MANUAL TÉCNICO

Información del sistema:

- • OS: Ubuntu 22.04.2 LTS
- • OS Type: 64-bit
- • CPU: Intel core i5 2.50 GHz
- • GPU: Mesa Intel HD Graphics 4400
- • Versión de java: 17.0.8
- • RAM: 8 GB
- • IDE: IntelliJ idea
- • Control de versiones: git 2.34.1
- • Github: https://github.com/Hatsune02/practice1-Compi.git

Descripción del proyecto:

SQL (Structured Query Language) es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una base de datos relacional almacena la información en tablas, las tablas son los objetos de base de datos que se comportan como contenedores de información, en los cuales la información será lógicamente organizada en formato de filas y columnas. Cada fila es considerada como una entidad que es descrita por las columnas que contienen los atributos de la entidad.

Podemos decir que SQL es un lenguaje interpretado de instrucciones en donde se pueden ejecutar varias instrucciones a la vez y donde un punto y coma indica el fin de una instrucción y se presiona la tecla [Enter] para confirmar la o las instrucciones. Al ser confirmada una instrucción, el intérprete revisa que los caracteres sean válidos (análisis léxico) y posterior a ello se verifica que tengan la estructura correcta (análisis sintáctico) para así ejecutar la tarea relacionada.

Para crear nuestro emulador de instrucciones SQL necesitamos nuestras expresiones regulares para tokens y producciones de gramática solidos para que no tengamos ningún problema.

Gramática Léxica.

KeyWords

- SELECCIONAR
- FILTRAR
- INSERTAR
- ACTUALIZAR
- ASIGNAR
- ELIMINAR
- VALORES
- EN

Operaciones Relacionales (OR)

- =
- <
- >
- <=
- >=
- <>

Operaciones logicas (OL)

- AND
- OR

Coma

- “ , ”

Punto

- “ . ”

Cierre de commandos (PC)

- “ ; ”

Cadenas de texto (strings STR)

- \"([^\"])\"

All (toda la tabla)

- *

Identificadores (Id)

- [a-zA-Z0-9_@+#*-]+

Numeros (NUM)

- [0-9]+

Path

- { **Identificadores** }("."{ **Identificadores** })+

Gramática Sintáctica

Terminales: (P_COMA, COMA, ASTERISCO, AND, OR, SELECCIONAR, FILTRAR, INSERTAR, ACTUALIZAR, ASIGNAR, ELIMINAR, EN, VALORES, CADENA, EQUAL, PARENT_1, PARENT_2, ID, REL_OP, DIGIT, PATH**)**

No Terminales: (instrucciones, instruccion, seleccionar, insertar, actualizar, eliminar, expr, listV, listC, and, or, conditionsP, conditions, filter, asignar**)**

instrucciones ::= instruccion

| instrucciones instruccion

;

instruccion ::= seleccionar P_COMA

| insertar P_COMA

| actualizar P_COMA

| eliminar P_COMA

seleccionar ::= SELECCIONAR ASTERISCO EN PATH filter

| SELECCIONAR listC EN PATH filter

insertar ::= INSERTAR EN PATH VALORES PARENT_1 listV PARENT_2

| INSERTAR EN PATH PARENT_1 listC PARENT_2 VALORES PARENT_1 listV PARENT_2

actualizar ::= ACTUALIZAR EN PATH ASIGNAR asignar filter

asignar ::= ID EQUAL expr

| asignar COMA ID EQUAL expr

eliminar ::= ELIMINAR EN PATH filter

filter ::= FILTRAR conditions

|

conditions ::= ID EQUAL expr conditionsP

| ID REL_OP expr conditionsP

conditionsP ::=

| AND ID EQUAL expr and

| AND ID REL_OP expr and

| OR ID EQUAL expr or

| OR ID REL_OP expr or

and ::=

| AND ID EQUAL expr and

| AND ID REL_OP expr and

or ::=

| OR ID EQUAL expr or

| OR ID REL_OP expr or

listV ::= expr

| listV COMA expr

listC ::= ID

| listC COMA ID

expr ::= DIGIT:

| CADENA: