

Python – manipularea bazelor de date folosind biblioteca *pymysql*

Se folosește biblioteca *pymysql*

- <https://pymysql.readthedocs.io/en/latest/>,
- <https://github.com/PyMySQL/PyMySQL>

O alta biblioteca similara este MySQLdb.

1. Conexiunea la o bază de date MySQL în Python

Pentru a putea crea o conexiune la o bază de date MySQL în Python, trebuie instalata și importata biblioteca **PyMySQL**:

```
import pymysql
```

Pentru accesarea datelor din tabelele unei baze de date MySQL, din Python, sunt necesari 4 pași:

1. Crearea conexiunii cu baza de date
2. Crearea unui obiect numit cursor
3. Execuția interogărilor SQL
4. Manipularea setului de date rezultat în urma interogărilor.

1. Prima etapă se realizează cu ajutorul metodei *connect()*, ce primește următorii parametri:

- **host**: numele sistemului (domeniului) unde rulează serverul MySQL. Poate fi un nume sau o adresă IP. Dacă nu este specificat, atunci va primi implicit valoare **localhost**.
- **user**: id-ul utilizatorului care trebuie să fie autentificat. Valoarea implicită este user-ul curent (de cele mai multe ori, root).
- **passwd**: parola
- **db**: baza de date
- **port**: portul prin care se permite accesul la serverul MySQL. Valoarea implicită este **3306**.

Exemplu:

```
db= pymysql.connect(host="Demo", user="root" , passwd="admln", db="test")
```

2. Crearea cursorului:

În ceea ce privește bazele de date, cursorul reprezintă zona de memorie alocată în urma unei interogări asupra bazei de date. Pentru a crea un cursor se apelează metoda **cursor()** asupra conexiunii.

```
cursor= db.cursor()
```

3. Execuția interogărilor SQL

Interogările de tip SQL pot fi executate cu ajutorul metodei *execute()* din clasa *Cursor*. Interogarea este transmisă sub formă de string, iar rezultatul interogării va fi stocat în obiectul cursor.

```
cursor.execute("select * from table")
```

4. Extragerea setului de date stocat în obiectul cursor

Clasa **Cursor** pune la dispoziție două funcții pentru manipularea datelor extrase într-un obiect de tip cursor: **fetchone()** și **fetchall()**. Prima funcție returnează o singură înregistrare sub formă de tuplu Python. Toate tipurile de date rezultate sunt automat mapate în tipurile de date Python, cu excepția tipului *unsigned integer* care este mapat către tipul *long*.

```
numrows = int(cursor.rowcount)

for x in range(0,numrows):
    row = cursor.fetchone()
    print row[0], "-->", row[1]
```

O altă variantă ar fi utilizarea funcției `fetchall()`:

```
result = cursor.fetchall()

for record in result:
    print record[0] , "=", record[1]
```

2. Exemplu de aplicație Python cu conexiune la o bază de date MySQL

Aplicația permite accesul la o bază de date în care sunt stocate informații despre o colecție de DVD-uri cu filme, precum: titlul filmului, actorul din rolul principal, actorul din rolul secundar, anul apariției și genul filmului. Sunt permise operațiile de adăugare în baza de date, de căutare după mai multe criterii, de modificare a înregistrărilor, de ștergere și de export în format CSV.

Aplicația este structurată sub forma mai multor scripturi, după cum urmează:

Scriptul principal, unde sunt importate toate celelalte scripturi, aplicatie.py

```
#aplicatie.py

import os
import adaugare
import cautare
import modificare
import stergere
import export

#Meniul principal
def Menu():
    os.system('cls')
    print """
=====
1 - Adauga in baza de date
2 - Cauta in baza de date
3 - Modifica inregistrare
4 - Sterge inregistrare
5 - Export in format CSV
6 - Iesire
=====
"""
    opt = raw_input("Introduceti optiunea si apasati 'Enter': ")
    return opt

#Rularea modulelor in functie de optiune
opt = ""
while opt != "6":
    opt = Menu()
    if opt == "1":
        os.system('cls')
        adaugare.MeniuAdaugaDVD()
    elif opt == "2":
        os.system('cls')
        cautare.MeniuCautaDVD()
    elif opt == "3":
        os.system('cls')
        modificare.MeniuModificaDVD()
    elif opt == "4":
        stergere.MeniuStergeDVD()
    elif opt == "5":
        export.ExportCSV()
```

