

Nota: este documento no pretende ser una palabra sagrada, es solo mi manera de pensar. Existen muchas formas de hacerlo e incluso los IDE son capaces de insertar los datos con un par de clicks, el motivo de este documento es instruir paso a paso una manera de lograr insertar un excel en una base de datos.

Link del código:

https://github.com/felipedelosh/DATA_VIH_CALDAS_CO

1 -> En este caso de estudio vamos a descargar la base de datos de mortalidad por VIH para CALDAS.

https://www.datos.gov.co/Salud-y-Protecci-n-Social/Mortalidad-VIH-2010-A-2016/yht4-tw4/about_data

2 -> Vamos a instalar la librería pandas para leer el .csv

Abrir la consola y escribir:

Python -m pip install pandas

Vamos a cargar el documento de la siguiente manera

Vamos a ingresar manualmente la ruta del .csv yo la tengo en la carpeta data

```
_ruta_excel = "DATA/Mortalidad_VIH_2010_A_2016_20240403.csv"
```

Vamos a importar pandas

```
import pandas as pd
```

Vamos a cargar los datos

```
dataframe = pd.read_csv(_ruta_excel)
```

Nota: una consulta insert de compone de la siguiente manera:

CREATE TABLE IF NOT EXISTS <NombreTabla>

```
(  
    nombre_col_0 TEXT,  
    nombre_col_1 TEXT,  
    nombre_col_2 TEXT  
    ...  
);
```

Se compone de 4 escenciales

Parte 1 la cabecera:

```
_SQL = f"CREATE TABLE IF NOT EXISTS {_nombre_tabla}(\n"
```

Parte 2 nombre de la tabla:

```
_nombre_tabla = "VIH"
```

Parte 3 los nombres de las columnas:

```
# Ver nombres de las cabeceras del dataframe  
✓ for columna in dataframe.columns:  
    _SQL = _SQL + f" {columna} TEXT,\n"
```

Parte 4 borrar la última coma y cerrar el query

```
# Borrar la ultima coma
_SQL = _SQL[:-2]

# Poner el cierre de consulta
_SQL = _SQL + "\n);"
```

Y si imprimimos la consulta podemos verla con todo como texto:

```
print(_SQL)
```

```
CREATE TABLE IF NOT EXISTS VIH(
  COD_DPTO TEXT,
  COD_MUNIC TEXT,
  A_DEFUN TEXT,
  COD_INSP TEXT,
  SIT_DEFUN TEXT,
  OTRSITIODE TEXT,
  COD_INST TEXT,
  NOM_INST TEXT,
  TIPO_DEFUN TEXT,
  FECHA_DEF TEXT,
  ANO TEXT,
  MES TEXT,
  HORA TEXT,
  MINUTOS TEXT,
  SEXO TEXT,
  FECHA_NAC TEXT,
  EST_CIVIL TEXT,
  EDAD TEXT,
```

...

Ahora vamos a crear los insert

Este es un ejemplo de un insert:

```
INSERT INTO <tableName>  
  
(col_a, col_b,...col_z)  
  
VALUES  
  
(data, data, data);
```

Se compone de 3 partes

Parte 1 de 3

la cabecera del insert "Insert into nombreTabla values ("

Esto lo podemos lograr de forma estática:

```
_HEAD_INSERT_SQL = f"INSERT INTO {_nombre_tabla} ("
```

Parte 2 de 3

Poner las columnas: lo podemos hacer mediante un for iterando sobre la cabecera del .csv con un for y aludiendo una coma

```
for columna in dataframe.columns:  
    _HEAD_INSERT_SQL = _HEAD_INSERT_SQL + columna + ","
```

Nota importante: si lo dejamos hasta ahí podremos ver que termina en col_a,.. col_z, esa última coma nos ocasiona un error por ello tenemos que borrarla:

```
# Borrar la ultima coma
_HEAD_INSERT_SQL = _HEAD_INSERT_SQL[:-1]
```

Y luego de ello tenemos que cerrar la parte del nombre de las columnas con un) y pegarle la palabra reservada VALUES

```
# Poner cierre de parentesis y keyword values
_HEAD_INSERT_SQL = _HEAD_INSERT_SQL + ") VALUES "
```

hasta el momento solo hemos hecho lo siguiente:

```
INSERT INTO <tableName>
(col_a, col_b,...col_z)
VALUES
```

Parte 3 de 3

Tenemos que tener las siguientes consideraciones:

- Siempre empieza por paréntesis que abre.
- Como todo es texto tiene que empezar y terminar con comilla simple
- Los datos nulos o faltantes los rellenamos con dos comillas simple
- Tenemos que eliminar la última coma
- Siempre termina con);
- Guardar los valores con su respectiva cabecera.

```
# Vamos a construir el resto de la consulta con un metodo
def poner_datos_en_query(cabecera_sql, datos):
    _VALUES = "("
    for i in datos:
        if str(i) == "nan":
            _VALUES = _VALUES + "'\''" + ","
        else:
            _VALUES = _VALUES + "'" + str(i) + "'" + ","

    # Borrar la ultima coma
    _VALUES = _VALUES[:-1]

    # Poner el final ) y ;
    _VALUES = _VALUES + ");"

    # Guardar
    _DATOS.append(cabecera_sql + _VALUES)
```

Y luego de hacer esto podemos ver la consulta con un print

```
94 | print(cabecera_sql + _VALUES)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

✓ TERMINAL

```
IDPERTET, IDPUEBIN, N_IDPUEBIN, CODPRES, CODPTORE, CODMUNRE, AREA_RES, BARRIOFAL, COD_LOCA, CODIGO, VEREDAFALL, SEG_SOC
IAL, IDADMISALU, IDCLASADMI, PMAN_MUER, CONS_EXP, MU_PARTO, T_PARTO, TIPO_EMB, T_GES, PESO_NAC, EDAD_MADRE, N_HIJOSV, N_
HIJOSM, EST_CIVM, NIV_EDUM, ULTCURMAD, EMB_FAL, EMB_SEM, EMB_MES, MAN_MUER, COMOCUHEC, CODOCUR, CODMUNOC, DIROCUHEC, LOC
ALOCUHE, C_MUERTE, C_MUERTEB, C_MUERTEC, C_MUERTEED, C_MUERTEE, ASIS_MED, N_DIR1, T_DIR1, M_DIR1, C_DIR12, N_ANT1
, T_ANT1, M_ANT1, C_ANT1, C_ANT12, N_ANT2, T_ANT2, M_ANT2, C_ANT2, C_ANT22, N_ANT3, T_ANT3, M_ANT3, C_ANT3, C_ANT32, N_PAT1
, T_PAT1, M_PAT1, C_PAT1, N_PAT2, C_PAT2, N_BAS1, C_BAS1, N_MCM1, C_MCM1, CAUSA_666, IDPROFCER, DD_EXP, MM_EXP, FECHA_EXP
, FECHAGRA, CAU_HOMOL, GRU_ED1, GRU_ED2, HORA_SE) VALUES ('17','1','Cabecera municipal','','Hospital/clínica','','
170,010,087,301','ESE HOSPITAL DEPARTAMENTAL SANTA SOFIA DE CALDAS','2','09/14/2012 12:00:00 AM','2,012','9'
,'10','0','Masculino','05/26/1956 12:00:00 AM','Estaba soltero(a)','56','Sin información ','99','No','','AG
RICULTORES DE CULTIVOS TRANSITORIOS','Otro','0','','170.0','Caldas','Aguadas','Rural disperso','','','EL
GUACO','Subsidiado','2.0','CAPRECOM - CAJA DE PREVISION SOCIAL DE COMUNICACIONES','Natural','Médico tratante
','','','','0','0','0','0','','','0','','','','','','','','','Historia clínica','Pruebas de laborat
orio','','','Si','NEUMONIA','8.0','3.0','J189','','INFECCION POR VIH','1.0','6.0','B24','','','','','','','
','','','','','','','','','ENFERMEDAD POR VIH, RESULTANTE EN OTRAS ENFERMEDADES INFECCIOSAS O PARASITAR
IAS','B208','','','108','Médico','17','1','09/14/2012 12:00:00 AM','09/14/2012 12:00:00 AM','9','19','5','N'
);
PS C:\Users\docto\Documents\BASES DE DATOS 2\2 - Convertir excel a SQL insert usando python>
```

Y procederemos a ensayar dicha consulta (OJO: recuerda primero crear la tabla antes de meter los datos)

Y entonces ahora necesitamos guardar esas consultas en un vector

```
_DATOS = []
```

```
# Vamos a construir el resto de la consulta con un metodo
def poner_datos_en_query(cabecera_sql, datos):
    _VALUES = "("
    > for i in datos:|...

    # Borrar la ultima coma
    _VALUES = _VALUES[:-1]

    # Poner el final ) y ;
    _VALUES = _VALUES + ");"

    # Guardar
    _DATOS.append(cabecera_sql + _VALUES)
```

Ahora procederemos a guardar todo lo que está contenido en un archivo plano:

```
# Guardar en un archivo plano
_FINAL_SQL = ""
for i in _DATOS:
    _FINAL_SQL = _FINAL_SQL + i + "\n"

with open(f"data_{_nombre_tabla}.sql", "w", encoding="UTF-8") as f:
    f.write(_FINAL_SQL)
```

Primero se pone todo en un texto, luego se guarda.