



## Enhanced Text to SQL Parser

S. Sairaam Prasad<sup>1</sup>, T. Antony Ashwin Daniel<sup>2</sup>, K. Mohamed Aasin<sup>3</sup>, Dr. G. Uma Maheswari<sup>4</sup>  
<sup>1,2,3</sup> Year, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India.

<sup>4</sup>Assistant Professor, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India.

**Emails:** 23ucs041@kamarajengg.edu.in<sup>1</sup>, 23ucs031@kamarajengg.edu.in<sup>2</sup>,  
23ucs074@kamarajengg.edu.in<sup>3</sup>, umamaheswaricse@kamarajengg.edu.in<sup>4</sup>

### Abstract

*Text-to-SQL parsing is an important research area in natural language processing that enables users to interact with databases using plain English queries. Traditional database systems require knowledge of structured query language (SQL), which limits accessibility for non-technical users. This paper proposes an Enhanced Text to SQL Parser that converts natural language questions into accurate and optimized SQL queries. The system integrates Natural Language Processing (NLP), semantic analysis, and schema understanding to generate reliable SQL statements. The proposed solution improves query accuracy, reduces human effort, and enables intelligent database interaction for academic, business, and enterprise applications.*

**Keywords:** Text-to-SQL; Natural Language Processing; SQL Generation; Database Systems; Query Optimization; Artificial Intelligence.

### 1. Introduction

Databases are fundamental components of modern information systems, storing and managing large volumes of structured data. However, interacting with databases traditionally requires users to write SQL queries, which demands technical expertise. Many users struggle to retrieve data efficiently due to limited knowledge of SQL syntax and database schema design. Recent advancements in artificial intelligence and NLP have enabled systems to interpret human language and convert it into structured commands. The Enhanced Text to SQL Parser aims to bridge the gap between human language and relational databases by translating user queries into syntactically correct and semantically meaningful SQL statements. This system enhances usability, reduces learning barriers, and simplifies database management through intelligent automation.

### 2. Related Works

#### 2.1. Text-to-SQL Systems

Previous research has explored rule-based and machine learning approaches for converting natural language into SQL queries. Early systems relied on predefined templates, which limited flexibility and scalability. Modern approaches use deep learning models to understand context and schema

relationships, improving performance and adaptability.

#### 2.2. Database Query Automation

Several automated query generation tools exist; however, many struggle with complex joins, nested queries, and ambiguous user input. The proposed system enhances parsing accuracy by incorporating semantic analysis and schema mapping techniques for improved query construction.

### 3. Problem Statement

- Existing database systems require users to possess technical knowledge of SQL.
- Natural language interfaces often misinterpret complex queries.
- Schema ambiguity reduces query accuracy.
- Limited support for nested and multi-table queries.

There is a need for an intelligent system that accurately converts natural language into optimized SQL queries while maintaining high reliability and scalability.

### 4. Proposed System

The proposed Enhanced Text to SQL Parser is an AI-powered system that translates user questions into SQL queries using NLP and schema analysis. The system processes user input, identifies entities and

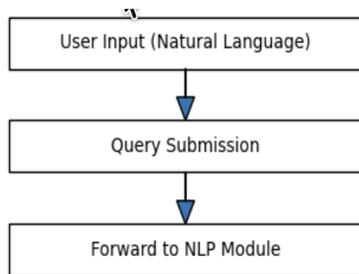
relationships, and constructs optimized SQL statements compatible with relational database systems.

**Key Objectives**

- Enable natural language interaction with databases
- Improve SQL query accuracy and efficiency
- Support multi-table and nested query generation
- Reduce dependency on technical SQL knowledge

**5. System Modules**

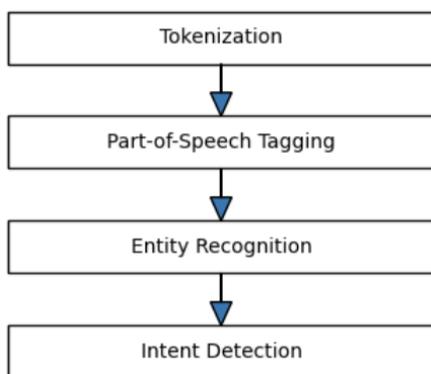
**5.1. User Interface Module**



**Figure 1** Allows Users to Enter Natural Language Queries Through a Simple Interface

**5.2. NLP Processing Module**

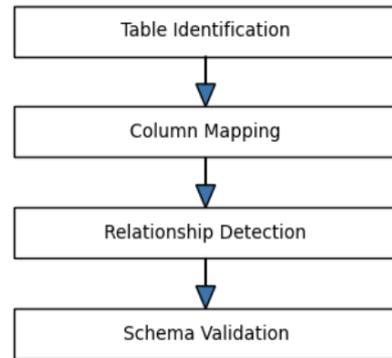
Processes input text by performing tokenization, part-of-speech tagging, and entity recognition to understand user intent.



**Figure 2** NLP Processing Module

**5.3. Schema Mapping Module**

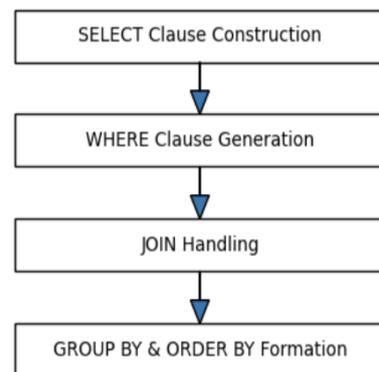
Maps identified entities and attributes to database tables and columns using metadata and schema relationships.



