

 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:92510133019

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 	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:92510133019

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:92510133019

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()
```

 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:92510133019

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 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:92510133019

Close the Database Connection

Always close the connection after completing your operations.

Example

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

Post Lab Exercise:

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

Code:-

```
import sqlite3

# Connect to database
conn = sqlite3.connect('student_record.db')
cursor = conn.cursor()

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

# Add New Student With Multiple Subjects
enroll = int(input("Enter Enrollment number: "))
name = input("Enter student name: ")
num_subjects = int(input("How many subjects to enroll? "))

subjects = []
for i in range(num_subjects):
    subject = input(f"Enter subject #{i+1}: ")
```

 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:92510133019

```
mark = int(input(f'Enter mark for {subject}: '))
subjects.append((enroll, name, subject, mark))
```

```
cursor.executemany("""
    INSERT OR REPLACE INTO student_record (Enrollment, name, Subject, Mark)
    VALUES (?, ?, ?, ?)
""", subjects)
conn.commit()
print("\nStudent data inserted successfully.\n")
```

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

```
# Students with Marks greater 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 Marks
update_name = input("\nEnter name to update marks: ")
update_subject = input("Enter subject to update: ")
new_mark = int(input("Enter new mark: "))
cursor.execute("""
    UPDATE student_record SET Mark = ?
    WHERE name = ? AND Subject = ?
""", (new_mark, update_name, update_subject))
conn.commit()
print(f'{update_name}'s mark in {update_subject} updated to {new_mark}.")
```

```
# Delete Student
delete_name = input("\nEnter name to delete from records: ")
cursor.execute('DELETE FROM student_record WHERE name = ?', (delete_name,))
conn.commit()
```

```
# Check deletion
cursor.execute('SELECT * FROM student_record WHERE name = ?', (delete_name,))
```

 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:92510133019

```
if cursor.fetchone() is None:
    print(f' {delete_name} has been successfully deleted.")
```

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

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

Output:-

```
PS D:\pwp> & C:/Users/premj/AppData/Local/Programs/Python/Python313/python.exe d:/pwp/exp_15.py
Enter Enrollment number: 92510133019
Enter student name: Prem Joshi
How many subjects to enroll? 2
Enter subject #1: PWP
Enter mark for PWP: 100
Enter subject #2: SS
Enter mark for SS: 99

Student data inserted successfully.

All Student Records:
(92510133019, 'Prem Joshi', 'PWP', 100)
(92510133019, 'Prem Joshi', 'SS', 99)

Students with Marks greater than 90:
('Prem Joshi', 'PWP', 100)
('Prem Joshi', 'SS', 99)

Enter name to update marks: 99
Enter subject to update: DMGT
Enter new mark: 100
99's mark in DMGT updated to 100.

Enter name to delete from records: SS
SS has been successfully deleted.

The average mark of all students is: 99.50
```