



SQLite

INSERT YOUR NAME HERE

Introduction

- ▶ SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring.
- ▶ It provides an SQL interface compliant
- ▶ You do not need to install this module separately because it is shipped by default
- ▶ 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.

Python sqlite3 module APIs

- ▶ `sqlite3.connect`
 - ▶ Opens a connection to the SQLite database file.
- ▶ `connection.cursor`
 - ▶ Creates a cursor which will be used throughout of your database programming with Python
- ▶ `cursor.execute`
 - ▶ This routine executes an SQL statement. The SQL statement may be parameterized
- ▶ `cursor.executescript`
 - ▶ executes multiple SQL statements at once provided in the form of script.

Python sqlite3 module APIs ...

- ▶ `connection.commit()`
 - ▶ This method commits the current transaction.
- ▶ `connection.rollback()`
 - ▶ Rolls back any changes to the database since the last call to `commit()`.
- ▶ `connection.close()`
 - ▶ Closes the database connection. Note that this does not automatically call `commit()`. If you just close your database connection without calling `commit()` first, your changes will be lost!

Connect To Database

- ▶ `#!/usr/bin/python`
- ▶ `import sqlite3`
- ▶ `conn = sqlite3.connect('test.db')`
- ▶ `print "Opened database successfully";`

Create a Table

```
▶ #!/usr/bin/python
▶ import sqlite3
▶ conn = sqlite3.connect('test.db')
▶ print "Opened database successfully";
▶ conn.execute("""CREATE TABLE COMPANY
▶      (ID INT PRIMARY KEY   NOT NULL,
▶      NAME      TEXT   NOT NULL,
▶      AGE       INT    NOT NULL,
▶      ADDRESS   CHAR(50),
▶      SALARY    REAL);""")
▶ print "Table created successfully";
▶ conn.close()
```

INSERT Operation

- ▶ Following Python program shows how to create records in the COMPANY table created in the above example.



SQLite-Insert.py

- ▶ When the above program is executed, it will create the given records in the COMPANY table and it will display the following two lines –
 - ▶ Opened database successfully
 - ▶ Records created successfully

SELECT Operation

- ▶ `#!/usr/bin/python`
- ▶ `import sqlite3`
- ▶ `conn = sqlite3.connect('test.db')`
- ▶ `print "Opened database successfully";`
- ▶ `cursor = conn.execute("SELECT id, name, address, salary from COMPANY")`
- ▶ `for row in cursor:`
- ▶ `print "ID = ", row[0]`
- ▶ `print "NAME = ", row[1]`
- ▶ `print "ADDRESS = ", row[2]`
- ▶ `print "SALARY = ", row[3], "\n"`
- ▶ `print "Operation done successfully";`
- ▶ `conn.close()`

Output

- ▶ When the above program is executed, it will produce the following result.
 - ▶ Opened database successfully
 - ▶ ID = 1
 - ▶ NAME = Paul
 - ▶ ADDRESS = California
 - ▶ SALARY = 20000.0

- ▶ ID = 2
- ▶ NAME = Allen
- ▶ ADDRESS = Texas
- ▶ SALARY = 15000.0

UPDATE Operation

- ▶ `#!/usr/bin/python`
- ▶ `import sqlite3`
- ▶ `conn = sqlite3.connect('test.db')`
- ▶ `print "Opened database successfully";`
- ▶ `conn.execute("UPDATE COMPANY set SALARY = 25000.00 where ID = 1")`
- ▶ `conn.commit`
- ▶ `print "Total number of rows updated :", conn.total_changes`

- ▶ `cursor = conn.execute("SELECT id, name, address, salary from COMPANY")`
- ▶ `for row in cursor:`
- ▶ `print "ID = ", row[0]`
- ▶ `print "NAME = ", row[1]`
- ▶ `print "ADDRESS = ", row[2]`
- ▶ `print "SALARY = ", row[3], "\n"`
- ▶ `print "Operation done successfully";`
- ▶ `conn.close()`

Output

- ▶ Opened database successfully
- ▶ Total number of rows updated : 1
- ▶ ID = 1
- ▶ NAME = Paul
- ▶ ADDRESS = California
- ▶ SALARY = 25000.0

- ▶ ID = 2
- ▶ NAME = Allen
- ▶ ADDRESS = Texas
- ▶ SALARY = 15000.0

- ▶ ID = 3
- ▶ NAME = Teddy
- ▶ ADDRESS = Norway
- ▶ SALARY = 20000.0

DELETE Operation

- ▶ `#!/usr/bin/python`
- ▶ `import sqlite3`
- ▶ `conn = sqlite3.connect('test.db')`
- ▶ `print "Opened database successfully";`
- ▶ `conn.execute("DELETE from COMPANY where ID = 2;")`
- ▶ `conn.commit()`
- ▶ `print "Total number of rows deleted :", conn.total_changes`
- ▶ `cursor = conn.execute("SELECT id, name, address, salary from COMPANY")`
- ▶ `for row in cursor:`
- ▶ `print "ID = ", row[0]`
- ▶ `print "NAME = ", row[1]`
- ▶ `print "ADDRESS = ", row[2]`
- ▶ `print "SALARY = ", row[3], "\n"`
- ▶ `print "Operation done successfully";`
- ▶ `conn.close()`

Output

- ▶ Opened database successfully
- ▶ Total number of rows deleted : 1
- ▶ ID = 1
- ▶ NAME = Paul
- ▶ ADDRESS = California
- ▶ SALARY = 20000.0

- ▶ ID = 3
- ▶ NAME = Teddy
- ▶ ADDRESS = Norway
- ▶ SALARY = 20000.0

Thanks