from tkinter import \*  
from PIL import ImageTk, Image  
import sqlite3  
from tkinter import messagebox  
from tkinter import ttk  
from tkcalendar import DateEntry  
  
ltt = []  
check, total, z = 0, 0, 0  
list1 = []  
listv = []  
  
table\_name = **"expense\_tracker"**conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
cur = conn.cursor()  
  
  
def initial():  
 global table\_name  
 try:  
 cur.execute(**"CREATE TABLE "** + str(  
 table\_name) + **" (Date\_Of\_Payment text,Method\_Of\_Payment text, Paid\_To text,Description text,Amount\_Paid REAL)"**)  
 except(sqlite3.OperationalError):  
 pass  
  
  
def tables():  
 global list1  
 list1 = []  
 cur.execute(**'SELECT name from sqlite\_master where type="table"'**)  
 dta = cur.fetchall()  
  
 for a in dta:  
 s = a[0]  
 if s in list1:  
 pass  
 else:  
  
 list1.append(a[0])  
  
  
  
def name():  
 global table\_name  
 table\_name = **""** table\_name = table\_entry.get()  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 try:  
  
 cur.execute(**"CREATE TABLE "** + str(  
 table\_name) + **" (Date\_Of\_Payment text,Method\_Of\_Payment text, Paid\_To text,Description text,Amount\_Paid REAL)"**)  
 messagebox.showinfo(**"SUCCESS"**, **"Table "** + str(table\_name) + **" has been created"**)  
 \_entry = StringVar()  
 \_entry.set(**"Current Table(in use): "** + table\_name)  
 Entry(screen, width=35, textvariable=\_entry, state=DISABLED, relief=SOLID, bg=**"black"**, fg=**"white"**).place(x=265,  
 y=100)  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 conn.commit()  
 conn.close()  
 new\_screen2.destroy()  
 win.destroy()  
 manage()  
 except(sqlite3.OperationalError):  
 messagebox.showwarning(**"WARNING"**, **"PLEASE ENTER A VALID TABLE NAME"**)  
  
  
def create\_table():  
 global table\_entry  
 global new\_screen2  
 new\_screen2 = Toplevel(screen)  
 new\_screen2.iconbitmap(**r'Requirements\icons8-money-box-50.ico'**)  
 new\_screen2.title(**"Create New Table"**)  
 Label(new\_screen2, text=**"Please enter the name of the new table"**).pack()  
  
 table\_entry = Entry(new\_screen2, width=30)  
 table\_entry.pack()  
 Button(new\_screen2, text=**"ENTER"**, command=name).pack()  
  
  
def submit(data\_list):  
 check = 0  
 for i in data\_list:  
 if i.get() == **""**:  
 pass  
 else:  
 check += 1  
  
 try:  
  
 if check == 4:  
 global total  
  
  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
  
 cur.execute(**"INSERT INTO "** + str(  
 table\_name) + **" VALUES (:Date\_Of\_Payment,:Method\_of\_Payment,:Paid\_To,:Description,:Amount\_Paid)"**,  
 {**'Date\_Of\_Payment'**: (data\_list[0]).get(),**'Method\_of\_Payment'**:Method\_Of\_Payment\_Entry.get(),  
 **'Paid\_To'**: (data\_list[1]).get(), **'Description'**: (data\_list[2]).get(),  
 **'Amount\_Paid'**: float(data\_list[3].get())})  
 conn.commit()  
  
 messagebox.showinfo(**"Success"**, **"Successfully added the record to the database"**)  
 for i in data\_list:  
 i.delete(0, END)  
 tot = 0  
  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += amount[0]  
  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 conn.close()  
  
 else:  
 messagebox.showwarning(**"WARNING!"**, **"One or more than one fields are empty**\n**Please Check Again"**)  
 pass  
  
 except(ValueError):  
 messagebox.showwarning(**"WARNING!"**, **"Amount Paid must be integer number**\n**Please Check Again"**)  
  
 new\_screen.destroy()  
 new\_Records()  
  
  
def main\_screen():  
 global screen  
 global table\_name  
 global \_entry  
 global render4  
 global render5  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
 screen = Tk()  
 screen.iconbitmap(**r'Requirements\icons8-money-box-50.ico'**)  
 screen.geometry(**"700x500+150+135"**)  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 conn.commit()  
 conn.close()  
 my\_img = ImageTk.PhotoImage(Image.open(**r"Requirements\Expense-Tracker.png"**))  
 button1\_img = ImageTk.PhotoImage(Image.open(**r"Requirements\button1.png"**))  
 button\_img = ImageTk.PhotoImage(Image.open(**r"Requirements\button.png"**))  
 bg\_img = ImageTk.PhotoImage(Image.open(**r"Requirements\bg2\_.png"**))  
 load = Image.open(**r"Requirements\Expense-Tracker.png"**)  
 image = load.resize((700, 80), Image.ANTIALIAS)  
 render = ImageTk.PhotoImage(image)  
  
