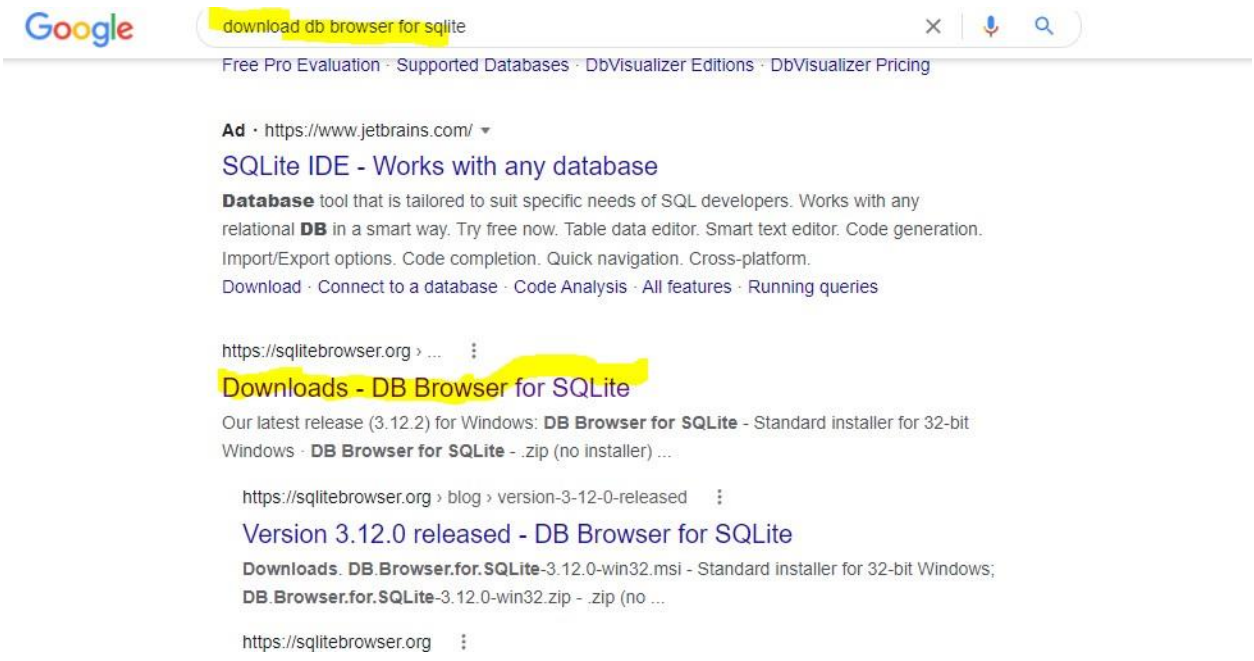
CRUD Operation with SQLite and Python



Downloads

(**Please** consider sponsoring us on Patreon 😄)

Windows

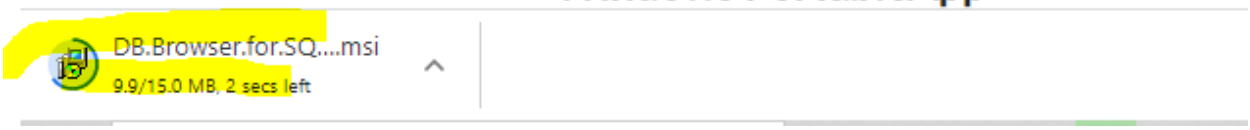
Our latest release (3.12.2) for Windows:

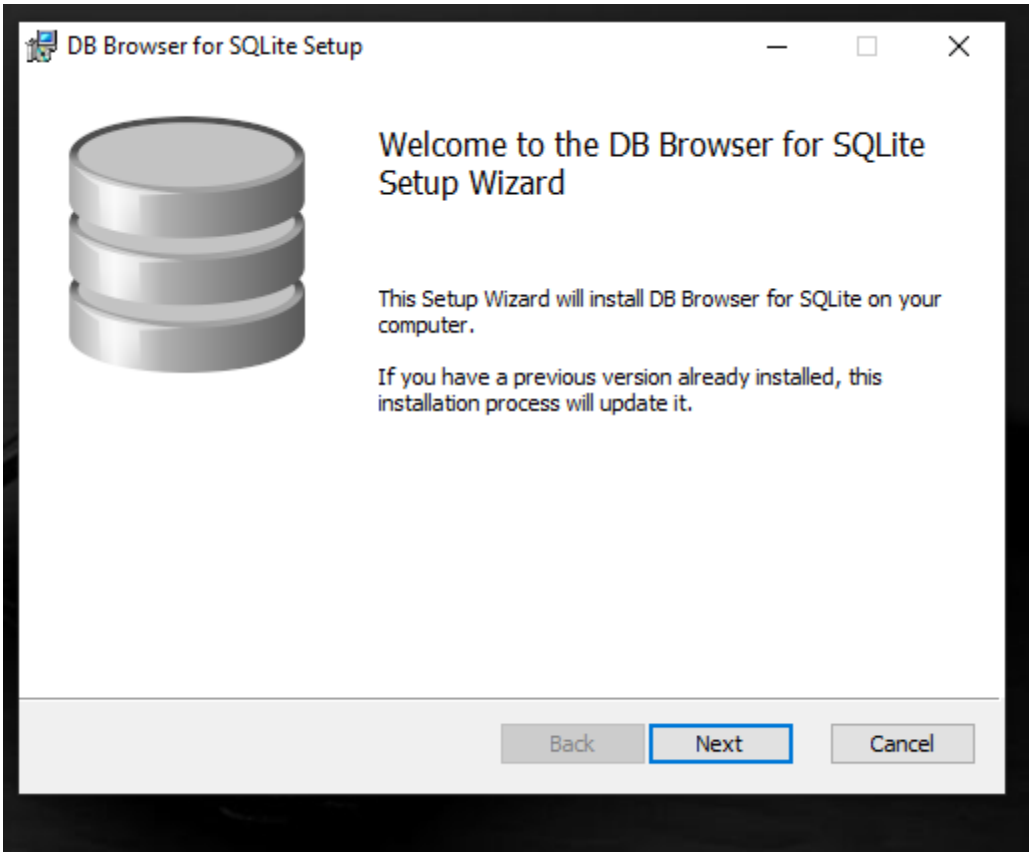
- DB Browser for SQLite - Standard installer for 32-bit Windows
- DB Browser for SQLite - .zip (no installer) for 32-bit Windows
- DB Browser for SQLite - Standard installer for 64-bit Windows
- DB Browser for SQLite - .zip (no installer) for 64-bit Windows

