

SQLBOT – AI CHATBOT FOR SQL QUERY ASSISTANCE USING CONTEXT-FREE GRAMMAR

ANUDEEP NALUKURTHI¹

1.Student, Dept. of Computer Science and Engineering (CSE), Usha Rama College of Engineering and Technology. Near Gannavaram Ungutur Mandalam, Telaprolu, Andhra Pradesh.

Dr. K. P N V SATYA SREE²

2. Professor, Dept. of Computer Science and Engineering (CSE), Usha Rama College of Engineering and Technology. Near Gannavaram Ungutur Mandalam, Telaprolu, Andhra Pradesh

ABSTRACT

Structured Query Language (SQL) is essential for interacting with relational databases, but many beginners struggle with its syntax and logical structure. This paper presents SQLBOT, an AI-based chatbot designed to assist users in generating and understanding SQL queries using Context-Free Grammar (CFG). The system enables users to interact with databases through natural language queries, which are interpreted and converted into valid SQL commands. By integrating Natural Language Processing (NLP) and grammar-based parsing, SQLBOT analyzes user input, validates SQL syntax, and provides accurate query suggestions. The chatbot improves user experience by simplifying database interactions and reducing syntax errors. The proposed system demonstrates how conversational AI combined with formal grammar rules can enhance SQL learning and database accessibility for students, analysts, and developers.

Keywords: SQLBOT, Chatbot, SQL Query Assistance, Context-Free Grammar, NLP, Database Systems

I. INTRODUCTION

With the rapid growth of digital data, relational databases have become a fundamental component of modern information systems. SQL is the standard language used to retrieve, update, and manage data stored in relational databases. Despite its importance, writing SQL queries can be challenging for beginners because it requires knowledge of database schema, syntax rules, and query logic. Many users prefer interacting with systems using natural language rather than structured query languages. Chatbots powered by Artificial Intelligence have become popular tools for providing conversational interfaces that simplify complex tasks. This research proposes SQLBOT, an intelligent chatbot that assists users in generating SQL

queries from natural language input. The system uses Context-Free Grammar (CFG) to define the syntax rules of SQL and validate the generated queries. By combining Natural Language Processing and grammar-based parsing, SQLBOT provides an efficient solution for assisting users in writing accurate SQL queries. The primary aim of the system is to reduce query errors, support SQL learning, and enable easier interaction with relational databases.

II. LITERATURE REVIEW

Early research in Natural Language Interfaces to Databases (NLIDB) focused on translating natural language queries into SQL statements. Systems such as LUNAR and CHAT-80 demonstrated the feasibility

of natural language querying for database retrieval. Recent advancements in Artificial Intelligence and NLP have enabled more sophisticated chatbot systems capable of understanding user intent. Studies have shown that AI-powered chatbots can improve accessibility and user interaction with complex software systems. Researchers have also used Context-Free Grammar (CFG) to define syntactic structures of programming languages and database queries. CFG provides a structured method for validating SQL syntax and ensuring that generated queries follow proper rules. Existing solutions often rely heavily on machine learning models but lack grammar-based validation, which may lead to incorrect SQL statements. Therefore, integrating CFG-based parsing with chatbot systems provides a more reliable approach for SQL query generation and validation. The proposed SQLBOT combines **AI-based conversational interfaces with grammar-based query validation**, making it a practical tool for SQL learning and database interaction.

III. SYSTEM ARCHITECTURE

The architecture of SQLBOT consists of several components that work together to process user input and generate SQL queries.

1. **User Interface (Chat Interface)**
Provides a conversational interface where users can type questions or commands.
2. **Natural Language Processing Module**
Interprets the user's natural language input and identifies the query intent.
3. **Context-Free Grammar Parser**
Validates the structure of generated SQL queries according to predefined grammar rules.

4. Query Generator

Converts interpreted user intent into SQL statements.

The interaction flow begins with the user entering a query, which is processed by the NLP module, validated by the CFG parser, executed in the database, and returned as a response.