 load3 = Image.open(**r"Requirements\button.png"**)  
 image3 = load3.resize((276, 50), Image.ANTIALIAS)  
 render3 = ImageTk.PhotoImage(image3)  
  
 load1 = Image.open(**r"Requirements\button1.png"**)  
 image1 = load1.resize((276, 50), Image.ANTIALIAS)  
 render1 = ImageTk.PhotoImage(image1)  
  
 load2 = Image.open(**"Requirements\exit.png"**)  
 image2 = load2.resize((100, 30), Image.ANTIALIAS)  
 render2 = ImageTk.PhotoImage(image2)  
  
 load4 = Image.open(**r"Requirements\create\_new\_table.png"**)  
 image4 = load4.resize((176, 50), Image.ANTIALIAS)  
 render4 = ImageTk.PhotoImage(image4)  
  
 load5 = Image.open(**r"Requirements\update.png"**)  
 image5 = load5.resize((176, 50), Image.ANTIALIAS)  
 render5 = ImageTk.PhotoImage(image5)  
  
 load6 = Image.open(**r"Requirements\manage.png"**)  
 image6 = load6.resize((176, 50), Image.ANTIALIAS)  
 render6 = ImageTk.PhotoImage(image6)  
  
 bg\_label = Label(screen, image=bg\_img).place(x=0, y=0, relwidth=1)  
 Label(screen, image=render, bg=**"#576E8D"**, relief=GROOVE, bd=0).place(x=0, y=0, relwidth=1)  
 Button(screen, image=render3, command=new\_Records, bg=**"#BDC9D9"**, relief=SOLID, bd=1).place(x=230, y=150)  
 Button(screen, image=render1, command=previous\_Records, bg=**"#E1E4EB"**, relief=SOLID, bd=1).place(x=230, y=250)  
 Button(screen, image=render2, command=screen.quit, bg=**"#EAEDF2"**, bd=0, relief=SUNKEN).place(x=315, y=430)  
  
 Button(screen, image=render6, command=manage, relief=SOLID, bd=1).place(x=275, y=345)  
  
 \_entry = StringVar()  
 \_entry.set(**"Current Table(in use): "** + table\_name)  
 Entry(screen, width=35, textvariable=\_entry, state=DISABLED, relief=SOLID, bg=**"black"**, fg=**"white"**).place(x=265,  
 y=100)  
  
 screen.mainloop()  
  
  
def manage():  
 global win  
 global lt  
 global lst  
 global n  
 tables()  
 win = Toplevel(screen)  
 win.iconbitmap(**r'Requirements\icons8-money-box-50.ico'**)  
 win.title(**"MANAGE TABLES"**)  
 Button(win, image=render4, command=create\_table, relief=SOLID, bd=1).pack(pady=30, expand=True)  
 Button(win, text=**"DELETE SELECTED TABLE"**, font=(**"BOLD"**, 10), bg=**"red"**, command=remove\_table, relief=SUNKEN,  
 bd=1).pack(padx=30, ipadx=30, pady=10, ipady=10)  
  
 Label(win, text=**"Choose the Table that you want to use:"**, font=(**"Bold"**, 15)).pack()  
 lt = []  
 lst = []  
 n = 0  
  
 for n in range(0, len(list1), 1):  
 lt.append(**"var"** + str(n))  
  
 n\_\_ = 0  
  
 for a\_ in list1:  
 lt[n\_\_] = IntVar()  
 Checkbutton(win, text=str(a\_), variable=lt[n\_\_]).pack(anchor=NW, ipadx=50)  
 lst.append(a\_)  
 n\_\_ += 1  
 Button(win, text=**"Done"**, font=(**"BOLD"**, 15), command=change, relief=SOLID, bd=1).pack(ipadx=90, padx=30, pady=10)  
  
 win.mainloop()  
  
  
def select():  
 global p  
 global z  
 global listv  
 listv = []  
 m = 0  
 selection = my\_tree.focus()  
 name\_box1.delete(0, END)  
 *# name\_box2.delete(0, END)* name\_box3.delete(0, END)  
 name\_box4.delete(0, END)  
 name\_box5.delete(0, END)  
  