Scriptul dbConnection unde este creată conexiunea la baza de date:

```

#dbConnection.py

# Cele mai importante module Python DB API

# MySQL - MySQLdb
# PostgreSQL - psycopg2
# SQLite - sqlite3
# Oracle - oracle
# MS SQL server - adodbapi

import pymysql

# variabile pentru accesul la baza de date
HOST = 'localhost'
USER = 'root'
PASSWD = 'root'
DATABASE = 'test1'

def CreeazaConexiune():
    # creeaza o conexiune la baza de date
    try:
        #db = pymysql.connect(host=HOST, user=USER, db=DATABASE)
        db = pymysql.connect(host='localhost', port=3306, user='root',
passwd='root', db='test1')

        #print "Conexiunea la baza de date s-a realizat cu succes!"
        return db
    except Exception, e:
        print "Eroare la conexiunea cu baza de date! - ", e

def CreeazaBazaDeDate():
    # creeaza baza de date
    db = CreeazaConexiune()
    if db == None:
        try:
            db = pymysql.connect(host=HOST, port=3306, user=USER, passwd=PASSWD)
            db.cursor().execute('CREATE DATABASE ' + DATABASE)
        except Exception as e:
            print("Eroare la crearea bazei de date!", e)

def CreeazaTabela():
    # creeaza tabel
    db=CreeazaConexiune()
    # pentru a executa interogari asupra bazei de date este nevoie de un obiect
de tip cursor
    cursor=db.cursor()
    cursor.execute('USE ' + DATABASE)

    cursor.execute("""
CREATE TABLE dvd(
titlu Varchar(100),
actor_principal VARCHAR(100),
actor_secundar VARCHAR(100),
an INT,
gen VARCHAR(100)
)
""")

    db.commit()
    cursor.close()

```

```
db.close()
```

```
CreeazaConexiune()
```

Scriptul de adăugare a unei înregistrări noi, este prezentat mai jos:

```
#adaugare.py

import dbConnection

#Executa o interogare SQL pentru a adauga in baza de date
def AdaugaDVD(titlu, actorPrincipal, actorSecundar, an, gen):
    sqlInsert = 'INSERT INTO dvd values ("%s", "%s", "%s", "%s", "%s")' % (titlu,
actorPrincipal, actorSecundar, an, gen)
    try:
        db=dbConnection.CreeazaConexiune()
        c=db.cursor()
        try:
            c.execute(sqlInsert)
        except Exception, e:
            print "Eroare: ",e
        db.commit()
        c.close()
        db.close()
        raw_input("Inregistrarea a fost adaugata cu success - apasati 'Enter'
pentru a continua: ")
    except:
        print "Eroare la adaugarea in baza de date"
        raw_input("Apasati 'Enter' pentru a continua.")

#Afiseaza meniul pentru adaugarea in baza de date
def MeniuAdaugaDVD():
    print "======"
    print "Adaugare DVD in baza de date:"
    print "======"
    titlu = raw_input("Introduceti titlul filmului: ")
    actorPrincipal = raw_input("Introduceti numele actorului din rolul principal:
")
    actorSecundar = raw_input("Introduceti numele actorului din rolul secundar:
")
    an = raw_input("Introduceti anul aparitiei: ")
    gen = raw_input("Introduceti genul filmului:\n - 1 - Drama, 2 - Horror, 3 -
Comedie, 4 - Romantic: ")
    if gen == "1":
        gen = "Drama"
    elif gen == "2":
        gen = "Horror"
    elif gen == "3":
        gen = "Comedie"
    elif gen == "4":
        gen = "Romantic"
    else:
        print "Ati introdus informatii gresite!"
        raw_input("Apasati 'Enter' pentru a va intoarce la meniu.")
        return

    AdaugaDVD(titlu, actorPrincipal, actorSecundar, an, gen)
```

