Experiment 7: Input:

Part 1:

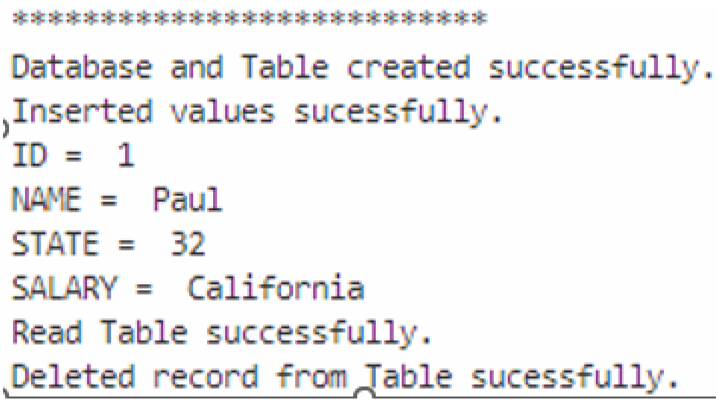
```
# Branch: Computer
# Year: 2025
# Sem: 4
# Name: Mohd Qayam
# UIN: 231P038
# Roll No.: 02

print("***************************")
print("CRUD Operation")
print("Mohd Qayam")
print("***************************")

import sqlite3

# Create
conn = sqlite3.connect('test.db')
conn.execute('CREATE TABLE IF NOT EXISTS COMPANY(ID INT PRIMARY KEY NOT NULL,
NAME TEXT NOT NULL, AGE INT NOT NULL, STATE CHAR(50), SALARY REAL)')
print("Database and Table created successfully.")

# Update
conn.execute("INSERT INTO COMPANY VALUES (1, 'Paul', 32, 'California',
20000.00)")
print("Inserted values sucessfully.")

# Read
cursor = conn.execute("SELECT * from COMPANY")
for row in cursor:
    print("ID = ", row[0])
    print("NAME = ", row[1])
    print("STATE = ", row[2])
    print("SALARY = ", row[3])
print("Read Table successfully.")

# Delete
conn.execute("DELETE from COMPANY where ID=1")
print("Deleted record from Table sucessfully.")

conn.close
```

Part 2:

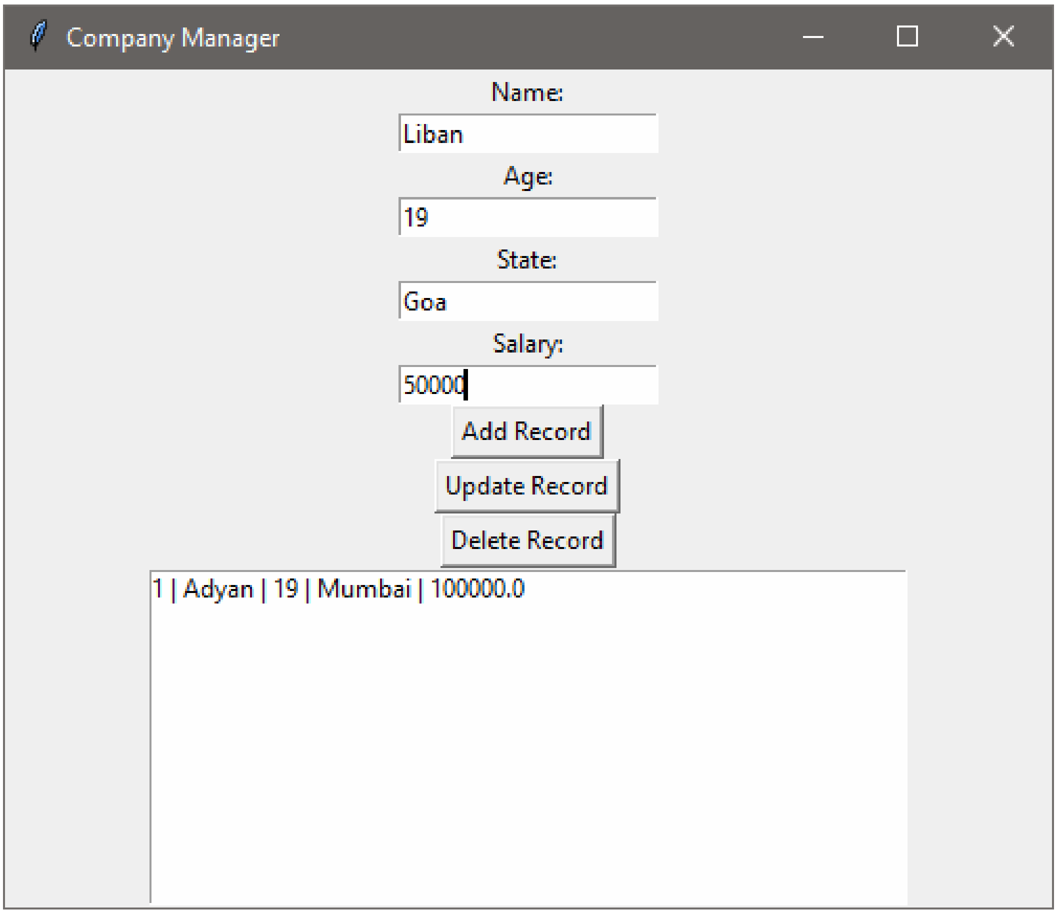
```
# AIM: Write a Python program to create, read, and delete data/task added from
an SQLite database within a Tkinter application. Read write and delete within
the application itself.
# Branch: Computer
# Year: 2025
# Sem: 4
# Name: Mohd Qayam
# UIN: 231P038
# Roll No.: 02

print("***************************")
print("CRUD Operation")
print("Mohd Qayam")
print("***************************")

import sqlite3
import tkinter as tk
from tkinter import messagebox

def init_db():
    conn = sqlite3.connect("test.db")
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS Company (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL,
            age INTEGER NOT NULL,
            state TEXT NOT NULL,
            salary REAL NOT NULL
        )
    """)
    conn.commit()
    conn.close()

# Ensure the database and table exist before any operations
init_db()

def add_record():
    name = name_entry.get()
    age = age_entry.get()
    state = state_entry.get()
    salary = salary_entry.get()
    if name and age and state and salary:
        conn = sqlite3.connect("test.db")
        cursor = conn.cursor()
        cursor.execute("INSERT INTO Company (name, age, state, salary) VALUES
(?, ?, ?, ?)", (name, age, state, salary))
        conn.commit()
        conn.close()
        clear_entries()
        show_records()
    else:
        messagebox.showwarning("Warning", "All fields must be filled!")
```

```
def show_records():
    listbox.delete(0, tk.END)
    conn = sqlite3.connect("test.db")
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM Company")
    records = cursor.fetchall()
    conn.close()
    for record in records:
        listbox.insert(tk.END, f"{record[0]} | {record[1]} | {record[2]} |
{record[3]} | {record[4]}")

def delete_record():
    selected = listbox.curselection()
    if selected:
        record_id = listbox.get(selected).split(" | ")[0]
        conn = sqlite3.connect("test.db")
        cursor = conn.cursor()
        cursor.execute("DELETE FROM Company WHERE id = ?", (record_id,))
        conn.commit()
        conn.close()
        show_records()
    else:
        messagebox.showwarning("Warning", "Please select a record to delete!")

def update_record():
    selected = listbox.curselection()
    if selected:
        record_id = listbox.get(selected).split(" | ")[0]
        name = name_entry.get()
        age = age_entry.get()
        state = state_entry.get()
        salary = salary_entry.get()
        if name and age and state and salary:
            conn = sqlite3.connect("test.db")
            cursor = conn.cursor()
            cursor.execute("UPDATE Company SET name = ?, age = ?, state = ?,
salary = ? WHERE id = ?", (name, age, state, salary, record_id))
            conn.commit()
            conn.close()
            clear_entries()
            show_records()
        else:
            messagebox.showwarning("Warning", "All fields must be filled to
update!")
    else:
        messagebox.showwarning("Warning", "Please select a record to update!")
```

```
def clear_entries():
    name_entry.delete(0, tk.END)
    age_entry.delete(0, tk.END)
    state_entry.delete(0, tk.END)
    salary_entry.delete(0, tk.END)

# GUI Setup
root = tk.Tk()
root.title("Company Manager")
root.geometry("500x400")
tk.Label(root, text="Name:").pack()
name_entry = tk.Entry(root)
name_entry.pack()
tk.Label(root, text="Age:").pack()
age_entry = tk.Entry(root)
age_entry.pack()
tk.Label(root, text="State:").pack()
state_entry = tk.Entry(root)
state_entry.pack()
tk.Label(root, text="Salary:").pack()
salary_entry = tk.Entry(root)
salary_entry.pack()
tk.Button(root, text="Add Record", command=add_record).pack()
tk.Button(root, text="Update Record", command=update_record).pack()
tk.Button(root, text="Delete Record", command=delete_record).pack()
listbox = tk.Listbox(root, width=60, height=10)
listbox.pack()
show_records()
root.mainloop()
```



```
***************************
Database and Table created successfully.
Inserted values sucessfully.
ID =  1
NAME =  Paul
STATE =  32
SALARY =  California
Read Table successfully.
Deleted record from Table sucessfully.
```



Name: Liban

Age: 19

State: Goa

Salary: 50000

```
1 | Adyan | 19 | Mumbai | 100000.0
```