**Curso:** Compiladores

**Semestre:** 2 - 2022

**Facilitador:** Wider Farid Sánchez Garzón

**Tema:** Prueba parcial

Llevar a cabo las siguientes tareas:

1. Identificar conjunto de reglas léxicas del lenguaje
2. Crear el diagrama de transiciones que permita construir un analizador léxico para el siguiente lenguaje SQL (En las palabras en negrilla no importan las mayúsculas, minúsculas o combinación de ambas)
3. Compilador implementado que acepte:
   1. Ingreso de programa de entrada por consola o a través de la carga de un archivo.
   2. Carga del programa de entrada a la caché para disponerlo de esta manera para el análisis.
4. Simular la formación de componentes léxicos para el programa de muestra que se adjunta:



**SELECT** <campos> **FROM** <tablas> [**WHERE** <condiciones>] [**ORDER BY** <ordenadores>]

**Dónde:**

* Los corchetes ([….]) son sólo una guía para indicar que ésta sección es opcional dentro de la sentencia SQL, pero cuando se presente, debe tener todo el contenido que se escriben dentro de los mismos.
* <campos>: Los campos tienen la siguiente estructura:
  + <campo> [ , <campos>]
* <campo>: Un campo es todo aquello que siga la siguiente estructura de formación
  + CAM\_<nombre>
  + Su categoría es **CAMPO**
  + Ejemplo de campos válidos:
    - CAM\_Campo1
    - Cam\_Campo\_2
    - CaM\_\_\_\_\_\_Campo\_\_\_\_\_
    - cAm\_Otro\_Campo
  + Ejemplo de campos inválidos:
    - CAMP\_Campo1
    - CAMPO\_Campo\_2
    - CAM\_\_ nombre campo
    - CAM\_Un$Campo
* <tablas>: Las tablas tienen la siguiente estructura:
  + <tabla> [ , <tablas>]
* <tabla>: Una tabla es todo aquello que siga la siguiente estructura de formación
  + TAB\_<nombre>
  + Su categoría es **TABLA**
  + Ejemplo de tablas válidos:
    - TaB\_Nombre\_Tabla
    - tAB\_2\_Otra\_Tabla
    - TAb\_Otra\_Tabla\_\_\_\_\_\_
    - TAB\_NombreTabla34
  + Ejemplo de tablas inválidos:
    - TaBla\_Nombre\_Tabla
    - tA\_2\_Otra\_Tabla
    - TAb\_ Otra Tabla
    - TABl\_NombreTabla 34
* <nombre>: Un nombre está conformado sólo por letras (Mayúsculas o Minúsculas), dígitos y/o guiones bajos (\_). No importa su longitud. No tiene categoría, ya que se combina con otros componentes léxicos para obtener la verdadera categoría a la que pertenece.
* <literal>: Un literal tiene la siguiente estructura (comilla simple que abre, contenido, comilla simple que cierra):
  + ‘<contenido>’
  + Su categoría es **LITERAL**
  + Ejemplo de literales válidos:
    - ‘Esto es un literal donde se muestra un @ y 12345’
    - ‘¿Esto es una pregunta?’
    - ‘Dirección: calle 56 # 98-98 int. 207’
* <contenido>: Un contenido hace referencia a cualquier combinación de caracteres. No tiene categoría, ya que se combina con otros componentes léxicos para obtener la verdadera categoría a la que pertenece.
* <numero>: Un número puede ser <numero\_entero> o <numero\_decimal>.cualquier número entero o cualquier número decimal, cuyo separador de la parte flotante es el punto (.)
* <numero\_entero>: Número entero:
  + Su categoría es **NUMERO ENTERO**
  + Ejemplo de números enteros válidos:
    - 45
* <numero\_decimal>: Número decimal, cuyo separador de la parte flotante es el punto (.)
  + Su categoría es **NUMERO DECIMAL**
  + Ejemplo de números decimales válidos:
    - 45.65
    - 0.33
  + Ejemplo de números decimales inválidos:
    - .33
    - 44.
* <condiciones>: Las condiciones tienen la siguiente estructura:
  + <operando> <operador> <operando> [<conector> <condiciones>]
* <operando>: Los operandos válidos pueden ser:
  + Un <campo>
  + Un <literal>
  + Un <numero>
* <operador>: Los operadores válidos son:

|  |  |
| --- | --- |
| **Operador** | **Categoría** |
| > | MAYOR QUE |
| < | MENOR QUE |
| = | IGUAL QUE |
| >= | MAYOR O IGUAL QUE |
| <= | MENOR O IGUAL QUE |
| <> | DIFERENTE QUE |
| != | DIFERENTE QUE |

* <conector>: Los conectores válidos son:

|  |  |
| --- | --- |
| **Conector** | **Categoría** |
| AND | CONECTOR Y |
| OR | CONECTOR O |

* <ordenadores>: Los ordenadores tienen la siguiente estructura:
  + <campos>[ <criterio>] o <indices>[ <criterio>], sólo una de ellas puede estar o los campos o los índices, pero no los dos.
* <indices>: Los indices tienen la siguiente estructura:
  + <numero\_entero> [ , <indices>]
* <criterio>: Los conectores válidos son:

|  |  |
| --- | --- |
| **Conector** | **Categoría** |
| ASC | ASCENDENTE |
| DESC | DESCENDENTE |

* Acepta los siguientes caracteres sobre el archivo leído y los interpreta:

|  |  |
| --- | --- |
| **Carácter especial** | **Categoría** |
| @FL@ | SALTO LINEA |
| @EOF@ | EOF |