

Agentic AI Framework for Natural Language to SQL Query Generation

Satish Bhaskar¹, U. Karthi², CH. Rajesh³, T. Tejaswini⁴

Assistant Professor, Department of CSE, ACE Engineering College, Hyderabad, Telangana, India¹

IV B. Tech. Students, Department of CSE, ACE Engineering College, Hyderabad, Telangana, India^{2,3,4}

ABSTRACT: This project presents an agentic Natural Language to SQL (NL-to-SQL) system that enables users to query relational databases using plain English instead of complex SQL statements. The system uses an agent-based architecture to understand user intent, extract live database schema, and generate accurate, schema-validated SQL queries. A hybrid combination of rule-based logic and AI-assisted intent extraction ensures safe handling of both simple and complex queries. The generated SQL is executed on a MySQL database, and results are displayed with explanations and insights. The system achieves high accuracy, low latency, and reliable performance, making it suitable for real-time database analytics. It is an intelligent, production-ready web application designed to simplify database interaction for non-technical users. Traditional database querying requires knowledge of Structured Query Language (SQL), which creates a barrier for business users and stakeholders. This system addresses the challenge by enabling users to interact with complex MySQL databases using natural language queries. Overall, this project bridges the gap between natural language understanding and database analytics, empowering users to make data-driven decisions quickly and effectively while maintaining high standards of security and reliability.

KEYWORDS: NL2SQL Systems, Transformer-based Models, Prompt Engineering, Semantic Parsing, Agent-based Architecture, Database Interaction Systems, SQL Synthesis, Context-Aware Querying, AI-driven Analytics, Secure Query Execution.

I. INTRODUCTION

In the modern data-driven world, organizations rely heavily on databases to store, manage, and analyze large volumes of structured information. However, accessing and extracting meaningful insights from these databases typically requires proficiency in Structured Query Language (SQL), which poses a significant challenge for non-technical users such as business analysts, managers, and decision-makers. This gap between technical database systems and user accessibility often leads to delays in decision-making and underutilization of valuable data resources.

Natural language access to databases is essential for enabling non-technical users to retrieve information efficiently. This project introduces an agentic Natural Language to SQL system that automatically converts plain English queries into SQL statements. By leveraging AI-assisted intent understanding and schema-aware validation, the system improves query accuracy and makes database interaction faster, safer, and more user-friendly.

The system allows users to input queries in natural language. It incorporates features such as schema awareness, query validation, role-based access control, and interactive visualization of results.

II. LITERATURE SURVEY

Recent studies have explored the use of Artificial Intelligence, Natural Language Processing (NLP), and deep learning techniques to develop Natural Language to SQL (NL2SQL) systems, enabling users to query databases using plain English instead of structured SQL queries.

The research paper “Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning” proposes a model that uses reinforcement learning to generate SQL queries from natural language. While it improves query accuracy, the approach depends heavily on reinforcement learning and is limited to simpler query structures.[1]

The research work “SQLNet: Generating Structured Queries from Natural Language Without Reinforcement Learning” introduces a sequence-to-set model to overcome limitations of reinforcement learning. Although it reduces dependency on reinforcement learning and improves efficiency, it still lacks the ability to generalize across complex and cross-domain database schemas.[2]

Recent research “Large Language Models for Text-to-SQL: Challenges and Opportunities” highlights the effectiveness of transformer-based LLMs in generating SQL queries using prompt engineering. However, direct SQL generation using LLMs introduces issues such as hallucination, lack of reliability, and potential security risks.[5]

In summary, existing research primarily focuses on improving the accuracy of SQL generation using machine learning and deep learning techniques. However, many approaches suffer from limitations such as high computational cost, lack of generalization, latency, and security concerns.

III. METHODOLOGY

The proposed Agentic Natural Language to SQL (NL2SQL) Analytics System is designed using a hybrid, agent-based architecture that combines rule-based processing with Large Language Model (LLM)-driven intent extraction. The methodology focuses on ensuring high accuracy, low latency, and strong security while translating natural language queries into executable SQL statements.

To ensure database safety, the system enforces a strict read-only policy. Potentially harmful SQL operations such as INSERT, UPDATE, DELETE, DROP, ALTER, and TRUNCATE are filtered and blocked at the backend level. Token-based validation is applied before executing any query, ensuring that only safe and valid SQL statements are processed.

The system incorporates schema-aware query generation by dynamically extracting database metadata (e.g., table names, columns) using SQL commands such as SHOW TABLES and DESCRIBE. This schema information is embedded into LLM prompts along with few-shot examples, enabling accurate mapping between user intent and database structure.

Depending on the dataset, the system generates bar charts, line graphs, pie charts, or tabular views. Additionally, it produces concise, human-readable insights summarizing the query results.

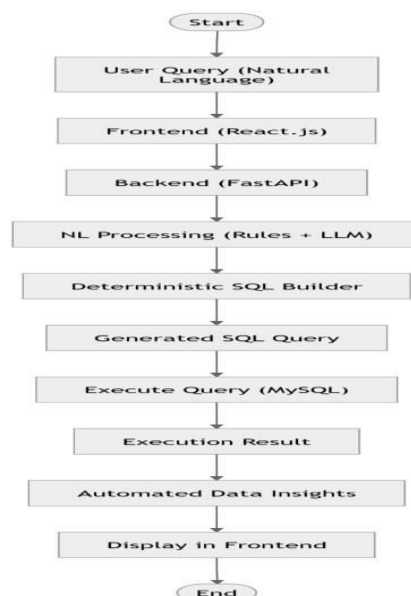


Fig 3.1 Natural Language to SQL Query Generation Methodology

IV. SYSTEM ARCHITECTURE

The Agentic Natural Language to SQL (NL2SQL) Analytics System is designed using a structured approach that separates the application into three main layers: frontend, backend, and database. This design helps the system convert user questions written in simple language into secure SQL queries and generate meaningful results efficiently.

The frontend is the user-facing part of the system. It allows users to enter their queries in plain language and view the output in an easy-to-understand format. The interface is designed to be simple so that even non-technical users can interact with the system comfortably.

The backend acts as the brain of the system. It processes the user's input, understands its intent, transforms it into an SQL query, verifies its safety, and executes it. This layer ensures that the entire workflow is handled correctly and securely.

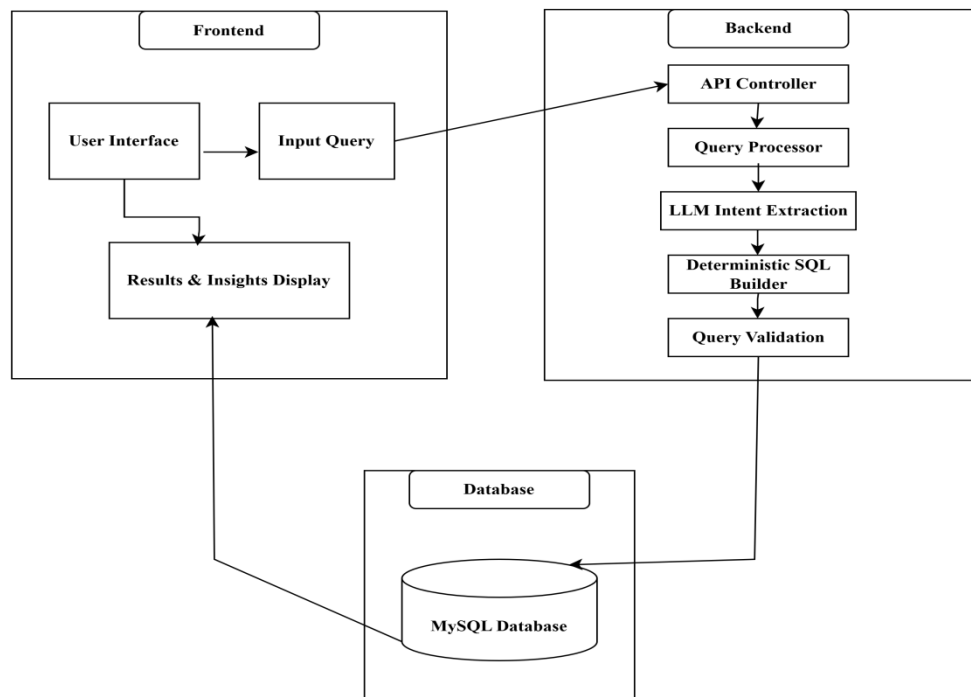


Fig 4.1 System Architecture

V. OUTPUT SCREENS

The system offers a simple and user-friendly interface that allows users to interact with databases using natural language. The interface is divided into two main sections: the Database Connection screen and the Query Analysis screen.

