NAME: Meet Anand Raut

DIV: S2-1

ROLL NO: 2201084

EXPERIMENT – 12:

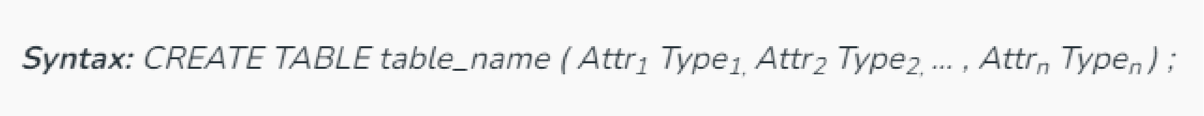
- ***AIM:*** **To study and implement program on demonstrating CRUD (create, red, update and delete) operations on database (SQLite/MySQL) using python**
- ***THEORY:***

Python SQLite – CRUD Operations:

CRUD Operations: The abbreviation CRUD expands to Create, Read, Update and Delete. These four are fundamental operations in a database. In the sample database, we will create it, and do some operations. Let’s discuss these operations one by one with the help of examples.

CREATE

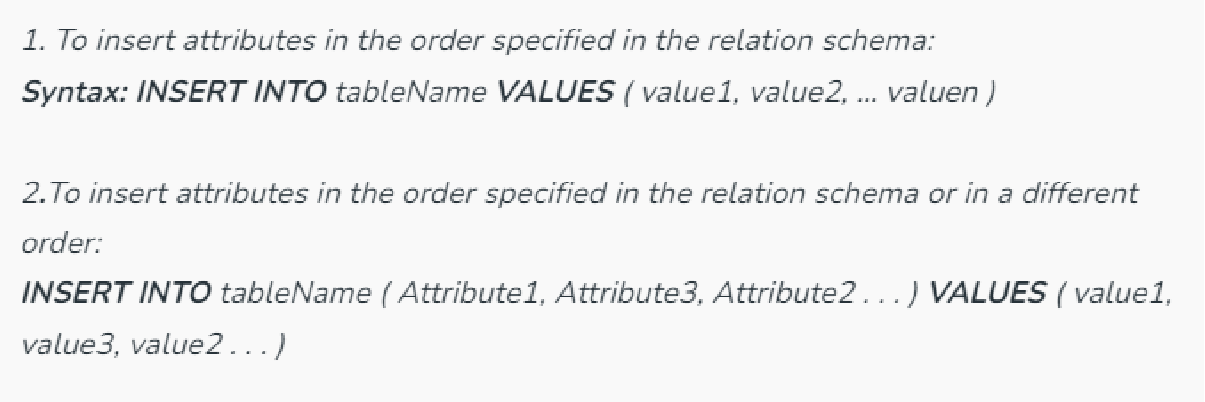
The create command is used to create the table in database.



Syntax: *CREATE TABLE table_name (Attr$_1$ Type$_1$, Attr$_2$ Type$_2$, … , Attr$_n$ Type$_n$) ;*

INSERT

This refers to the insertion of new data into the table. Data is inserted in the form of a tuple. The number of attributes in the tuple must be equal to that defined in the relation schema while creating the table.



1. To insert attributes in the order specified in the relation schema:

Syntax: INSERT INTO *tableName* ***VALUES*** *(value1, value2, … valuen)*

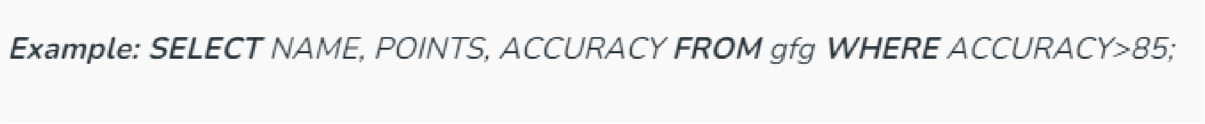
2.To insert attributes in the order specified in the relation schema or in a different order:

INSERT INTO *tableName (Attribute1, Attribute3, Attribute2 . . .)* ***VALUES*** *(value1, value3, value2 . . .)*

READ

This refers to reading data from a database. A read statement has three clauses:

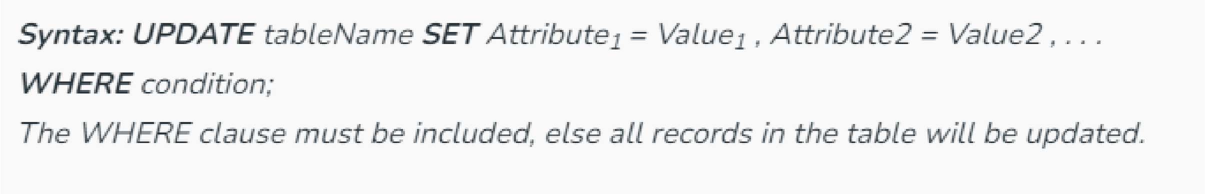
1. **SELECT:** Takes as the predicate the attributes to be queried, use ***** for all attributes.
2. **FROM:** Takes as the predicate a relation.
3. **WHERE:** Takes as the predicate a condition, this is not compulsory.



Example: SELECT *NAME, POINTS, ACCURACY* ***FROM*** *gfg* ***WHERE*** *ACCURACY>85;*

UPDATE

This refers to the updating of tuple values already present in the table.