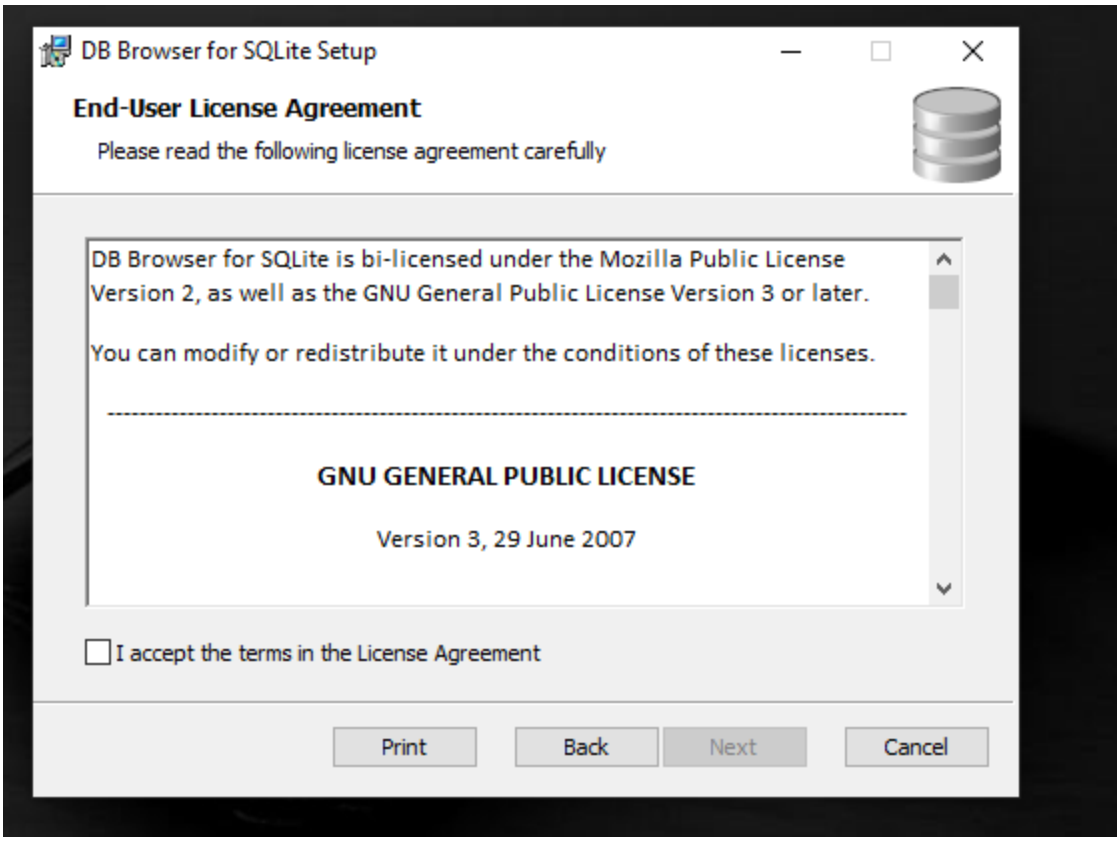
Windows PortableApp

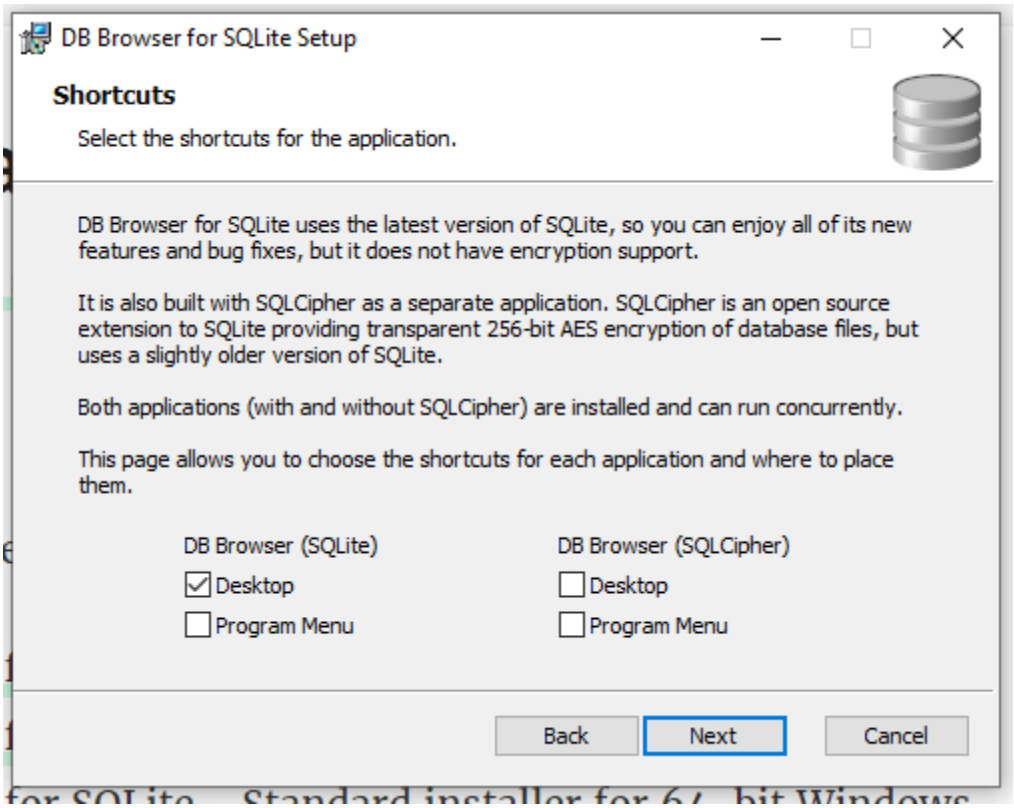
- DB Browser for SQLite - .zip (no
- DB Browser for SQLite - Standar
- DB Browser for SQLite - .zip (no

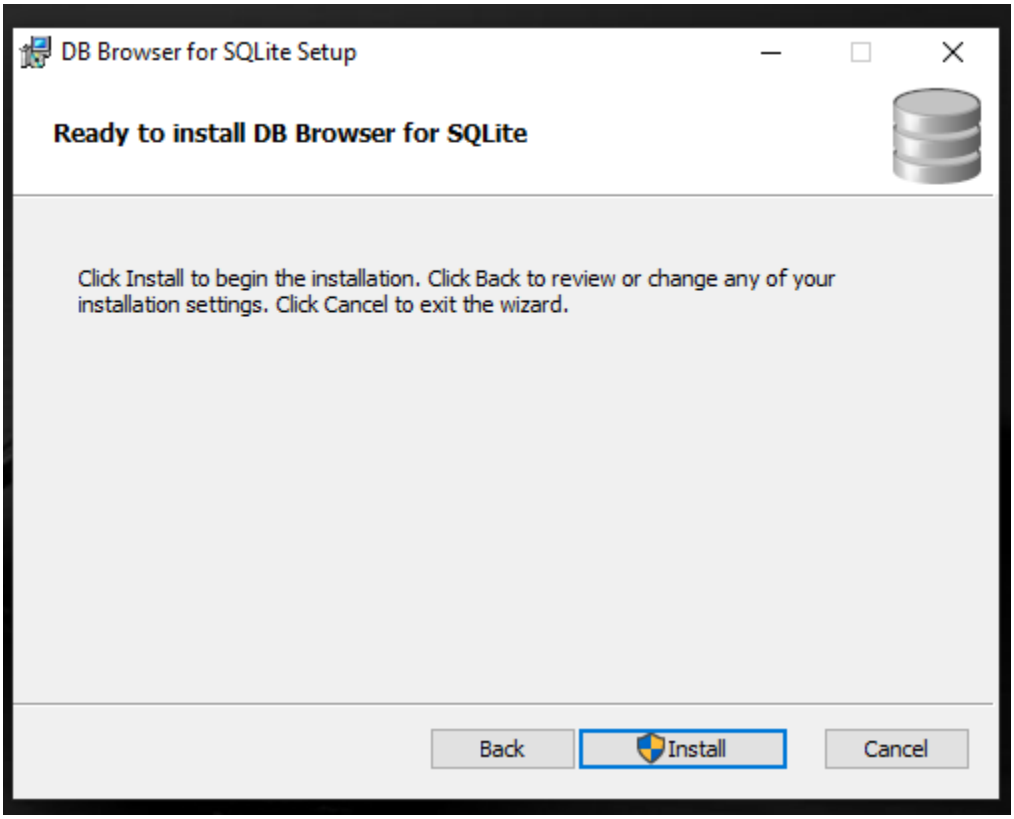
Windows PortableApp

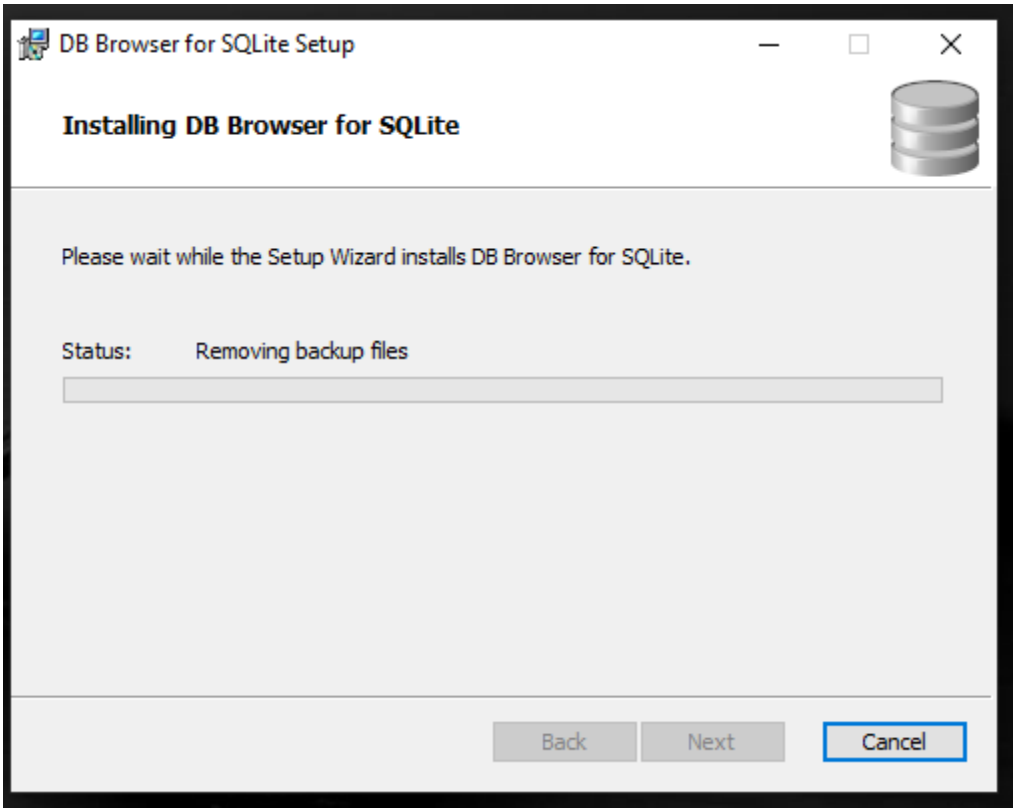


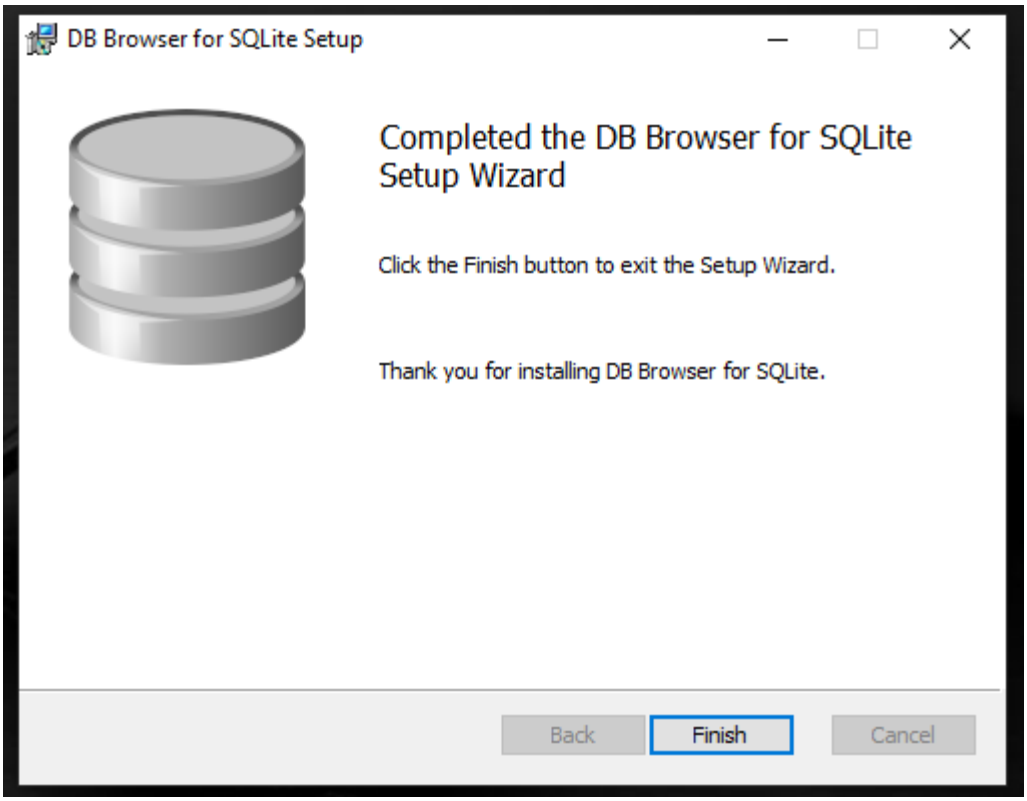

DB Browser for SQLite Setup
Welcome to the DB Browser for SQLite Setup Wizard
This Setup Wizard will install DB Browser for SQLite on your computer.
If you have a previous version already installed, this installation process will update it.
Back
Next
Cancel


