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: 92400133110**

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()

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

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: 92400133110**

```
In [1]: 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.

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()
```

```
 7    # Create students table if it doesn't exist
 8    cursor.execute('''CREATE TABLE IF NOT EXISTS student_record (
 9                        Enrollment INTEGER PRIMARY KEY AUTOINCREMENT,
10                        name TEXT NOT NULL,
11                        Subject TEXT NOT NULL,
12                        Mark INTEGER NOT NULL
13                    )''')
14    # Commit the changes
15    conn.commit()
```

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: 92400133110**

```
In [2]:
   ...: 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: 92400133110**

```
# 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()
```

Output:

```
In [3]:
   ...: 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()
   ...:
```

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 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:

```
In [4]:
   ...: cursor.execute('SELECT * FROM student_record')
   ...: rows = cursor.fetchall()
   ...: # Display the results
   ...: print("All Student Records:")
   ...: for row in rows:
   ...:     print(row)
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)
```

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)
```

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: 92400133110**

```
# 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)
```

Output:

```
In [5]:
   ...: 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)

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:

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()

```
# 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()
```

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: 92400133110**

```
In [6]:
   ...: 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]}")

```
# 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:

```
In [7]:
   ...: 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]}")
   ...:

Updated Mark for ASHUTOSH KUMAR SINGH: 98
```

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()

```
# 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()
```

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: 92400133110**

```
In [8]:
   ...: 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.")

```
# 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.")
```

Output:

```
In [9]:
   ...: 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.")
   ...:

DEVENDRASINH 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 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: 92400133110**

```
# 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}")
```

Output:

```
In [10]:
    ...: cursor.execute('''SELECT AVG(Mark) FROM student_record''')
    ...: avg_mark = cursor.fetchone()[0]
    ...:
    ...: print(f"\nThe average mark of students is: {avg_mark:.2f}")

The average mark of students is: 91.50
```

Close the Database Connection

Always close the connection after completing your operations.

Example

# Close the connection
conn.close()

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

Output:

```
In [11]:
    ...: 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: 92400133110**

Post Lab Exercise:

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

Code:

```
1   import sqlite3
2
3   conn = sqlite3.connect('my_record.db')
4   cursor = conn.cursor()
```

Output:

```
In [1]: import sqlite3
   ...:
   ...: conn = sqlite3.connect('my_record.db')
   ...: cursor = conn.cursor()
   ...:
```

Create an Student Table:

Code:

```
5
6   # Create table
7   cursor.execute('''
8       CREATE TABLE IF NOT EXISTS my_record (
9           Enrollment INTEGER,
10          name TEXT NOT NULL,
11          Subject TEXT NOT NULL,
12          Mark INTEGER NOT NULL
13      )
14  ''')
15  conn.commit()
16
```

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: 92400133110**

Output:

```
In [2]:
   ...: cursor.execute('''
   ...:     CREATE TABLE IF NOT EXISTS my_record (
   ...:         Enrollment INTEGER,
   ...:         name TEXT NOT NULL,
   ...:         Subject TEXT NOT NULL,
   ...:         Mark INTEGER NOT NULL
   ...:     )
   ...: ''')
   ...: conn.commit()
```

Insert Student Data:

Code:

```
# Student records
my_record = [
    (92400133147, 'Deva Harsha Veeranki', 'PWP', 98),
    (92400133147, 'Deva Harsha Veeranki', 'ICE', 95),
    (92400133147, 'Deva Harsha Veeranki', 'DMGT', 92),
    (92400133147, 'Deva Harsha Veeranki', 'DSC', 89),
    (92400133147, 'Deva Harsha Veeranki', 'SS', 86),
    (92400133147, 'Deva Harsha Veeranki', 'COA', 84)
]

# Insert records
cursor.executemany('''
    INSERT INTO my_record (Enrollment, name, Subject, Mark)
    VALUES (?, ?, ?, ?)
''', my_record)
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: 92400133110**

Output:

```
In [21]:
    ...: my_record = [
    ...:     (92400133147, 'Deva Harsha Veeranki', 'PWP', 98),
    ...:     (92400133147, 'Deva Harsha Veeranki', 'ICE', 95),
    ...:     (92400133147, 'Deva Harsha Veeranki', 'DMGT', 92),
    ...:     (92400133147, 'Deva Harsha Veeranki', 'DSC', 89),
    ...:     (92400133147, 'Deva Harsha Veeranki', 'SS', 86),
    ...:     (92400133147, 'Deva Harsha Veeranki', 'COA', 84)
    ...: ]
    ...:
    ...: # Insert records
    ...: cursor.executemany('''
    ...:     INSERT INTO my_record (Enrollment, name, Subject, Mark)
    ...:     VALUES (?, ?, ?, ?)
    ...: ''', my_record)
    ...: conn.commit()
    ...:
```

Fetch Student Data:

Code:

