

# Interface Homme Machine Python-Tkinter-SQLite3 Travaux Pratiques N° 10

## Objetif :

- Conception et réalisation d'une application graphique qui réalise les 4 opérations de base sur une base de données (CRUD):  
C = Create, R = Read, U = Update, D = Delete

### Interface de l'application au démarrage

Python-tkinter-SQLite3: Application CRUD

NumInscription:

Nom:

Prenom:

Genre:  Homme  Femme

Promotion:

Username:

Password:

Insert Read Update Delete Exit

NumInscription	Nom	Prenom	Genre	Promotion	Username	Password
----------------	-----	--------	-------	-----------	----------	----------

## Python-tkinter-SQLite3: Application CRUD

Insert   Read   Update   Delete   Exit

Données lues avec succès à partir de la base

NumInscription:

Nom:

Prenom:

Genre:  Homme  Femme

Promotion:

Username:

Password:

NumInscription	Nom	Prenom	Genre	Promotion	Username	Password
Etu001	HABIT	Noura	F	2018-2019	HABTI	topsecret
Etu002	TALI	Houda	F	2018-2019	TALI	topsecret
Etu003	BALDI	Hassan	H	2018-2019	BALDI	topsecret
Etu005	FADLANE	Radia	H	2018-2019	FADLANE	topsecret
Etu010	ZAROUTI	Hassan	H	2018-2019	ZAROUTI	topsecret
Etu011	KAZAOUI	Tarik	H	2018-2019	KAZAOUI	topsecret
Etu004	MAZRAOUI	Nadine	F	2017-2018	MAZRAOUI	topsecret
Etu007	BOUTAL	Reda	H	2018-2019	BOUTAL	topsecret
Etu12	EL ATRACH	Farid	H	2017-2018	FARID	topsecret

Application CRUD simple

**i** Données lues avec Succès à partir de la base

OK

## Python-tkinter-SQLite3: Application CRUD

NumInscription	Nom	Prenom	Genre	Promotion	Username	Password
Etu001	HABTI	Noura	F	2018-2019	HABTI	topsecret
Etu002	TALII	Houda	F	2018-2019	TALII	topsecret
Etu003	BALDI	Hassan	H	2018-2019	BALDI	topsecret
Etu005	FADLANE	Radia	H	2018-2019	FADLANE	topsecret
Etu010	ZAROUTI	Hassan	H	2018-2019	ZAROUTI	topsecret
Etu011	KAZAOUI	Tarik	H	2018-2019	KAZAOUI	topsecret
Etu004	MAZRAOUI	Nadine	F	2017-2018	MAZRAOUI	topsecret
Etu007	BOUTAL	Reda	H	2018-2019	BOUTAL	topsecret
Etu12	EL ATRACH	Farid	H	2017-2018	FARID	topsecret

  

Données lues avec succès à partir de la base

Insert    Read    Update    Delete    Exit

NumInscription:

Nom:

Prenom:

Genre:  Homme  Femme

Promotion:

Username:

Password:

### Code complet de l'application

```

from tkinter import *
import sqlite3
from tkinter import *
from tkinter import messagebox

