

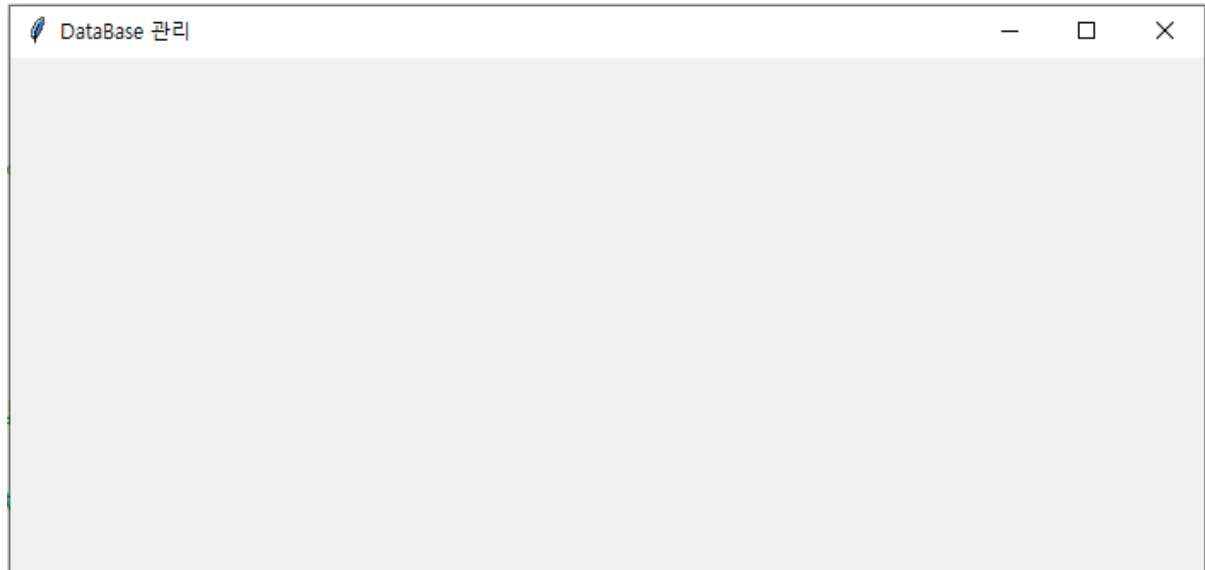
PYTHON

프로그래밍

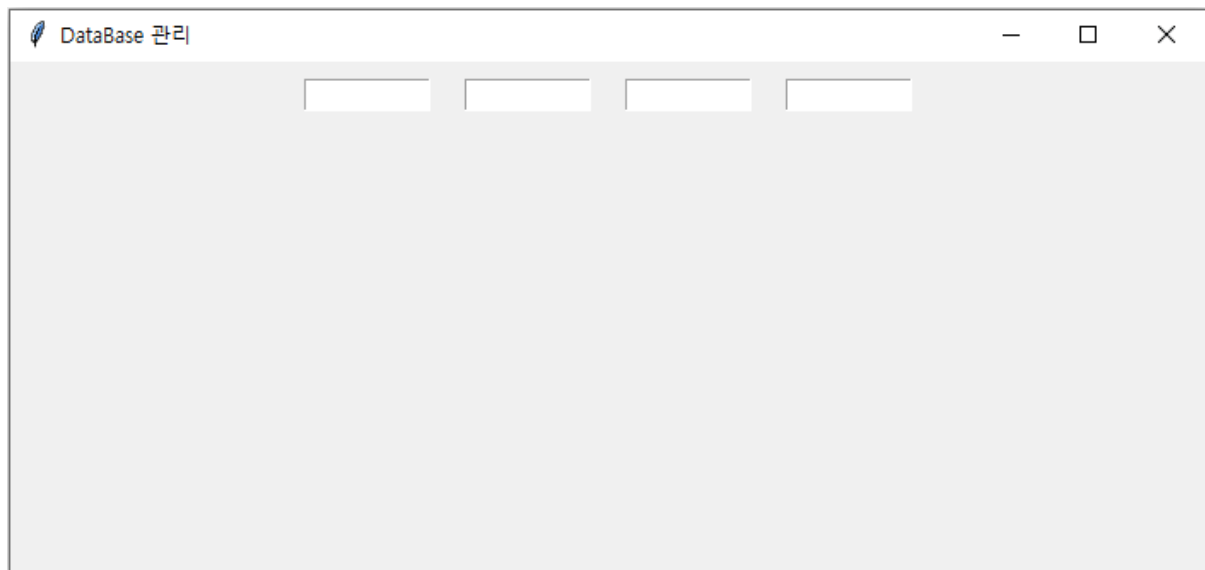
김세운

2024-01-17

과제1> 처음 칸부터 아이디, 이름, 이메일, 연도를 입력 받을 수 있다.