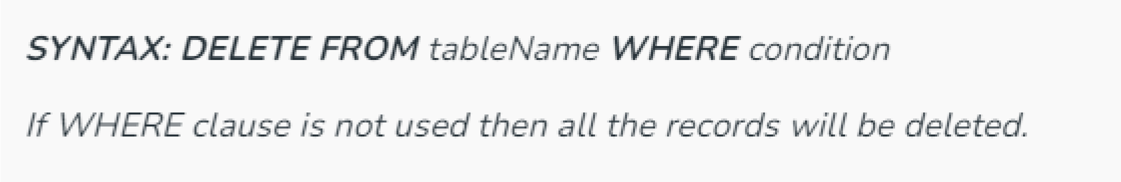


Syntax: UPDATE *tableName* ***SET*** *Attribute$_1$ = Value$_1$, Attribute2 = Value2 , . . .*
WHERE *condition;*

The WHERE clause must be included, else all records in the table will be updated.

DELETE

This refers to the deletion of the tuple present in the table.



SYNTAX: DELETE FROM *tableName* ***WHERE*** *condition*

If WHERE clause is not used then all the records will be deleted.

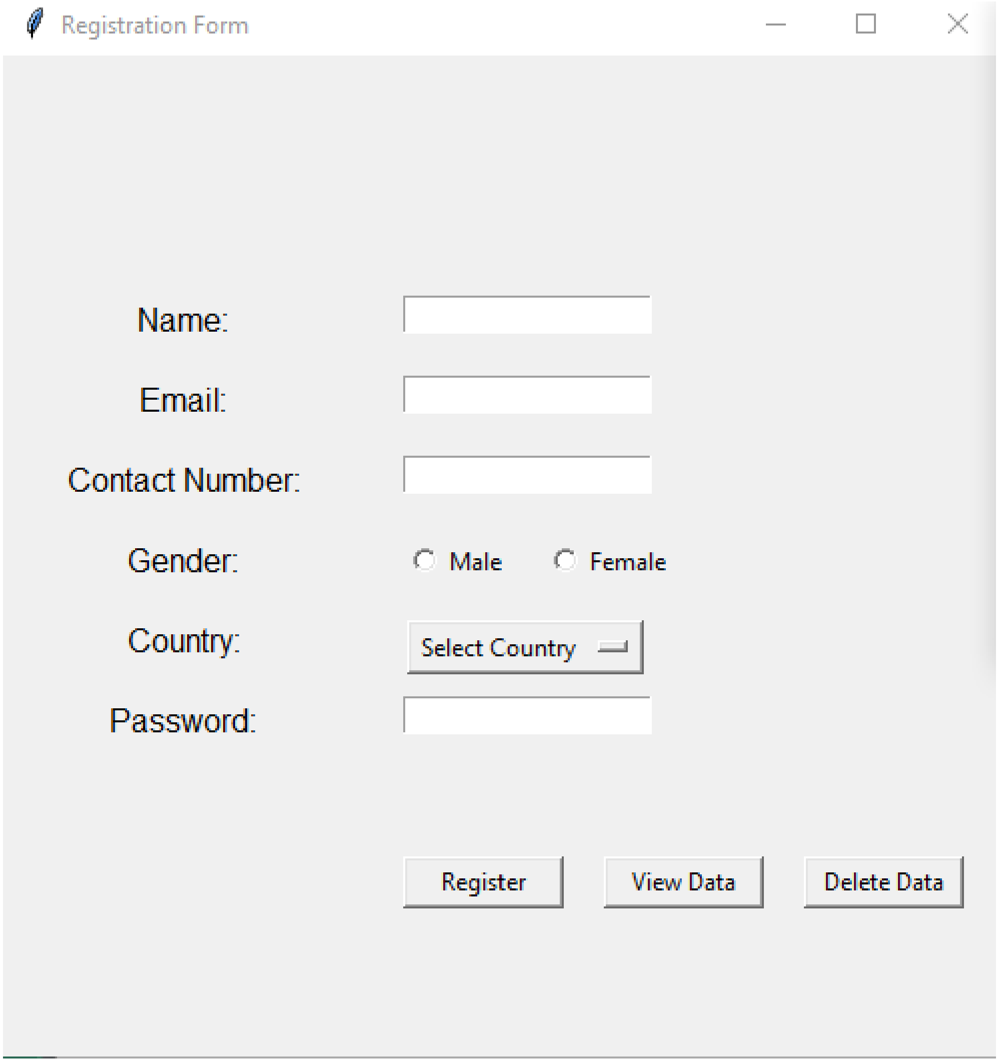
- **PROGRAM:**

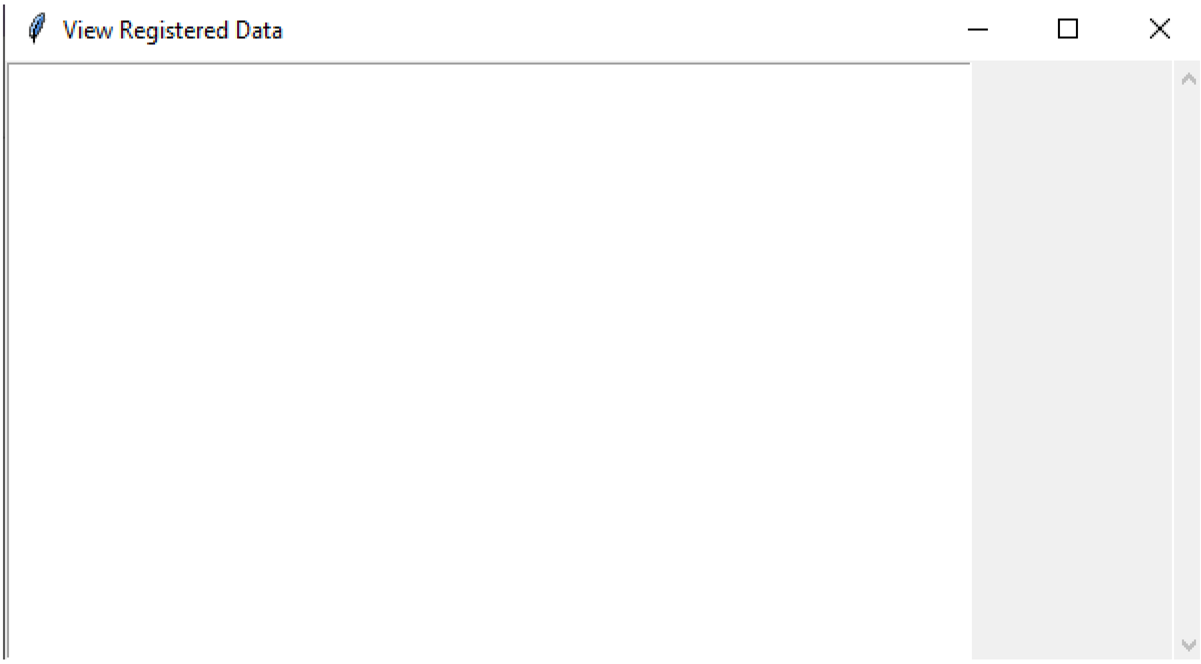
```
import sqlite3
from tkinter import *

# Create an SQLite database connection
conn = sqlite3.connect('registration.db')
cursor = conn.cursor()

# Create the users table if it doesn't exist
cursor.execute('''
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY,
        name TEXT,
        email TEXT,
        contact_number TEXT,
        gender TEXT,
        country TEXT,
        password TEXT
    )
''')

def register_user():
    # Get form input values
    name = en1.get()
    email = en2.get()
    contact_number = en3.get()
    gender = var.get()  # Assumes var is an IntVar from radio buttons
    country = var_country.get()  # Assumes var_country is a StringVar from the
dropdown
    password = en6.get()

    # Insert user data into the database
    cursor.execute('''
        INSERT INTO users (name, email, contact_number, gender, country,