DB Browser for SQLite Setup
End-User License Agreement
Please read the following license agreement carefully
DB Browser for SQLite is bi-licensed under the Mozilla Public License Version 2, as well as the GNU General Public License Version 3 or later.
You can modify or redistribute it under the conditions of these licenses.
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
I accept the terms in the License Agreement
Print
Back
Next
Cancel

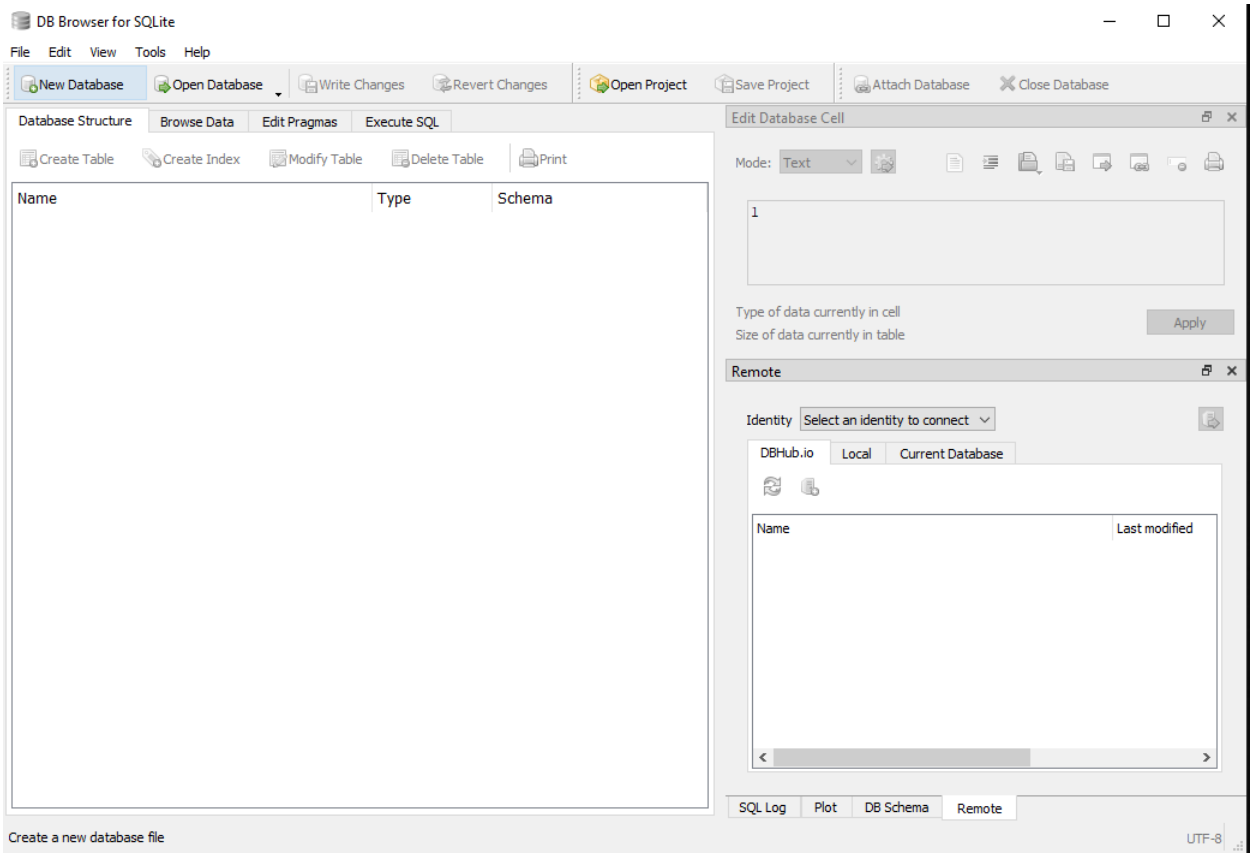

DB Browser for SQLite Setup
Shortcuts
Select the shortcuts for the application.
DB Browser for SQLite uses the latest version of SQLite, so you can enjoy all of its new features and bug fixes, but it does not have encryption support.
It is also built with SQLCipher as a separate application. SQLCipher is an open source extension to SQLite providing transparent 256-bit AES encryption of database files, but uses a slightly older version of SQLite.
Both applications (with and without SQLCipher) are installed and can run concurrently.
This page allows you to choose the shortcuts for each application and where to place them.
DB Browser (SQLite)
Desktop
Program Menu
DB Browser (SQLCipher)
Desktop
Program Menu
Back
Next
Cancel

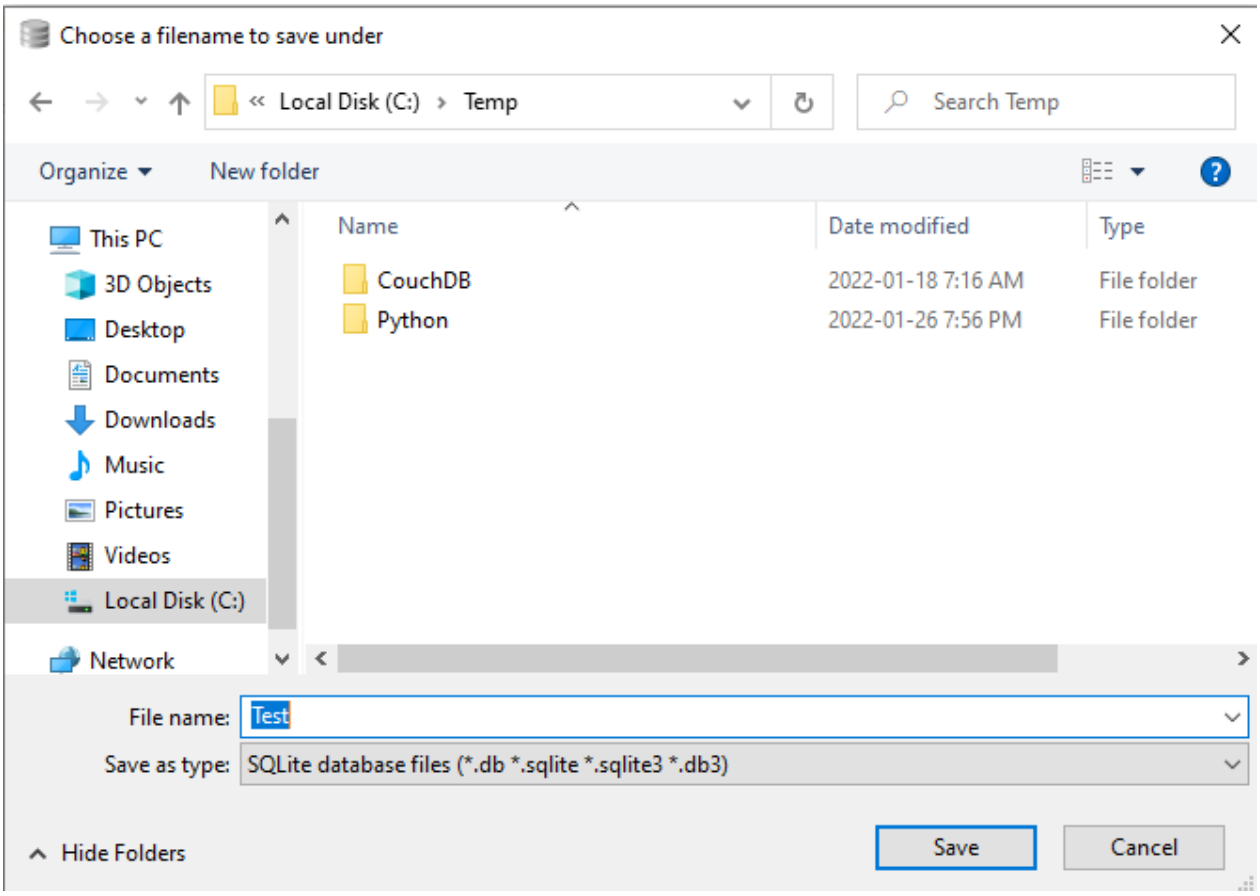

DB Browser for SQLite Setup
Ready to install DB Browser for SQLite
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

