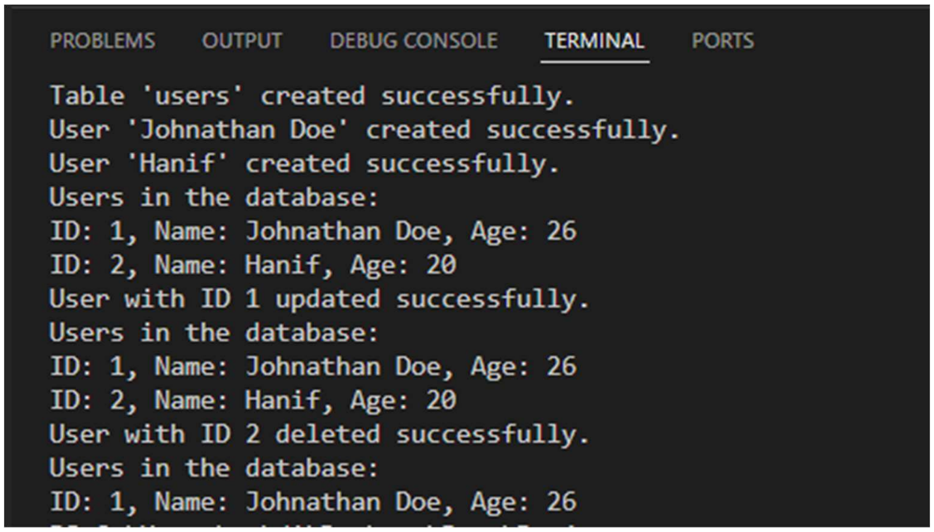
7a.

```
"""
WAP to demonstrate CRUD (create, read, update, delete) operation on database using sqlite3.
Hanif 231P044 / 01
"""
import sqlite3
conn = sqlite3.connect('example.db')
cursor = conn.cursor()
def create_table():
  cursor.execute('''CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT NOT NULL, age INTEGER NOT NULL);''')
  print("Table 'users' created successfully.")
def create_user(name, age):
  cursor.execute('''INSERT INTO users (name, age) VALUES (?, ?);''', (name, age))
  conn.commit()
  print(f"User '{name}' created successfully.")
def read_users():
  cursor.execute('SELECT * FROM users;')
  rows = cursor.fetchall()
  if rows:
    print("Users in the database:")
    for row in rows:
      print(f"ID: {row[0]}, Name: {row[1]}, Age: {row[2]}")
  else:
    print("No users found in the database.")
def update_user(user_id, name, age):
  cursor.execute('''UPDATE users SET name = ?, age = ? WHERE id = ?;''', (name, age, user_id))
```

```
    conn.commit()
    print(f"User with ID {user_id} updated successfully.")
def delete_user(user_id):
    cursor.execute('''DELETE FROM users WHERE id = ?;''', (user_id,))
    conn.commit()
    print(f"User with ID {user_id} deleted successfully.")
if __name__ == "__main__":
    create_table()
    create_user("Hanif ", 20)
    read_users()
    update_user(1, "Hanif", 20)
    read_users()
    delete_user(1)
    read_users()
    conn.close()
```

OUTPUT:



```
Table 'users' created successfully.
User 'Johnathan Doe' created successfully.
User 'Hanif' created successfully.
Users in the database:
ID: 1, Name: Johnathan Doe, Age: 26
ID: 2, Name: Hanif, Age: 20
User with ID 1 updated successfully.
Users in the database:
ID: 1, Name: Johnathan Doe, Age: 26
ID: 2, Name: Hanif, Age: 20
User with ID 2 deleted successfully.
Users in the database:
ID: 1, Name: Johnathan Doe, Age: 26
```

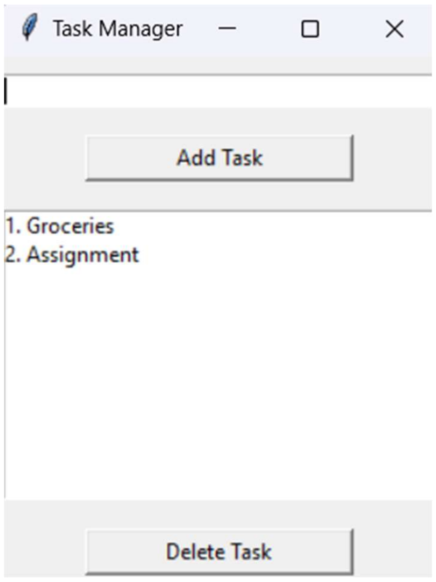
7b.

```
"""

Write a Python program to create, read, and delete data/task added from an SQLite database

within a Tkinter application.

HANIF 231P44 / 01

"""

import sqlite3

import tkinter as tk

from tkinter import messagebox

conn = sqlite3.connect('tasks.db')

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS tasks (id INTEGER PRIMARY KEY, task TEXT NOT NULL);''')

conn.commit()

def add_task():

  task = task_entry.get()

  if task:

    cursor.execute('''INSERT INTO tasks (task) VALUES (?);''', (task,))

    conn.commit()
```

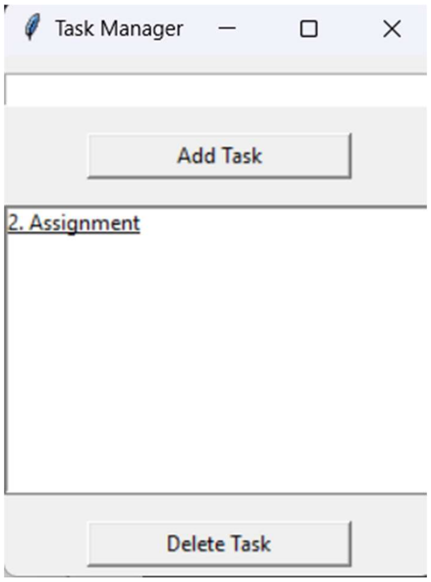
```
        task_entry.delete(0, tk.END)
        load_tasks()
    else:
        messagebox.showwarning("Input Error", "Please enter a task.")
def load_tasks():
    cursor.execute('SELECT * FROM tasks;')
    rows = cursor.fetchall()
    task_listbox.delete(0, tk.END)
    for row in rows:
        task_listbox.insert(tk.END, f"{row[0]}. {row[1]}")
def delete_task():
    try:
        selected_task = task_listbox.get(task_listbox.curselection())
        task_id = selected_task.split('.')[0]
        cursor.execute('''DELETE FROM tasks WHERE id = ?;''', (task_id,))
        conn.commit()
        load_tasks()
    except:
        messagebox.showwarning("Selection Error", "Please select a task to delete.")
root = tk.Tk()
root.title("Task Manager")
task_entry = tk.Entry(root, width=40)
task_entry.pack(pady=10)
add_button = tk.Button(root, text="Add Task", width=20, command=add_task)
add_button.pack(pady=5)
task_listbox = tk.Listbox(root, width=40, height=10)
task_listbox.pack(pady=10)
delete_button = tk.Button(root, text="Delete Task", width=20, command=delete_task)
```

```
delete_button.pack(pady=5)

load_tasks()

root.mainloop()
```

OUTPUT:



1. Groceries
2. Assignment



2. Assignment