Lex Programming

SQL to MongoDB Translator Using Lex

Introduction

The project aims to create a Lex-based translator that converts simple SQL queries into MongoDB queries. The system reads an SQL query from an input file, processes the query using lexical rules defined in Lex, and outputs the corresponding MongoDB query into an output file.

Lexical Analysis

The Lex program reads and tokenizes SQL queries. It recognizes SQL keywords, operators, and symbols, and converts them into their MongoDB equivalents. The rules and tokens defined include SQL keywords (SELECT, FROM, WHERE, etc.), operators (>, <, =, etc.), and identifiers (table names, column names).

Example Translational rules:
SQL Keywords:

- SELECT → Translates to db.collection.find().
- FROM → Identifies the collection name (table name).
- WHERE → Translates to .filter().

File Handling

- The input SQL query is read from an input file (sql.txt).
- The output MongoDB query is written to an output file (mongo.txt).

Conclusion

This project demonstrates the use of Lex in building a simple SQL-to-MongoDB translator. The implementation successfully reads SQL queries, applies lexical rules for translation, and writes MongoDB queries to an output file. Future work can expand the system's capabilities to support more SQL features and provide better error handling and user interaction.

Mytreyan Jp
2022115102