A screenshot of a web browser window titled "DataBase 관리" (Database Management). The window has a light gray background and a white header bar with the title and standard window controls (minimize, maximize, close). The main content area is empty.



A screenshot of the same "DataBase 관리" window, but now with four white input fields arranged horizontally at the top. The fields are empty, and the rest of the window remains the same as in the previous image.

과제1-1> 데이터 삽입 코드.

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

##함수선언##

def insertData():
    con, cur = None, None
    data1, data2, data3, data4 = "", "", "", ""
    sql = ""

    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    data1 = edt1.get()
    data2 = edt2.get()
    data3 = edt3.get()
    data4 = edt4.get()

    try :
        sql = "insert into userTable values('"+data1+"','"+data2+"','"+data3+"','"+data4+"')"
        cur.execute(sql)

    except :
        messagebox.showerror('오류', '데이터 입력 오류')

    else :
        messagebox.showinfo('성공', '데이터 입력 성공')

    con.commit()
    selectData()
    con.close()
```

과제1-2> 데이터 입력창 윈도우 프레임 생성

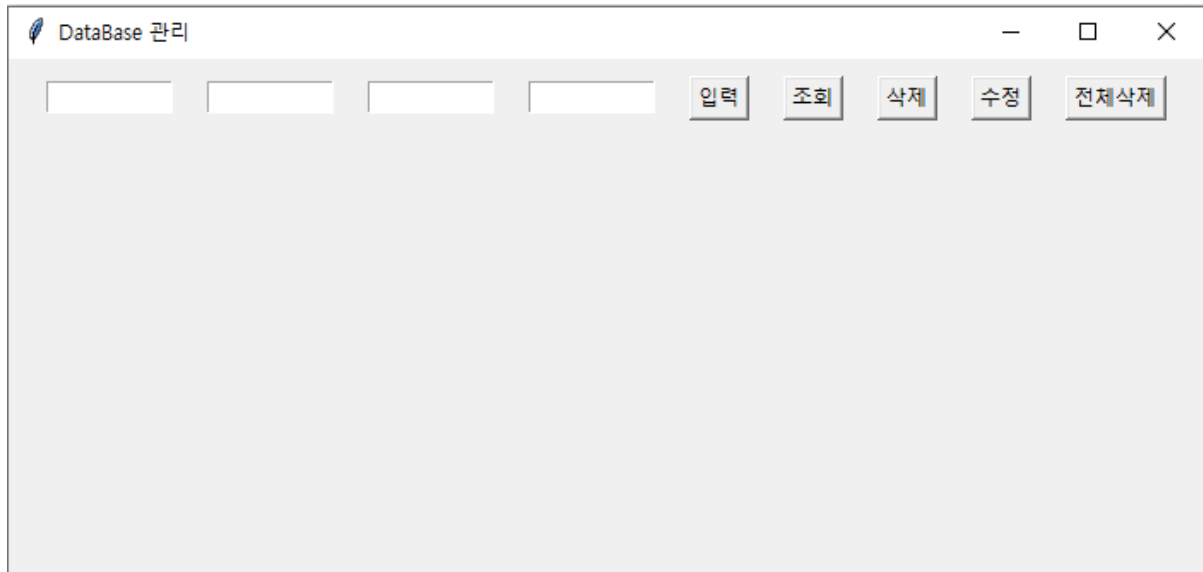
```
window = Tk()
window.geometry("700x300")
window.title("DataBase 관리")

edtFrame = Frame(window)
edtFrame.pack()

edt1 = Entry(edtFrame, width = 10)
edt1.pack(side=LEFT, padx = 10, pady = 10)
edt2 = Entry(edtFrame, width = 10)
edt2.pack(side=LEFT, padx = 10, pady = 10)
edt3 = Entry(edtFrame, width = 10)
edt3.pack(side=LEFT, padx = 10, pady = 10)
edt4 = Entry(edtFrame, width = 10)
edt4.pack(side=LEFT, padx = 10, pady = 10)

window.mainloop()
```

과제2> 입력, 조회, 삭제, 수정, 전체삭제 버튼생성.



