@dynamic.coding

Python SQL Dynamic Coding

Intro To SQLite Module In Python

Python SQLite Module is a lightweight library that provides an easy way to do the often difficult task of SQL type Database Management. This, unlike other Database Systems, does not require a dedicated server process/machine.

SQLite is a built-in module so we don't need to install it separately.

List of common methods which we require in order to perform any SQL opration.

connect(database.db):

To create or to establish a connection to a database file.

cursor()

To create cursor object.

execute(query:str)

To execute a sql query.

Create the DataBase:

create-Database.py

```python
import sqlite3
# Establish a connection to the Database and create
# a connection object
conn = sqlite3.connect('database.db')
```

Here, database.db is the database file, to where the data will be stored.

we can use the conn object to perform all the operations on our database file.

Create Table:

In order to create a table we can use the SQL CREATE TABLE query and execute it using execute function.

For simplicity, I'm creating a table with just two columns.

create-Table.py

```python
import sqlite3

with sqlite3.connect("StudentDB.db") as conn:
    print('Created the connection!')

    # SQL Query to create a table named Student
    query = """CREATE TABLE Student(
            id int primary key,
            name text);"""

    # Execute the SQL query to create the table
    conn.execute(query)
```

Inserting Records:

In order to insert into database table we can make use of SQL INSERT query.

insert-records.py

```
import sqlite3

with sqlite3.connect("StudentDB.db") as conn:
    print('Created the connection!')

    # SQL Query to Insert records
    query = """insert into Student (id, name)
                    values
                    (10, "Alex"), (20, "Alok"), (30, "Arwind");"""

    # Execute the SQL query to insert records
    conn.execute(query)
```

Reading records:

For reading records we first have to create a cursor object and execute the SQL SELECT Query using the cursor.

The use the fetchall() method to get all the records from table.

read-records.py

```python
import sqlite3

with sqlite3.connect("StudentDB.db") as conn:
    cursor = conn.cursor()
    cursor.execute("""select * from Student""")
    for row in cursor.fetchall():
        id, name = row
        print(f"{id = }, {name = }")
```