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: 92400133037**

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: 92400133037**

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	**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: 92400133037**

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}")

Close the Database Connection

Always close the connection after completing your operations.

Example

# Close the connection
conn.close()

Output:

lab15 > sql.py > ...

```python
import sqlite3
conn=sqlite3.connect('student_record.db')
cursor=conn.cursor()
print("Database connected successfully!")

cursor.execute("DROP TABLE IF EXISTS student_record")
#added this drop line as to exceute everytime it will nor show dupliacte error
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)''')


#CREATE TABLE-Creates a new table in the database
#IF NOT EXISTS-Prevents error if the table already exist
#PRIMARY KEY-Each row will have a unique value here (no duplicates).
#AUTOINCREMENT-Values are automatically assigned (1, 2, 3..)

conn.commit()#stors permanantly

```

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: 92400133037**

```
22
23  student_record=[
24      (92301733016,'ASHUTOSH KUMAR SINGH','PWP',95),
25      (92301733017,'HARSH VISHALBHAI TRIVEDI','PWP',85),
26      (92301733027,'VIRAJ PRAKASHBHAI VAGHASIYA','PWP',90),
27      (92301733046,'SHIVAM ATULKUMAR BHATT', 'PWP',93),
28      (92301733058,'DEVENDRASINH DOLATSINH JADEJA','PWP',75)
29  ]
30  cursor.executemany('''INSERT INTO student_record(Enrollment,name,subject,Mark)
31                        VALUES(?,?,?,?)''',student_record)
32  conn.commit()
33
34  cursor.execute('SELECT*FROM student_record')
35  #*-fetch all rows
36  rows=cursor.fetchall()#will be list of tuples
37  #.cursor-as data will be stored in cursor pointer
38  #fetchall-all rows
39  print("All Student Records:")
40  for i in rows:
41      print(i)
42
```

```
43  cursor.execute('SELECT name,subject,Mark FROM student_record WHERE Mark>90')
44  high_marks=cursor.fetchall()
45  print("\n Students with Marks greater than 90:")
46  for i in high_marks:
47      print(i)
48
49  cursor.execute('''UPDATE student_record SET Mark=98
50                    WHERE name='ASHUTOSH KUMAR SINGH' AND subject='PWP' ''')
51  conn.commit()
52  cursor.execute('SELECT name,Mark FROM student_record WHERE name="ASHUTOSH KUMAR SINGH"')
53  update_mark=cursor.fetchone()
54  print(f"\nUpdate Mark for {update_mark[0]}:{update_mark[1]}")
55
56  cursor.execute('''DELETE FROM student_record WHERE name='DEVENDRASINH DOLATSINH JADEJA' ''')
57  conn.commit()
58  cursor.execute('SELECT*FROM student_record WHERE name="DEVENDRASINH DOLATSINH JADEJA"')
59  deleted_name=cursor.fetchone()
60  if(deleted_name is None):
61      print("\n DEVENDRASINH DOLATSINH JADEJA has been succefully deleted.")
62
```

```
63  cursor.execute('''SELECT AVG(Mark) FROM student_record''')
64  avg_mark=cursor.fetchone()[0]
65  print(f"\n The averge mark of student is:${avg_mark:.2f}")
66  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: 92400133037**

```
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab15\sql.py"
Database connected successfully!
All Student Records:
(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)

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

Update Mark for ASHUTOSH KUMAR SINGH:98

 DEVENDRASINH DOLATSINH JADEJA has been succefully deleted.

 The averge mark of student is:$91.50
```

Post Lab Exercise:

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

```python
import sqlite3
conn=sqlite3.connect('student_record.db')
cursor=conn.cursor()
print("Database connected successfully!")

cursor.execute("DROP TABLE IF EXISTS student_record")
cursor.execute("DROP TABLE IF EXISTS subjects")

cursor.execute('''CREATE TABLE IF NOT EXISTS student_record(
                Enrollment INTEGER PRIMARY KEY AUTOINCREMENT,
                name TEXT NOT NULL)''')
cursor.execute('''
                CREATE TABLE subjects(
                SubjectID INTEGER PRIMARY KEY AUTOINCREMENT,
                Enrollment INTEGER,
                Subject TEXT NOT NULL,
                Mark INTEGER NOT NULL,
                FOREIGN KEY (Enrollment) REFERENCES student_record(Enrollment)
                )''')

```

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: 92400133037**

```python
students = [
    ('ASHUTOSH KUMAR SINGH',),
    ('HARSH VISHALBHAI TRIVEDI',),
    ('VIRAJ PRAKASHBHAI VAGHASIYA',),
]
cursor.executemany('INSERT INTO student_record(name) VALUES(?)', students)

subjects = [
    (1, 'PWP', 95),
    (2, 'DBMS', 90),
    (3, 'PWP', 85),
    (4, 'OOP', 88),
    (5, 'PWP', 92),
    (6, 'Maths', 89)
]
cursor.executemany('INSERT INTO subjects( Enrollment,Subject, Mark) VALUES(?,?,?)', subjects)

cursor.execute('''
SELECT s.name, sub.Subject, sub.Mark
FROM student_record s
JOIN subjects sub ON s.Enrollment = sub.Enrollment
''')
rows = cursor.fetchall()
for row in rows:
    print(row)
conn.commit()
conn.close()
```

```
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab15\PostLab.py"
Database connected successfully!
('ASHUTOSH KUMAR SINGH', 'PWP', 95)
('HARSH VISHALBHAI TRIVEDI', 'DBMS', 90)
('VIRAJ PRAKASHBHAI VAGHASIYA', 'PWP', 85)
PS G:\sem-3\python_lab>
```

GITHUB LINK
https://github.com/Heer972005/Python_Lab