Modificarea unei înregistrări este realizată în scriptul următor:

```
#modificare.py
#modificare.py

import dbConnection

def MeniuModificaDVD():
    print "======"
    print "Modificare inregistrare in baza de date:"
    print "======"

    titluDVD = raw_input("\n Introduceți titlul filmului pe care doriți sa-l
modificati: ")

    querySelectTitlu = "SELECT * FROM dvd WHERE titlu = \"%s\"" % titluDVD

    try:
        db = dbConnection.CreeazaConexiune()
        c = db.cursor()
        c.execute(querySelectTitlu)
        rezultat = c.fetchall()
        #if rezultat[0] == ():
            # raise
    except:
        print "Eroare la accesarea inregistrarii in baza de date!"
        raw_input("Apasati 'Enter' pentru a continua.")
        return

    try:
        print "======"
        print "Filmul care va fi modificat:"
        print "======"
        print "1 - Titlu:\t", rezultat[0][0]
        print "2 - Actor principal:\t", rezultat[0][1]
        print "3 - Actor secundar:\t", rezultat[0][2]
        print "4 - An:\t", rezultat[0][3]
        print "5 - Gen:\t", rezultat[0][4]
        print "======"

        opt = raw_input("Introduceți numele campului pe care doriți sa-l
modificati, si apasati 'Enter': ")

        titluModificat = False
        campModificat = ""
        nouaValoare = ""
        if opt == "1":
            campModificat = "titlu"
            nouaValoareTitlu = raw_input("Introduceți noul titlu: ")
            nouaValoare = "\"%s\"" % nouaValoareTitlu
            titluModificat = True
        elif opt == "2":
            campModificat = "actor_principal"
```

```

        nouaValoare = raw_input("Introduceti noul nume pentru actorul
principal: ")
        nouaValoare = "\"%s\"" % nouaValoare
        elif opt == "3":
            campModificat = "actor_secundar"
            nouaValoare = raw_input("Introduceti noul nume pentru actorul
secundar: ")
            nouaValoare = "\"%s\"" % nouaValoare
        elif opt == "4":
            campModificat = "an"
            nouaValoare = raw_input("Introduceti anul: ")
            nouaValoare = "\"%s\"" % nouaValoare
        elif opt == "5":
            campModificat = "gen"
            print "======"
            print "Alegeti genul pe care doriti sa-l modificati:"
            print "1 - Drama"
            print "2 - Horror"
            print "3 - Comedy"
            print "4 - Romance"

    optiune = raw_input("Introduceti noua valoare pentru genul filmului:
")

    if optiune == "1":
        nouaValoare = "\"Drama\""
    elif optiune == "2":
        nouaValoare = "\"Horror\""
    elif optiune == "3":
        nouaValoare = "\"Comedie\""
    elif optiune == "4":
        nouaValoare = "\"Romantic\""

    queryUpdate = "UPDATE dvd SET %s = %s WHERE titlu = \"%s\"" %
(campModificat, nouaValoare, titluDVD)

    db = dbConnection.CreeazaConexiune()
    c = db.cursor()
    c.execute(queryUpdate)
    db.commit()

    if titluModificat:
        querySelect = "SELECT * FROM dvd WHERE titlu = \"%s\"" %
nouaValoareTitlu
        c = db.cursor()
        c.execute(querySelect)
        rezultatModificare = c.fetchall()
        c.close()
        db.close()

except:
    print "Eroare la modificare inregistrarii!"
    raw_input("Apasati 'Enter' pentru a continua.")
    return

print "======"
print "Intregistrare modificata:"
print "======"
print "1 - Titlu:\t", rezultatModificare[0][0]
print "2 - Actor principal:\t", rezultatModificare[0][1]
print "3 - Actor secundar:\t", rezultatModificare[0][2]
print "4 - An:\t", rezultatModificare[0][3]
print "5 - Gen\t", rezultatModificare[0][4]

