**Introducción**

Para el año 2021 en la historia de Venezuela sé llevaría a cabo la tercera reconversión implementada bajo decreto No. 4.553, mediante el cual el Ejecutivo Nacional decretó la nueva expresión monetaria. Dicho decreto fue publicado en la Gaceta Oficial No. 42.185 del 6 de agosto de 2021.

El siguiente procedimiento intenta describir de manera general los pasos a seguir para lograr aplicar una función matemática y eliminar seis (6) ceros a campos numéricos en tablas de dos tipos de bases de datos distintos, una de ella PostgreSQL y la otra Microsoft SQL Server. Cabe destacar que a través del uso del lenguaje de programación Python, fue la estrategia ideal a emplementar, para aplicar una función matemática a ambas bases de datos.

Este procedimiento más el documento fue escrito y desarrollado por mi persona, quien estuvo a cargo además del desarrollo del programa en python para actualizar los valores en campos numéricos de ambas base de datos SQL Server y PostgreSQL.

**Procedimiento para reconvertir las bases de datos de SQL Server**

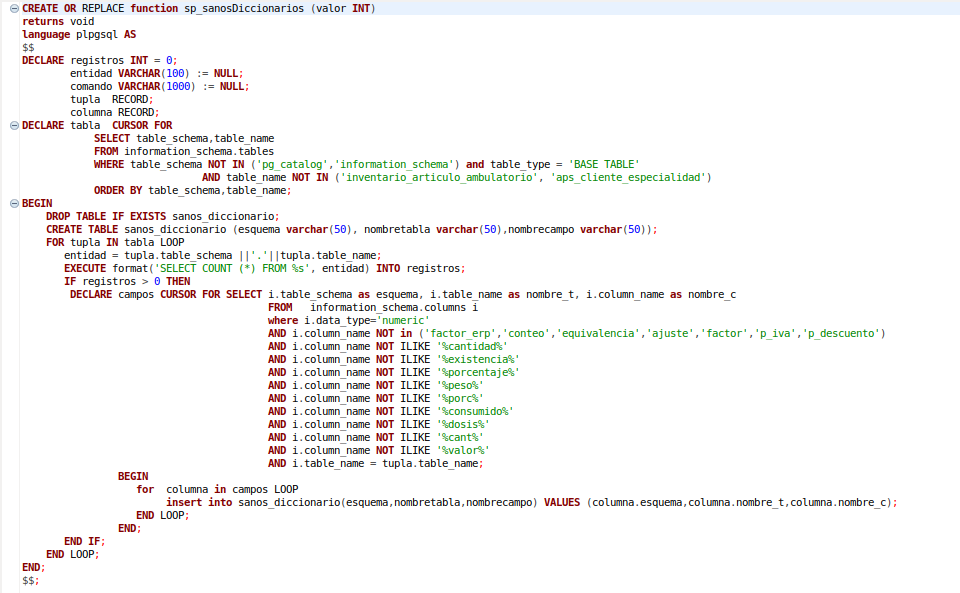
**Base de Datos PostgreSQL**

**Paso 0. (Preparación)**

* Crear copia de seguridad de la base de datos
* Crear un duplicado de la base de datos sanos : CREATE DATABASE sanos\_bs WITH TEMPLATE sanos OWNER postgres; (Opcional)
* Dar permiso al usuario **reconversion\_user** en **pg\_hba.conf**
* **Ejecutar el programa de reconversión (python) con privilegios root**
* **Editar y colocar el nombre de la base de datos en el programa python**

**Paso 1.** Crear función sp\_sanosDiccionarios en la base de datos PostgreSQL

La función analiza la metadata, tablas y campos de la base de datos, realizando inicialmente sobre las tablas un conteo del número de registros, excluyendo las tablas sin registro, luego hará un análisis sobre los campos de tipo numérico, creando una tabla con estos campos, llamada **<nombre\_tabla>\_diccionario**, este store procedure tiene la función de excluir tablas y campos.



**Paso 2.** Ejecutar la función : SELECT sp\_sanosDiccionarios (1);

**Paso 3.** Verificar la creación de la tabla :

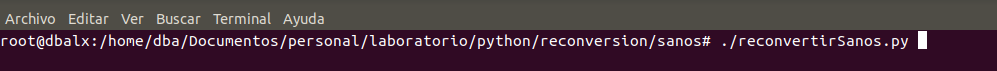
SELECT \* FROM sanos\_diccionario;