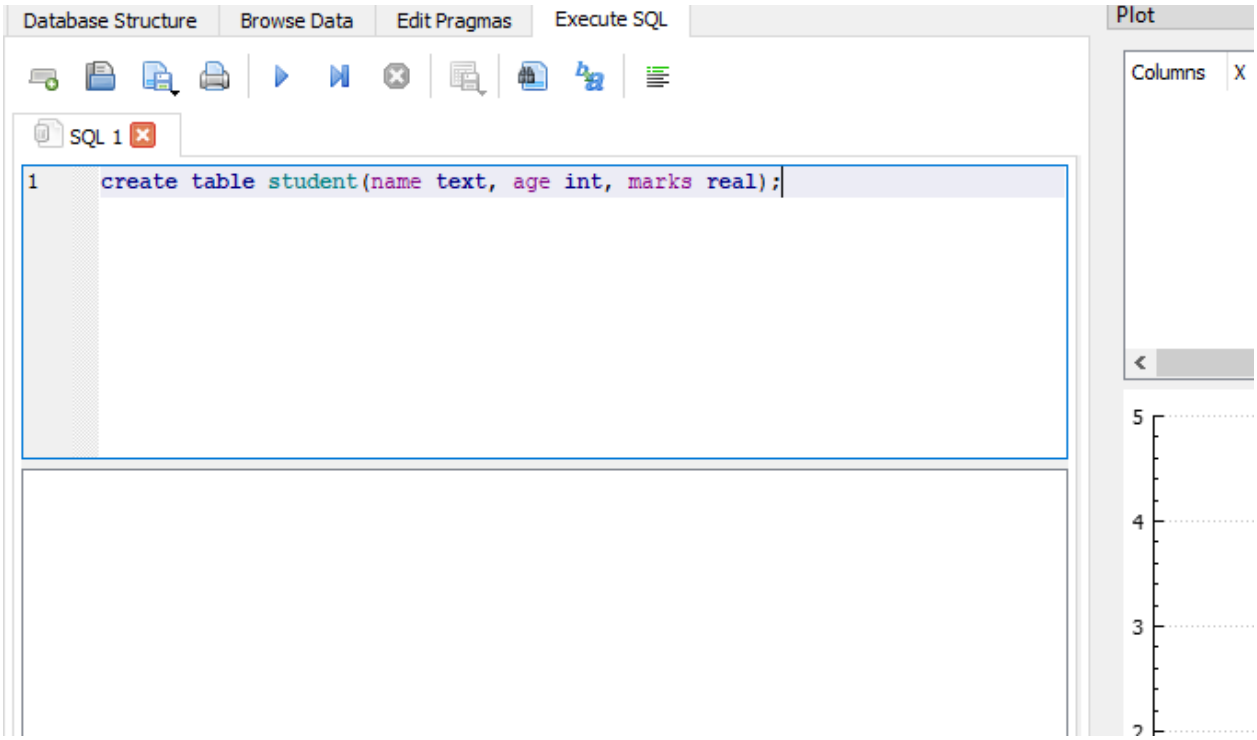

DB Browser for SQLite Setup
Installing DB Browser for SQLite
Please wait while the Setup Wizard installs DB Browser for SQLite.
Status:
Removing backup files
Back
Next
Cancel

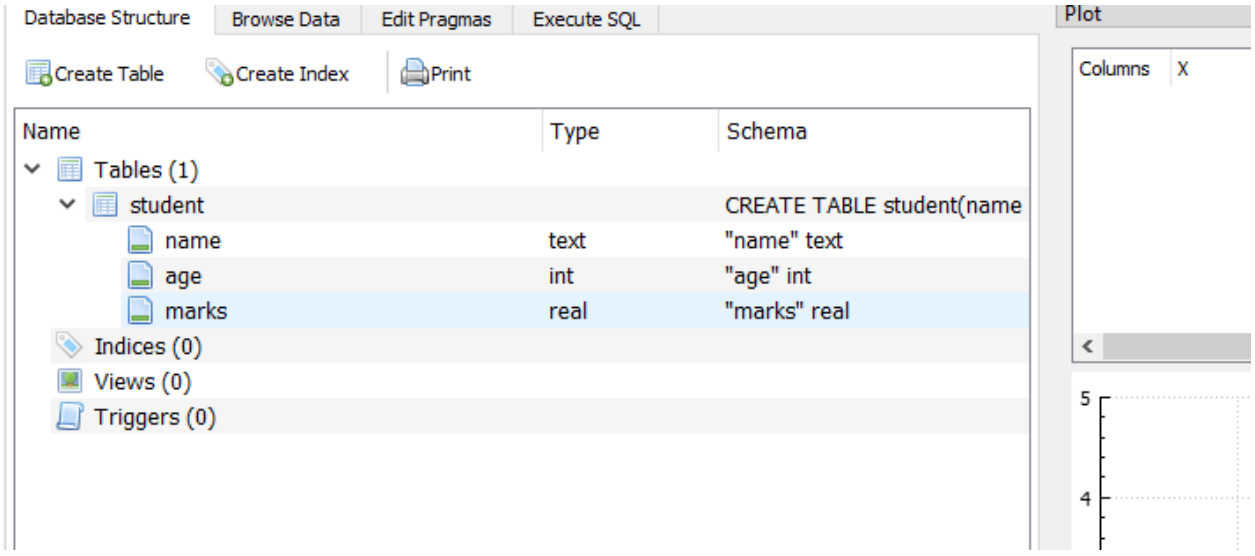

DB Browser for SQLite Setup
Completed the DB Browser for SQLite Setup Wizard
Click the Finish button to exit the Setup Wizard.
Thank you for installing DB Browser for SQLite.
Back
Finish
Cancel

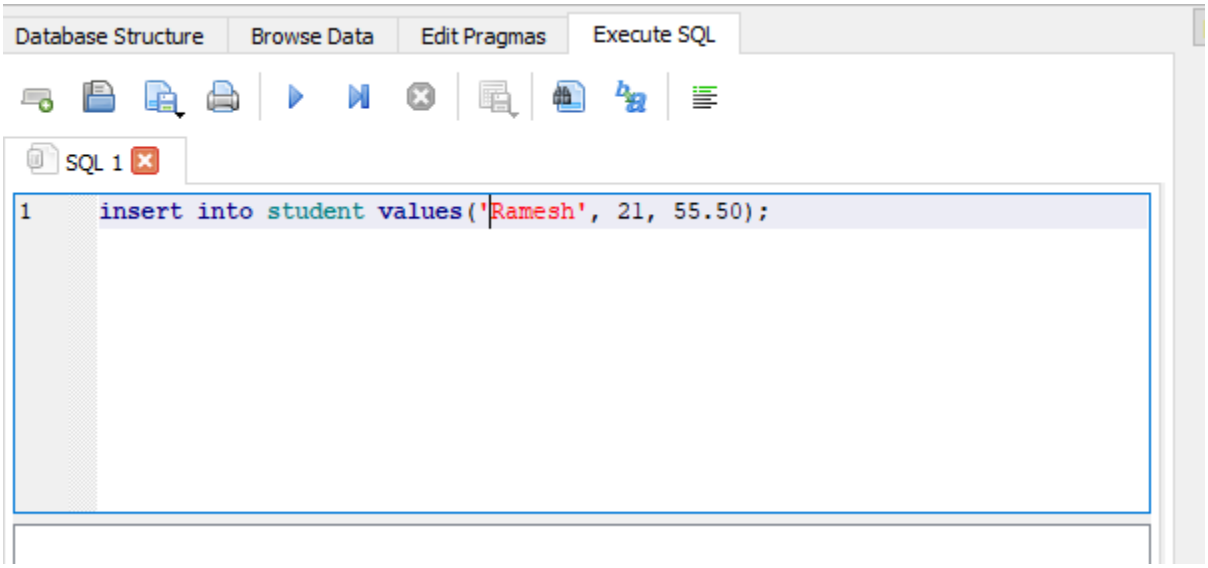

Recycle Bin
CCleaner
Apache NetBeans I...
DB Browser (SQLite)
SumatraPDF
Notepad++
Kindle
LAMB_22W
python2
Word_2016
TBC_22W
ImageGlass Spider

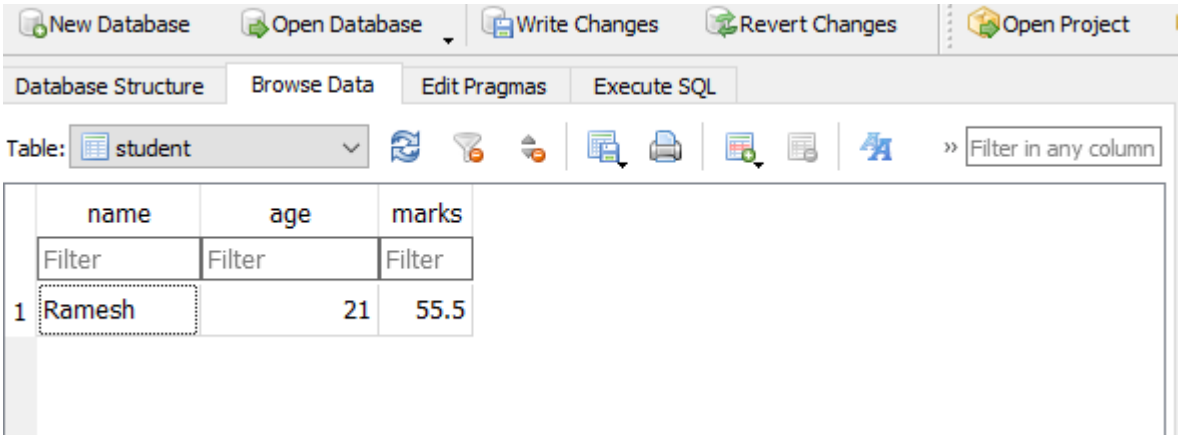

DB Browser for SQLite
File
Edit
View
Tools
Help
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Create Table
Create Index
Modify Table
Delete Table
Print
Name
Type
Schema
Edit Database Cell
Mode:
Text
1
Type of data currently in cell
Size of data currently in table
Apply
Remote
Identity
Select an identity to connect
DBHub.io
Local
Current Database
Name
Last modified
SQL Log
Plot
DB Schema
Remote
Create a new database file
UTF-8

