# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 9

**1) Write a python program to create a Database and table using SQLite3**

CODE:

import sqlite3

conn = sqlite3.connect("bo3.db")

print("Connected to database successfully")

import sqlite3

conn = sqlite3.connect("bo3.db")

print("Connected to database successfully")

conn.execute('''CREATE TABLE SPECIALISTS

         (

         ID INT PRIMARY KEY      NOT NULL,

         NAME           TEXT     NOT NULL,

         GENDER         CHAR(50) NOT NULL,

         WEAPON         CHAR(50),

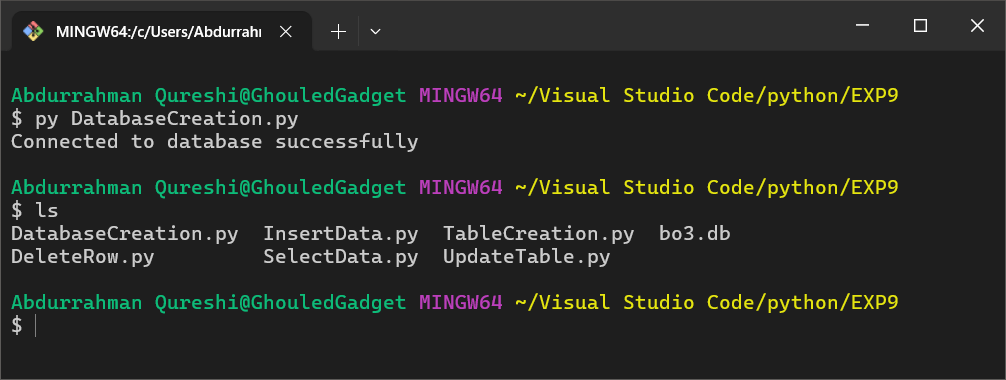
         ABILITY        CHAR(50)

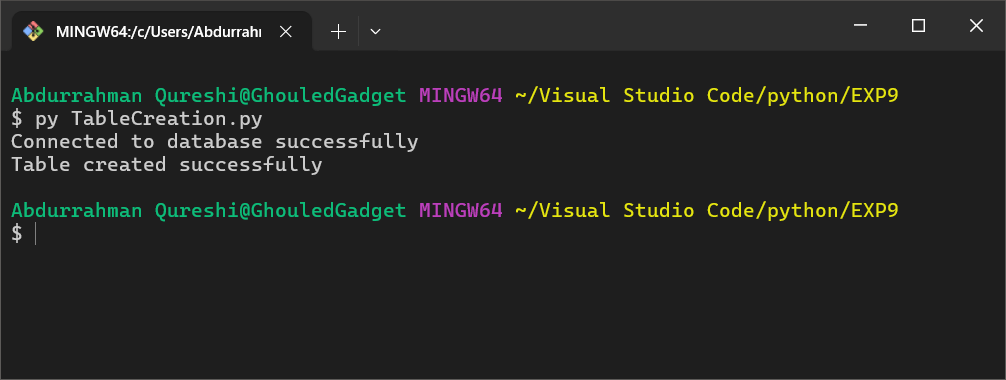
         );''')

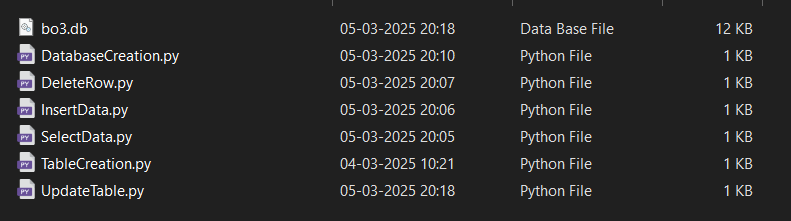
print ("Table created successfully")

conn.close()

OUTPUT:







**2) Write a python program to insert data into a table in the Database using SQLite3**

CODE:

import sqlite3

conn = sqlite3.connect('bo3.db')

print ("Opened database successfully")

conn.execute('''