password)
        VALUES (?, ?, ?, ?, ?, ?)
    ''', (name, email, contact_number, gender, country, password))
    conn.commit()
    clear_entries()

def clear_entries():
    en1.delete(0, 'end')
    en2.delete(0, 'end')
    en3.delete(0, 'end')
    en6.delete(0, 'end')
    var.set(0)
    var_country.set("Select Country")

def view_data():
    view_window = Toplevel()
    view_window.title("View Registered Data")
    view_window.geometry("600x300")  # Adjust window size here

    scrollbar = Scrollbar(view_window)
    scrollbar.pack(side=RIGHT, fill=Y)
    listbox = Listbox(view_window, yscrollcommand=scrollbar.set, width=80)  # Adjust width here
    listbox.pack(side=LEFT, fill=BOTH)

    scrollbar.config(command=listbox.yview)

    cursor.execute("SELECT * FROM users")
    data = cursor.fetchall()
    for row in data:
        listbox.insert(END, row)

def delete_data():
    delete_window = Toplevel()
    delete_window.title("Delete Data")

    Label(delete_window, text="Enter Name to Delete:", width=20, font=("arial", 12)).pack()
    entry_name = Entry(delete_window)
    entry_name.pack()

    def delete():
        name = entry_name.get()
        cursor.execute("DELETE FROM users WHERE name=?", (name,))
        conn.commit()
        delete_window.destroy()

    Button(delete_window, text="Delete", command=delete).pack()
```

