



# Lenguajes formales y autómatas:

## Trabajo práctico grupal

### 1. Enunciado

Se debe escribir un programa que genere un parser capaz de analizar una sentencia SQL de tipo SELECT y extraer de la misma los siguientes elementos:

- listado de tablas utilizadas
- columnas utilizadas por cada una de las tablas, independientemente de la cláusula en la que se la utilice

Las particularidades que debe contemplar la consulta de entrada se describen a continuación:

**Consulta:** SELECT

**Cláusulas opcionales a ser contempladas:**

- FROM (puede contener uno o más INNER JOIN y/o LEFT JOIN, y la opción AS como alias para las tablas)
- WHERE (puede contener consultas anidadas IN)
- GROUP BY
- HAVING
- ORDER BY

**Funciones a ser contempladas (~~en las cláusulas que permiten utilizarlas~~):**

- MIN
- MAX
- COUNT (podría utilizarse el operador DISTINCT)

**Restricciones y aclaraciones:**

- Solo se deben contemplar comprobaciones contra tipos de datos numéricos o cadenas de texto.
- Contemplar los siguientes operadores de comparación: <, =, >, <=, >=, <>.
- Contemplar los siguientes operadores booleanos: AND y OR.
- No se deberá contemplar la utilización de \* como abreviación de todas las columnas existentes.
- Cada columna será utilizada a cada momento anteponiendo el alias o nombre de la tabla correspondiente.
  - Ejemplos válidos:
    - select personas.nombre from personas;



- select p.nombre from personas as p;
- Ejemplo inválido:
  - select personas.nombre from personas as p;
- La definición del alias de cada tabla puede o no contener la palabra reservada AS.
- Se asume que las sentencias deberían ser ejecutadas contra una BBDD de tipo SQL Server (T-SQL).
- En la sección del select solo pueden aparecer campos o las funciones listadas en "Funciones a ser contempladas"
- Las cadenas se delimitan con comillas simples.

## 2. Consignas

- a. Utilizando la herramienta PLY, escriba el código necesario correspondiente a los módulos lex (tokens) y yacc (reglas) que codifiquen una gramática capaz de reconocer la sentencia requerida en el enunciado. (60 puntos)
- b. Incorporar al código desarrollado en el punto previo la lógica necesaria para obtener las tablas y columnas utilizadas en la consulta. (30 puntos)
- c. Desarrolle una función denominada **parse\_select\_statement** que reciba como parámetro una consulta SELECT y devuelva como resultado un diccionario. Dicho diccionario tendrá como claves los nombres de las tablas y como valor el listado de columnas ordenado alfabéticamente. (10 puntos)

## Generalidades

- El trabajo se aprueba con 80 puntos.
- Fecha límite de entrega: **11/06/2021 23:59**
- Coloquio: **15/06/2021**
- Todo el código desarrollado debe incluirse en un archivo denominado grupox.py donde xx es el número de grupo correspondiente.
- El código entregado no debe imprimir ningún valor en pantalla (excluir sentencias **print**).
- Forma de entrega: correo electrónico ([marianoferrero.mf@gmail.com](mailto:marianoferrero.mf@gmail.com) y [arielrossanigo@gmail.com](mailto:arielrossanigo@gmail.com)) incluyendo un enlace a un repositorio de código público (GitHub, BitBucket o GitLab)
- Se debe utilizar Python 3
- Número máximo de integrantes por grupo: 3
- Número mínimo de integrantes por grupo: 2



**Universidad Católica de Santiago del Estero**

Departamento Académico Rafaela

Bv. Hipólito Irigoyen 1502 - Rafaela - Santa Fe

Teléfono: 03492 - 432832/433408/433550

Email: [ucsedar@ucse.edu.ar](mailto:ucsedar@ucse.edu.ar)

---

- Los integrantes de cada grupo se deben comunicar antes del **23/04/2021** en el foro de la materia. El orden en el que se envíe esa comunicación definirá el número de grupo (ejemplo: el primero en enviar será automáticamente el grupo 1, y así sucesivamente).