SYSTEM ARCHITECTURE

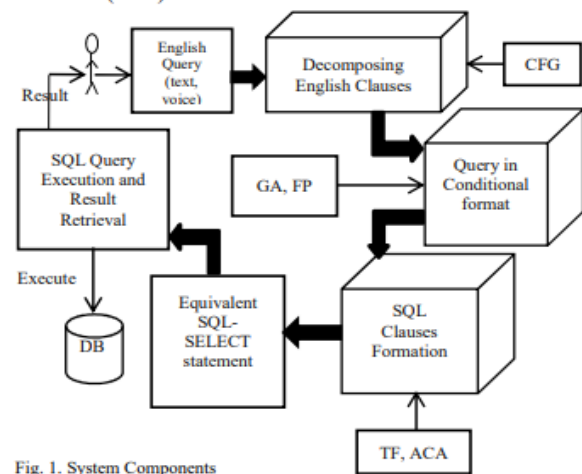
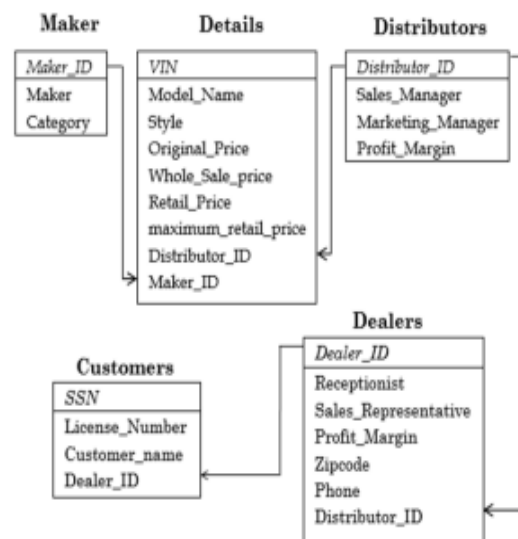


Fig. 1. System Components



IV. METHODOLOGY

The proposed system follows a structured approach to convert natural language queries into valid SQL statements.

Step 1: Input Processing

The user enters a natural language query such as: “Show all students whose marks are greater than 80.”

Step 2: Natural Language Analysis

The NLP module extracts keywords such as:

- Table name
- Column names
- Conditions
- Operators

Step 3: Grammar Validation

The extracted elements are validated using **Context-Free Grammar rules** to ensure that the SQL query structure is correct.

Example CFG rule:

Query → SELECT Columns FROM Table WHERE Condition
 Columns → * | ColumnList
 ColumnList → Column | Column , ColumnList
 Condition → Column Operator Value

Step 4: SQL Query Generation

After validation, the system constructs the SQL query:

```
SELECT * FROM Students WHERE Marks > 80;
```

Step 5: Query Execution

The generated query is executed in the database and the results are retrieved.

Step 6: Response Delivery

The chatbot returns the results to the user in conversational format.

Objectives

The primary objectives of the proposed SQLBot system are:

- To design a **Context-Free Grammar (CFG)** capable of parsing complex English queries used for database interaction.
- To implement a system that accurately translates the **parsed semantic structure into syntactically correct SQL queries**.
- To support advanced query components such as **logical operators (AND, OR, NOT)** and **aggregate functions**.

To develop an **interactive chatbot interface** that allows users to input natural English queries and receive immediate SQL query assistance.

Scope

The scope of the proposed system includes the following capabilities:

- The system accepts natural English questions as text input from the user.
- The chatbot analyzes the input and generates a corresponding syntactically valid SQL query.
- The system supports SQL queries containing SELECT, FROM, WHERE, GROUP BY, and HAVING clauses.
- The chatbot handles **complex conditions** including conjunctive (AND), disjunctive (OR), and negative (NOT) conditions within a single query.

PROBLEM STATEMENT

Although SQL is a fundamental skill in data analysis and database management, beginners often struggle with its strict syntax and logical structure. Translating a conceptual question into a valid SQL query can be difficult for students and novice professionals. This challenge creates a barrier to learning and effectively using database systems. While modern AI-based solutions using Large Language Models (LLMs) provide conversational assistance, they are inherently probabilistic and cannot guarantee syntactic accuracy. In applications where precise query generation is required, this lack of reliability becomes a limitation. Therefore, there is a need for a deterministic and education-oriented system that can reliably convert natural language queries into correct SQL statements while helping users understand the underlying query structure.

Existing System

Recent advancements in educational technology have led to the development of chatbot-based support systems using no-code platforms such as Chatbase.

These platforms integrate powerful Large Language Models (LLMs) through APIs such as those provided by OpenAI.

In such systems, educators can build chatbots by uploading course-related documents such as lecture notes, syllabi, and assignments.

The chatbot then uses this contextual information to answer students' questions about course materials, deadlines, or definitions. While these systems are effective for information retrieval and conversational assistance, their primary function is to retrieve and summarize information from a predefined knowledge base. They are not specifically designed to translate natural language queries into structured SQL statements.

Limitations of Existing System

• Lack of Syntactic Guarantee:

LLM-based systems generate outputs probabilistically and cannot guarantee that the generated SQL query will be syntactically correct.

• Non-Deterministic Results:

The same input query may produce different SQL outputs on different attempts, which reduces reliability for educational purposes.

• Difficulty with Complex Logic:

These systems often struggle to correctly interpret queries containing multiple logical operators such as **AND, OR, and NOT**.

