

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.

To use the module, start by creating a Connection object that represents the database. Here the data will be stored in the example.db file:

Python SQLite – Cursor Object:

Cursor Object

It is an object that is used to make the connection for executing SQL queries. It acts as middleware between SQLite database connection and SQL query. It is created after giving connection to SQLite database.

Syntax: `cursor_object=connection_object.execute("sql query");`

Python SQLite – Create Table

we will discuss how can we create tables in the SQLite database from the Python program using the sqlite3 module. In SQLite database we use the following syntax to create a table:

```
CREATE TABLE database_name.table_name(  
    column1 datatype PRIMARY KEY(one or more columns),  
    column2 datatype,  
    column3 datatype,  
    .....,  
    columnN datatype);
```

Python SQLite – Insert Data

we insert data in a table in the SQLite database from Python using the sqlite3 module. The SQL INSERT INTO statement of SQL is used to insert a new row in a table. There are two ways of using the INSERT INTO statement for inserting rows:

- **Only values:** The first method is to specify only the value of data to be inserted without the column names.

```
INSERT INTO table_name VALUES (value1, value2, value3,...);
```

table_name: name of the table.

value1, value2,.. : value of first column, second column,... for the new record

- **Column names and values both:** In the second method we will specify both the columns which we want to fill and their corresponding values as shown below:

```
INSERT INTO table_name (column1, column2, column3,..) VALUES ( value1, value2, value3,..);
```

table_name: name of the table.

column1: name of first column, second column ...

value1, value2, value3 : value of first column, second column,... for the new record

Python SQLite – Select Data from Table

we will discuss, select statement of the Python SQLite module. This statement is used to retrieve data from an SQLite table and this returns the data contained in the table. In SQLite the syntax of Select Statement is:

```
SELECT * FROM table_name;
```

** : means all the column from the table*

To select specific column replace * with the column name or column names.

Python SQLite – Deleting Data in Table

we will discuss how we can delete data in the table in the SQLite database from the Python program using the sqlite3 module. In SQLite database we use the following syntax to delete data from a table:

```
DELETE FROM table_name [WHERE Clause]
```

```
#delete data
```

```
cursor_obj.execute("DELETE FROM GEEK WHERE Score < 15")
```

/* Insert Data into Table*/

```
# Import module
```

```
import sqlite3
```

```
# Connecting to sqlite
```

```
conn = sqlite3.connect('geeks2.db')
```

```
# Creating a cursor object using the
```

```
# cursor() method
```

```
cursor = conn.cursor()
```

```
# Creating table
```

```
table = """CREATE TABLE STUDENT(NAME VARCHAR(255), CLASS VARCHAR(255),  
SECTION VARCHAR(255));"""
```

```
cursor.execute(table)
```

```
# Queries to INSERT records.
```

```
cursor.execute("INSERT INTO STUDENT VALUES ('Raju', '7th', 'A')")
```

```
cursor.execute("INSERT INTO STUDENT VALUES ('Shyam', '8th', 'B')")
```

```
cursor.execute("INSERT INTO STUDENT VALUES ('Baburao', '9th', 'C')")
```

```
# Display data inserted
```

```
print("Data Inserted in the table: ")
```

```
data=cursor.execute("SELECT * FROM STUDENT")
```

```
for row in data:
```

```
print(row)
```

```
# Commit your changes in the database
```

```
conn.commit()
```

```
# Closing the connection
```

```
conn.close()
```

/* Select Data from Table*/

```
import sqlite3
```

```
# Connecting to sqlite
```

```
# connection object
```

```
connection_obj = sqlite3.connect('geek.db')
```

```
# cursor object
```

→ create

→ insert

→ display

```
cursor_obj = connection_obj.cursor()
```

```
# to select all column we will use  
statement = "SELECT * FROM GEEK"
```

```
cursor_obj.execute(statement)
```

```
print("All the data")
```

```
output = cursor_obj.fetchall()
```

```
for row in output:
```

```
print(row)
```

```
connection_obj.commit()
```

```
# Close the connection
```

```
connection_obj.close()
```

/* Deleting the Data */

```
import sqlite3
```

```
# Connecting to sqlite
```

```
# connection object
```

```
connection_obj = sqlite3.connect('geek.db')
```

```
# cursor object
```

```
cursor_obj = connection_obj.cursor()
```

```
#delete data
```

```
cursor_obj.execute("DELETE FROM GEEK WHERE Score < 15")
```

```
connection_obj.commit()
```

```
# Close the connection
```

```
connection_obj.close()
```

Python SQLite – Update Data

we can update data in tables in the SQLite database using Python – sqlite3 module.

