

SimpleQL

Objetivos

- General

Entender y aplicar los conceptos básicos de los lenguajes de programación, así como con los conceptos de los paradigmas a los que estos pertenecen.

- Específicos

1. Implementar una solución de software utilizando Python como lenguaje.
2. Implementar una solución en consola que utilice el paradigma declarativo.
3. Crear una aplicación con la capacidad de cargar información en formato json y entregar reportes atractivos utilizando html.

Descripción

SimpleQL es un lenguaje de consultas que funciona únicamente a nivel de consola, su propósito es facilitar al usuario la búsqueda de registros completos en archivos json, en los que buscar registro por registro podría ser muy tedioso y cansado.

No es el objetivo de SimpleQL convertirse en una versión de SQL, en su lugar, SimpleQL funciona como una versión minimalista con algunas similitudes que permiten al usuario cargar información a memoria por medio de comandos y obtener algunos datos generales acerca de esta, como el número de registros, el valor máximo de un atributo o incluso un reporte de en html de un conjunto de registros.

*Nota: Los comandos y campos serán case insensitive.

**Nota: Los comandos serán ejecutados únicamente en consola, uno por uno, por lo que no habrá carga de archivos bash ni nada similar.

Comandos

1. Cargar:

Este comando permitirá la carga de diferentes archivos a memoria, el único parámetro que lo conforma es una lista de direcciones a los archivos que cargará a memoria. *Nota: Todos los archivos tendrán la misma estructura, la misma se detalla más adelante.

Estructura:

CARGAR archivo1, archivo2, archivo3, archivoN

2. Seleccionar

Permite seleccionar uno o más registros o atributos de los mismos con base en condiciones simples que pueden aplicarse a los atributos de los mismos.

Estructura:

SELECCIONAR nombre, edad, promedio, activo DONDE nombre = "Francisco"

SELECCIONAR *

SELECCIONAR nombre, edad DONDE promedio = 14.45

*Nota: Si se utiliza * en lugar de nombrar los atributos uno por uno, entonces automáticamente se seleccionan todos los atributos. Puede solicitarse cualquier combinación de atributos, siempre y cuando no se repitan y pertenezcan a los atributos listados en el archivo JSON, es decir, nunca se podrá solicitar un atributo "ubicacion", porque el mismo no existe.

*Condiciones: Las condiciones serán sencillas, únicamente de igualdad, solo es posible calcular una selección con una condición a la vez, es decir, nunca se podrá realizar una selección donde la condición sea promedio = 0.0 & nombre = "Jose", solamente un atributo por selección.

*Nombres: Para utilizar un nombre en una condición este siempre debe venir dentro de comillas.

3. Maximo

Permite encontrar el valor máximo que se encuentre en el atributo de uno de los registros del conjunto en memoria.

Estructura:

MAXIMO edad

MAXIMO promedio

*Nota: Las listadas arriba son las únicas opciones, no existe ninguna otra combinación para el comando Maximo.

4. Minimo

Permite encontrar el valor mínimo que se encuentre en el atributo de uno de los registros del conjunto en memoria.

Estructura:

MINIMO edad

MINIMO promedio

*Nota: Las listadas arriba son las únicas opciones, no existe ninguna otra combinación para el comando Minimo.

5. Suma

Permite obtener la suma de todos los valores de un atributo especificado en el comando.

Estructura:

SUMA edad

SUMA promedio

*Nota: Las listadas arriba son las únicas opciones, no existe ninguna otra combinación para el comando Minimo.

6. Cuenta

Permite contar el número de registros que se han cargado a memoria.

Estructura:

CUENTA

7. Reportar

Este comando permite crear un reporte en html que contiene N cantidad de registros.

Estructura:

REPORTAR N

*Donde N es un número que representa la cantidad de registros que se deben tener en el reporte.

Estructura del Archivo JSON

Puede encontrar en el siguiente enlace un [ejemplo](#) sencillo de cómo estará conformado el archivo json, la estructura de cualquier archivo json a utilizar será la misma sin variación alguna.

Entregables

- Manual de Usuario en formato .md, este debe ser escrito en el archivo README del repositorio. El mismo debe enseñar a una persona a utilizar la aplicación desde cero.
- Código Fuente
- Manual Técnico, con una explicación clara del flujo de su aplicación.

Consideraciones

- La aplicación debe ser realizada utilizando Python como lenguaje de programación.
- El reporte en html debe ser entendible y atractivo a la vista.
- La práctica debe ser desarrollada en forma individual.
- La entrega se realizará por medio de un repositorio localizado en Github, el link al mismo se entregará en Uedi.
- No se permite la modificación de código durante la calificación.
- Copias totales o parciales tendrán una nota automática de 0 puntos y serán notificadas a la escuela de Ciencias y Sistemas.
- No se dará prórroga para la entrega de la práctica.
- **La entrega de la práctica será el 02 de Septiembre, a más tardar a las 23:59.**