In the first screen, users can connect to a MySQL database by providing necessary details such as Host (for example, localhost), Port (default: 3306), Username (such as root), and Password. Once the user clicks the Connect button, the system checks the entered credentials and establishes a secure connection with the database.

This step ensures that only authorized users are able to access the system, thereby maintaining data security and controlled usage.

After a successful connection, the system moves to the query analysis interface. This section includes features like user query input, where users can enter their questions in natural language, and the generated SQL query, which shows how the system converts the input into an executable database query.

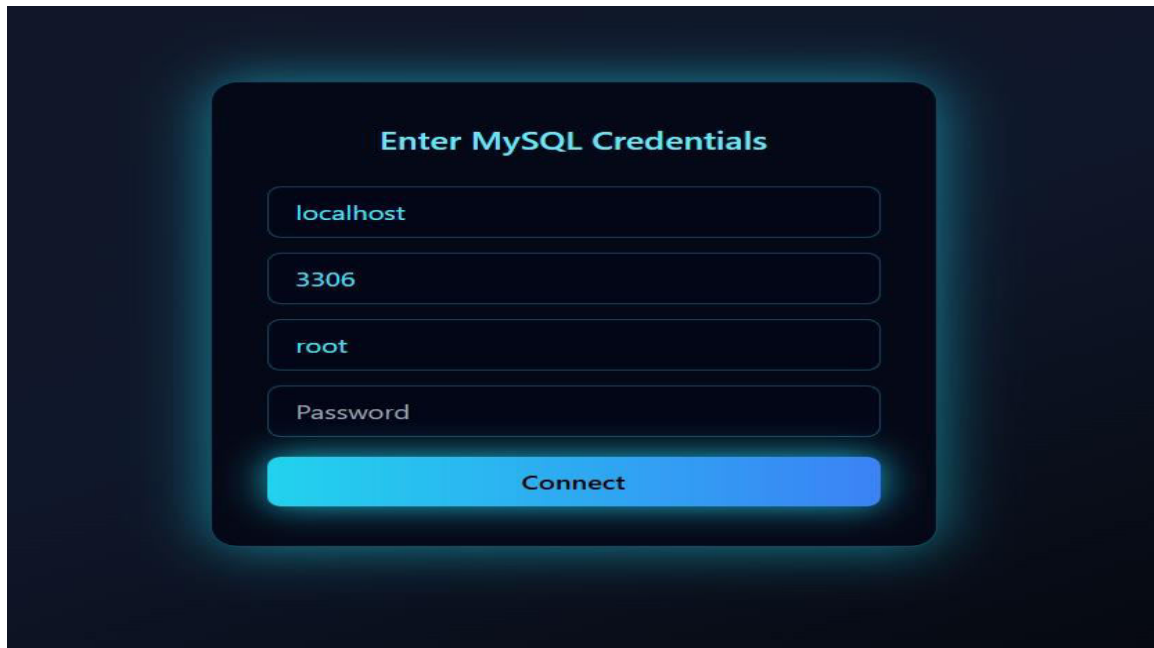


Fig 5.1 Database Connection Interface

Query Analysis

Question

give me the names whose cgpa is greater than 9

Generated SQL

```
SELECT Name FROM `cse_c` WHERE cgpa > 9 LIMIT 10
```

Execution Result

Name
Rajesh
Sneha
Neha
Meena
Bhavya

Insights

Returned 5 rows based on your query.