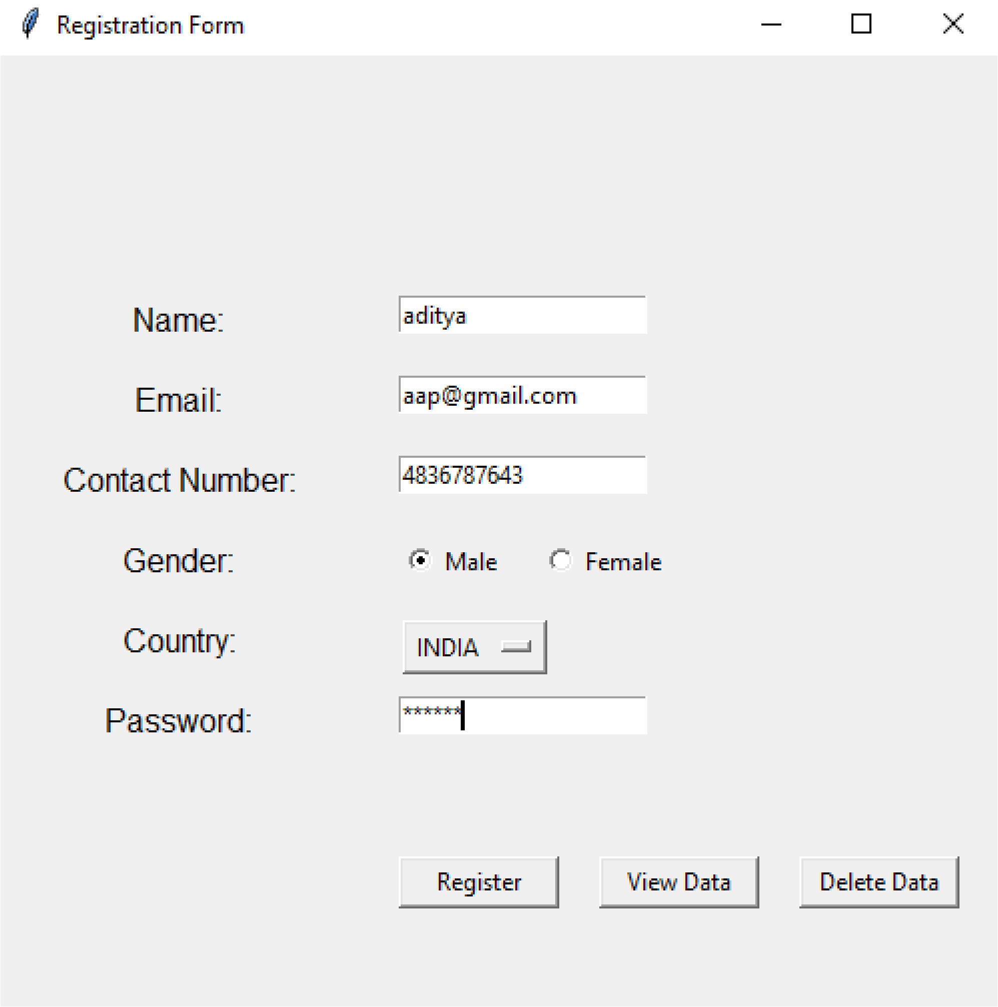
```
# Create the registration form
base = Tk()
base.geometry("500x500")
base.title("Registration Form")

Label(base, text="Name:", width=15, font=("arial", 12)).place(x=20, y=120)
en1 = Entry(base)
en1.place(x=200, y=120)

Label(base, text="Email:", width=15, font=("arial", 12)).place(x=20, y=160)
en2 = Entry(base)
en2.place(x=200, y=160)

Label(base, text="Contact Number:", width=15, font=("arial", 12)).place(x=20, y=200)
en3 = Entry(base)
en3.place(x=200, y=200)

Label(base, text="Gender:", width=15, font=("arial", 12)).place(x=20, y=240)
var = IntVar()
Radiobutton(base, text="Male", variable=var, value=1).place(x=200, y=240)
Radiobutton(base, text="Female", variable=var, value=2).place(x=270, y=240)

Label(base, text="Country:", width=15, font=("arial", 12)).place(x=20, y=280)
var_country = StringVar(base)
var_country.set("Select Country")
```

```
dropdown = OptionMenu(base, var_country, "INDIA", "Canada", "UK", "Australia")
dropdown.place(x=200, y=280)

Label(base, text="Password:", width=15, font=("arial", 12)).place(x=20, y=320)
en6 = Entry(base, show='*')
en6.place(x=200, y=320)

Button(base, text="Register", width=10, command=register_user).place(x=200, y=400)
Button(base, text="View Data", width=10, command=view_data).place(x=300, y=400)
Button(base, text="Delete Data", width=10, command=delete_data).place(x=400, y=400)

base.mainloop()

# Close the cursor and connection
cursor.close()
conn.close()
```

- **OUTPUT:**


Registration Form
Name:
Email:
Contact Number:
Gender:
○ Male
○ Female
Country:
Select Country
Password:
Register
View Data
Delete Data