```
btnInsert = Button(edtFrame, text = "입력", command = insertData)
btnInsert.pack(side=LEFT, padx = 10, pady = 10)
btnSelect = Button(edtFrame, text = "조회", command = selectData)
btnSelect.pack(side=LEFT, padx = 10, pady = 10)
btnselect = Button(edtFrame, text = "삭제", command = deleteData)
btnselect.pack(side = LEFT, padx = 10, pady = 10)
btnselect = Button(edtFrame, text = "수정", command = updateData)
btnselect.pack(side = LEFT, padx = 10, pady = 10)
btnselect = Button(edtFrame, text = "전체삭제", command = allDeleteData)
btnselect.pack(side = LEFT, padx = 10, pady = 10)
```

과제3> 데이터베이스 내용을 출력하기 위한 프레임 4개의 영역생성 및 색깔

```
listFrame = Frame(window)
listFrame.pack(side = BOTTOM, fill = BOTH, expand = 1)

listData1 = Listbox(listFrame, bg = 'lightgrey')
listData1.pack(side=LEFT, fill = BOTH, expand = 1)
listData2 = Listbox(listFrame, bg = 'lightgrey')
listData2.pack(side=LEFT, fill = BOTH, expand = 1)
listData3 = Listbox(listFrame, bg = 'lightgrey')
listData3.pack(side=LEFT, fill = BOTH, expand = 1)
listData4 = Listbox(listFrame, bg = 'lightgrey')
listData4.pack(side=LEFT, fill = BOTH, expand = 1)
```

과제4> 각 영역에는 각각 아이디, 이름, 이메일, 연도를 조회하여 표시.

DataBase 관리

입력

조회

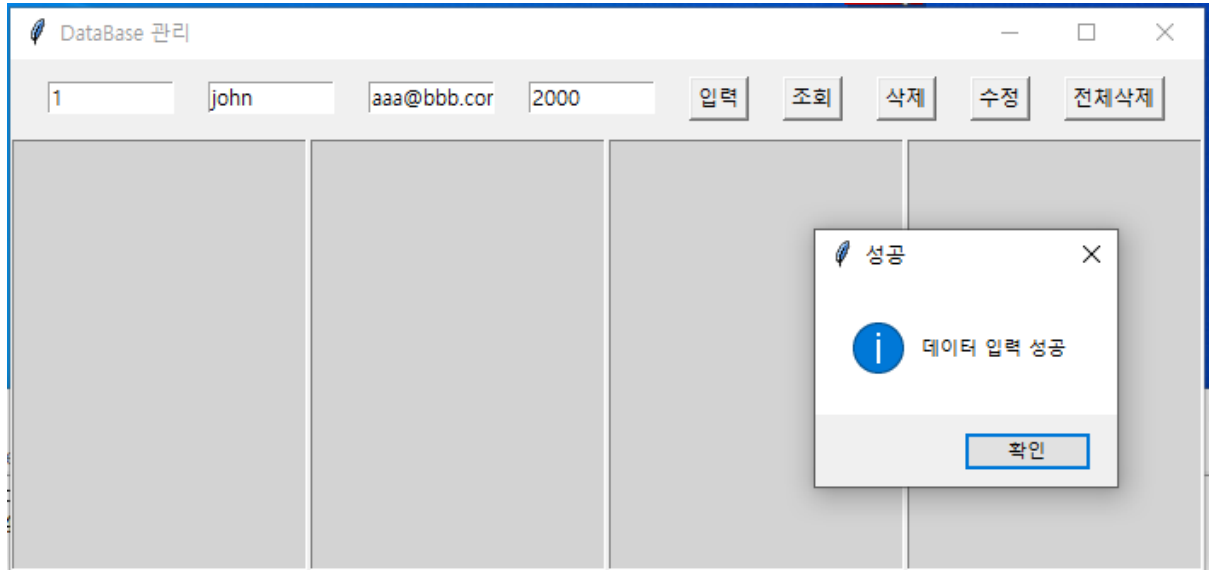
삭제

수정

전체 삭제

사용자 ID	이름	이메일	생일

과제5>데이터를 입력하고 '입력' 버튼을 클릭하면 정보가 데이터베이스 테이블에 저장된다.



과제5-1>인서트 코드 데이터를 받아서 데이터베이스에 기록 저장.

