from tkinter import \*

from tkinter import messagebox

import sqlite3

#funciones

def conexionBBDD():

miConexion=sqlite3.connect("Usuarios")

miCursor=miConexion.cursor()

try:

miCursor.execute('''

CREATE TABLE DATOSUSUARIOS(

ID INTEGER PRIMARY KEY AUTOINCREMENT,

NOMBRE\_USUARIO VARCHAR(50),

PASSWORD VARCHAR(50),

COMENTARIOS VARCHAR(100))

''')

messagebox.showinfo("BBDD", "BBDD creada con exito")

except:

messagebox.showwarning("Atencion!", "La BBDD ya existe")

def salirAplicacion():

valor=messagebox.askquestion("Salir", "Deseas salir?")

if valor=="yes":

root.destroy()

def limpiarCampos():

miID.set("")

miNombre.set("")

miPassword.set("")

def crear():

miConexion=sqlite3.connect("Usuarios")

miCursor=miConexion.cursor()

miCursor.execute("INSERT INTO DATOSUSUARIOS VALUES(NULL, '" + miNombre.get() +

"','" + miPassword.get() +

"','" + textoComentario.get("1.0", END)+ "')")

#OTRA FORMA DE HACERLO "con consulta parametrica\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#datos=miNombre.get(),miPassword.get(),textoComentario.get("1.0", END)#CONSULTA PARAMETRICA

#miCursor.execute("INSERT INTO DATOSUSUARIOS VALUES(NULL,?,?,?)",(datos))

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

miConexion.commit()

messagebox.showinfo("BBDD", "Registro insertado con exito")

def leer():

miConexion=sqlite3.connect("Usuarios")

miCursor=miConexion.cursor()

miCursor.execute("SELECT \* FROM DATOSUSUARIOS WHERE ID=" + miID.get())

elUsuario=miCursor.fetchall()

for usuario in elUsuario:

miID.set(usuario[0])

miNombre.set(usuario[1])

miPassword.set(usuario[2])

textoComentario.insert(1.0, usuario[3])

miConexion.commit()

def actualizar():

miConexion=sqlite3.connect("Usuarios")

miCursor=miConexion.cursor()

miCursor.execute("UPDATE DATOSUSUARIOS SET NOMBRE\_USUARIO='" + miNombre.get() +

"', PASSWORD='" + miPassword.get() +

"', COMENTARIOS='" + textoComentario.get("1.0", END) +

"' WHERE ID=" + miID.get())

'''OTRA FORMA DE HACERLO "con consulta parametrica\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

datos=miNombre.get(),miPassword.get(),textoComentario.get("1.0", END)

miCursor.execute("UPDATE DATOSUSUARIOS SET NOMBRE\_USUARIO=?, PASSWORD=?, COMENTARIOS=? " +

"WHERE ID=" + miID.get(),(datos))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'''

miConexion.commit()

messagebox.showinfo("BBDD", "Actualizado con exito")

def eliminar():

miConexion=sqlite3.connect("Usuarios")

miCursor=miConexion.cursor()

miCursor.execute("DELETE FROM DATOSUSUARIOS WHERE ID =" + miID.get())

miConexion.commit()

messagebox.showinfo("BBDD", "Registro borrado")

root=Tk()

barraMenu=Menu(root)

root.config(menu=barraMenu, width=300, height=300)

bbddMenu=Menu(barraMenu,tearoff=0)

bbddMenu.add\_command(label="Conectar", command=conexionBBDD)

bbddMenu.add\_command(label="Salir", command=salirAplicacion)

borrarMenu=Menu(barraMenu,tearoff=0)

borrarMenu.add\_command(label="Borrar campos", command=limpiarCampos)

crudmenu=Menu(barraMenu,tearoff=0)

crudmenu.add\_command(label="Crear", command=crear)

crudmenu.add\_command(label="Leer", command=leer)

crudmenu.add\_command(label="Actualizar", command=actualizar)

crudmenu.add\_command(label="Borrar", command=eliminar)

barraMenu.add\_cascade(label="BBDD", menu=bbddMenu)

barraMenu.add\_cascade(label="Borrar", menu=borrarMenu)

barraMenu.add\_cascade(label="CRUD", menu=crudmenu)

#campos

miFrame=Frame(root)

miFrame.pack()

miID=StringVar()

miNombre=StringVar()

miPassword=StringVar()

cuadroID=Entry(miFrame, textvariable=miID)

cuadroID.grid(row=0,column=1,padx=10,pady=10)

cuadroNombre=Entry(miFrame, textvariable=miNombre)

cuadroNombre.grid(row=1,column=1,padx=10,pady=10)

cuadroNombre.config(fg="red",justify="right")

cuadroPass=Entry(miFrame, textvariable=miPassword)

cuadroPass.grid(row=2,column=1,padx=10,pady=10)

cuadroPass.config(show="?")

textoComentario=Text(miFrame,width=16, height=5)

textoComentario.grid(row=5,column=1,padx=10,pady=10)

scrollVert=Scrollbar(miFrame,command=textoComentario.yview)

scrollVert.grid(row=5,column=2,sticky="nsew")

textoComentario.config(yscrollcommand=scrollVert.set)

#labels

idLabel=Label(miFrame,text="ID:")

idLabel.grid(row=0,column=0,sticky="e",padx=10,pady=10)

NombreLabel=Label(miFrame,text="Nombre:")

NombreLabel.grid(row=1,column=0,sticky="e",padx=10,pady=10)

PassLabel=Label(miFrame,text="Password:")

PassLabel.grid(row=2,column=0,sticky="e",padx=10,pady=10)

ComentarioLabel=Label(miFrame,text="Comentario:")

ComentarioLabel.grid(row=5,column=0,sticky="e",padx=10,pady=10)

#botones

miFrame2=Frame(root)

miFrame2.pack()

botonCrear=Button(miFrame2, text="Create", command=crear)

botonCrear.grid(row=1,column=0,sticky="e",padx=10,pady=10)

botonLeer=Button(miFrame2,text="Leer", command=leer)

botonLeer.grid(row=1,column=1,sticky="e",padx=10,pady=10)

botonActualizar=Button(miFrame2,text="Actualizar", command=actualizar)

botonActualizar.grid(row=1,column=2,sticky="e",padx=10,pady=10)

botonBorrar=Button(miFrame2,text="Borrar", command=eliminar)

botonBorrar.grid(row=1,column=3,sticky="e",padx=10,pady=10)

root.mainloop()

#conecion.close()