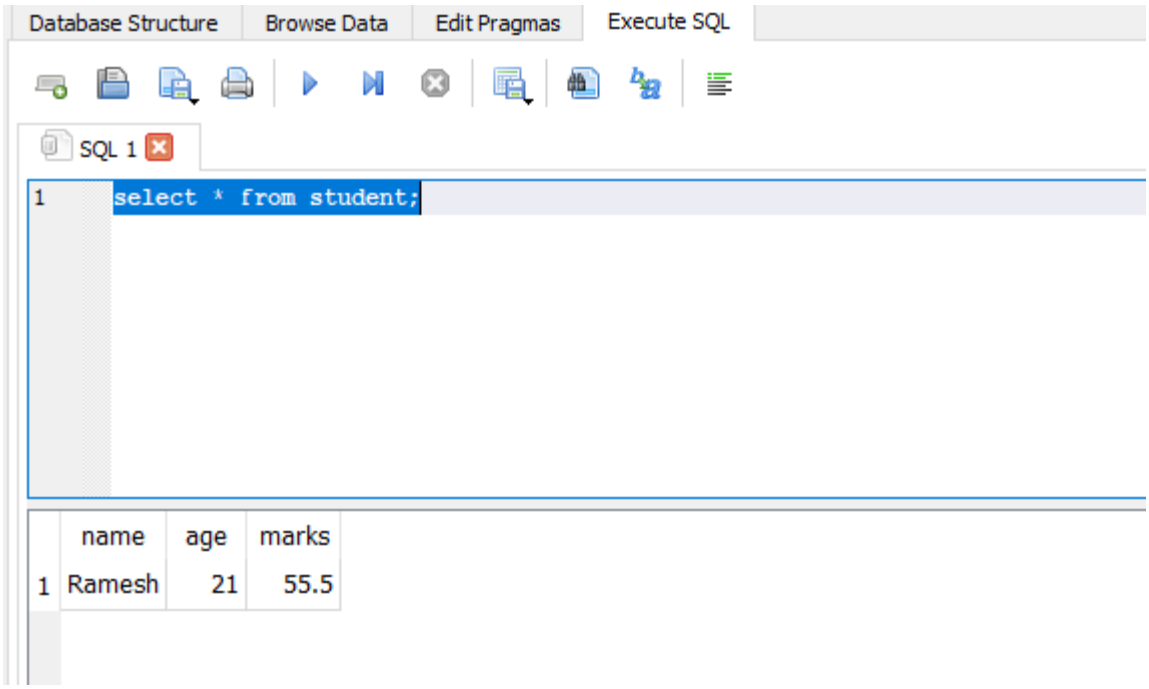

Choose a filename to save under
« Local Disk (C:) › Temp
Search Temp
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network
Name
Date modified
Type
CouchDB
2022-01-18 7:16 AM
File folder
Python
2022-01-26 7:56 PM
File folder
File name:
Test
Save as type:
SQLite database files (*.db *.sqlite *.sqlite3 *.db3)
Hide Folders
Save
Cancel


Database Structure
Browse Data
Edit Pragmas
Execute SQL
Plot
Columns
X
SQL 1
1
create table student(name text, age int, marks real);
5
4
3

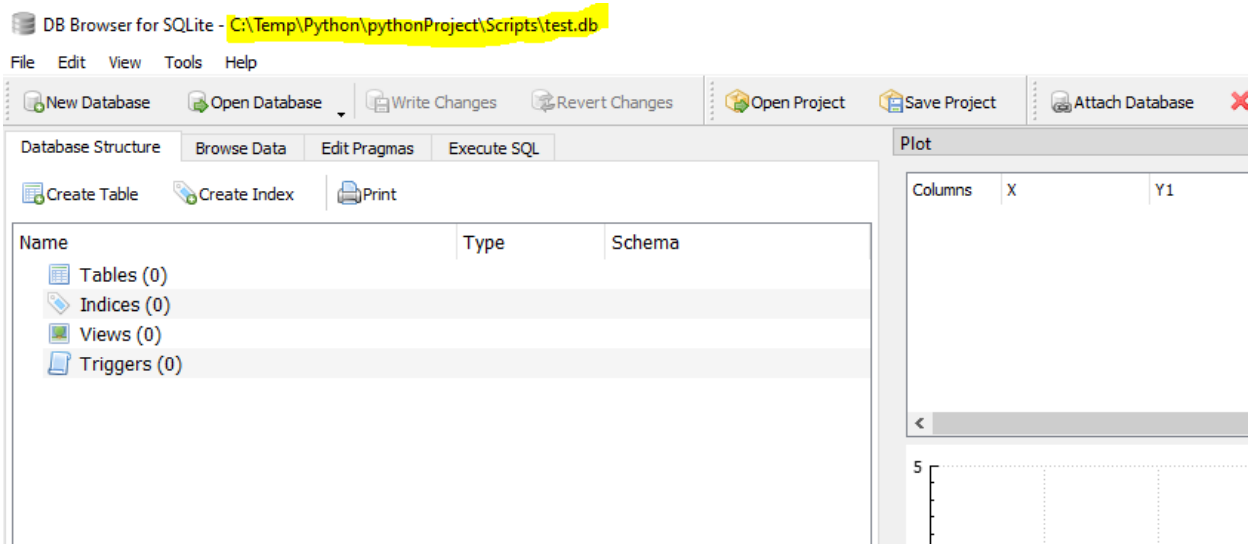

Database Structure
Browse Data
Edit Pragmas
Execute SQL
Plot
Create Table
Create Index
Print
Columns
X
Name
Type
Schema
Tables (1)
student
CREATE TABLE student(name
name
text
"name" text
age
int
"age" int
marks
real
"marks" real
Indices (0)
Views (0)
Triggers (0)
5
4

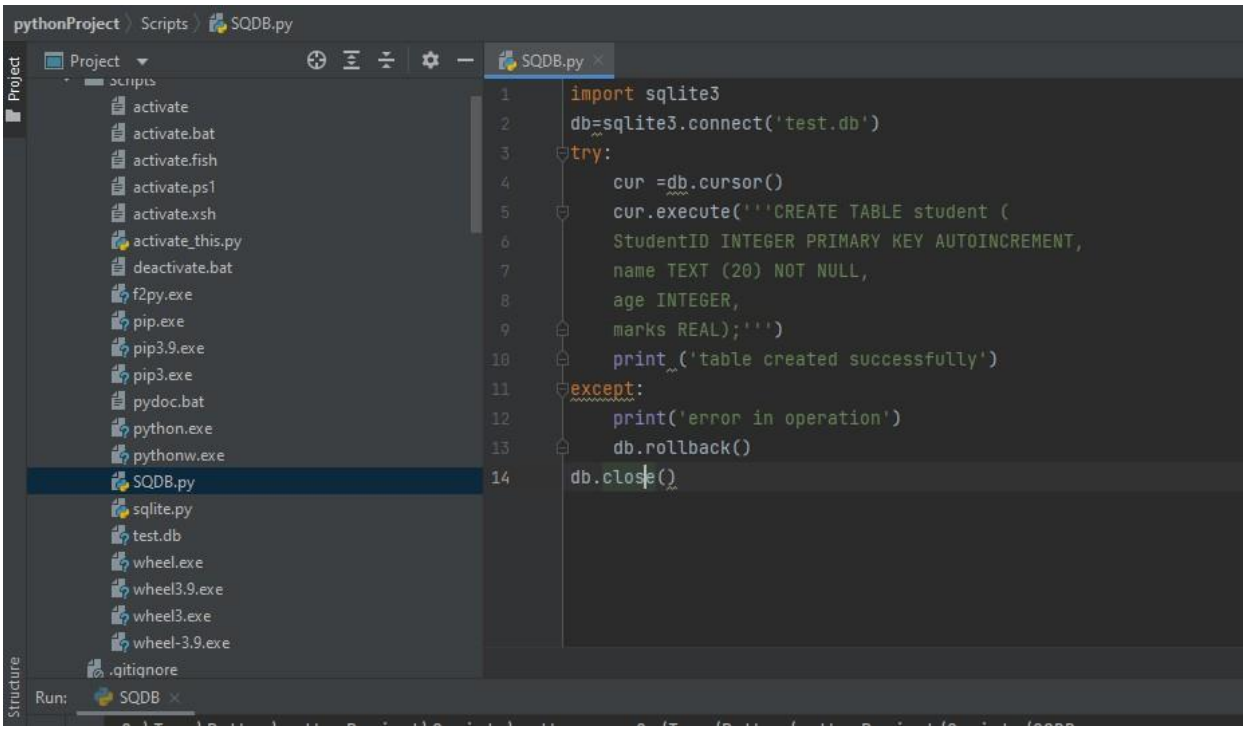

Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
1
insert into student values('Ramesh', 21, 55.50);

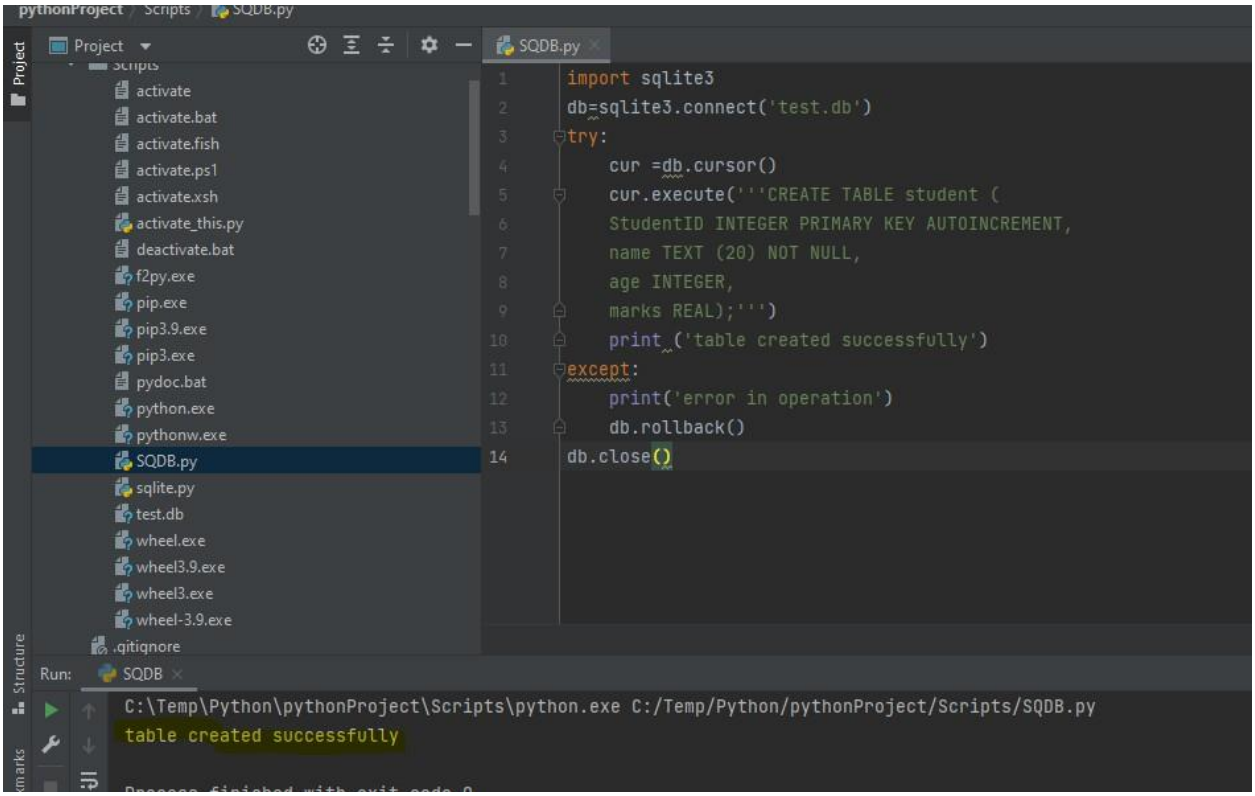

New Database
Open Database
Write Changes
Revert Changes
Open Project
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table:
student
Filter in any column
name
age
marks
Filter
Filter
Filter
1
Ramesh
21
55.5