The UPDATE statement in SQL is used to update the data of an existing table in the database. We can update single columns as well as multiple columns using UPDATE statement as per our requirement.

Syntax:

```
UPDATE table_name SET column1 = value1, column2 = value2,...
```

WHERE condition;

In the above syntax, the SET statement is used to set new values to the particular column, and the WHERE clause is used to select the rows for which the columns are needed to be updated.

Example 1: Python SQLite program to update a particular column. In this example, we are first going to create an EMPLOYEE table and insert values into it. Then we are going to set the income of employees to 5000 whose age is less than 25

```
# Import module
```

```

import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('gfg1.db')

# Creating a cursor object using
# the cursor() method
cursor = conn.cursor()

# Creating table
table =
"""CREATE TABLE EMPLOYEE(FIRST_NAME VARCHAR(255),LAST_NAME VARCHAR(255),AGE int, SEX
VARCHAR(255), INCOME int);"""
cursor.execute(table)

# Queries to INSERT records.
cursor.execute( "INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('Anand', 'Choubey', 25, 'M', 10000)")

cursor.execute( "INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('Mukesh', 'Sharma', 20, 'M', 9000)")

cursor.execute( "INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('Ankit', 'Pandey', 24, 'M', 6300)")

cursor.execute( "INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('Subhdra ', 'Singh', 26, 'F', 8000)")

cursor.execute( "INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME)
VALUES ('Tanu', 'Mishra', 24, 'F', 6500)")

# Display data inserted
print("EMPLOYEE Table: ")
data = cursor.execute("SELECT * FROM EMPLOYEE")
for row in data:
    print(row)

# Updating
cursor.execute("UPDATE EMPLOYEE SET INCOME = 5000 WHERE Age<25;")
print('\nAfter Updating...\n')

# Display data
print("EMPLOYEE Table: ")
data = cursor.execute("SELECT * FROM EMPLOYEE")
for row in data:
    print(row)

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()

```

Example 3: In the below program we update multiple columns using the UPDATE statement. In this example, we are first going to create a STAFF table and insert values into it. Then we are going to update all the columns i.e. all the attributes of the staff whose department is Computer

```
# Import module
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('gfg3.db')

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Creating table
table = """CREATE TABLE STAFF(NAME VARCHAR(255), AGE int,DEPARTMENT VARCHAR(255));"""
cursor.execute(table)

# Queries to INSERT records.
cursor.execute("""INSERT INTO STAFF VALUES('Anand', 45, 'Chemistry')""")
cursor.execute("""INSERT INTO STAFF VALUES('Ravi', 32, 'Physics')""")
cursor.execute("""INSERT INTO STAFF VALUES('Chandini', 32, 'Computer')""")
cursor.execute("""INSERT INTO STAFF VALUES('Latika', 40, 'Maths')""")

# Display data inserted
print("STAFF Table: ")
data=cursor.execute("""SELECT * FROM STAFF""")
for row in data:
    print(row)

# Updating
cursor.execute("""UPDATE STAFF SET NAME = 'Ram', AGE = 30,DEPARTMENT = 'Biology' WHERE
DEPARTMENT = 'Computer';""")
print('\nAfter Updating...\n')

# Display data
print("STAFF Table: ")
data=cursor.execute("""SELECT * FROM STAFF""")
for row in data:
    print(row)

# Commit your changes in the database
conn.commit()

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

Example 4: In the below program we create the previous table and update the name and age of the staff whose department is Chemistry.