View Registered Data



Name: aditya

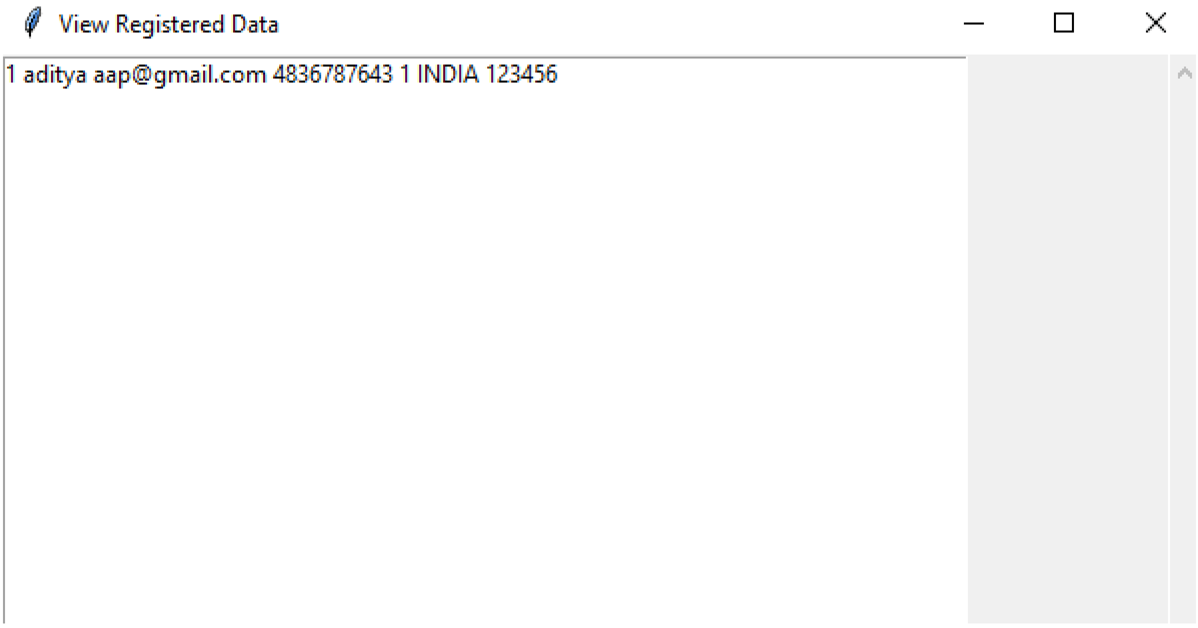
Email: aap@gmail.com

Contact Number: 4836787643

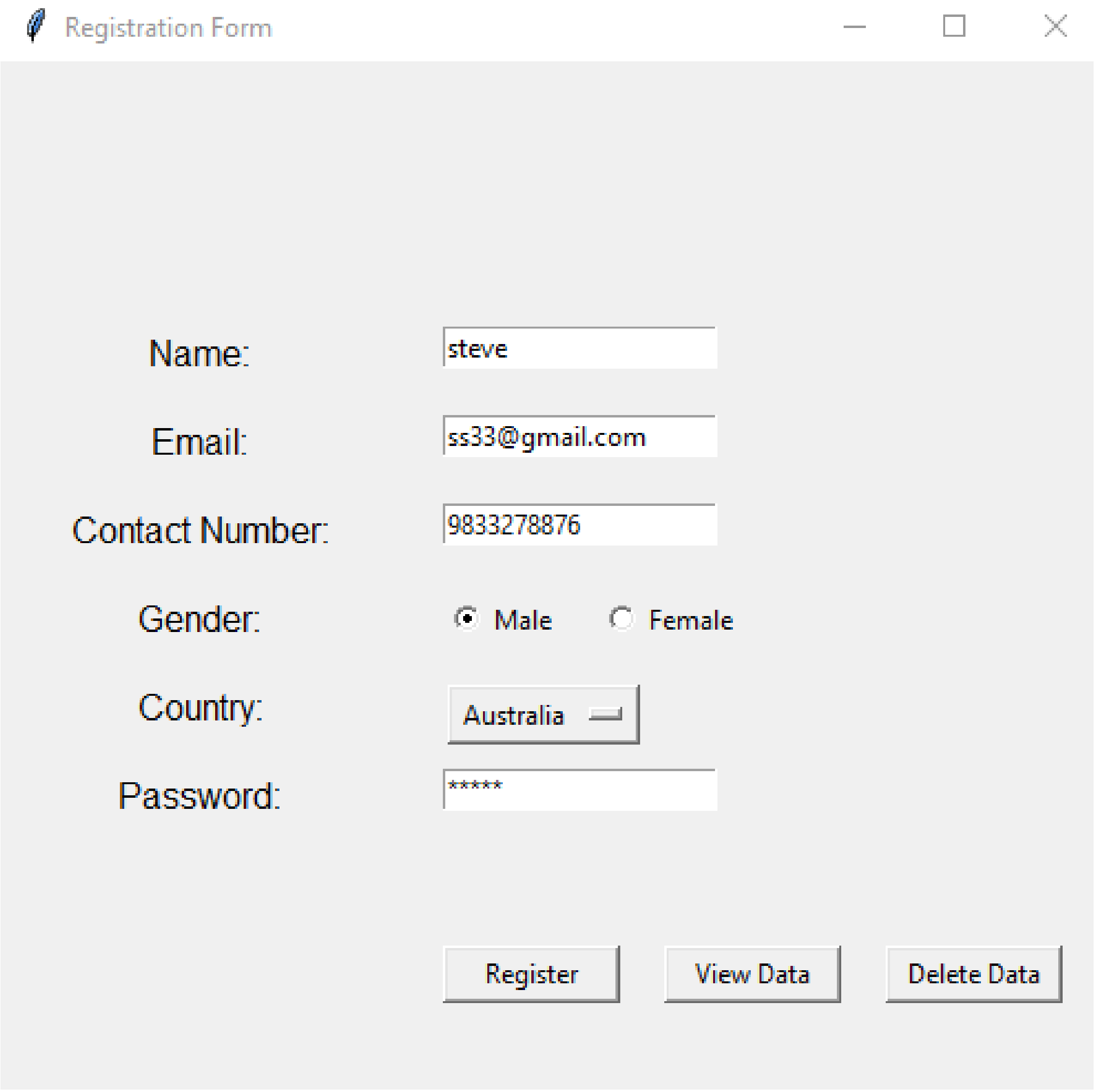
Gender: ● Male ○ Female

Country: INDIA

Password: ******



```
1 aditya aap@gmail.com 4836787643 1 INDIA 123456
```



Registration Form

Name: steve

