import mysql.connector as mysql

import sys

import re

import traceback

import datetime

import os

class DatabaseHelper():

def log(self,parsename, detail):

if not os.path.exists(path="log" + "/"):

os.makedirs(name="log" + "/")

path = "log" + "/" + parsename +".log"

if os.path.isfile(path) is not True:

f= open(path, "w+")

f.close()

with open(path, "a") as f:

f.write("Hora:" + str(datetime.datetime.now()) + "\n")

f.write(str(detail) + "\n")

f.write("\n ")

def \_\_init\_\_(self):

self.server = "192.168.0.234" #186.18.178.206

self.database = "moneda"

self.username = "root"

self.password = "retsam77"

if self.password is None:

self.password = ""

self.conn = mysql.connect(user=self.username, password=self.password, host=self.server, database=self.database)

self.cursor = self.conn.cursor(dictionary=True)

def commit(self):

self.conn.commit()

def DBQuery(self, query):

contador=0

while True:

contador+=1

try:

self.cursor.execute(query)

if "SELECT" in query or "select" in query:

result = self.cursor.fetchall()

#self.conn.commit()

return result

else:

#self.conn.commit()

return True

break

except:

text=traceback.format\_exc()+ "\n"

text+=query+ "\n"

self.log("DB", text)

try:

self.conn = mysql.connect(user=self.username, password=self.password, host=self.server, database=self.database)

self.cursor = self.conn.cursor(dictionary=True)

except:

text=traceback.format\_exc() + "\n"

text+=query+ "\n"

self.log("DB", text)

if contador == 3:

break

return None

def cerrarConexion(self):

self.cursor.close()

self.conn.close()

def ArreglarFecha(self, date):

if date == 'null' or date == '-':

return 'null'

date=date.replace(' ','')

listDate = date.split("/")

return str(listDate[2]) + "/" + str(listDate[1]) + "/" + str(listDate[0])

def constructorInsert(self, tabla, arrayValores):

columnas=''

valores=[]

query=''

valoresstring=''

for valor in arrayValores:

for (col,val) in valor.items():

columnas+=col +','

valores.append(val)

for value in valores:

if value is None:

valoresstring += "null" + ","

continue

value = str(value).replace("\n","").replace(" ","").replace("'",'"')

if value is None or value =="None" or value =="none" or value =="NONE" or value == "S/N" or value == "s/n" or value == "-" or value =="null" or value=="Null" or value =="NULL" or value =='':

valoresstring += "null" + ","

elif isinstance(value, int):

valoresstring += str(value).replace(',','') + ","

elif len(re.findall(r"[\d]{1,2}/[\d]{1,2}/[\d]{4}", value))>0:

valoresstring += "'" +self.ArreglarFecha(value)+"',"

elif(re.match("^[^a-zA-Z]\*[^a-zA-Z]$", value)):

valoresstring += "'" + value.replace('.','').replace(',','.').replace('$','') + "',"

elif(re.match("^[A-Za-z0-9\_-]\*$", value)):

valoresstring += "'" + value + "',"

else:

valoresstring +="'"+ value + "',"

query="REPLACE into "+ tabla +"("+columnas[:-1]+") values("+valoresstring[:-1]+") "

return query