  
 values = my\_tree.item(selection, **'values'**)  
 if selection == **""**:  
 messagebox.showwarning(**"Attention"**, **"You must select atleast one record to perform this action"**)  
 else:  
 for k in range(0, 5):  
 listv.append(values[k])  
  
 try:  
 name\_box1.insert(0, values[0])  
 name\_box2.set(values[1])  
 name\_box3.insert(0, values[2])  
 name\_box4.insert(0, values[3])  
 name\_box5.insert(0, values[4])  
 z = 1  
 except(IndexError):  
 pass  
  
def update1():  
 global p  
 global z  
 global listv  
  
 m = 0  
 selection = my\_tree.focus()  
  
  
 if [name\_box1.get(), name\_box2.get(), name\_box3.get(), name\_box4.get(), str(name\_box5.get())] == listv:  
 messagebox.showinfo(**"Attention"**, **"Seems as if you didn't make any change to the existing record"**)  
 else:  
  
 if z == 1 and selection != **""**:  
 z = 0  
 try:  
  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 selection = my\_tree.focus()  
 values = my\_tree.item(selection, text=**""**, values=(  
 name\_box1.get(), name\_box2.get(), name\_box3.get(), name\_box4.get(), name\_box5.get()))  
 x = my\_tree.selection()  
 for record in x:  
 cur.execute(**"UPDATE "** + str(  
 table\_name) + **" SET Date\_Of\_Payment = '"**+str(name\_box1.get()) + **"' , Method\_of\_Payment = '"**+str(name\_box2.get()) +**"' , Paid\_To = '"**+ str(name\_box3.get())+**"' , Description = '"**+str(name\_box4.get())+**"' , Amount\_Paid = "**+str(name\_box5.get())+**" WHERE oid = "** + str(ltt[(int(record) - 1)])+**" ;"**)  
  
  
  
  
 name\_box1.delete(0, END)  
 *# name\_box2.delete(0, END)* name\_box3.delete(0, END)  
 name\_box4.delete(0, END)  
 name\_box5.delete(0, END)  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += amount[0]  
  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 conn.commit()  
 conn.close()  
 except(sqlite3.OperationalError):  
 messagebox.showwarning(**"WARNING!"**, **"Amount Paid must be a number**\n**Please Check Again"**)  
 else:  
  
 messagebox.showwarning(**"Attention"**, **"Seems You didn't select any record**\n**Please Check Again"**)  
  
  
  
  
  
def remove():  
 ch = 0  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 x = my\_tree.selection()  
  
 for record in x:  
  
 cur.execute(**"DELETE FROM "** + str(table\_name) + **" WHERE oid="** + str(ltt[(int(record) - 1)]))  
 my\_tree.delete(record)  
 ch += 1  
 if ch == 0:  
 messagebox.showwarning(**"Attention"**, **"You must select atleast one record to perform this action"**)  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 conn.commit()  
 conn.close()  
  
  
def remove\_table():  
 \_i = 0  
 \_n = 0  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 global table\_name  
  
  
 for \_i in range(0, len(lst)):  
  
 if (lt[\_i].get()) == 1:  
 table\_name = lst[\_n]  
 if table\_name == **"expense\_tracker"**:  
 messagebox.showwarning(**"Warning"**, **"Can't delete the default table 'expense\_tracker'"**)  
  
 else:  
 cur.execute(**"DROP TABLE "** + table\_name)  
 messagebox.showinfo(**"SUCCESS"**, **"Successfully delete table "** + table\_name)  
 table\_name=**"expense\_tracker"** \_entry = StringVar()  
 \_entry.set(**"Current Table(in use): "** + table\_name)  
 Entry(screen, width=35, textvariable=\_entry, state=DISABLED, relief=SOLID, bg=**"black"**,  
 fg=**"white"**).place(x=265,  
 y=100)  
  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
 win.destroy()  
 manage()  
 break  
 \_n += 1  
  
  
 conn.commit()  
 conn.close()  
  
  
def change():  
 \_i = 0  
 \_n = 0  
 global table\_name  
 tempp = 0  
  
 for \_i in range(0, len(lst)):  
  
 if (lt[\_i].get()) == 1:  
 table\_name = lst[\_n]  
 messagebox.showinfo(**"SUCCESS"**, **"The current table has been changed to- '"** + table\_name + **"'"**)  
 entry = StringVar()  
 \_entry.set(**"Current Table(in use): "** + table\_name)  
 Entry(screen, width=35, textvariable=\_entry, state=DISABLED, relief=SOLID, bg=**"black"**, fg=**"white"**).place(  
 x=265, y=100)  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
  