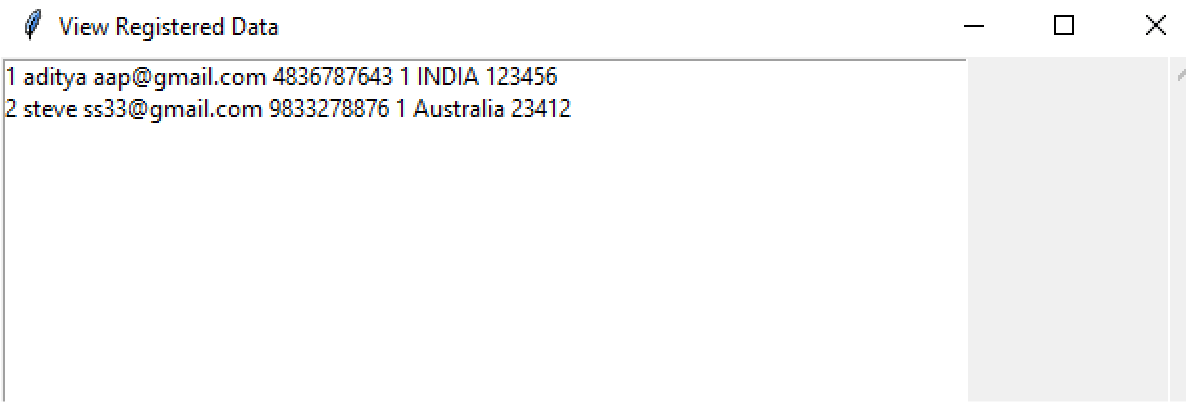
Email: ss33@gmail.com

Contact Number: 9833278876

Gender: ● Male ○ Female

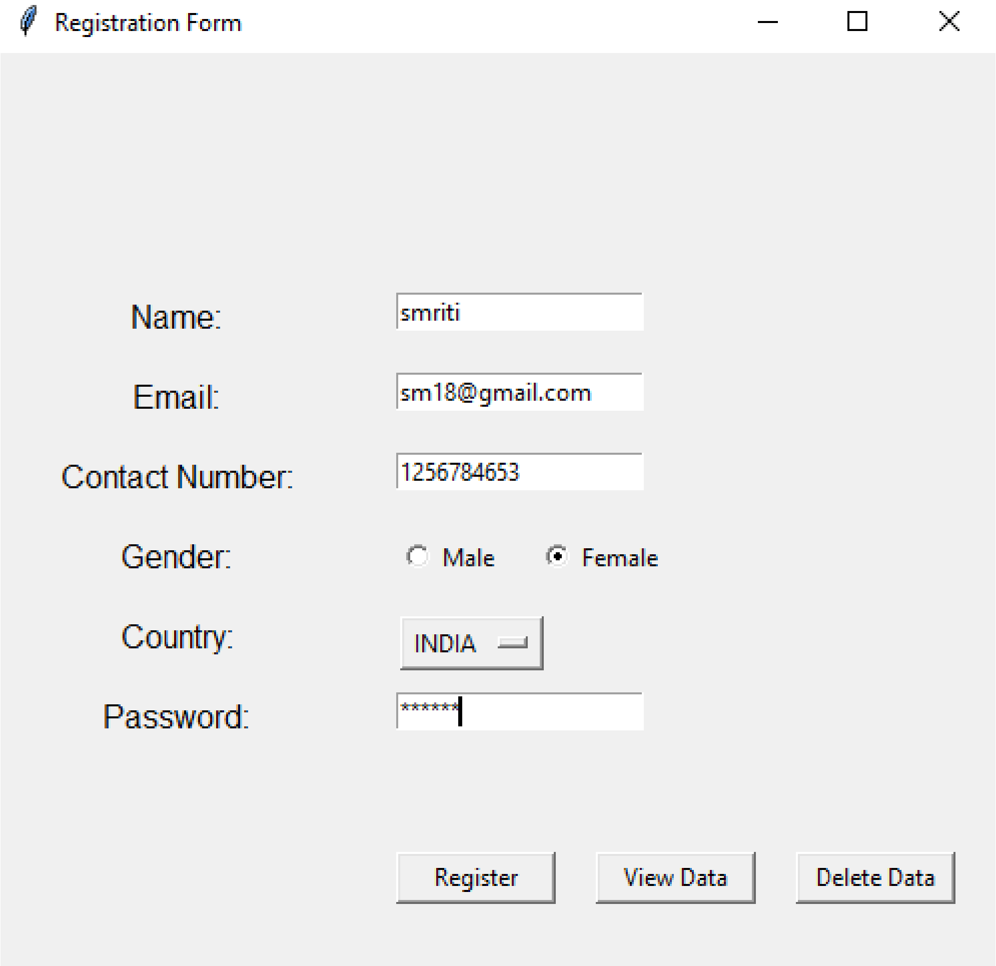
Country: Australia

Password: *****



View Registered Data

```
1 aditya aap@gmail.com 4836787643 1 INDIA 123456
2 steve ss33@gmail.com 9833278876 1 Australia 23412
```



