Pyton

Creare una tabella SQLite3

```python
import sqlite3
 conn = sqlite3.connect('tutorial.db')  # Don't forget to connect to db!
c = conn.cursor()  # think of it like a mouse cursor

# def create_table():
#     c.execute("CREATE TABLE example(Language VARCHAR, Version REAL, Skill TEXT)")

def enter_data():
    c.execute("INSERT INTO example VALUES('Python', 2.7, 'Beginner')")
    c.execute("INSERT INTO example VALUES('Python', 3.3, 'Intermediate')")
    c.execute("INSERT INTO example VALUES('Python', 3.4, 'Expert')")
    conn.commit()

# create_table()
enter_data()
conn.close()
```

Aggiungere dati a una tabella SQLite3

```python
import sqlite3

conn = sqlite3.connect('tutorial.db')  # Don't forget to connect to db!
c = conn.cursor()  # think of it like a mouse cursor

# def create_table():
#     c.execute("CREATE TABLE example(Language VARCHAR, Version REAL, Skill TEXT)")

def enter_data():
    c.execute("INSERT INTO example VALUES('Python', 2.7, 'Beginner')")
    c.execute("INSERT INTO example VALUES('Python', 3.3, 'Intermediate')")
    c.execute("INSERT INTO example VALUES('Python', 3.4, 'Expert')")
    conn.commit()  # Instead of closing the db  every time you enter data,
    # and then reopen it, you run .commit() to update the db table with your data
```

```python
# create_table()  # Runs the function create_table()
enter_data()  # Runs the function enter_data()
conn.close()  # this is only needed at the very end of the program.  Use the .commit() instead.
```

Leggere dati da una tabella SQLite3

```python
import sqlite3

conn = sqlite3.connect('tutorial.db')  # Don't forget to connect to db!
c = conn.cursor()  # think of it like a mouse cursor


# def create_table():
#    c.execute("CREATE TABLE example(Language VARCHAR, Version REAL, Skill TEXT)")
#
#
# def enter_data():
#    c.execute("INSERT INTO example VALUES('Python', 2.7, 'Beginner')")
#    c.execute("INSERT INTO example VALUES('Python', 3.3, 'Intermediate')")
#    c.execute("INSERT INTO example VALUES('Python', 3.4, 'Expert')")
#    conn.commit()  # Instead of closing the db  every time you enter data,
#    # and then reopen it, you run .commit() to update the db table with your data
#
#
# def enter_dynamic_data():
#    lang = input("What language?: ")
#    version = float(input("What version?: "))
#    skill = input("What skill level?: ")
#
#    c.execute("INSERT INTO example (Language, Version, Skill) VALUES (?, ?, ?) ", (lang, version, skill))
#    # This is the actual interface of the input data, placing it into the db.
#
#    conn.commit()


def read_from_database():
    # Example 1: Universal Print All
    sql = "SELECT * FROM example"
    for row in c.execute(sql):
        print(row)

    # Example 2: Print only WHERE Skill == 'Beginner'
    sql = "SELECT * FROM example WHERE Skill == 'Beginner'"
    for row in c.execute(sql):
        print(row)

    # Example 3: Print User Input Choice of Skill
```

```python
    what_skill = input("What skill are you looking for?: ")
    sql = "SELECT * FROM example WHERE Skill = ?"
    for row in c.execute(sql, [(what_skill)]):
        print(row)


    # Example 4: Print User Input Multiple Choices of Skill and Language
    what_skill = input("What skill are you looking for?: ")
    what_lang = input("What language?: ")
    sql = "SELECT * FROM example WHERE Skill = ? AND Language = ?"
    for row in c.execute(sql, [(what_skill), (what_lang)]):
        print(row)



# create_table()  # Only needs to be run once!  More than once creates an error.
# enter_data()  # Runs the function enter_data()
# enter_dynamic_data()
read_from_database()

conn.close()  # this is only needed at the very end of the program.  Use the .commit() instead.
```