```
# Fetch all records
cursor.execute('SELECT * FROM my_record')
rows = cursor.fetchall()
print("All Student Subjects Records:")
for row in rows:
    print(row)
```

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: 92400133110**

Output:

```
In [22]:
    ...: cursor.execute('SELECT * FROM my_record')
    ...: rows = cursor.fetchall()
    ...: print("All Student Subjects Records:")
    ...: for row in rows:
    ...:     print(row)
    ...:
All Student Subjects Records:
(92400133147, 'Deva Harsha Veeranki', 'PWP', 98)
(92400133147, 'Deva Harsha Veeranki', 'ICE', 95)
(92400133147, 'Deva Harsha Veeranki', 'DMGT', 92)
(92400133147, 'Deva Harsha Veeranki', 'DSC', 89)
(92400133147, 'Deva Harsha Veeranki', 'SS', 86)
(92400133147, 'Deva Harsha Veeranki', 'COA', 84)
```

Fetch Data with Specific Criteria:

Code:

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

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: 92400133110**

Output:

```
In [23]:
    ...: cursor.execute('SELECT name, Subject, Mark FROM my_record WHERE Mark > 90')
    ...: high_marks = cursor.fetchall()
    ...: print("\nSubjects with Marks greater than 90:")
    ...: for subject in high_marks:
    ...:     print(subject)

Subjects with Marks greater than 90:
('Deva Harsha Veeranki', 'PWP', 98)
('Deva Harsha Veeranki', 'ICE', 95)
('Deva Harsha Veeranki', 'DMGT', 92)
```

Update Student Information:

Code:

```
# Update Mark for COA
cursor.execute('''
    UPDATE my_record
    SET Mark = 98
    WHERE Enrollment = 92400133147 AND Subject = 'COA'
''')
conn.commit()
```

Output:

```
In [24]:
    ...: cursor.execute('''
    ...:     UPDATE my_record
    ...:     SET Mark = 98
    ...:     WHERE Enrollment = 92400133147 AND Subject = 'COA'
    ...: ''')
    ...: conn.commit()
    ...:
```

Verify the update:

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: 92400133110**

Code:

```
# Verify the update
cursor.execute('''
    SELECT Subject, Mark FROM my_record
    WHERE Enrollment = 92400133147 AND Subject = 'COA'
''')
updated = cursor.fetchone()
print(f"\nUpdated Mark for COA: {updated[1]}")

cursor.execute('DELETE FROM my_record WHERE Enrollment = 92400133147')
conn.commit()
```

Output:

```
In [25]:
    ...: cursor.execute('''
    ...:     SELECT Subject, Mark FROM my_record
    ...:     WHERE Enrollment = 92400133147 AND Subject = 'COA'
    ...: ''')
    ...: updated = cursor.fetchone()
    ...: print(f"\nUpdated Mark for COA: {updated[1]}")

Updated Mark for COA: 98
```

Delete a subject:

Code:

```
# Delete marks for 'SS' subject
cursor.execute('''
    DELETE FROM my_record
    WHERE Enrollment = 92400133147 AND Subject = 'SS'
''')
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: 92400133110**

Output:

```
In [46]:
    ...: cursor.execute('''
    ...:     DELETE FROM my_record
    ...:     WHERE Enrollment = 92400133147 AND Subject = 'SS'
    ...: ''')
    ...: conn.commit()
    ...:
```

Verify deletion:

Code:

```
# Verify deletion
cursor.execute('''
    SELECT * FROM my_record
    WHERE Enrollment = 92400133147 AND Subject = 'SS'
''')
deleted = cursor.fetchone()
if deleted is None:
    print("\n' SS ' subject record has been successfully deleted")
```

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: 92400133110**

Output:

```
In [48]:
    ...: cursor.execute('''
    ...:     SELECT * FROM my_record
    ...:     WHERE Enrollment = 92400133147 AND Subject = 'SS'
    ...: ''')
    ...: deleted = cursor.fetchone()
    ...: if deleted is None:
    ...:     print("\n' SS ' subject record has been successfully deleted")
    ...:

' SS ' subject record has been successfully deleted
```

Calculate the average marks:

Code:

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

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

Output:

```
In [49]:
    ...: cursor.execute('''SELECT AVG(Mark) FROM my_record''')
    ...: avg_mark = cursor.fetchone()[0]
    ...:
    ...: print(f"\nThe average mark of students is: {avg_mark:.2f}")

The average mark of students is: 94.40
```

Close the connection:

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: 92400133110**

Code:

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

Output:

```
In [50]:
    ...: conn.close()
    ...:
```

GitHub:

https://github.com/hemanthsingampalli/PWP-Lab-Exercises.git