```

```

=====Fonctions utilisées par le programme principal=====
#-----Fonction appelée par les autres fonctions pour établir une connection avec la base de données-----
def Database () :
    global conn, cursor #Déclare deux variables globales qui vont être utilisées dans d'autres fonctions
    conn = sqlite3.connect('filieres.db') #Crée un objet connection à la base de données "filieres.db"
    cursor = conn.cursor() #Crée un objet curseur qui permet d'exécuter des instructions SQL
#-----
#-----Fonction Exécutée lorsqu'on clique sur le bouton Insert pour insérer une ligne dans la table Etudiants-----
#Définition d'une fonction Insert qui est exécuté lorsqu'on clique sur le bouton "Insert"
#Permet d'insérer une ligne dans la table Etudiants
def Insert () :
    #Avant d'exécuter l'instruction SQL Insert on teste que l'utilisateur a rempli toutes les znes de saisie
    if NUMINSCRIPTION.get() == "" or NOM.get() == "" or PRENOM.get() == "" or GENRE.get() == "" or PROMOTION.get() == "" or
    USERNAME.get() == "" or PASSWORD.get() == "" :
        lbl_result.config(text="Prière de remplir tous les champs!", fg="red")
    else:
        Database() #pour se connecter à la base et créer un objet curseur
        #l'instruction ci-dessous exécute l'instruction SQL INSERT qui prend les données à partir des varaiblaes de contrôles
        associées aux zones de saiaie Entry
        #Faites bien attention à la syntaxe
        cursor.execute("INSERT INTO Etudiants (NumInscription, NomEtudiant, PrenomEtudiant, Genre, Promotion, UserName,
        Password) VALUES(?, ?, ?, ?, ?, ?)",
            (str(NUMINSCRIPTION.get()), str(NOM.get()), str(PRENOM.get()), str(GENRE.get()), str(PROMOTION.get()),
            str(USERNAME.get()), str(PASSWORD.get())))
        conn.commit() #Pour rendre permanentes les données en les transférant de la RAM vers la base de données sur disque

        tree.delete(*tree.get_children()) #Pour supprimer toutes les lignes du treeview
        cursor.execute("SELECT * FROM Etudiants") #Pour extraire toutes les lignes de la table Etudiants
        fetch = cursor.fetchall() #Crée une liste qui contient toutes les lignes résultats du SELECT
        for data in fetch: #pour traiter chaque ligne de la liste
            #l'instruction ci-dessous insère dans le treeview les valeurs des colonnes de la ligne date
            #data[0]=Numinscription, data[1] = NomEtudiant,..... data[6]=Password
            tree.insert('', 'end', 'values=(data[0], data[1], data[2], data[3], data[4], data[5], data[6])')
        cursor.close()
        conn.close()

#Effacer le contenu des zones de saisie apr l'intermédiaire de s variables de contrôle associées
NUMINSCRIPTION.set("")

```

```

NOM.set("")
PRENOM.set("")
GENRE.set("H")
PROMOTION.set("")
USERNAME.set("")
PASSWORD.set("")

lbl_result.config(text="Données Enregistrées!", fg="green") #Affiche un message dans le frame des boutons
messagebox.showinfo("Application CRUD simple", "Données Enregistrées avec Succès")

#
-----Fonction Exécutée lorsqu'on clique sur le bouton Read pour afficher le contenu de la table Etudiants-----
def Read():
    tree.delete(*tree.get_children()) #Supprime toutes les lignes contenues dans le treeview
    Database()
    cursor.execute("SELECT * FROM Etudiants")
    fetch = cursor.fetchall()
    for data in fetch:
        tree.insert('', 'end', values=(data[0],data[1], data[2], data[3], data[4], data[5], data[6]))
    cursor.close()
    conn.close()
    lbl_result.config(text="Données lues avec succès à partir de la base", fg="green")
    messagebox.showinfo("Application CRUD simple", "Données lues avec Succès à partir de la base")

#
-----Fonction Exécutée lorsqu'on clique sur le bouton Delete pour supprimer la ligne sélectionné-----
def Delete():
    if not tree.selection(): #Teste si une ligne a été sélectionnée avant d'exécuter l'instruction SQL Delete
        lbl_result.config(text="Prière de sélectionner d'abord une ligne", fg="red")
        messagebox.showinfo("Application CRUD simple", "Prière de sélectionner d'abord une ligne")
    else:
        result = messagebox.askquestion('Python: Application CRUD Simple', 'Etes vous sûr de vouloir supprimer cet
enregistrement?', icon="warning")
        if result == 'yes':
            iid = tree.focus() #Obtient l'id de la ligne sélectionnée et qui a le focus
            numasupprimer = tree.set(iid, "NumInscription")
            tree.delete(iid)
            Database()
            cursor.execute("DELETE FROM Etudiants WHERE NumInscription = ?", (numasupprimer,))