5.2 Query Analysis

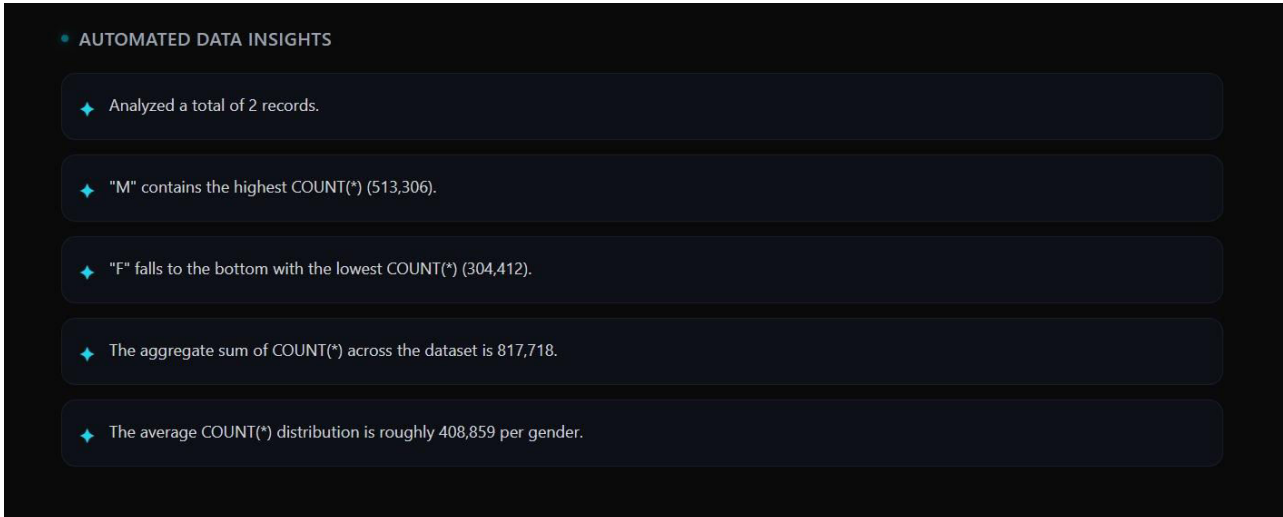


Fig 5.3 Data Insights page



Fig 5.4 visualization Result

VI. CONCLUSION

The Agentic Natural Language to SQL (NL2SQL) Analytics System presents an effective way to connect everyday language with structured database queries. By using a hybrid dual-pass approach that combines rule-based processing with LLM-based understanding, the system achieves both speed and accuracy in handling user requests.

The use of a structured SQL generation process improves consistency and ensures that the queries produced are reliable and secure. In addition, strong validation checks are applied to allow only safe, read-only operations, which helps protect the database from any unwanted changes.

VII. FUTURE SCOPE

The proposed system can be further enhanced by incorporating several advanced features to improve usability and functionality. One important extension is the integration of voice-based query input, which would allow users to interact with the system using speech, making it more accessible and user-friendly. Additionally, the system can be improved to support advanced query handling, including complex SQL operations such as joins, subqueries, and nested queries, thereby increasing its capability to handle real-world analytical tasks. Expanding the system to include multilingual support would enable users to input queries in different languages, broadening its accessibility to a global audience. Furthermore, integration with business intelligence (BI) tools can provide advanced dashboards, reporting features, and deeper analytical insights, making the system more suitable for enterprise-level applications. These enhancements will significantly improve the system's efficiency, flexibility, and overall user experience

REFERENCES

- [1] V. Zhong, C. Xiong, and R. Socher, (2017)“Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning,”arXiv.Link: <https://arxiv.org/abs/1709.00103>
- [2] X. Xu, C. Liu, and D. Song, (2017) “SQLNet: Generating Structured Queries from Natural Language Without Reinforcement Learning,”arXiv.Link: <https://arxiv.org/abs/1711.04436>
- [3] B. Wang et al.,“RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers,” Proceedings of ACL, (2020)Link: <https://aclanthology.org/2020.acl-main.677/>
- [4] T. Yu et al.,“Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task,”Proceedings of EMNLP, (2018)Link: <https://aclanthology.org/D18-1425/>
- [5] Xiaohu Zhu, Qian Li, Lizhen Cui, Yongkang Liu, (2024). “Large Language Models for Text-to-SQL: Challenges and Opportunities,” IEEE Transactions on Knowledge and Data Engineering.Link: <https://arxiv.org/html/2410.06011v1>



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