	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: 92400133055**

GITHUB

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: 92400133055**

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: 92400133055**

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)
```

OUTPUT:

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: 92400133055**

```python
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 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 multiple employee records
student_record = [
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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)
PS E:\SEM 3\PWP>
```

Fetch Data with Specific Criteria

Let's fetch employees who earn more than 90.

Example

Fetch student got more than 90

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: 92400133055**

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)

OUTPUT:

```
34   conn.commit()
35
36   # Fetch all student records
37   cursor.execute('SELECT * FROM student_record')
38   rows = cursor.fetchall()
39
40   print("All Student Records:")
41   for row in rows:
42       print(row)
43
44   # Fetch students with marks > 90
45   cursor.execute('SELECT name, Subject, Mark FROM student_record WHERE Mark > 90')
46   high_marks = cursor.fetchall()
47
48   print("\nStudents with Marks greater than 90:")
49   for student in high_marks:
50       print(student)
51
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
(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 Student Information

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

Example:

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: 92400133055**

```
# 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]}")
```

OUTPUT:

```
51
52  # Update MArk for Ashutosh kumar (PWP)
53  cursor.execute('''UPDATE student_record SET Mark = 98
54                    WHERE name = 'ASHUTOSH KUMAR SINGH' AND subject = 'PWP' ''')
55
56  # Commit the changes
57  conn.commit()
58  # Verify the update
59  cursor.execute('SELECT name, MArk FROM student_record WHERE name = "ASHUTOSH KUMAR SINGH"')
60  updated_mark = cursor.fetchone()
61  print(f"\nUpdated Mark for {updated_mark[0]}: {updated_mark[1]}")
62
63  # Close connection
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

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

Updated Mark for ASHUTOSH KUMAR SINGH: 98
```

Delete a Student

Let's remove a student from the database.

Example:

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: 92400133055**

# 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.")

```
64   cursor.execute('''DELETE FROM student_record WHERE name = 'DEVENDRASINH DOLATSINH JADEJA' '''
65
66   # Commit the changes
67   conn.commit()
68
69   # Verify the deletion
70   cursor.execute('SELECT * FROM student_record WHERE name = "DEVENDRASINH DOLATSINH JADEJA"')
71   deleted_name = cursor.fetchone()
72
73   if deleted_name is None:
74       print("\nDEVENDRASINH DOLATSINH JADEJA has been successfully deleted.")
75
76   # Close connection
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
(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', 98)
('SHIVAM ATULKUMAR BHATT', 'PWP', 93)

Updated Mark for ASHUTOSH KUMAR SINGH: 98

DEVENDRASINH DOLATSINH JADEJA has been successfully deleted.
```

Calculate Average 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: 92400133055**

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}")

```
76   # Calculate the average Mark
77   cursor.execute('''SELECT AVG(Mark) FROM student_record''')
78   avg_mark = cursor.fetchone()[0]
79
80   print(f"\nThe average mark of students is: ${avg_mark:.2f}")
81
82
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(92301733058, 'DEVENDRASINH DOLATSINH JADEJA', 'PWP', 75)

Students with Marks greater than 90:
('ASHUTOSH KUMAR SINGH', 'PWP', 98)
('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
```

Close the Database Connection

Always close the connection after completing your operations.

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: 92400133055**

Example

# Close the connection
conn.close()

Post Lab Exercise:

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

Class Tutorials > post_lab_15.py > ...

```
47    cursor.execute( SELECT name, Subject, Mark FROM student_record WHERE name =  'ASHUTOSH
48    updated_marks = cursor.fetchall()
49    print("\nUpdated Marks for ASHUTOSH KUMAR SINGH:")
50    for mark in updated_marks:
51        print(mark)
52
53    cursor.execute('''DELETE FROM student_record
54                      WHERE name = 'HARSH VISHALBHAI TRIVEDI' AND Subject = 'Maths' ''')
55    conn.commit()
56
57    cursor.execute('SELECT * FROM student_record WHERE name = "HARSH VISHALBHAI TRIVEDI"')
58    remaining = cursor.fetchall()
59    print("\nRemaining records for HARSH VISHALBHAI TRIVEDI:")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
('ASHUTOSH KUMAR SINGH', 'PWP', 95)
('HARSH VISHALBHAI TRIVEDI', 'Maths', 92)

Updated Marks for ASHUTOSH KUMAR SINGH:
('ASHUTOSH KUMAR SINGH', 'PWP', 98)
('ASHUTOSH KUMAR SINGH', 'DBMS', 88)

Remaining records for HARSH VISHALBHAI TRIVEDI:
(92301733017, 'HARSH VISHALBHAI TRIVEDI', 'PWP', 85)

The average mark across all subjects is: 90.00
PS F:\SEM 3\PWP>
```