INSERT INTO SPECIALISTS (ID, NAME, GENDER, WEAPON, ABILITY) VALUES

(1, 'Ruin', 'Male', 'Gravity Spikes', 'Overdrive'),

(2, 'Outrider', 'Female', 'Sparrow', 'Vision Pulse'),

(3, 'Prophet', 'Male', 'Tempest', 'Glitch'),

(4, 'Battery', 'Female', 'War Machine', 'Kinetic Armor'),

(5, 'Seraph', 'Female', 'Annihilator', 'Combat Focus'),

(6, 'Nomad', 'Male', 'H.I.V.E.', 'Rejack'),

(7, 'Reaper', 'Robot', 'Scythe', 'Psychosis'),

(8, 'Spectre', 'Unknown', 'Ripper', 'Active Camo'),

(9, 'Firebreak', 'Male', 'Purifier', 'Heat Wave'),

(10, 'Blackjack', 'Male', 'Rogue', 'Gambler')

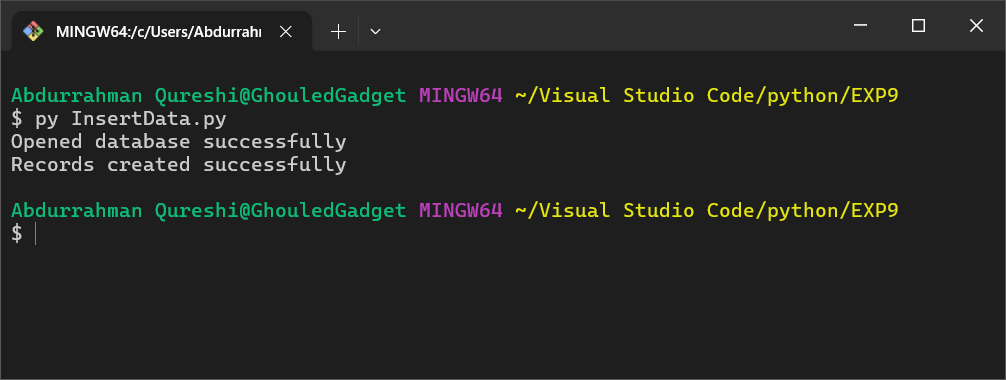
''');

conn.commit()

print ("Records created successfully")

conn.close()

OUTPUT:



**3) Write a python program to select and display data from a table in the Database using SQLite3**

CODE:

import sqlite3

def print\_table(cursor):

    print("-" \* 67)

    print(f"| {'ID':<3} | {'NAME':<12} | {'GENDER':<8} | {'WEAPON':<15} | {'ABILITY':<13} |")

    print("-" \* 67)

    for row in cursor:

        print(f"| {row[0]:<3} | {row[1]:<12} | {row[2]:<8} | {row[3]:<15} | {row[4]:<13} |")

    print("-" \* 67)

conn = sqlite3.connect('bo3.db')

print ("Opened database successfully")

cursor = conn.execute("SELECT id, name, gender, weapon, ability from SPECIALISTS")

print\_table(cursor)

print ("Operation done successfully")

conn.close()

OUTPUT:



**4) Write a python program to update data from a table in the Database using SQLite3**

CODE:

import sqlite3

conn = sqlite3.connect('bo3.db')

print("Opened database successfully")

def print\_table(cursor):

    print("-" \* 67)

    print(f"| {'ID':<3} | {'NAME':<12} | {'GENDER':<8} | {'WEAPON':<15} | {'ABILITY':<13} |")

    print("-" \* 67)

    for row in cursor:

        print(f"| {row[0]:<3} | {row[1]:<12} | {row[2]:<8} | {row[3]:<15} | {row[4]:<13} |")

    print("-" \* 67)

cursor = conn.execute("SELECT id, name, gender, weapon, ability FROM SPECIALISTS")

print("\nInitial Data:")

print\_table(cursor)

conn.execute("UPDATE SPECIALISTS SET id = 100 WHERE ID = 10")

conn.commit()

cursor = conn.execute("SELECT id, name, gender, weapon, ability FROM SPECIALISTS")