```
# Import module
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('gfg4.db')
```

```

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Creating table
table = """CREATE TABLE STAFF(NAME VARCHAR(255), AGE int,DEPARTMENT VARCHAR(255));"""
cursor.execute(table)

# Queries to INSERT records.
cursor.execute("""INSERT INTO STAFF VALUES('Anand', 45, 'Chemistry')""")
cursor.execute("""INSERT INTO STAFF VALUES('Ravi', 32, 'Physics')""")
cursor.execute("""INSERT INTO STAFF VALUES('Chandini', 32, 'Computer')""")
cursor.execute("""INSERT INTO STAFF VALUES('Latika', 40, 'Maths')""")

# Display data inserted
print("STAFF Table: ")
data=cursor.execute("""SELECT * FROM STAFF""")
for row in data:
    print(row)

# Updating
cursor.execute("""UPDATE STAFF SET NAME = 'Chandini',AGE = 32 WHERE DEPARTMENT =
'Chemistry';""")
print("\nAfter Updating...\n")

# Display data
print("STAFF Table: ")
data=cursor.execute("""SELECT * FROM STAFF""")
for row in data:
    print(row)

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()

```

Example 5: Below program depicts the use of the UPDATE statement without the WHERE statement. In this program, we create the STUDENT table and insert values into it. After that, we update the SECTION column of all the students by assigning it to X.

```

# Import module
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('gfg5.db')

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Creating table
table = """CREATE TABLE STUDENT(NAME VARCHAR(255), CLASS VARCHAR(255),SECTION
VARCHAR(255));"""
cursor.execute(table)

```

```

# Queries to INSERT records.
cursor.execute("""INSERT INTO STUDENT VALUES ('Raju', '7th', 'A')""")
cursor.execute("""INSERT INTO STUDENT VALUES ('Shyam', '8th', 'B')""")
cursor.execute("""INSERT INTO STUDENT VALUES ('Baburao', '9th', 'C')""")

# Display data inserted
print("STUDENT Table: ")
data=cursor.execute("""SELECT * FROM STUDENT""")
for row in data:
    print(row)

# Updating
cursor.execute("""UPDATE STUDENT SET SECTION = 'X'""")
print("\nAfter Updating...\n")

# Display data
print("STUDENT Table: ")
data=cursor.execute("""SELECT * FROM STUDENT""")
for row in data:
    print(row)

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()

```

Python SQLite – Update Specific Column

In order to update a particular column in a table in SQL, we use the **UPDATE query**. The UPDATE statement in SQL is used to update the data of an existing table in the database. We can update single columns as well as multiple columns using UPDATE statement as per our requirement.

Syntax:

UPDATE table_name

SET column1 = value1, column2 = value2..., column n = value n

WHERE [condition];

importing sqlite3 module

```
import sqlite3
```

create connection by using object to

connect with gfg database

```
connection = sqlite3.connect('gfg.db')
```

query to create a table named STUDENT

```
connection.execute(""" CREATE TABLE STUDENTS
(SID INT PRIMARY KEY NOT NULL,
SNAME TEXT NOT NULL,
SAGE INT NOT NULL,
ADDRESS CHAR(50));
""")
```

insert query to insert student details

in the above table

```
connection.execute( "INSERT INTO STUDENTS VALUES (1, 'mohan pavan', 22, 'ponnur' )")

connection.execute( "INSERT INTO STUDENTS VALUES (2, 'sudheer', 28, 'chebrolu' )")

connection.execute("INSERT INTO STUDENTS VALUES (3, 'mohan', 22, 'tenali' )")

# creating cursor object to display all
# the data in the table
cursor = connection.execute("SELECT * from STUDENTS")

# display data
print('\nOriginal Table:')
for row in cursor:
    print(row)

# update query to update sname to sravan
# where id = 1
connection.execute("UPDATE STUDENTS set SNAME = 'sravan' where SID = 1")
connection.commit()

# creating cursor object to display all
# the data in the table
cursor = connection.execute("SELECT * from STUDENTS")

# display data
print('\nUpdated Table:')
for row in cursor:
    print(row)
```