If you want to create a table using python

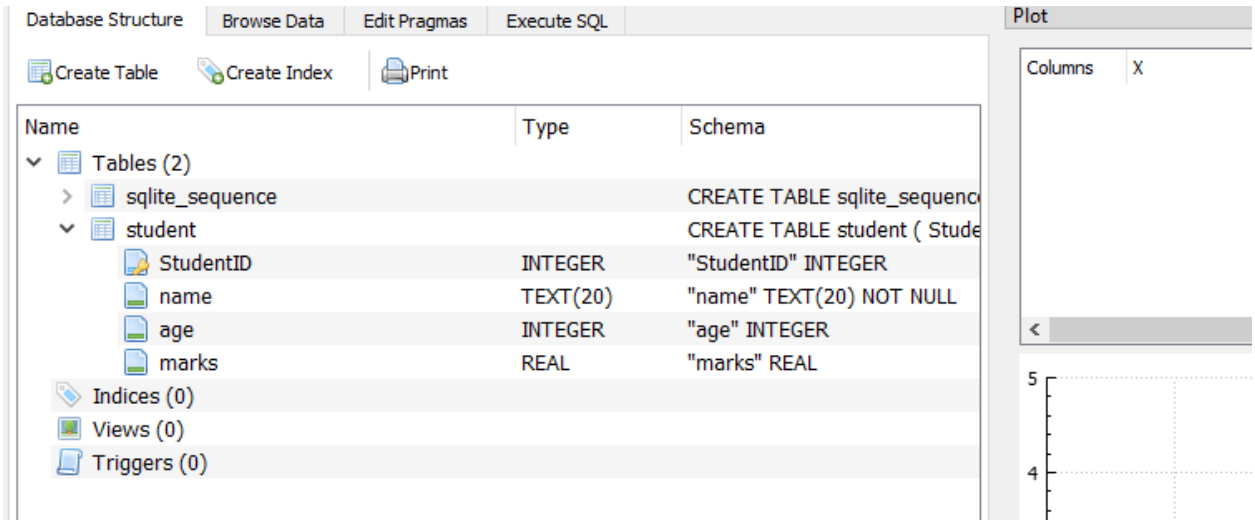


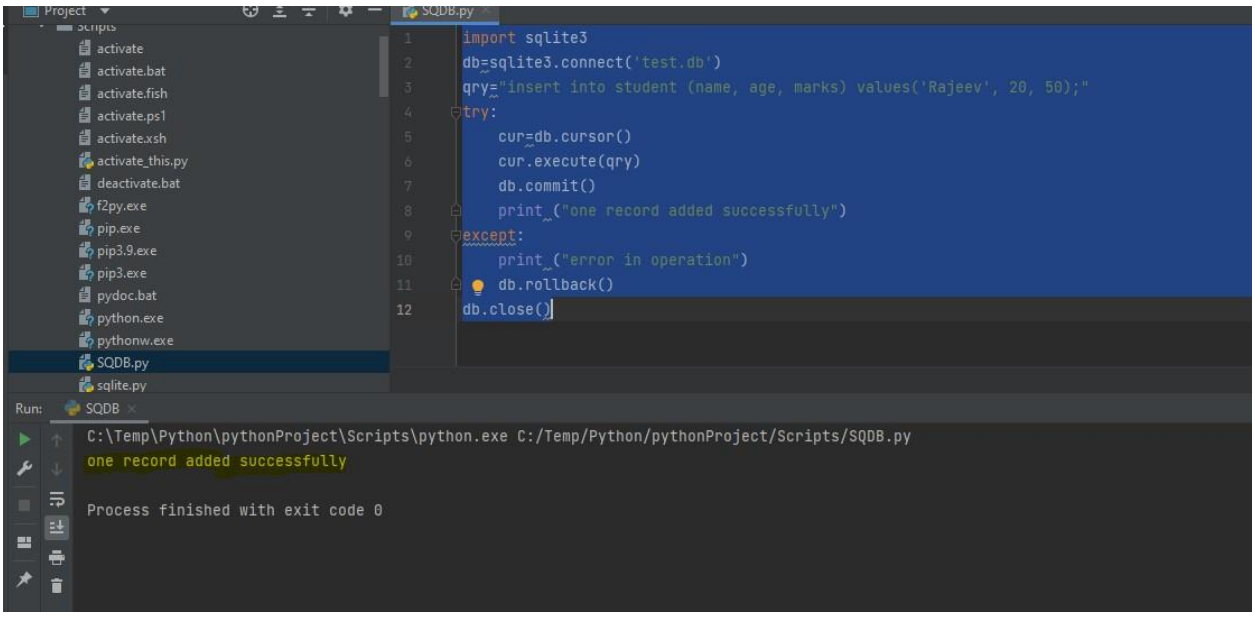

pythonProject
Scripts
SQDB.py
Project
activate
activate.bat
activate.fish
activate.ps1
activate.xsh
activate_this.py
deactivate.bat
f2py.exe
pip.exe
pip3.9.exe
pip3.exe
pydoc.bat
python.exe
pythonw.exe
SQDB.py
sqlite.py
test.db
wheel.exe
wheel3.9.exe
wheel3.exe
wheel-3.9.exe
.qitignore
import sqlite3
db=sqlite3.connect('test.db')
try:
cur =db.cursor()
cur.execute('''CREATE TABLE student (
StudentID INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT (20) NOT NULL,
age INTEGER,
marks REAL);''')
print ('table created successfully')
except:
print('error in operation')
db.rollback()
db.close()
Run:
SQDB
Structure

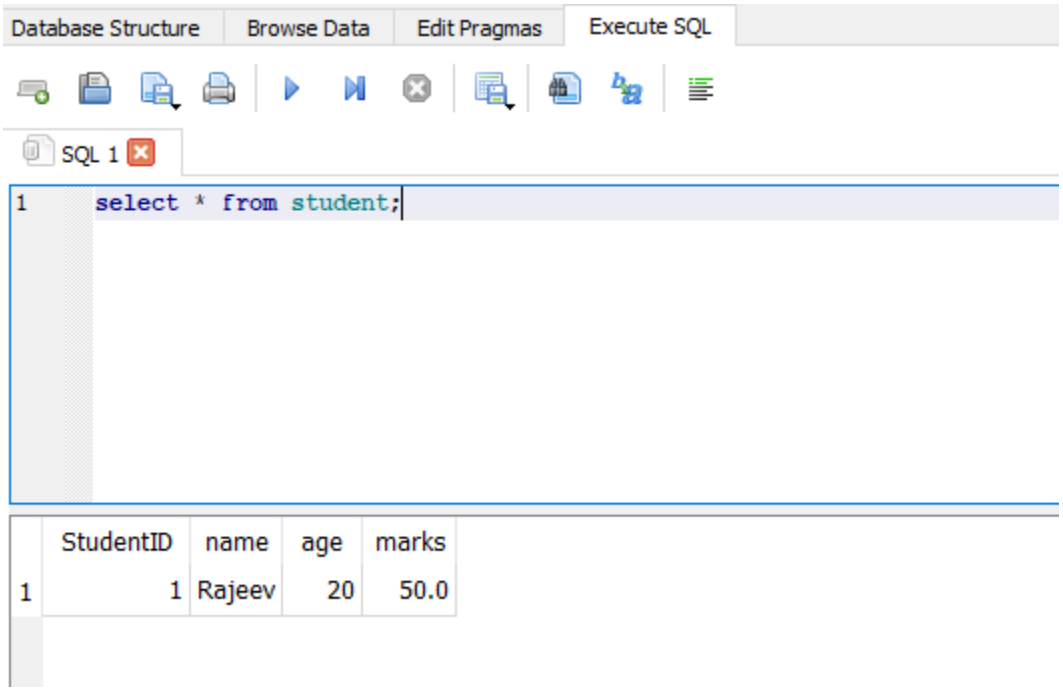

pythonProject
Scripts
SQDB.py
Project
activate
activate.bat
activate.fish
activate.ps1
activate.xsh
activate_this.py
deactivate.bat
f2py.exe
pip.exe
pip3.9.exe
pip3.exe
pydoc.bat
python.exe
pythonw.exe
SQDB.py
sqlite.py
test.db
wheel.exe
wheel3.9.exe
wheel3.exe
wheel-3.9.exe
.qitignore
import sqlite3
db=sqlite3.connect('test.db')
try:
cur =db.cursor()
cur.execute('''CREATE TABLE student (
StudentID INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT (20) NOT NULL,
age INTEGER,
marks REAL);''')
print ('table created successfully')
except:
print('error in operation')
db.rollback()
db.close()
Run:
SQDB
C:\Temp\Python\pythonProject\Scripts\python.exe C:/Temp/Python/pythonProject/Scripts/SQDB.py
table created successfully
Structure

Insert a Record (Create)