**Paso 4.** Otorgar permisos al usuario **reconversion\_user** sobre toda la base de datos

**Paso 5.** Desactivar triggers



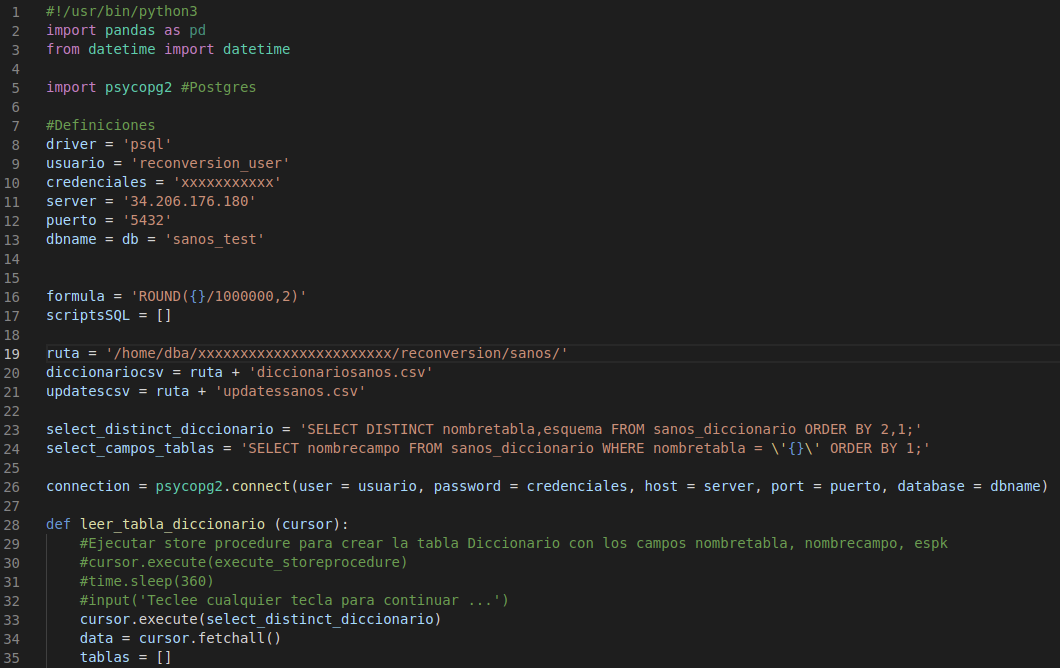
**Paso 6.** Ejecutar el programa **reconvertirSanos.py** para iniciar el proceso de reconversión. Desde un terminal de linux y usuario con privilegio root.



**Paso 7.** Al culminar el proceso, se debe habilitar los triggers



**Programa reconvertirSanos.py**



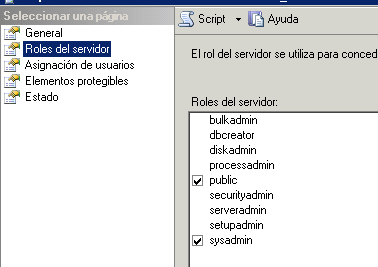
**Base de Datos Microsoft SQL Server**

**Paso 0. (Preparación)**

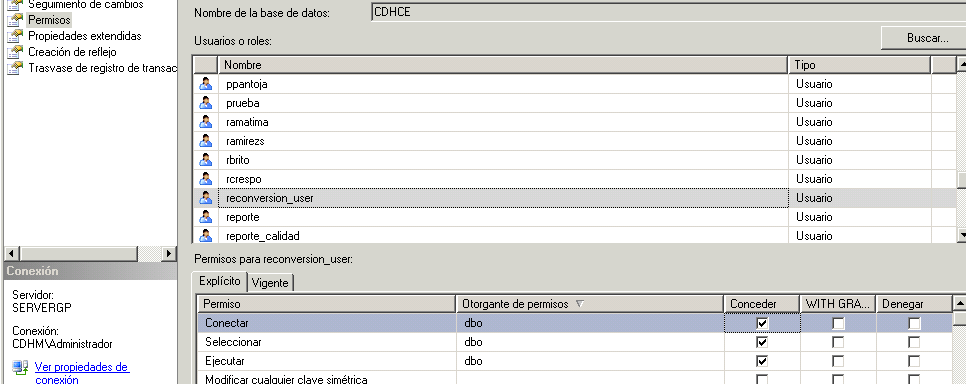
* Verificar que no existan las tablas **BDNDiccionario** y **CDHDiccionario** en la **tempdb** (Opcional)
* Crear copia de seguridad de la base de datos
* Dar permiso al usuario **reconversion\_user** en **BDNBS | CDHBS**
* **Ejecutar el programa de reconversión (python) con privilegios root**
* **Editar y colocar el nombre de la base de datos en el programa python**

**Asignación del usuario y roles de base de la base de datos**

**Asignación de roles de servidor**



**Asignación de permisos**



**Paso 1.** Crear un procedimiento almacenado en la base de datos a reconvertir

El procedimiento almacenado analiza los objetos, tablas y campos de la base de datos, realizando inicialmente sobre las tablas un conteo del número de registros, excluyendo las tablas sin registro, luego hará un análisis sobre los campos de tipo numérico, creando una tabla solo con estos campos, llamada **BDNDiccionario** o **CDHDiccionario** en la base de datos **tempdb**, el procedimiento almacenado tiene la facilidad de excluir tablas y campos.

**Paso 2.** Ejecución del procedimiento almacenado **spBDNDiccionario**



**Paso 3.** Ejecutar el programa python **reconvertirHigea.py** para iniciar el proceso de reconversión. Desde un terminal de linux y usuario con privilegio root.

****

**Paso 3.** Repetir los pasos para la siguiente base de datos a reconvertir

Programa reconvertirFundacion.py