```
def insertData():
    con, cur = None, None
    data1, data2, data3, data4 = "", "", "", ""
    sql = ""

    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    data1 = edt1.get()
    data2 = edt2.get()
    data3 = edt3.get()
    data4 = edt4.get()

    try :
        sql = "insert into userTable values('"+data1+"','"+data2+"','"+data3+"','"+data4+"')"
        cur.execute(sql)

    except :
        messagebox.showerror('오류', '데이터 입력 오류')

    else :
        messagebox.showinfo('성공', '데이터 입력 성공')

    con.commit()
    selectData()
    con.close()
```

과제6> '조회' 버튼을 클릭시 데이터베이스에 있는 데이터를 불러오는 기능.

사용자 ID	이름	이메일	생일
1	john	aaa@bbb.com	2000
2	kang	aaa@bbb.com	2000
3	peter	qqqq@bbb.com	2001
4	jack	qqqq123@bbb.com	2002
5	park	kkqw3@bbb.com	2002
6	jin	jin122@bbb.com	2002

과제6-1> 슬렉트 코드 데이터베이스에 들어갈 영역 설정 및 데이터를 불러오는 기능.

```
def selectData():
    strData1, strData2, strData3, strData4 = [], [], [], []
    con, cur = None, None
    data1, data2, data3, data4 = "", "", "", ""
    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    sql = "select * from userTable"
    cur.execute(sql)

    strData1.append("사용자 ID")
    strData2.append("이름")
    strData3.append("이메일")
    strData4.append("생일")

    strData1.append("-----")
    strData2.append("-----")
    strData3.append("-----")
    strData4.append("-----")

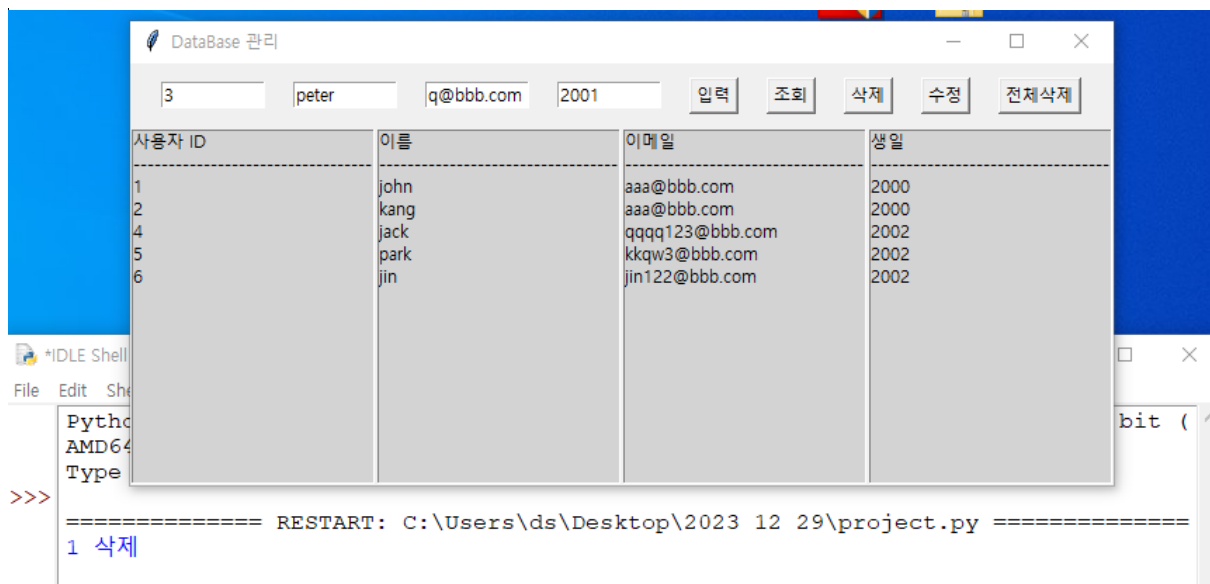
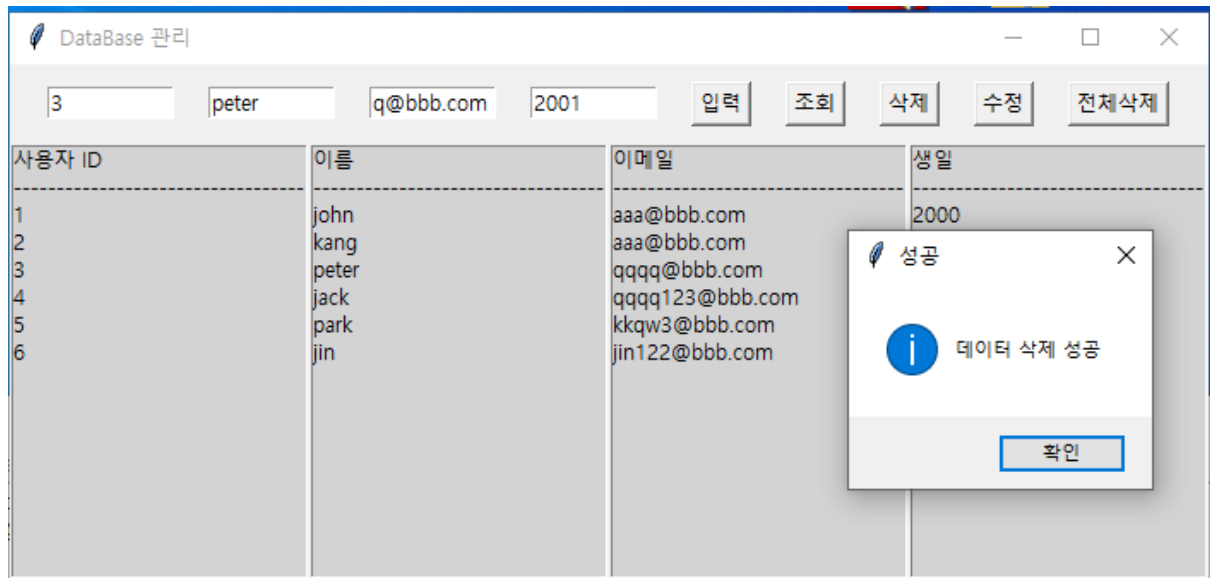
    while True:
        row = cur.fetchone()
        if row == None:
            break
        strData1.append(row[0])
        strData2.append(row[1])
        strData3.append(row[2])
        strData4.append(row[3])

    listData1.delete(0, listData1.size() - 1)
    listData2.delete(0, listData2.size() - 1)
    listData3.delete(0, listData3.size() - 1)
    listData4.delete(0, listData4.size() - 1)

    for item1, item2, item3, item4 in zip(strData1, strData2, strData3, strData4):
        listData1.insert(END, item1)
        listData2.insert(END, item2)
        listData3.insert(END, item3)
        listData4.insert(END, item4)

    con.close()
```

과제7> 삭제 버튼을 누르면 아이디 입력 값을 기준으로 데이터베이스 테이블의 데이터가 삭제.



과제7-1>딜리트 코드 데이터베이스에 아이디 값을 기준으로 삭제가능.

```
def deleteData():

    con, cur = None, None
    data1, data2, data3, data4 = "", "", "", ""
    sql = ""

    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    data1 = edt1.get()
    data2 = edt2.get()
    data3 = edt3.get()
    data4 = edt4.get()

