Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

Aim: Understand how to create a SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.

IDE:

SQLite3 can be integrated with Python using sqlite3 module. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards. To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements.

Let's enhance the examples with a more practical use case, focusing on **Student Record Management**. We will simulate managing student_record by storing student data like their enrollment, **name**, subject, and mark in the database, and include additional operations like calculating the average mark.

Install sqlite-database

pip install sqlite-database

Database Setup
We'll set up an SQLite database to manage student record information.

Example

```
import sqlite3
# Connect to database (or create it)
conn = sqlite3.connect('student_record.db')
# Create a cursor object using the cursor() method
cursor = conn.cursor()
```

Create an Student Table
We'll create a student_record table to store student details such as Enrollment, name, subject, and Mark.

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

Example

```
# Create students table if it doesn't exist
cursor.execute('''CREATE TABLE IF NOT EXISTS student_record (
                Enrollment INTEGER PRIMARY KEY AUTOINCREMENT,
                name TEXT NOT NULL,
                Subject TEXT NOT NULL,
                Mark INTEGER NOT NULL
            )''')

# Commit the changes
conn.commit()
```

Insert Student Data

Let's insert multiple students into the table.

Example

```
# Insert multiple employee records
student_record = [
   (92301733016,'ASHUTOSH KUMAR SINGH','PWP',95),
   (92301733017,'HARSH VISHALBHAI TRIVEDI','PWP',85),
   (92301733027,'VIRAJ PRAKASHBHAI VAGHASIYA','PWP',90),
   (92301733046,'SHIVAM ATULKUMAR BHATT', 'PWP',93),
   (92301733058,'DEVENDRASINH DOLATSINH JADEJA','PWP',75)
]
# Using executemany to insert multiple records
cursor.executemany('''INSERT INTO student_record (Enrollment, name, subject,Mark)
                VALUES (?, ?, ?,?)''', student_record)

# Commit the changes
conn.commit()
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

Fetch Student Data

Let's retrieve and display all student records.

Example

```
# Fetch all student records
cursor.execute('SELECT * FROM student_record')
rows = cursor.fetchall()
# Display the results
print("All Student Records:")
for row in rows:
    print(row)
```

Fetch Data with Specific Criteria

Let's fetch employees who earn more than 90.

Example

```
# Fetch student got more than 90
cursor.execute('SELECT name, subject, Mark FROM student_record WHERE Mark > 90')
high_marks = cursor.fetchall()

print("\nStudents with Marks greater than 90:")
for student in high_marks:
    print(student)
```

Update Student Information

Suppose a student gets a raise in mark. We can update their mark using an UPDATE statement.

Example:

```
# Update MArk for Ashutosh kumar (PWP)
cursor.execute('''UPDATE student_record SET Mark = 98
            WHERE name = 'ASHUTOSH KUMAR SINGH' AND subject = 'PWP' ''')

# Commit the changes
conn.commit()
```

```
# Verify the update
cursor.execute('SELECT name, MArk FROM student_record WHERE name = "ASHUTOSH KUMAR SINGH"')
updated_mark = cursor.fetchone()
print(f"\nUpdated Mark for {updated_mark[0]}: {updated_mark[1]}")
```

Delete a Student

Let's remove a student from the database.

Example:

```
# Delete a student record (e.g.,DEVENDRASINH DOLATSINH JADEJA )
cursor.execute('''DELETE FROM student_record WHERE name = 'DEVENDRASINH DOLATSINH JADEJA' ''')

# Commit the changes
conn.commit()

# Verify the deletion
cursor.execute('SELECT * FROM student_record WHERE name = "DEVENDRASINH DOLATSINH JADEJA"')
deleted_name = cursor.fetchone()

if deleted_name is None:
   print("\nDEVENDRASINH DOLATSINH JADEJA has been successfully deleted.")
```

Calculate Average Mark

Let's calculate the average mark of all students.

Example:

```
# Calculate the average Mark
cursor.execute('''SELECT AVG(Mark) FROM student_record''')
avg_mark = cursor.fetchone()[0]

print(f"\nThe average mark of students is: ${avg_mark:.2f}")
```

	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

Close the Database Connection

Always close the connection after completing your operations.

Example

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

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

exp_15.py > ...

