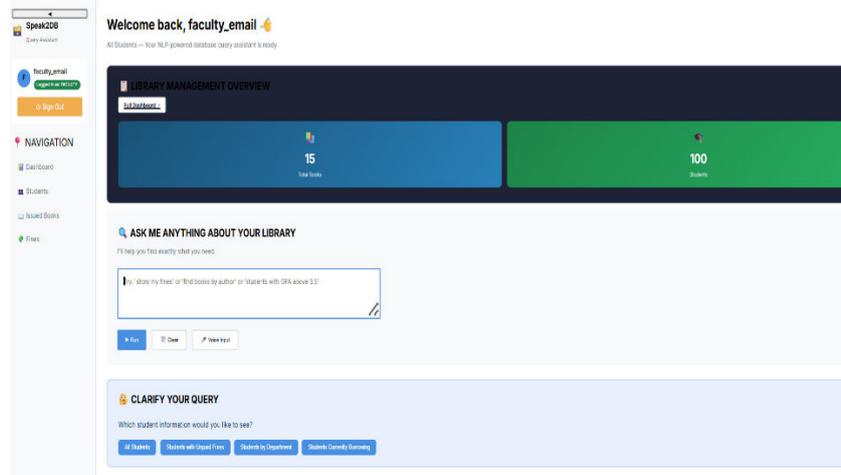


Fig 5.6

VI. CONCLUSION

The Speak2DB system was developed to simplify the process of interacting with databases by allowing users to use natural language queries instead of traditional SQL commands. Many users who need information from databases may not have the technical knowledge required to write SQL queries. This project addresses that limitation by providing a system that can understand user queries in natural language and convert them into SQL statements automatically.

During the development of this project, different components such as the user interface, NLP processing module, SQL query generation module, and database interaction layer were integrated to create a complete working system. The system accepts user queries through text or speech input and processes them using natural language processing techniques. After interpreting the user's request, the system generates the appropriate SQL query and retrieves the relevant information from the database.

The implementation of Speak2DB demonstrates how natural language interfaces can make database systems more accessible and user-friendly. By removing the need for users to write complex SQL queries, the system improves usability and reduces the barrier for non-technical users who want to retrieve data quickly and efficiently. The system also provides structured results that are easy for users to understand.

Overall, the project highlights the potential of combining natural language processing with database management systems to create smarter and more interactive data retrieval tools. The results obtained from the system show that natural language based database querying can be a practical solution for improving user interaction with databases in real-world applications.

VII. FUTURE SCOPE

Although the Speak2DB system successfully demonstrates natural language based database querying, there are several areas where the system can be further improved and expanded. One possible enhancement is improving the accuracy of the natural language processing module so that it can better understand complex and nested queries. This would allow the system to handle more advanced database operations.

Another potential improvement is supporting multiple database management systems such as MySQL, PostgreSQL, and MongoDB. Currently, the system may be designed for a specific database structure, but future versions could be made more flexible so that they can work with different database platforms without requiring major modifications.

The system can also be enhanced by improving its speech recognition capabilities. By integrating more advanced speech recognition technologies, the system could better understand spoken queries even in noisy environments or when users have different accents. This would make the system more reliable and convenient for voice-based interactions.

In addition, future development could include the integration of machine learning models that learn from previous queries and user interactions. This would allow the system to provide more accurate results over time and better understand user intent. With these improvements, Speak2DB has the potential to evolve into a more intelligent and adaptive database interaction system.

REFERENCES

- [1] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the NAACL-HLT Conference*, 2019.
- [2] Zhong, V., Xiong, C., and Socher, R., "Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning," *arXiv preprint arXiv:1709.00103*, 2017.
- [3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [4] Sandhu, R., Coyne, E., Feinstein, H., and Youman, C., "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.
- [5] Halfond, W., Viegas, J., and Orso, A., "A Classification of SQL Injection Attacks and Countermeasures," *Proceedings of the IEEE International Symposium on Software Reliability Engineering (ISSRE)*, 2006.
- [6] Grinberg, M., *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 2018.
- [7] Jurafsky, D., and Martin, J. H., *Speech and Language Processing*, 3rd ed., Pearson, 2023.
- [8] Li, Y., et al., "SQLCoder: Language Models for Text-to-SQL Generation," *arXiv preprint*, 2023.
- [9] Iyer, S., Konstas, I., Cheung, A., and Krishnamurthy, J., "Learning a Neural Semantic Parser from User Feedback," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.
- [10] Yu, T., et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [11] Dong, L., and Lapata, M., "Language to Logical Form with Neural Attention," *Proceedings of the Association for Computational Linguistics (ACL)*, 2016.
- [12] Wang, C., Tatwawadi, K., Brockschmidt, M., Huang, P. S., Mao, Y., Polozov, O., and Singh, R., "Robust Text-to-SQL Generation with Execution-Guided Decoding," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [13] Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J., Liu, T., and Zhang, D., "Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation," *Proceedings of the Association for Computational Linguistics (ACL)*, 2019.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details