Once again, the `execute()` method of the cursor object should be called with a string argument representing the INSERT query syntax. We have created a student table having three fields: name, age and marks. The string holding the INSERT query is defined as:



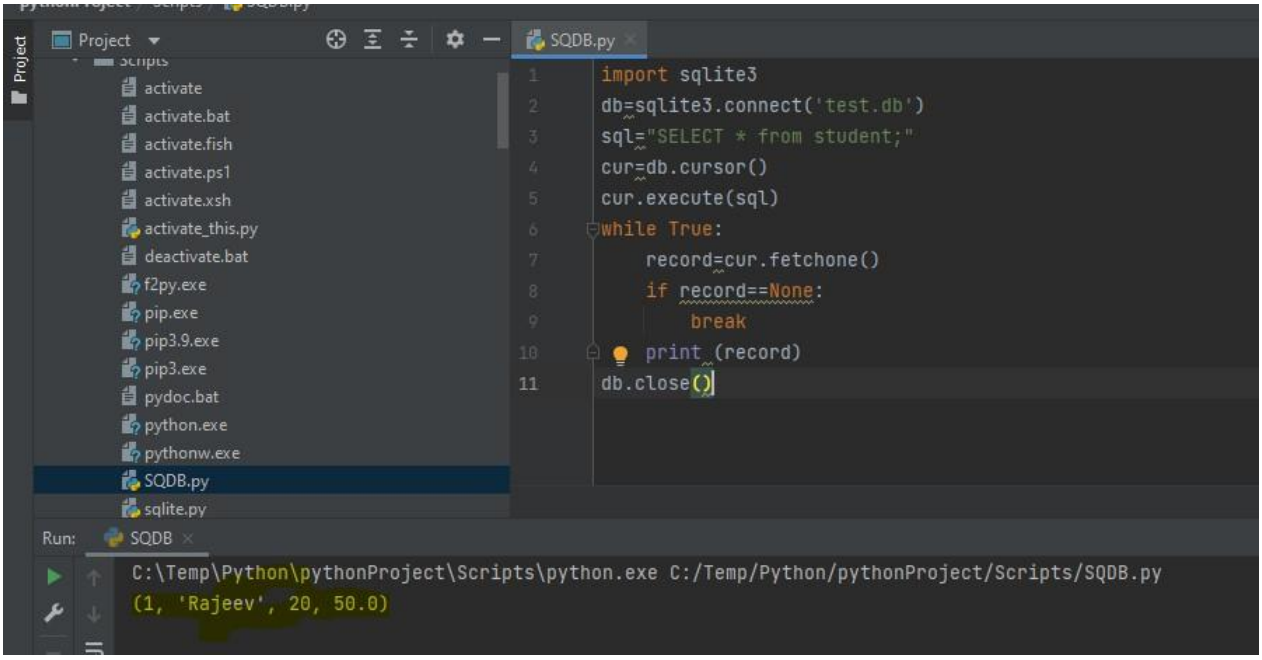




Retrieve Records (Read)

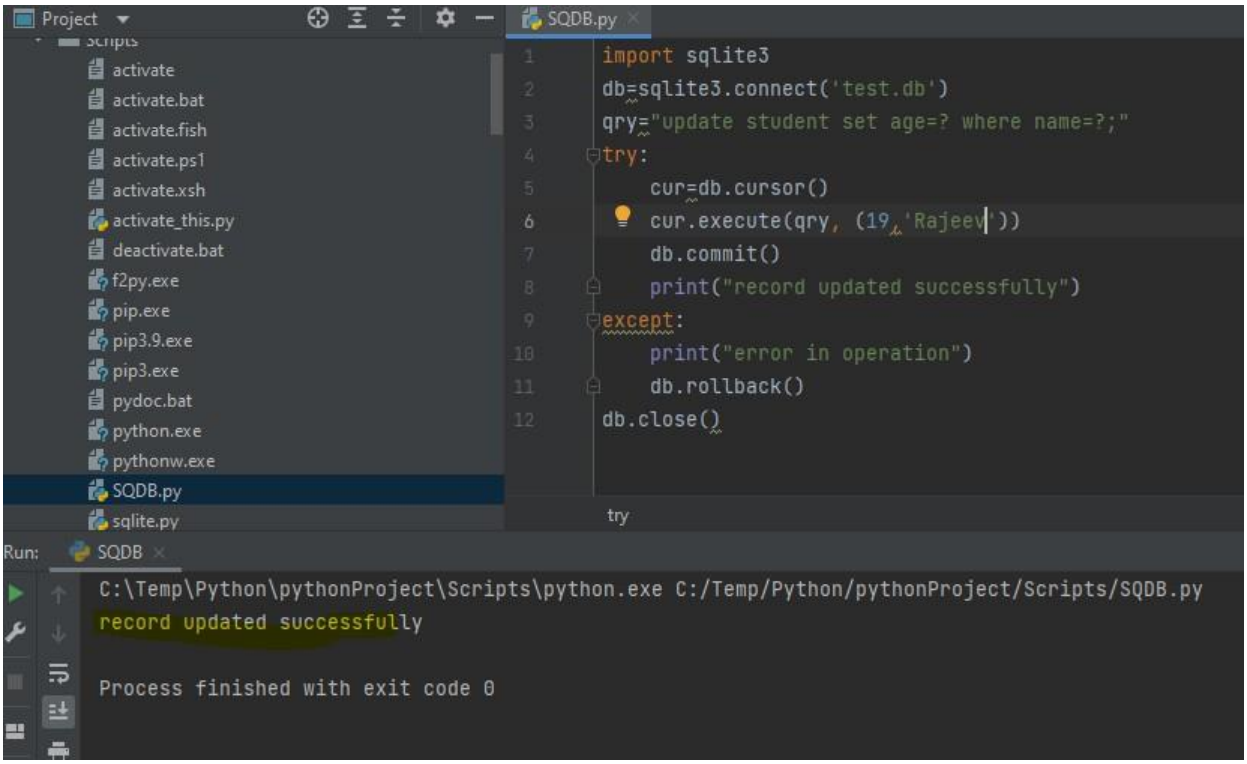
When the query string holds a SELECT query, the `execute()` method forms a result set object containing the records returned. Python DB-API defines two methods to fetch the records:

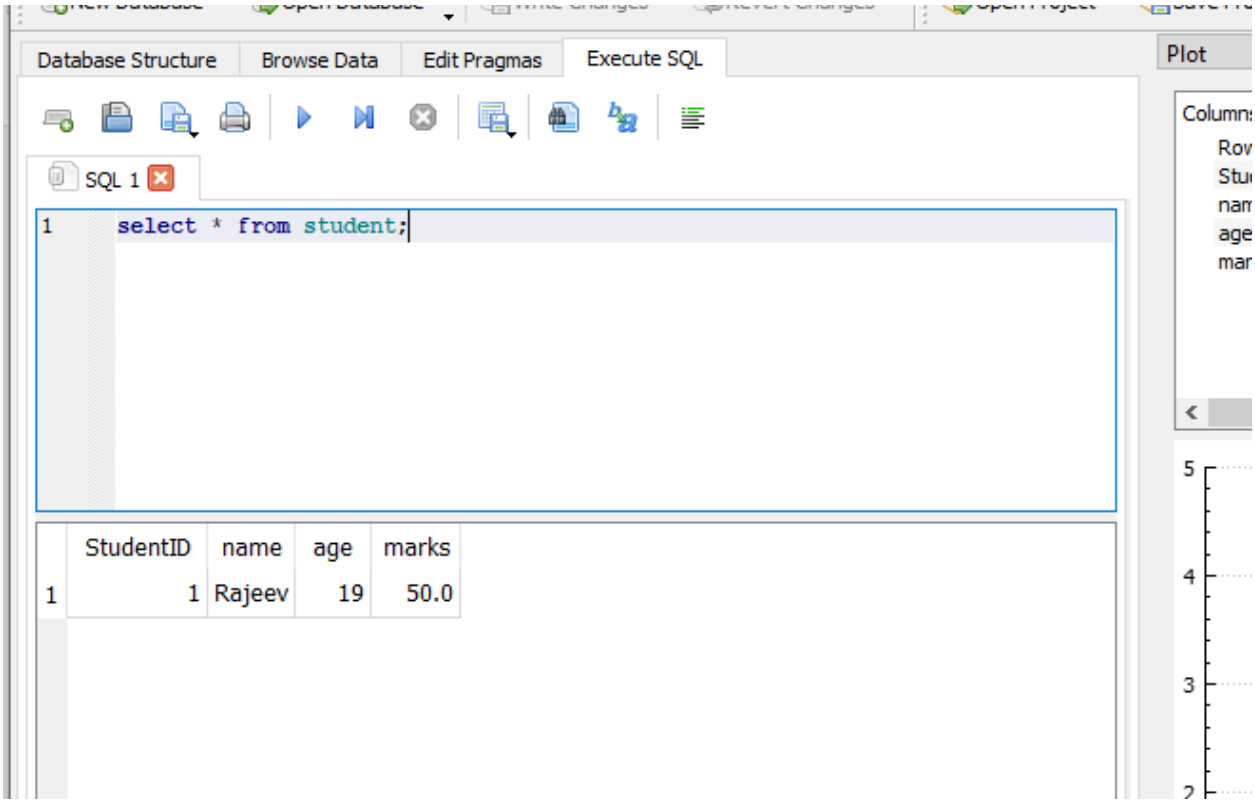
1. fetchone(): Fetches the next available record from the result set. It is a tuple consisting of values of each column of the fetched record.
2. fetchall(): Fetches all remaining records in the form of a list of tuples. Each tuple corresponds to one record and contains values of each column in the table.