 tempp += 1  
 break  
  
 \_n += 1  
 if tempp == 0:  
 messagebox.showwarning(**"WARNING"**, **"SELECT ATLEAST ONE TABLE TO WORK ON"**)  
 win.destroy()  
 manage()  
 else:  
 win.destroy()  
 tot = 0  
 cur.execute(**"SELECT Amount\_Paid FROM "** + table\_name)  
 amt = cur.fetchall()  
 for amount in amt:  
 tot += int(amount[0])  
 screen.title(**"Current Total Expense : Rupees "** + str(tot))  
  
  
def new\_Records():  
 global new\_screen  
 global Method\_Of\_Payment\_Entry  
  
 new\_screen = Toplevel(screen)  
 new\_screen.iconbitmap(**r'Requirements\icons8-money-box-50.ico'**)  
 new\_screen.title(**"New Records"**)  
 Amount\_Paid = DoubleVar()  
 Date\_Of\_Payment\_Entry = DateEntry(new\_screen, width=12,  
background=**'darkblue'**, foreground=**'white'**, borderwidth=2)  
  
 Date\_Of\_Payment\_Entry.grid(row=0, column=1, padx=20)  
 Method\_Of\_Payment\_Entry=StringVar()  
 Method\_Of\_Payment\_Entry.set(**"CASH"**)  
 OptionMenu(new\_screen,Method\_Of\_Payment\_Entry,**"CASH"**,**"CARD"**,**"PAYTM"**,**"CHEQUE"**,**"ONLINE TRANSACTION"**).grid(padx=30,row=1, column=1)  
  
 Paid\_To\_Entry = Entry(new\_screen, width=30)  
 Paid\_To\_Entry.grid(row=2, column=1)  
 Description\_Entry = Entry(new\_screen, width=30)  
 Description\_Entry.grid(row=3, column=1)  
 Amount\_Paid\_Entry = Entry(new\_screen, width=30, textvariable=Amount\_Paid)  
 Amount\_Paid\_Entry.grid(row=4, column=1)  
  
 Date\_Of\_Payment\_Label = Label(new\_screen, text=**"Date Of Payment**\n **(MM\DD\YY)"**).grid(row=0, column=0)  
 Method\_Of\_Payment\_Label = Label(new\_screen, text=**"Method Of Payment**\n**"**).grid(row=1, column=0)  
 Paid\_To\_Label = Label(new\_screen, text=**"Paid To"**).grid(row=2, column=0)  
 Description\_Label = Label(new\_screen, text=**"Description"**).grid(row=3, column=0)  
 Amount\_Paid\_Label = Label(new\_screen, text=**"Amount Paid(In Rupees)"**).grid(row=4, column=0)  
  
 \_list = [Date\_Of\_Payment\_Entry, Paid\_To\_Entry, Description\_Entry, Amount\_Paid\_Entry]  
  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 submit\_btn = Button(new\_screen, text=**"Add Record to Database"**, command=lambda: submit(\_list)).grid(row=5, column=0,  
 columnspan=2,  
 pady=10, padx=10,  
 ipadx=100)  
 conn.commit()  
 conn.close()  
 new\_screen.mainloop()  
  
  
def previous\_Records():  
 global new\_screen1  
 global my\_tree  
 global p  
 global name\_box1  
 global name\_box2  
 global name\_box3  
 global name\_box4  
 global name\_box5  
 global name\_box  
 global ltt  
 ltt = []  
  
 new\_screen1 = Toplevel(screen)  
 new\_screen1.iconbitmap(**r'Requirements\icons8-money-box-50.ico'**)  
 new\_screen1.title(**"Previous Records"**)  
  
 conn = sqlite3.connect(**r"Database\ExpenseTracker.db"**)  
 cur = conn.cursor()  
 cur.execute(**"SELECT \* FROM "** + table\_name)  
 records = cur.fetchall()  
 cur.execute(**"SELECT oid FROM "** + table\_name)  
 rec = cur.fetchall()  
  
 for recc in rec:  
  
 ltt.append(recc[0])  
 style = ttk.Style()  
 style.theme\_use(**"clam"**)  
 style.configure(**"Treeview"**, background=**"white"**, foreground=**"black"**, rowheight=25, fieldbackground=**"white"**)  
 style.map(**'Treeview'**, background=[(**'selected'**, **'green'**)])  
  
 tree\_scroll = Scrollbar(new\_screen1)  
 tree\_scroll.pack(side=RIGHT, fill=Y)  
  
 my\_tree = ttk.Treeview(new\_screen1, yscrollcommand=tree\_scroll.set)  
  
 tree\_scroll.config(command=my\_tree.yview)  
  
 my\_tree.tag\_configure(**"oddrow"**, background=**"WHITE"**)  
 my\_tree.tag\_configure(**"evenrow"**, background=**"lightblue"**)  
 my\_tree[**'columns'**] = (**'Date Of Payment'**, **'Method Of Payment'**, **'Paid To'**, **'Description'**, **'Amount Paid(In Rupees)'**)  
 my\_tree.column(**"#0"**, width=0, anchor=W, minwidth=0, stretch=NO)  
 my\_tree.column(**"Date Of Payment"**, width=150, anchor=W, minwidth=25 )  
 my\_tree.column(**"Method Of Payment"**, width=150, anchor=W, minwidth=25)  
 my\_tree.column(**"Paid To"**, width=150, anchor=W, minwidth=25)  
 my\_tree.column(**"Description"**, width=150, anchor=W, minwidth=50)  
 my\_tree.column(**"Amount Paid(In Rupees)"**, width=150, anchor=W, minwidth=25)  
  
 my\_tree.heading(**"#0"**, text=**"Label"**, anchor=W)  
 my\_tree.heading(**"Date Of Payment"**, text=**"Date Of Payment (MM/DD/YY)"**, anchor=W)  
 my\_tree.heading(**"Method Of Payment"**, text=**"Method Of Payment"**, anchor=W)  
 my\_tree.heading(**"Paid To"**, text=**"Paid To"**, anchor=W)  
 my\_tree.heading(**"Description"**, text=**"Description"**, anchor=W)  
 my\_tree.heading(**"Amount Paid(In Rupees)"**, text=**"Amount Paid(In Rupees)"**, anchor=W)  
  
 p, q = 1, 0  
 for record in records:  
 if p % 2 == 0:  
 my\_tree.insert(parent=**''**, index=**'end'**, iid=p, text=**'Parent'**, values=record, tags=(**'evenrow'**,))  
 else:  
 my\_tree.insert(parent=**''**, index=**'end'**, iid=p, text=**'Parent'**, values=record, tags=(**'oddrow'**,))  
  
 p += 1  
 my\_tree.pack(padx=10, ipadx=50, ipady=20)  
 Button(new\_screen1, text=**"DELETE THE SELECTED RECORD"**, bg=**"red"**, font=(**"BOLD"**, 15), command=remove).pack(padx=10,  
 pady=10,  
 ipadx=220)  
 Label(new\_screen1, text=**"UPDATE RECORDS :-"**, font=(**"BOLD"**, 15)).pack(pady=10)  
 add\_frame = Frame(new\_screen1)  
 add\_frame.pack(pady=10, ipady=10)  
  
 Button(new\_screen1, text=**"PRESS THIS BUTTON TO EDIT THE SELECTED RECORD"**, bg=**"yellow"**, command=select).pack(padx=30,  
 pady=10,  
 ipadx=100)  
  
 Button(new\_screen1, text=**"UPDATE RECORD"**, bg=**"blue"**, font=(**"BOLD"**, 15), command=update1).pack(padx=25, pady=10,  
 ipadx=300)  
  
 n1 = Label(add\_frame, text=**"Date Of Payment (MM/DD/YY)"**)  
 n1.grid(row=1, column=0)  
 n2 = Label(add\_frame, text=**"Method Of Payment"**)  
 n2.grid(row=1, column=1)  
 n3 = Label(add\_frame, text=**"Paid To"**)  
 n3.grid(row=1, column=2)  
 n4 = Label(add\_frame, text=**"Description"**)  
 n4.grid(row=1, column=3)  
 n5 = Label(add\_frame, text=**"Amount Paid"**)  
 n5.grid(row=1, column=4)  
  
 name\_box = IntVar()  
 name\_box.set(**""**)  
 name\_box1 = DateEntry(add\_frame,  
background=**'darkblue'**, foreground=**'white'**, borderwidth=2)  
 name\_box1.grid(row=2, column=0)  
  
 name\_box2 = StringVar()  
 name\_box2.set(**"CASH"**)  
 OptionMenu(add\_frame, name\_box2,**"CASH"**,**"CARD"**,**"PAYTM"**,**"CHEQUE"**,**"ONLINE TRANSACTION"**).grid(row=2, column=1)  
 name\_box3 = Entry(add\_frame)  
 name\_box3.grid(row=2, column=2)  
 name\_box4 = Entry(add\_frame)  
 name\_box4.grid(row=2, column=3)  
 name\_box5 = Entry(add\_frame, textvariable=name\_box)  
 name\_box5.grid(row=2, column=4)  
  
 conn.commit()  
 conn.close()  
  
  
initial()  
main\_screen()  
  
conn.commit()  
conn.close()