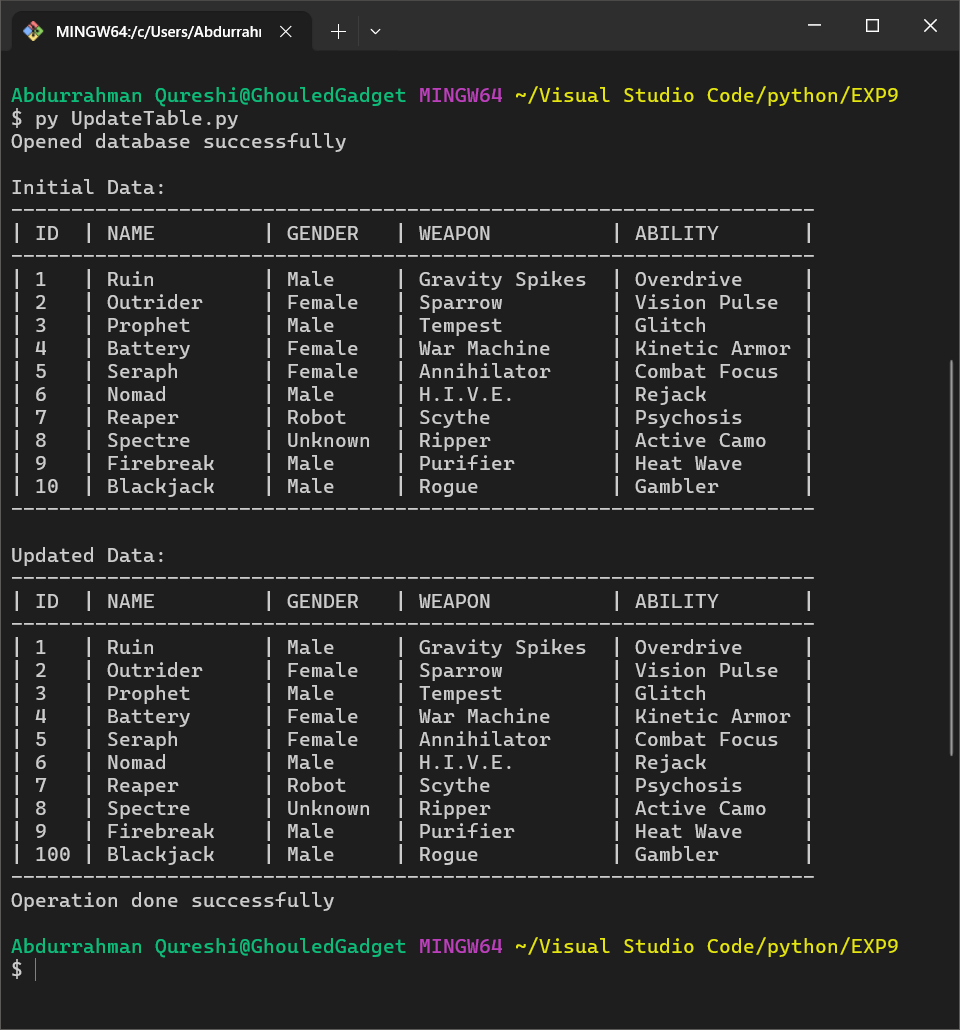
print("\nUpdated Data:")

print\_table(cursor)

print("Operation done successfully")

conn.close()

OUTPUT:



**5) Write a python program to delete data from a table in the Database using SQLite3**

CODE:

import sqlite3

conn = sqlite3.connect('bo3.db')

print("Opened database successfully")

def print\_table(cursor):

    print("-" \* 67)

    print(f"| {'ID':<3} | {'NAME':<12} | {'GENDER':<8} | {'WEAPON':<15} | {'ABILITY':<13} |")

    print("-" \* 67)

    for row in cursor:

        print(f"| {row[0]:<3} | {row[1]:<12} | {row[2]:<8} | {row[3]:<15} | {row[4]:<13} |")

    print("-" \* 67)

cursor = conn.execute("SELECT id, name, gender, weapon, ability FROM SPECIALISTS")

print("\nInitial Data:")

print\_table(cursor)

conn.execute("DELETE FROM SPECIALISTS WHERE id = 100")

conn.commit()

cursor = conn.execute("SELECT id, name, gender, weapon, ability FROM SPECIALISTS")

print("\nUpdated Data:")

print\_table(cursor)

print("Operation done successfully")

conn.close()

OUTPUT:

