What is SQLite?

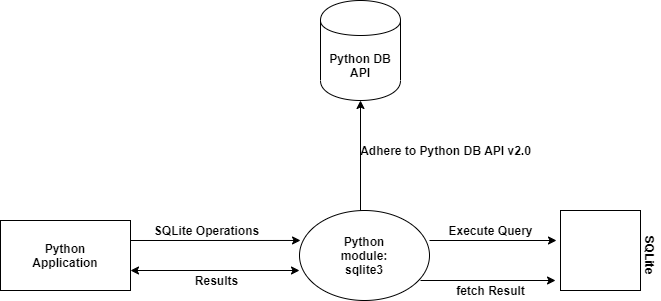
SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

Why SQLite?

* SQLite does not require a separate server process or system to operate (serverless).
* SQLite comes with zero-configuration, which means no setup or administration needed.
* A complete SQLite database is stored in a single cross-platform disk file.
* SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
* SQLite is self-contained, which means no external dependencies.
* SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
* SQLite supports most of the query language features found in SQL92 (SQL2) standard.
* SQLite is written in ANSI-C and provides simple and easy-to-use API.
* SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

**Python SQLite3** module is used to integrate the SQLite database with Python. It is a standardized Python DBI API 2.0 and provides a straightforward and simple-to-use interface for interacting with SQLite databases. There is no need to install this module separately as it comes along with Python after the 2.5x version.



Implementation:

First Create Database From Python

**EmployeeDB.py**

import sqlite3  
  
con = sqlite3.connect("employee.db")  
print("Database opened successfully")  
  
con.execute(  
 "create table Employees (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, email TEXT UNIQUE NOT NULL, address TEXT NOT NULL)")  
  
print("Table created successfully")  
  
con.close()

**Then Create following HTML Pages under templates Directory**

**Index.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>home</title>  
</head>  
<body>  
 <h2>Hi, welcome to the website</h2>  
 <a href="/add">Add Employee</a><br><br>  
 <a href ="/view">List Records</a><br><br>  
 <a href="/delete">Delete Record</a><br><br>  
</body>  
</html>

**Add.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>Add Employee</title>  
</head>  
<body>  
 <h2>Employee Information</h2>  
 <form action = "/savedetails" method="post">  
 <table>  
 <tr><td>Name</td><td><input type="text" name="name"></td></tr>  
 <tr><td>Email</td><td><input type="email" name="email"></td></tr>  
 <tr><td>Address</td><td><input type="text" name="address"></td></tr>  
 <tr><td><input type="submit" value="Submit"></td></tr>  
 </table>  
 </form>  
</body>  
</html>

**Success.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>save details</title>  
</head>  
<body>  
 <h3>Hi Admin, {{msg}}</h3>  
 <a href="/view">View Employees</a>  
</body>  
</html>

**Delete.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>delete record</title>  
</head>  
<body>  
  
 <h3>Remove Employee from the list</h3>  
  
<form action="/deleterecord" method="post">  
Employee Id <input type="text" name="id">  
<input type="submit" value="Submit">  
</form>  
</body>  
</html>

**delete\_record.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>delete record</title>  
</head>  
<body>  
<h3>{{msg}}</h3>  
<a href="/view">View List</a>  
</body>  
</html>

**View.html**

<!DOCTYPE html>  
<html>  
<head>  
 <title>List</title>  
</head>  
<body>  
  
<h3>Employee Information</h3>  
<table border=5>  
 <thead>  
 <td>ID</td>  
<td>Name</td>  
 <td>Email</td>  
 <td>Address</td>  
 </thead>  
  
 {% for row in rows %}  
  
 <tr>  
 <td>{{row["id"]}}</td>  
 <td>{{row["name"]}}</td>  
 <td>{{row["email"]}}</td>  
 <td>{{row["address"]}}</td>  
 </tr>  
  
 {% endfor %}  
</table>  
<br><br>  
  
<a href="/">Go back to home page</a>  
</body>  
</html>

**Then create and execute crud.py with following Code**

from flask import \*  
import sqlite3  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route("/")  
def index():  
 return render\_template("index.html");  
  
  
@app.route("/add")  
def add():  
 return render\_template("add.html")  
  
  
@app.route("/savedetails", methods=["POST", "GET"])  
def saveDetails():  
 msg = "msg"  
 if request.method == "POST":  
 try:  
 name = request.form["name"]  
 email = request.form["email"]  
 address = request.form["address"]  
 with sqlite3.connect("employee.db") as con:  
 cur = con.cursor()  
 cur.execute("INSERT into Employees (name, email, address) values (?,?,?)", (name, email, address))  
 con.commit()  
 msg = "Employee successfully Added"  
 except:  
 con.rollback()  
 msg = "We can not add the employee to the list"  
 finally:  
 return render\_template("success.html", msg=msg)  
 con.close()  
  
  
@app.route("/view")  
def view():  
 con = sqlite3.connect("employee.db")  
 con.row\_factory = sqlite3.Row  
 cur = con.cursor()  
 cur.execute("select \* from Employees")  
 rows = cur.fetchall()  
 return render\_template("view.html", rows=rows)  
  
  
@app.route("/delete")  
def delete():  
 return render\_template("delete.html")  
  
  
@app.route("/deleterecord", methods=["POST"])  
def deleterecord():  
 id = request.form["id"]  
 with sqlite3.connect("employee.db") as con:  
 try:  
 cur = con.cursor()  
 cur.execute("delete from Employees where id = ?", id)  
 msg = "record successfully deleted"  
 except:  
 msg = "can't be deleted"  
 finally:  
 return render\_template("delete\_record.html", msg=msg)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)