PARSER FOR SQL

SUBMITTED BY:

NAME	USN
PRATYUSH OAK	1PI10IS129
SAMARTH BS	1PI10IS092
SRUJAN SANTHOSH	1PI10IS104

INCHARGE:

PROF. DHEERAJ

ABSTRACT

WHAT IS PARSER?

In computing, a parser is one of the components in an interpreter or compiler that checks for correct syntax and builds a data structure(often some kind of parse tree,abstract syntax tree or other hierarchical structure) implicit in the input tokens. The parser often uses a separate lexical analyzer to create tokens from the sequence of input characters. Parsers may be programmed by hand or may be (semi-)automatically generated (in some programming languages) by a tool.

The most common use of a parser is as a component of a compiler or interpreter. This parses the source code of a computer programming language to create some form of internal representation. Programming languages tend to be specified in terms of a context free grammar because fast and efficient parsers can be written for them. Parsers are written by hand or generated by parser generators.

WHAT IS SQL:

SQL or Structured Query Language) is a special-purpose programming language designed for managing data in relational database management systems (RDBMS).

Originally based upon relational algebra and tuple relational calculus, its scope includes data insert, query, update and delete, schema creation and modification, and data access control.

CONSTRUCTS:

1.create
2.alter
3.drop

INTRODUCTION

The purpose of this project was to create a basic SQL parser,that would parse some very basic SQL queries.A parser is a syntax-checker,which is a phase of a compiler,that checks if the input string is syntactically correct according to the rules of the language.In this project, a parser has been implemented using a Deterministic Finite Automata.

Input data can only be parsed after an initial phase of lexical analysis.The project also include this phase,however at a much superficial level.This phase of the project behaves as a tokenizer, and tokenizes the input string.

The parsing phase uses the DFA to parse the tokenized string.Hence the parser can verify if the input string will be accepted or rejected according to whether it will land on a final state of the DFA.

Therefore this project basically accepts from the user a string,(SQL query) and parses it to determine whether its valid SQL query or not.This project is an earnest effort to recreate a basic SQL parser for some selected queries.

The 'CREATE','ALTER' and 'DROP' queries are supported by this parser.

REQUIREMENTS AND SPECIFICATIONS

TOOLS USED: The FLEX tool,which is the fast lexical analyzer generator was used to perform the lexical analysis on the input.

OPERATING SYSTEM: The project was developed on the MAC OS X operating system.which is a Linux based OS.Hence the project should work seamlessly on any other Linux based OS.

LANGUAGE USED: The project was coded in C.

IMPLEMENTATION

CONSTRUCTS: This project was developed to support the CREATE,ALTER and DELETE constructs of SQL.

INPUT: The input is accepted from the user.It is an SQL query that needs to be parsed.The input is basically a query that a user who is interacting with a database would enter.This query,if valid,will be accepted.

OUTPUT: The output generated includes a token table that lists each individual word/lexeme along with its corresponding token and token id.Then the input string is converted to a string of thokens to be parsed and this string of tokens is displayed. Finally the it is displayed whether the input string was a valid SQL CREATE,ALTER or DROP query.

LANGUAGE OF IMPLEMENTATION: The parser was implemented in C.The C code also uses the FLEX tool and hence the code is stored in a '.l' file,which on compilation will be created into a C file.

SYNTAX:

CREATE:

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
)
```

Example:

```
CREATE TABLE Persons
(
P_Id integer,
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)
```

ALTER:

```
ALTER TABLE table_name
ADD column_name datatype

ALTER TABLE table_name
MODIFY column_name datatype
```

Example:

```
ALTER TABLE Persons
ADD DateOfBirth date
```

DROP:

```
DROP TABLE table_name

DROP DATABASE database_name
```

Example:

```
DROP TABLE Employee
```

CONCLUSION

This project is thus a parser for the **CREATE,ALTER** and **DROP SQL** queries.The project initially posed some difficulties on how the input had to be parsed.There were many options such as using algorithms for Top down or Bottom up parsing.However,DFA has been used for parsing.SQL is a case insensitive language however,the parser in the project is not completely case insensitive.The parser generated correct results for almost all the input cases(for the give constructs).

ACKNOWLEDGEMENTS

We would like to thank the all the teachers in the compiler design lab for their support.We would also like to thank the HOD of the ISE deparment for the use of department facilities.We would like to thank all our classmates for their help and cooperation.
And finally a big thank you to Dheeraj sir for all the help,support and patience he provided us.