• **Black-Box Decision Process:**

The internal reasoning process of LLMs is not transparent, making it difficult for learners to understand how the SQL query was generated.

Proposed System

The proposed system, **SQLBot**, is an intelligent chatbot designed specifically to assist users in generating SQL queries from natural language input. The core innovation of this system is a rule-based approach called **Grammar-Driven Semantic Mapping (GDSM)**. Unlike probabilistic neural network approaches, the GDSM method relies on a **deterministic Context-Free Grammar (CFG)**. The grammar defines formal rules that parse an English sentence into its grammatical components.

During this process:

- **Nouns** are mapped to **database tables and columns**
- **Verbs** represent **database operations**
- **Adjectives and conditions** are mapped to the **WHERE clause**

By applying these grammar rules, SQLBot constructs SQL queries that are both **logically accurate and syntactically valid**. This rule-based design enables the system to reliably handle complex SQL queries involving multiple conditions and aggregate operations.

Advantages of Proposed System

• **Deterministic and Reliable:**

The grammar-based approach ensures that the same

natural language query always produces the same SQL output.

• **Guaranteed Syntactic Accuracy:**

Since the SQL query is constructed according to formal grammar rules, the generated query is always syntactically valid.

• **Better Handling of Complex Queries:**

The CFG is designed to handle nested conditions and multiple logical operators effectively.

• **Transparent and Explainable System:**

The parsing process can be traced through grammar rules, allowing users to understand how their natural language query is converted into SQL.

Modules Description

1. Query Preprocessing and Lexical Analysis Module

This module receives the raw English query from the user through the chatbot interface. It performs preprocessing tasks such as **text normalization, lowercasing, and tokenization**. The input sentence is divided into individual words or lexemes that can be further analyzed.

2. Part-of-Speech (POS) Tagging Module

The tokenized sentence is analyzed to identify the grammatical role of each word. POS tagging assigns labels such as **noun, verb, adjective, and preposition**, which helps the system understand the structure and meaning of the query.

3. GDSM Parsing Engine

The **Grammar-Driven Semantic Mapping (GDSM) engine** is the core component of SQLBot. It applies predefined **Context-Free Grammar rules** to the POS-tagged sentence and generates a **parse tree** representing the grammatical structure of the query. This parse tree identifies relationships between words and determines how they correspond to SQL elements.

4. SQL Construction Module

The SQL Construction Module traverses the parse tree generated by the GDSM parser. It extracts relevant entities such as:

- Table names
- Column names
- Conditions
- Aggregate functions

These elements are then combined to construct a **final SQL query that follows correct syntax and structure.**

V. IMPLEMENTATION

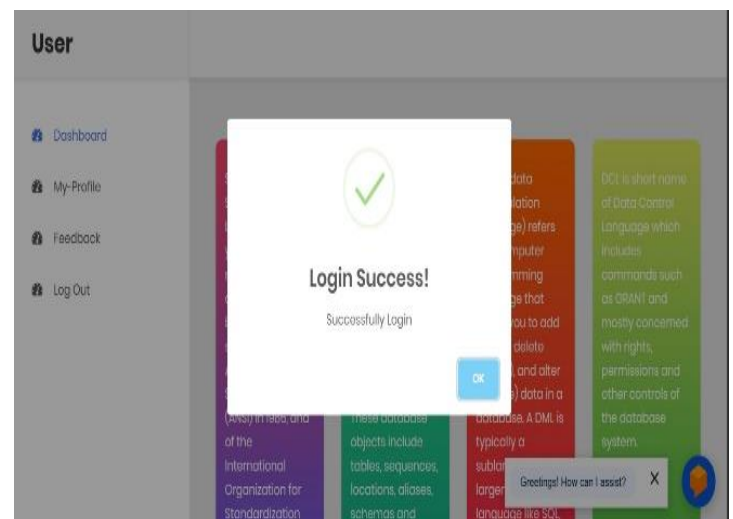
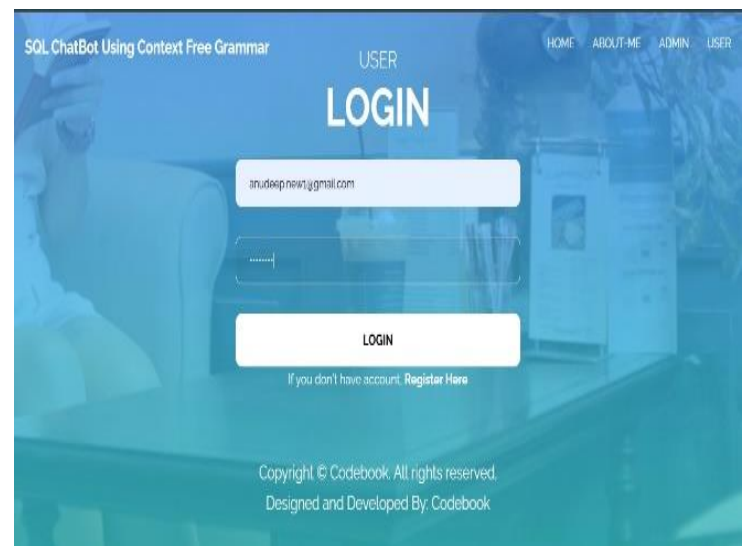
The SQLBOT system can be implemented using the following technologies:

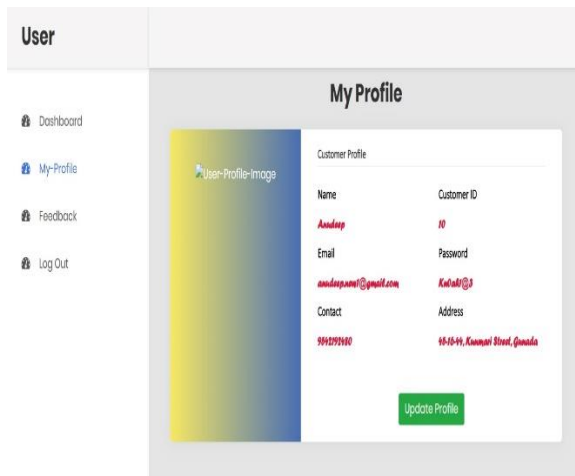
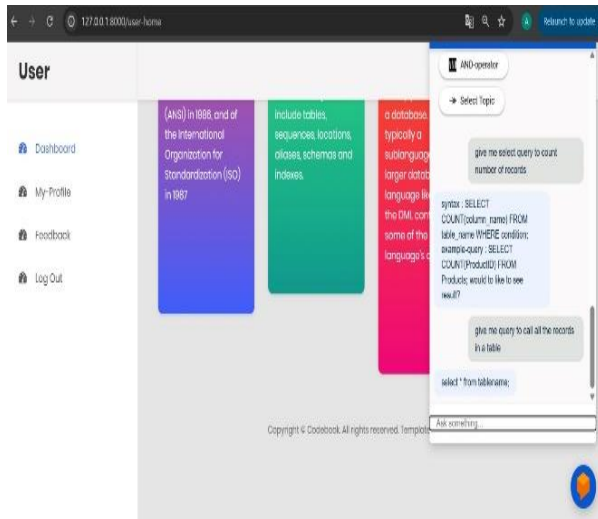
Component	Technology Used
Programming Language	Python
NLP Processing	NLTK / SpaCy
Chat Interface	Flask / Web Interface
Database	MySQL / SQLite
Grammar Parser	CFG Parser
Query Execution	SQL Engine

The chatbot interface accepts user queries and processes them using NLP techniques. The CFG parser checks the query structure before executing it in the database. The system returns query results along with explanations when necessary.

VI. RESULTS AND DISCUSSION

The performance of SQLBOT was evaluated by testing different natural language queries and verifying the correctness of generated SQL statements.





Example test cases:

User Input	Generated SQL Query
Show all students	SELECT * FROM Students
Show students with marks > 80	SELECT * FROM Students WHERE Marks > 80
List student names	SELECT Name FROM Students

The results show that SQLBOT successfully converts natural language queries into valid SQL statements with minimal syntax errors. The use of **Context-Free Grammar ensures proper query structure**, which improves the reliability of the system. Additionally,

the chatbot provides an interactive learning environment for users who are new to SQL.

VII. CONCLUSION

This paper presented **SQLBOT**, an AI-powered chatbot designed to assist users in generating SQL queries using Context-Free Grammar. The system enables natural language interaction with databases while ensuring syntactic correctness through grammar-based validation. SQLBOT reduces the complexity of writing SQL queries and provides an effective learning tool for beginners. The integration of NLP and CFG improves query accuracy and enhances user experience in database interaction.

FUTURE WORK

Future improvements to SQLBOT may include:

- Support for more complex SQL queries such as JOIN, GROUP BY, and nested queries
- Integration with voice-based assistants
- Use of machine learning models for better intent detection
- Multi-database support (PostgreSQL, Oracle, MongoDB)
- Advanced query optimization and error correction features

REFERENCES

1. A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, Pearson, 2007.
2. R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Pearson, 2016.

3. T. Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," MIT, 1971.
4. Jurafsky, D., and Martin, J., *Speech and Language Processing*, Pearson, 2019.
5. A. Gupta and R. Kumar, "Natural Language Interface for Databases," International Journal of Computer Applications, 2018.