Update a Record

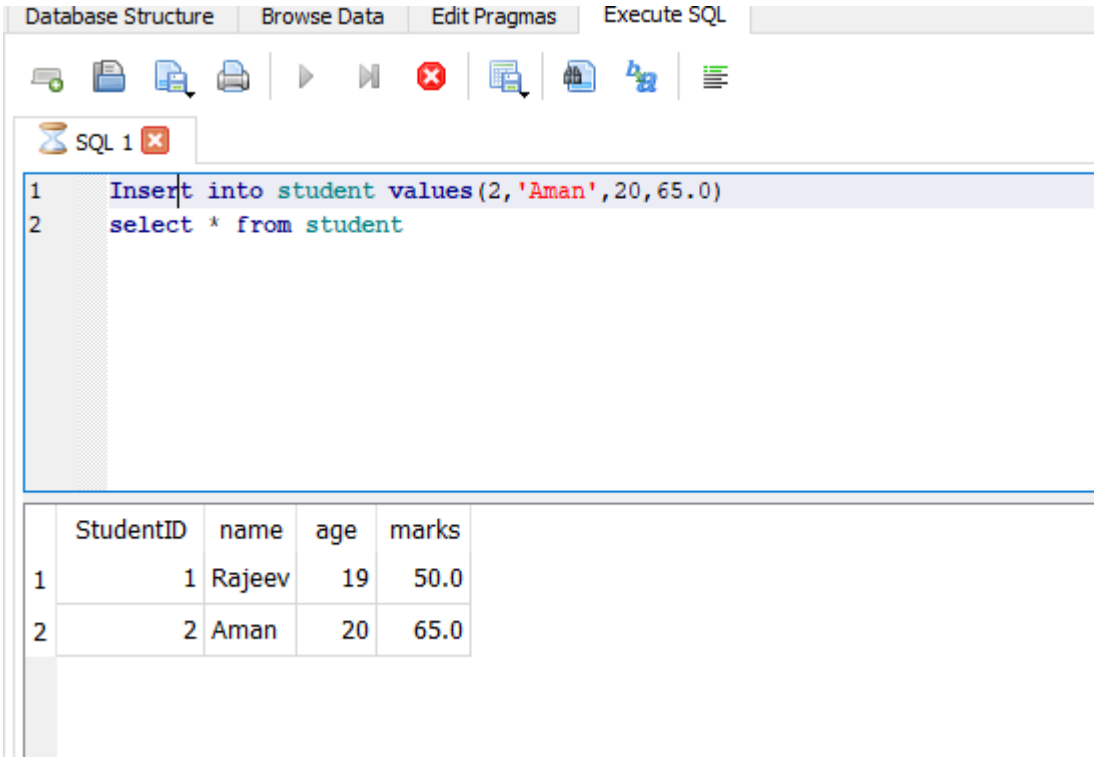
The query string in the `execute()` method should contain an UPDATE query syntax. To update the value of 'age' to 17 for 'Rajeev', define the string as below:

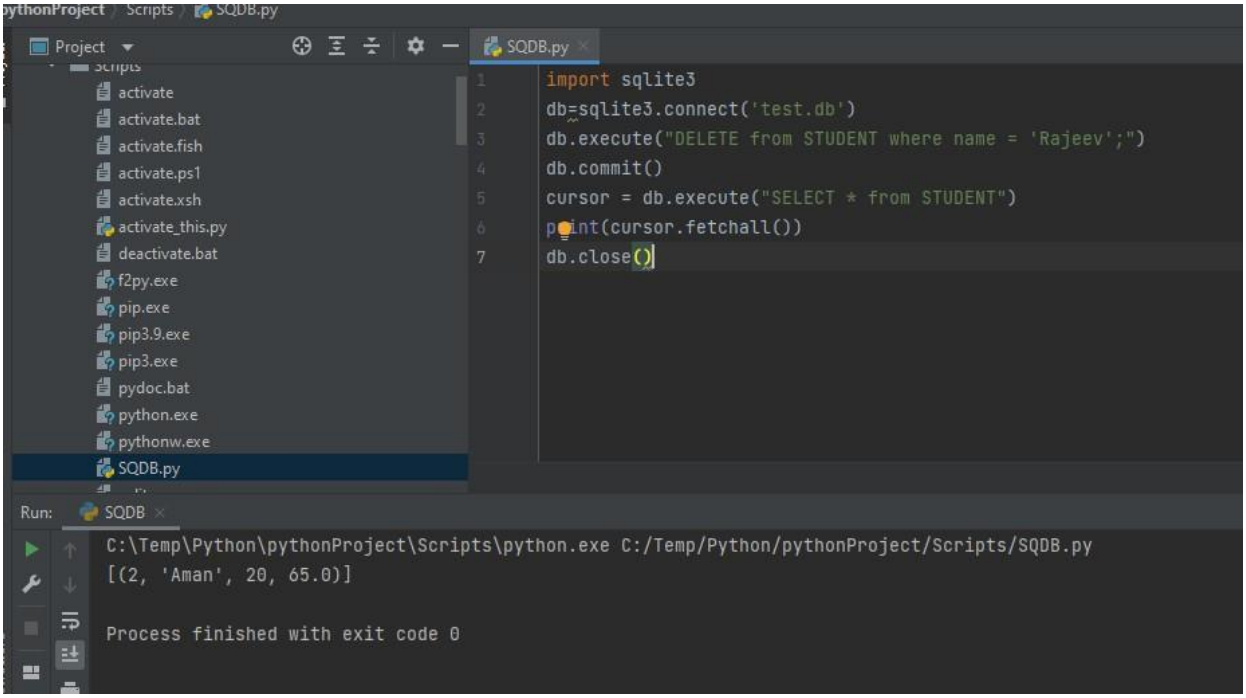

Project
activate
activate.bat
activate.fish
activate.ps1
activate.xsh
activate_this.py
deactivate.bat
f2py.exe
pip.exe
pip3.9.exe
pip3.exe
pydoc.bat
python.exe
pythonw.exe
SQDB.py
sqlite.py
SQDB.py
import sqlite3
db=sqlite3.connect('test.db')
qry="update student set age=? where name=?;"
try:
cur=db.cursor()
cur.execute(qry, (19,'Rajeev'))
db.commit()
print("record updated successfully")
except:
print("error in operation")
db.rollback()
db.close()
try
Run:
SQDB
C:\Temp\Python\pythonProject\Scripts\python.exe C:/Temp/Python/pythonProject/Scripts/SQDB.py
record updated successfully
Process finished with exit code 0

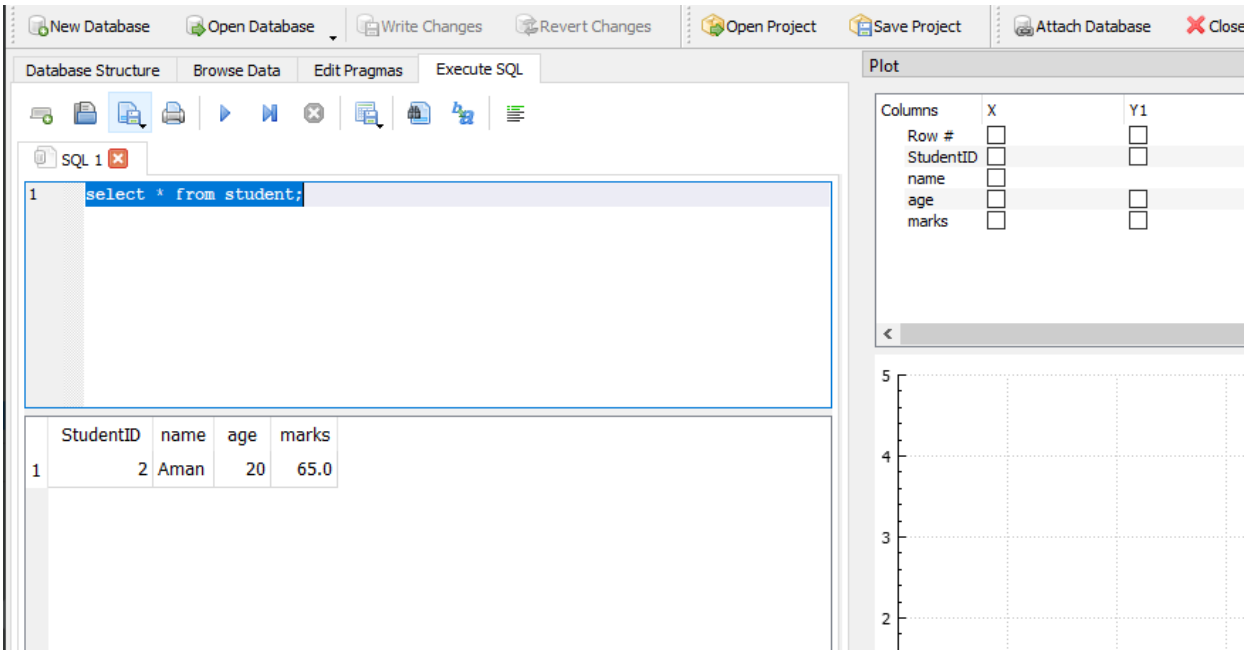

Database Structure
Browse Data
Edit Pragmas
Execute SQL
Plot
SQL 1
1 select * from student;
StudentID name age marks
1 1 Rajeev 19 50.0

Delete a Record

The query string should contain the DELETE query syntax. For example, the below code is used to delete 'Bill' from the student table.




pythonProject
Scripts
SQDB.py
Project
SQDB.py
activate
activate.bat
activate.fish
activate.ps1
activate.xsh
activate_this.py
deactivate.bat
f2py.exe
pip.exe
pip3.9.exe
pip3.exe
pydoc.bat
python.exe
pythonw.exe
SQDB.py
import sqlite3
db=sqlite3.connect('test.db')
db.execute("DELETE from STUDENT where name = 'Rajeev';")
db.commit()
cursor = db.execute("SELECT * from STUDENT")
print(cursor.fetchall())
db.close()
Run:
SQDB
C:\Temp\Python\pythonProject\Scripts\python.exe C:/Temp/Python/pythonProject/Scripts/SQDB.py
[(2, 'Aman', 20, 65.0)]
Process finished with exit code 0


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
select * from student;
StudentID
name
age
marks
1
2
Aman
20
65.0
Plot
Columns
X
Y1
Row #
StudentID
name
age
marks
5
4
3
2