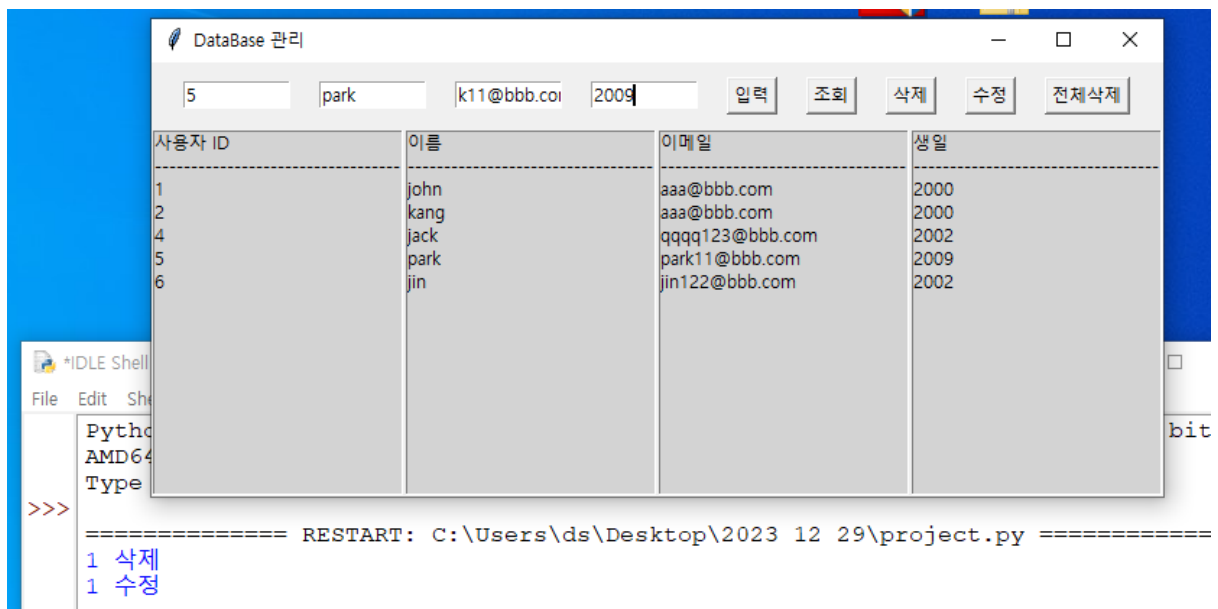
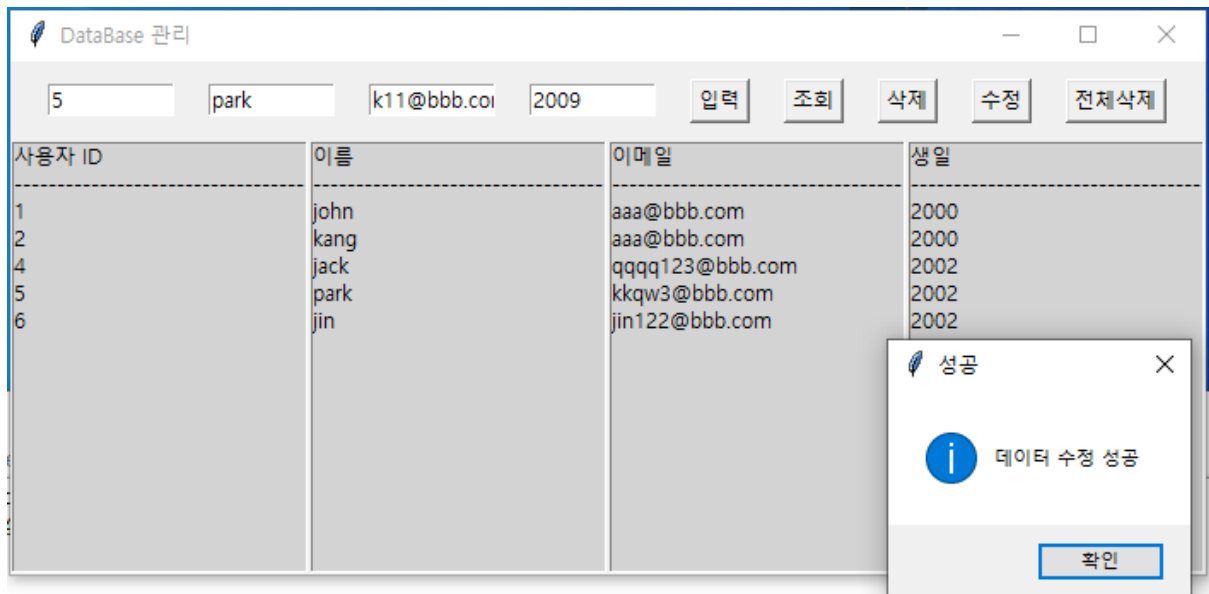
    try:
        sql = "delete from userTable where id = '"+data1+"'"
        cur.execute(sql)

    except :
        messagebox.showerror('오류', '데이터 입력 오류')

    else :
        messagebox.showinfo('성공', '데이터 삭제 성공')

    con.commit()
    print(cur.rowcount, "삭제")
    con.close()
```

과제8> 수정 코드 데이터베이스에 들어갈 영역 설정 및 데이터를 불러오는 기능.



과제8-1>업데이트 코드 데이터베이스에 들어갈 영역 설정 및 수정, 저장 기능.