```

```

conn.commit()
cursor.close()
conn.close()
lbl_result.config(text="Suppression effectuée avec succès", fg="green")
messagebox.showinfo("Application CRUD simple", "Enregistrement supprimé avec Succès")
#
#-----Fonction Exécutée lorsqu'on clique sur le bouton Update-----
def Update():
    Database()
    # Avant d'exécuter l'instruction SQL Update on teste que l'utilisateur a rempli toutes les zones de saisie
    if NUMINSCRIPTION.get() == "" or NOM.get() == "" or PRENOM.get() == "" or GENRE.get() == "" or PROMOTION.get() == "" or
    USERNAME.get() == "" or PASSWORD.get() == "":
        lbl_result.config(text="Prière de remplir tous les champs!", fg="red")
    else:
        tree.delete(*tree.get_children())
        cursor.execute()
        "UPDATE Etudiants SET NomEtudiant = ?, PrenomEtudiant = ?, Genre =?, Promotion = ?, USERNAME = ?, PASSWORD = ?
        WHERE NumInscription = ?",
        (str(NOM.get()), str(PRENOM.get()), str(GENRE.get()), str(PROMOTION.get()), str(USERNAME.get()),
        str(PASSWORD.get()), str(NUMINSCRIPTION.get()))
        conn.commit()
        cursor.execute("SELECT * FROM Etudiants")
        fetch = cursor.fetchall()
        for data in fetch:
            tree.insert('', 'end', values=(data[0], data[1], data[2], data[3], data[4], data[5], data[6]))
        cursor.close()
        conn.close()
        NUMINSCRIPTION.set("")
        NOM.set("")
        PRENOM.set("")
        GENRE.set("")
        PROMOTION.set("")
        USERNAME.set("")
        PASSWORD.set("")
        #Active tous les boutons sauf le bouton Update
        btn_insert.config(state=NORMAL)
        btn_read.config(state=NORMAL)
        btn_update.config(state=DISABLED)
        btn_delete.config(state=NORMAL)
        lbl_result.config(text="Mise à jour des données réussie", fg="green")

```

```

# -----messagebox.showinfo("Application CRUD simple", "Mise à jour effectuée avec Succès")
# -----
# -----Fonction Exécutée lorsqu'on clique sur le bouton gauche de la souris en vue de supprimer une ligne de la
table-----
def OnSelected(event):
    id = tree.focus()
    NUMINSCRIPTION.set(tree.set(id, "NumInscription"))
    NOM.set(tree.set(id, "Nom"))
    PRENOM.set(tree.set(id, "Prenom"))
    GENRE.set(tree.set(id, "Genre"))
    PROMOTION.set(tree.set(id, "Promotion"))
    USERNAME.set(tree.set(id, "Username"))
    PASSWORD.set(tree.set(id, "Password"))

#Désactive tous les boutons sauf le bouton Update
btn_insert.config(state=DISABLED)
btn_read.config(state=DISABLED)
btn_update.config(state=NORMAL)
btn_delete.config(state=DISABLED)

#-----
#-----Fonction Exécutée lorsqu'on clique sur le bouton Exit-----
def Exit():
    result = messagebox.askquestion('Python-tkinter-SQLite: Application CRUD', 'Etes vous sûr de vouloir quitter?',
    icon="warning")
    if result == 'yes':
        root.destroy()
        exit()

#-----
#-----Début du programme principal-----
root = Tk()
root.title("Python-tkinter-SQLite: Application CRUD")
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
width = 1200
height = 600
x = (screen_width/2) - (width/2)
y = (screen_height/2) - (height/2)

```

```

root.geometry('%dx%dt+d' % (width, height, x, y))
root.resizable(0, 0)

# =====Définition des VARIABLES de contrôle associées aux zones de saisie=====
NUMINSCRIPTION = StringVar()
NOM = StringVar()
PRENOM = StringVar()
GENRE = StringVar(value="H")
PROMOTION = StringVar()
USERNAME = StringVar()
PASSWORD = StringVar()

# =====Définition de la FRAMES Top=====
Top = Frame(root, width=900, height=50, bd=8, relief="raise")
Top.pack(side=TOP)
Lbl_title = Label(Top, width=900, font=('arial', 24), text="Python-tkinter-SQLite3: Application CRUD")
Lbl_title.pack()

# =====Définition de la FRAMES LEFT qui contient les frames Forms (qui contient le frame RadioGroup), Boutons=====
Left = Frame(root, width=300, height=500, bd=8, relief="raise")
Left.pack(side=LEFT)

# =====Définition de la FRAMES Forms=====
Forms = Frame(Left, width=300, height=450)
Forms.pack(side=TOP)

# -----BUTTONS WIDGET Label et Entry le frame Forms-----
lbl_NumInscription = Label(Forms, text="NumInscription:", font=('arial', 16), bd=15)
lbl_NumInscription.grid(row=0, stick="e")
entry_numInscription = Entry(Forms, textvariable=NUMINSCRIPTION, width=30)
entry_numInscription.grid(row=0, column=1)

lbl_Nom = Label(Forms, text="Nom:", font=('arial', 16), bd=15)
lbl_Nom.grid(row=1, stick="e")
entry_nom = Entry(Forms, textvariable=NOM, width=30)
entry_nom.grid(row=1, column=1)

lbl_Prenom = Label(Forms, text="Prenom:", font=('arial', 16), bd=15)
lbl_Prenom.grid(row=2, stick="e")
entry_prenom = Entry(Forms, textvariable=PRENOM, width=30)
entry_prenom.grid(row=2, column=1)

