

Manual Técnico del Analizador Léxico de SQL

Descripción del Proyecto

El analizador léxico y sintáctico tiene como objetivo reconocer y tokenizar la sintaxis de SQL. Este analizador convierte las cadenas de texto SQL en una serie de tokens que pueden ser interpretados por el analizador sintáctico.

Requisitos del Sistema

Para compilar y ejecutar este proyecto, se requiere:

- **Java Development Kit (JDK):** Versión 17 o superior.
- **JFlex:** Versión 1.9.1.

Estructura del Código

El código fuente se organiza en una clase principal que utiliza JFlex para definir el analizador léxico. A continuación se detallan las secciones clave:

Variables y Métodos

```
int linea = 1;
int columna = 1;
ArrayList<Token> tokens = new ArrayList<>();
```

- **línea:** Contador para las líneas en el texto de entrada.
- **columna:** Contador para las columnas, que se reinicia al detectar un salto de línea.
- **tokens:** Lista que almacena los tokens generados.

```
public void manejarToken(TipoToken tipo, String valor) {
    tokens.add(new Token(valor, linea, columna, tipo));
}
```

- Método para agregar un nuevo token a la lista con su tipo, valor, línea y columna.

Definición de Tokens

Las definiciones de tokens se agrupan en diferentes categorías, como palabras clave, identificadores y números. A continuación se detallan estas categorías.

Explicación de las Expresiones Regulares

Palabras Clave de SQL

```
"CREATE"           { manejarToken(TipoToken.CREATE, yytext()); }  
// ...  
"JOIN"            { manejarToken(TipoToken.JOIN, yytext()); }
```

Cada línea define una palabra clave de SQL. Cuando se encuentra una coincidencia, se llama a `manejarToken()` con el tipo correspondiente.

Identificadores

```
[a-z_][a-z0-9_]*   { manejarToken(TipoToken.IDENTIFICADOR,  
yytext()); }
```

Coincide con identificadores en formato `snake_case`.

Números Enteros y Decimales

```
[0-9]+             { manejarToken(TipoToken.ENTERO, yytext()); }  
[0-9]+\.[0-9]+     { manejarToken(TipoToken.DECIMAL_VAL,  
yytext()); }
```

Define cómo se representan los números en SQL.

Fechas

```
" "[0-9]{4}"-"[0-9]{2}"-"[0-9]{2}" " {  
manejarToken(TipoToken.FECHA, yytext()); }
```

Coincide con fechas en el formato `YYYY-MM-DD`.

Cadenas de Texto

```
"'\"([^\\"|\\'|\\\\\\\\)*\"'" { manejarToken(TipoToken.CADENA,
yytext()); }
```

Coincide con cadenas de texto entre comillas simples.

Operadores y Signos

Define cómo se representan los operadores aritméticos, relacionales y lógicos en SQL.

Comentarios de Línea

```
"--"["^\n"]* { manejarToken(TipoToken.COMENTARIO_LINEA,
yytext()); nuevaLinea(); }
```

Maneja los comentarios de línea que comienzan con --.

Espacios y Nuevas Líneas

```
[ \t\r]+ { /* Ignorar espacios y tabs */ }
\n { nuevaLinea(); }
```

Ignora espacios y tabulaciones y maneja nuevas líneas.

Uso del Analizador

Para utilizar el analizador léxico, se debe instanciar la clase generada y pasarle el texto SQL que se desea tokenizar. El método `getTokens()` devolverá la lista de tokens generados.

Ejemplo de Uso

java

Copy code

```
AnalizadorSQL analizador = new AnalizadorSQL(input);
analizador.yylex(); // Iniciar el proceso de análisis
ArrayList<Token> tokensGenerados = analizador.getTokens();
```