Name: smriti

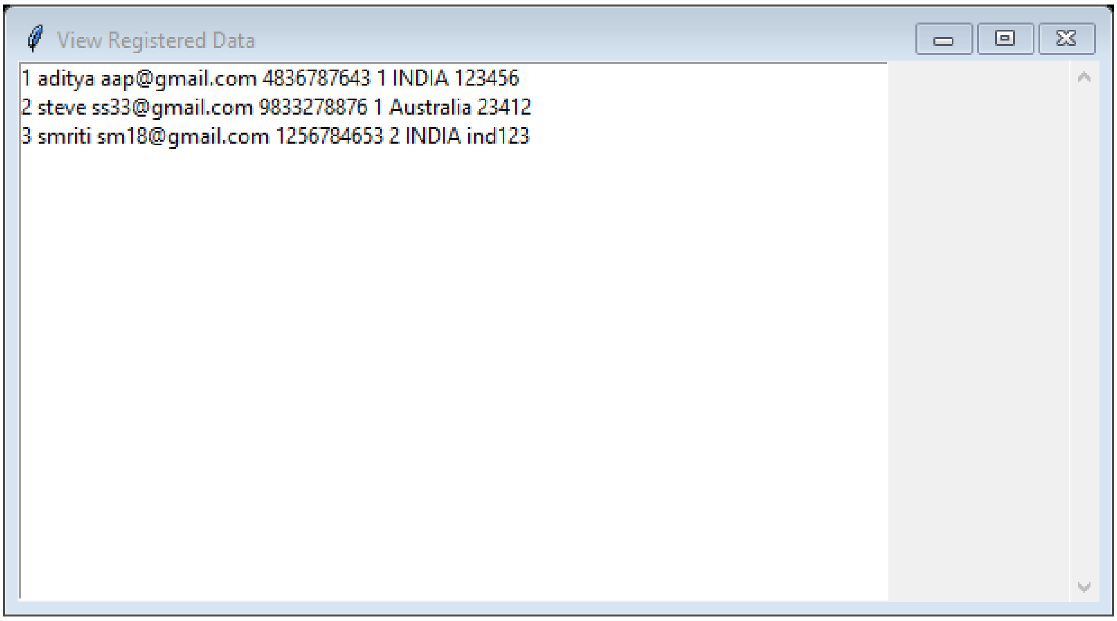
Email: sm18@gmail.com

Contact Number: 1256784653

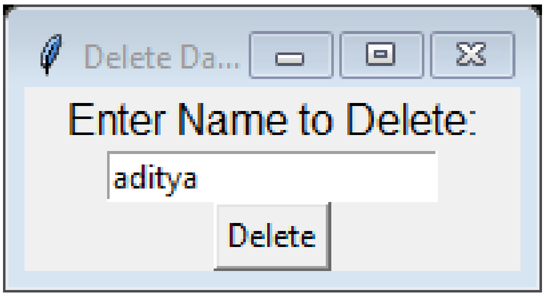
Gender: ○ Male ● Female

Country: INDIA

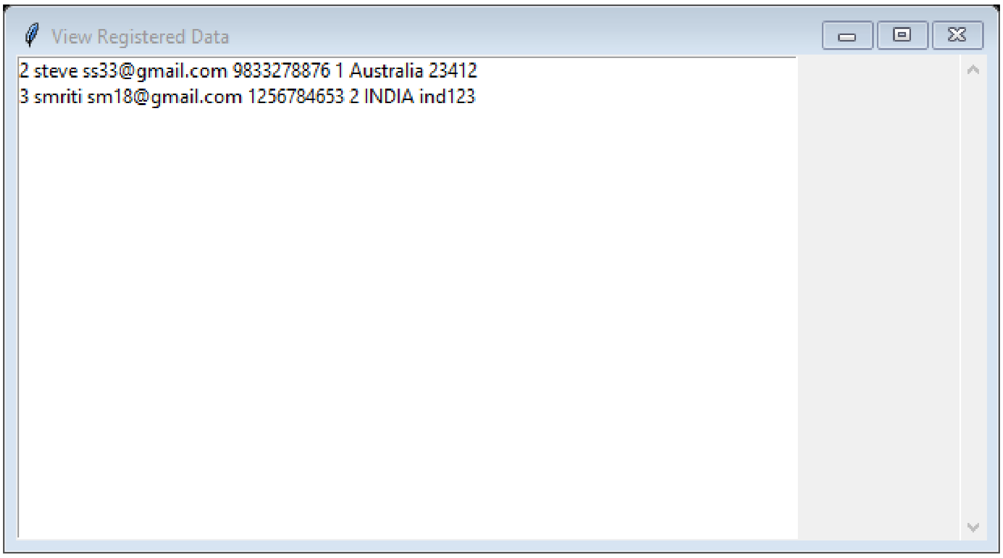
Password: ******



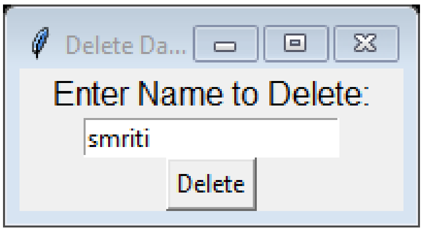
```
1 aditya aap@gmail.com 4836787643 1 INDIA 123456
2 steve ss33@gmail.com 9833278876 1 Australia 23412
3 smriti sm18@gmail.com 1256784653 2 INDIA ind123
```



