**Inserting a new Record**

**Example 2:** To accept user input for the values in the table:

1. Instead of adding known values, you can also accept user input for these values. Assuming that the database MySchool is created and contains the table student, we start by creating a connection:

```
import sqlite3
MySchool=sqlite3.connect('schooltest.db')
curschool=MySchool.cursor()
```

2. To accept user input, we use variables to store each of the values.

```
import sqlite3
MySchool=sqlite3.connect('schooltest.db')
curschool=MySchool.cursor()
mysid= int(input("Enter ID: "))
myname=input("Enter name: ")
myhouse=int(input("Enter house: "))
mymarks=float(input("Enter marks: "))
```

3. We now replaces the fixed VALUES in the INSERT query with the variables, mysid, myname, myhouse and mymarks. To do this, we use the DB-API's parameter substitution. We put a ? as a placeholder wherever we want to use a value and then give a tuple of values as the second argument to the cursor's execute() method.

```
import sqlite3
MySchool=sqlite3.connect('schooltest.db')
curschool=MySchool.cursor()
mysid= int(input("Enter ID: "))
myname=input("Enter name: ")
myhouse=int(input("Enter house: "))
mymarks=float(input("Enter marks: "))
curschool.execute("INSERT INTO student (StudentID, name, house, marks)
VALUES (?,?,?,?);", (mysid,myname,myhouse,mymarks))
```

4. We now commit the changes.

```
import sqlite3
MySchool=sqlite3.connect('schooltest.db')
curschool=MySchool.cursor()
mysid= int(input("Enter ID: "))
myname=input("Enter name: ")
myhouse=int(input("Enter house: "))
mymarks=float(input("Enter marks: "))
curschool.execute("INSERT INTO student (StudentID, name, house, marks)
VALUES (?,?,?,?);", (mysid,myname,myhouse,mymarks))
MySchool.commit()
```