```
75    avg_mark = cursor.fetchone()[0]
76
77    print(f"\nThe average mark of students is: {avg_mark:.2f}")
78
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

TERMINAL Code

```
All Student Records:
(92301733109, 'ASHUTOSH KUMAR SINGH', 'PWP', 95)
(92301733110, 'HARSH VISHALBHAI TRIVEDI', 'PWP', 85)
(92301733111, 'VIRAJ PRAKASHBHAI VAGHASIYA', 'PWP', 90)
(92301733112, 'SHIVAM ATULKUMAR BHATT', 'PWP', 93)
(92301733113, 'DEVENDRASINH DOLATSINH JADEJA', 'PWP', 75)

Students with Marks greater than 90:
('ASHUTOSH KUMAR SINGH', 'PWP', 95)
('SHIVAM ATULKUMAR BHATT', 'PWP', 93)
```

```
(92301733113, 'DEVENDRASINH DOLATSINH JADEJA', 'PWP', 75)

Students with Marks greater than 90:
('ASHUTOSH KUMAR SINGH', 'PWP', 95)
('SHIVAM ATULKUMAR BHATT', 'PWP', 93)

Updated Mark for ASHUTOSH KUMAR SINGH: 98

DEVENDRASINH DOLATSINH JADEJA has been successfully deleted.

The average mark of students is: 91.50
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

Post Lab Exercise:

Modify the system to allow a student to enroll in multiple subjects at once

```
import sqlite3

# Connect to database (or create it)

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

# Create a cursor object using the cursor() method

cursor = conn.cursor()


# Drop old table for clean start

cursor.execute('DROP TABLE IF EXISTS multiple_student_subjects')


# Create table with composite primary key

cursor.execute('''
    CREATE TABLE multiple_student_subjects (
        Enrollment INTEGER,
        name TEXT NOT NULL,
        Subject TEXT NOT NULL,
        Mark INTEGER NOT NULL,
        PRIMARY KEY (Enrollment, Subject)
    )
```

```
''')
conn.commit()


# Student records
multiple_student_subjects = [
    (92400133189, 'om', 'PWP', 99),
    (92400133189, 'om', 'ICE', 99),
    (92400133189, 'om', 'DMGT', 99),
    (92400133189, 'om', 'DSC', 99),
    (92400133189, 'om', 'SS', 99),
    (92400133189, 'om', 'SPDT', 99),
    (92400133189, 'om', 'APTI', 99),
    (92400133189, 'om', 'COA', 99)
]
cursor.executemany('''
    INSERT INTO multiple_student_subjects (Enrollment, name, Subject, Mark)
    VALUES (?, ?, ?, ?)
''', multiple_student_subjects)
conn.commit()


# Fetch all records
```

```
cursor.execute('SELECT * FROM multiple_student_subjects')
rows = cursor.fetchall()
print("All Student Subjects Records:")
for row in rows:
    print(row)


# Subjects with Marks > 90
cursor.execute('SELECT name, Subject, Mark FROM multiple_student_subjects WHERE Mark > 90')
high_marks = cursor.fetchall()
print("\nSubjects with Marks greater than 90:")
for subject in high_marks:
    print(subject)


# Update Mark for COA
cursor.execute('''
    UPDATE multiple_student_subjects
    SET Mark = 98
    WHERE Enrollment = 92400133189 AND Subject = 'ICE'
''')
conn.commit()
```

```
# Verify the update
cursor.execute('''
    SELECT Subject, Mark FROM multiple_student_subjects
    WHERE Enrollment = 92400133189 AND Subject = 'ICE'
''')
updated = cursor.fetchone()
print(f"\nUpdated Mark for ICE: {updated[1]}")


# Delete marks for 'SS' subject
cursor.execute('''
    DELETE FROM multiple_student_subjects
    WHERE Enrollment = 92400133189 AND Subject = 'DMGT'
''')
conn.commit()


# Verify deletion
cursor.execute('''
    SELECT * FROM multiple_student_subjects
    WHERE Enrollment = 92400133189 AND Subject = 'DMGT'
''')
deleted = cursor.fetchone()
```

```
if deleted is None:
    print("\n'DMGT' subject record has been successfully deleted")


# Calculate the average Mark
cursor.execute("'SELECT AVG(Mark) FROM multiple_student_subjects'")
avg_mark = cursor.fetchone()[0]

print(f"\nThe average mark of students is: {avg_mark:.2f}")
```

OUPUT :

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Understand how to create an SQLite database and perform basic **CRUD** (Create, Read, Update, Delete) operations using Python.	
Experiment No: 15	**Date:**	**Enrollment No:92400133189**

```
Subjects with Marks greater than 90:
('om', 'PWP', 99)
('om', 'ICE', 99)
('om', 'DMGT', 99)
('om', 'DSC', 99)
('om', 'SS', 99)
('om', 'SPDT', 99)
('om', 'APTI', 99)
('om', 'COA', 99)

Updated Mark for ICE: 98

'DMGT' subject record has been successfully deleted

The average mark of students is: 98.86
```

Github : [PythonPostLab/15 at main · Om-Lathigara/PythonPostLab]