```
def updateData():

    con, cur = None, None
    data1, data2, data3, data4 = "", "", "", ""
    sql = ""

    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    data1 = edt1.get()
    data2 = edt2.get()
    data3 = edt3.get()
    data4 = edt4.get()

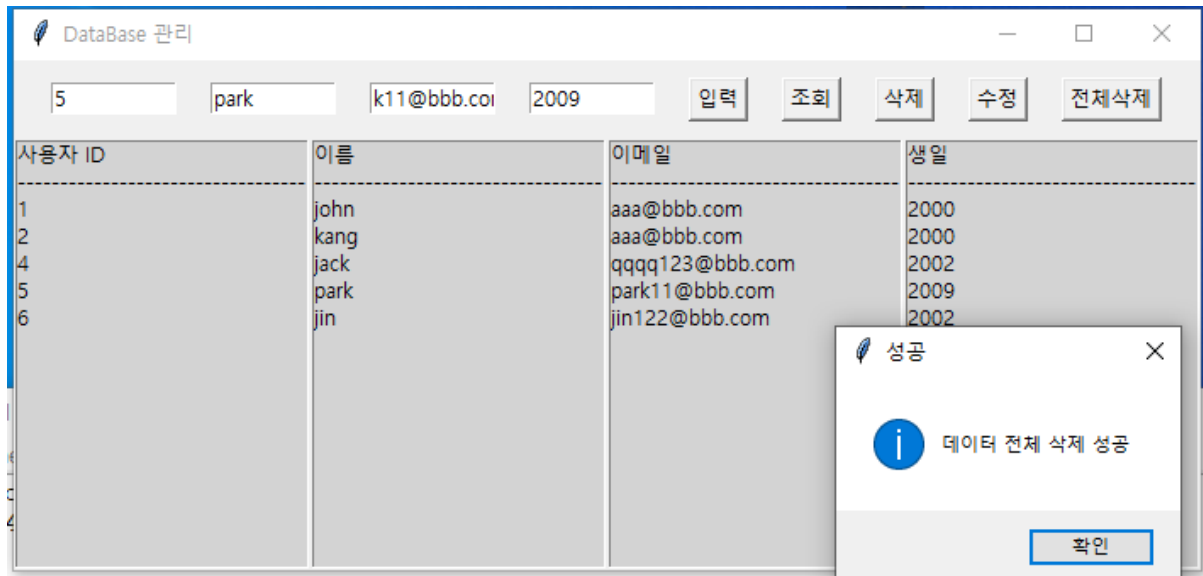
    try:
        sql = "UPDATE userTable SET username = '"+data2+"',email='"+data3+"',birthyear = '"+data4+"' WHERE id = '"+data1+"'"
        cur.execute(sql)

    except :
        messagebox.showerror('오류', '데이터 입력 오류')

    else :
        messagebox.showinfo('성공', '데이터 수정 성공')

    con.commit()
    print(cur.rowcount, "수정")
    con.close()
```

과제9> 전체삭제 데이터베이스에 들어갈 영역 설정 및 데이터를 불러오는 기능.



과제9-1> 전체삭제 코드 테이블에 있는 데이터를 전체 삭제한다.

```
def allDeleteData():
    con = sqlite3.connect("C:/Users/ds/Desktop/2023 12 29/DBtest")
    cur = con.cursor()

    try:
        sql = "delete from userTable"
        cur.execute(sql)

    except :
        messagebox.showerror('오류', '데이터 입력 오류')

    else :
        messagebox.showinfo('성공', '데이터 전체 삭제 성공')

    con.commit()
    print(cur.rowcount, "모든 DB삭제")
    con.close()
```