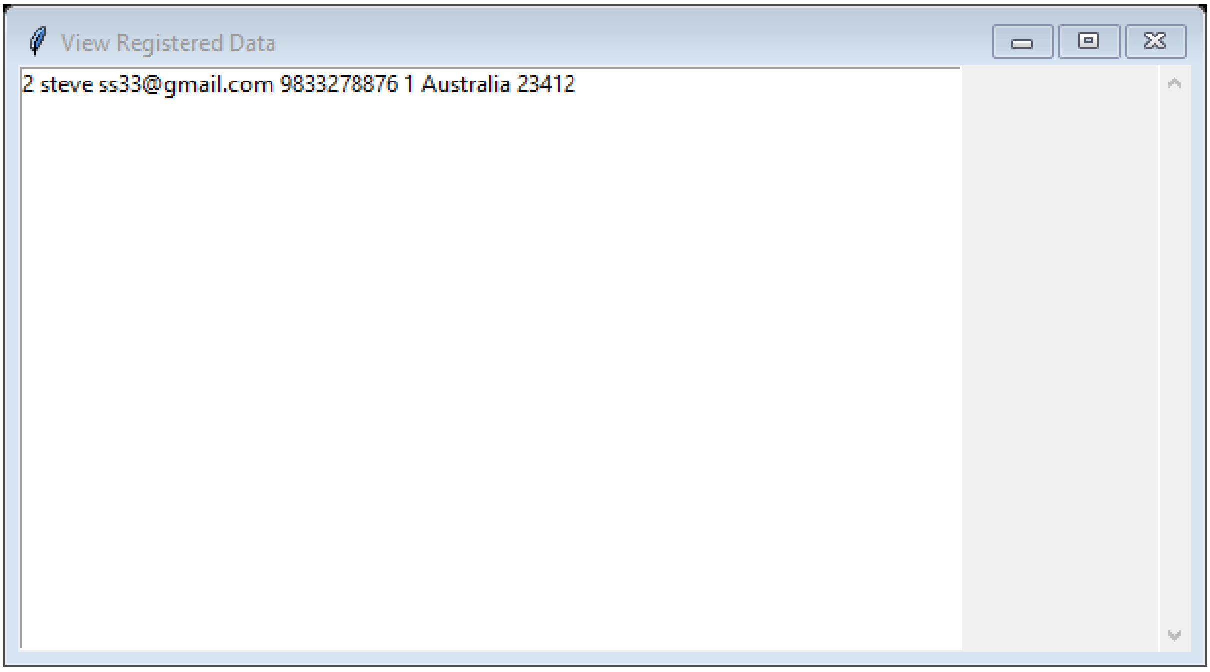
aditya



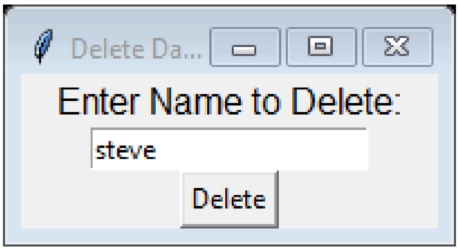
```
2 steve ss33@gmail.com 9833278876 1 Australia 23412
3 smriti sm18@gmail.com 1256784653 2 INDIA ind123
```



smriti

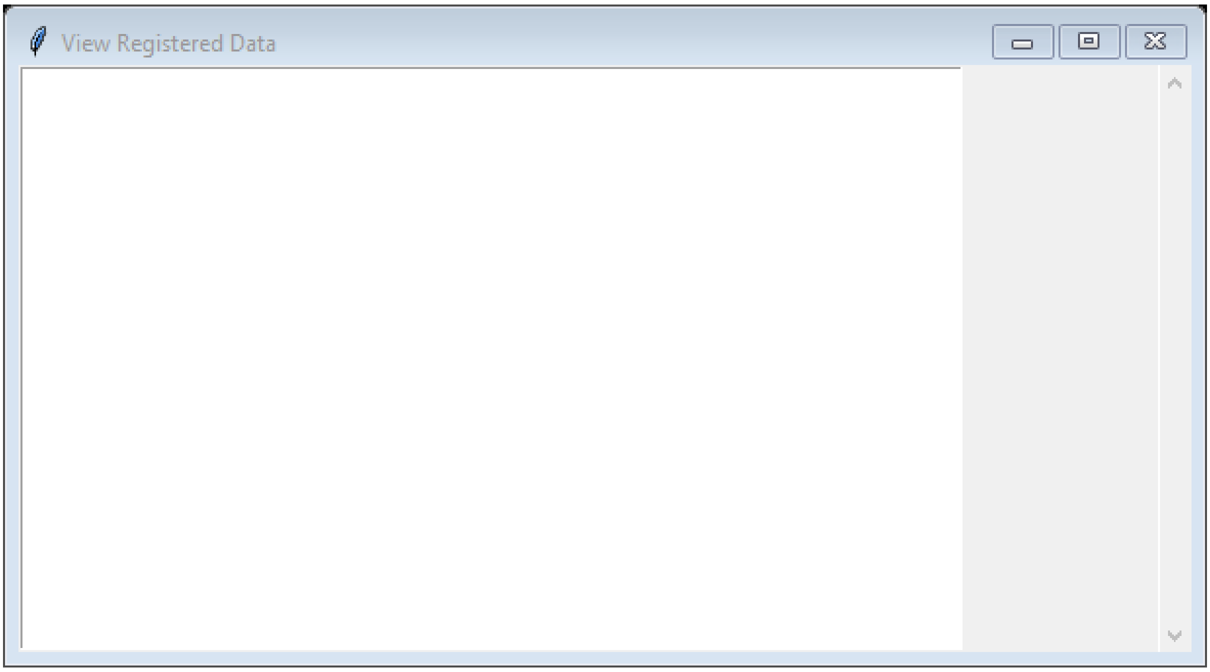


```
2 steve ss33@gmail.com 9833278876 1 Australia 23412
```



Enter Name to Delete:

steve



- ***CONCLUSION:*** **Hence, we have successfully implemented program on demonstrating CRUD operations on Registration form using SQLite python; LO 1.**