```



```

lbl_Genre = Label(Forms, text="Genre:", font=('arial', 16), bd=15)
lbl_Genre.grid(row=3, stick="e")
RadioGroup = Frame(Forms) #Définitio d'un frame qui va contenir les radioboutons
Male = Radiobutton(RadioGroup, text="Homme", variable=GENRE, value="H", font=('arial', 16)).pack(side=LEFT)
Female = Radiobutton(RadioGroup, text="Femme", variable=GENRE, value="F", font=('arial', 16)).pack(side=LEFT)
RadioGroup.grid(row=3, column=1)

lbl_Promotion = Label(Forms, text="Promotion:", font=('arial', 16), bd=15)
lbl_Promotion.grid(row=4, stick="e")
entry_promotion = Entry(Forms, textvariable=PROMOTION, width=30)
entry_promotion.grid(row=4, column=1)

lbl_username = Label(Forms, text="Username:", font=('arial', 16), bd=15)
lbl_username.grid(row=5, stick="e")
entry_username = Entry(Forms, textvariable=USERNAME, width=30)
entry_username.grid(row=5, column=1)

lbl_password = Label(Forms, text="Password:", font=('arial', 16), bd=15)
lbl_password.grid(row=6, stick="e")
entry_password = Entry(Forms, textvariable=PASSWORD, show="*", width=30)
entry_password.grid(row=6, column=1)

# =====Définition de la FRAMES Boutons=====
Buttons = Frame(Left, width=300, height=100, bd=8, relief="raise")
Buttons.pack(side=BOTTOM)
lbl_result = Label(Buttons)
lbl_result.pack(side=TOP)

# -----BUTTONS WIDGET dans le frame Buttons-----
btn_insert = Button(Buttons, width=10, text="Insert", command=Insert)
btn_insert.pack(side=LEFT)
btn_read = Button(Buttons, width=10, text="Read", command=Read)
btn_read.pack(side=LEFT)
btn_update = Button(Buttons, width=10, text="Update", command=Update)
btn_update.pack(side=LEFT)
btn_delete = Button(Buttons, width=10, text="Delete", command=Delete)
btn_delete.pack(side=LEFT)
btn_exit = Button(Buttons, width=10, text="Exit", command=Exit)
btn_exit.pack(side=LEFT)

# =====Définition de la FRAMES Right qui va contenir le widget treeview avec ses barres de défillements=====
Right = Frame(root, width=600, height=500, bd=8, relief="raise")

```

```

Right.pack(side=RIGHT)
# -----Définition du widget treeview avec deux barres de défillements verticale et horizontale-----
scrollbar = Scrollbar(Right, orient=VERTICAL)
scrollbarx = Scrollbar(Right, orient=HORIZONTAL)

tree = ttk.Treeview(Right, columns=("NumInscription", "Nom", "Prenom", "Genre", "Promotion", "Username", "Password"),
selectmode="extended", height=500, yscrollcommand=scrollbar.set, xscrollcommand=scrollbarx.set)

scrollbar.config(command=tree.yview)
scrollbar.pack(side=RIGHT, fill=Y)
scrollbarx.config(command=tree.xview)
scrollbarx.pack(side=BOTTOM, fill=X)

tree.heading('NumInscription', text="NumInscription", anchor=W)
tree.heading('Nom', text="Nom", anchor=W)
tree.heading('Prenom', text="Prenom", anchor=W)
tree.heading('Genre', text="Genre", anchor=W)
tree.heading('Promotion', text="Promotion", anchor=W)
tree.heading('Username', text="Username", anchor=W)
tree.heading('Password', text="Password", anchor=W)
tree.column('#0', stretch=NO, minwidth=0, width=0)
tree.column('#1', stretch=NO, minwidth=0, width=100)
tree.column('#2', stretch=NO, minwidth=0, width=80)
tree.column('#3', stretch=NO, minwidth=0, width=120)
tree.column('#4', stretch=NO, minwidth=0, width=80)
tree.column('#5', stretch=NO, minwidth=0, width=150)
tree.column('#6', stretch=NO, minwidth=0, width=120)
tree.column('#7', stretch=NO, minwidth=0, width=120)
tree.pack()
tree.bind('<Double-Button-1>', OnSelected)
#-----
root.mainloop()

```