**Figure 3** Schema Mapping Module

**5.4. SQL Generation Module**

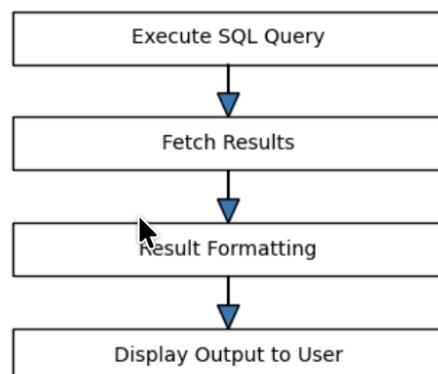
Constructs syntactically correct SQL queries, including SELECT, WHERE, JOIN, GROUP BY, and ORDER BY clauses when required.



**Figure 4** SQL Generation Module

**5.5. Database Execution Module**

Executes generated SQL queries on the database and retrieves accurate results for display.

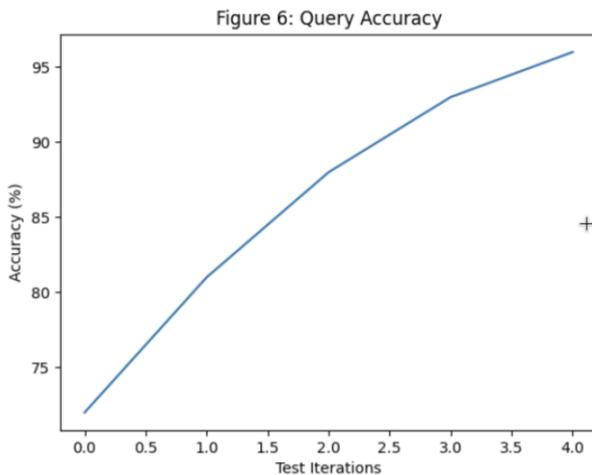


**Figure 5** Database Execution Module

## 6. Performance Analysis

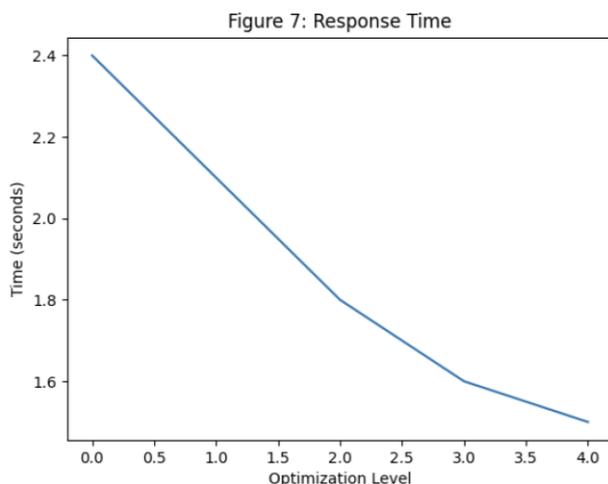
The system is evaluated based on the following parameters:

**Query Accuracy:** Measures correctness of generated SQL statements.



**Figure 6 Query Accuracy**

**Response Time:** Evaluates speed of query generation and execution.



**Figure 7 Response Time**

## Conclusion

The Enhanced Text to SQL Parser provides an intelligent solution for bridging the gap between human language and relational databases. By integrating NLP techniques with schema-aware query construction, the system simplifies database

interaction for users without SQL expertise. The proposed system improves accessibility, reduces manual errors, and enhances efficiency in data retrieval processes. Through accurate query generation and structured execution, it demonstrates strong potential for use in academic institutions, enterprises, and data-driven organizations. The integration of intelligent parsing mechanisms ensures reliable performance and supports scalable database operations. Overall, the system represents a significant advancement in natural language database interfaces and contributes to the development of user-friendly AI-driven database technologies.

## Future Enhancements

Future improvements can further enhance the system's performance and applicability. Incorporating advanced deep learning models such as transformer-based architectures can improve contextual understanding and query prediction accuracy. Multilingual support can be added to allow users to interact with databases in different languages. Integration with voice-based assistants can enable speech-to-SQL functionality for enhanced accessibility. Additionally, implementing adaptive learning mechanisms will allow the system to improve over time based on user interactions and feedback. Support for multiple database management systems such as MySQL, PostgreSQL, and Oracle can increase compatibility and adoption. Security enhancements, including role-based access control and query validation, can ensure safe database operations. These enhancements will strengthen system reliability, scalability, and usability, paving the way for intelligent and automated database management solutions.

## References

- [1]. A. Yaghmazadeh et al., "SQLizer: Querying Databases Using Natural Language," Proceedings of the VLDB Endowment, 2017.
- [2]. T. Yu et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing," EMNLP, 2018.
- [3]. C. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating Structured Queries from Natural Language," ICLR, 2017.
- [4]. X. Wang et al., "IRNet: A General



Framework for Text-to-SQL Generation,”  
ACL, 2019.

- [5]. D. Hakkani-Tür et al., “Natural Language  
Interfaces to Databases: A Survey,”  
Computational Linguistics Journal, 2020.