```

```
print "======"
raw_input("Apasati 'Enter' pentru a continua.")
```

Căutarea se poate face după mai multe criterii, așa cum este prezentată în următorul script:

```
#cautare.py

import pymysql, os
import dbConnection

#Executa o interogare pentru cautare in baza de date
def CautaDVD(campCautare, textCautare):
    querySelect = "SELECT * FROM dvd WHERE %s = %s" % (campCautare, textCautare)
    try:
        db=dbConnection.CreeazaConexiune()
        c=db.cursor()
        try:
            c.execute(querySelect)
        except Exception, e:
            print e
        output = c.fetchall()
        c.close()
        db.close()
    except:
        print "Eroare la conexiunea cu baza de date! ", e
        raw_input("Apasati tasta 'Enter' pentru a continua.")
        return

os.system('cls')
print "======"
print "Cautare DVD in baza de date:"
print "======"
if output == ():
    print "Nu a fost gasita nicio inregistrare!"
    print "======"
for entry in output:
    print "Titlu:\t", entry[0]
    print "Actor principal:\t", entry[1]
    print "Actor secundar:\t", entry[2]
    print "An:\t", entry[3]
    print "Gen:\t", entry[4]
    print "======"
raw_input("\n\nApasati 'Enter' pentru a continua: ")

#Afiseaza meniul pentru a cauta in baza de date
def MeniuCautaDVD():
    print ""
    =====
    Alegeti criteriul de cautare:
    1 - Dupa titlu
    2 - Dupa actorul principal
    3 - Dupa actorul secundar
    4 - Dupa an
    5 - Dupa gen""
    opt = raw_input("\nIntroduceti optiunea si apasati 'Enter': ")
```



```

campCautare = ""
textCautare = ""

if opt == "1":
    campCautare = "titlu"
    textCautare = raw_input("Introduceti titlul filmului: ")
    textCautare = "\"%s\"" % (textCautare)
elif opt == "2":
    campCautare = "actor_principal"
    textCautare = raw_input("Introduceti numele actorului principal: ")
    textCautare = "\"%s\"" % (textCautare)
elif opt == "3":
    campCautare = "actor_secundar"
    textCautare = raw_input("Introduceti numele actorului secundar: ")
    textCautare = "\"%s\"" % (textCautare)
elif opt == "4":
    campCautare = "an"
    textCautare = int(raw_input("Introduceti anul: "))
elif opt == "5":
    campCautare = "gen"
    print """
    Introduceti genul:
    1 - Drama
    2 - Horror
    3 - Comedie
    4 - Romantic
    """
    optiuneGen = raw_input("\t")
    if optiuneGen == "1":
        textCautare = "\"Drama\""
    elif optiuneGen == "2":
        textCautare = "\"Horror\""
    elif optiuneGen == "3":
        textCautare = "\"Comedie\""
    elif optiuneGen == "4":
        textCautare = "\"Romantic\""
    else:
        print "Ati introdus date gresite!"
        raw_input("Apasati 'Enter' pentru a reveni la meniu.")
        return
CautaDVD(campCautare, textCautare)

```

Ștergerea unei înregistrări:

```

#stergere.py

import pymysql, os, dbConnection

# Executa o interogare SQL pentru stergerea din baza de date
def StergeDVD(dvdStergere):
    try:
        queryDelete = 'DELETE FROM dvd WHERE titlu = %s' % dvdStergere
        db = dbConnection.CreeazaConexiune()
        c = db.cursor()
        c.execute(queryDelete)
        db.commit()
        c.close()
        db.close()
    
```

```

        raw_input("Inregistrarea a fost stearsa cu success! Apasati 'Enter' pentru
a continua.")
    except:
        print "Eroare la stergerea inregistrarii"
        raw_input("Apasati 'Enter' pentru a continua.")

# Afisare meniu stergere
def MeniuStergeDVD():
    os.system('cls')
    print "======"
    print "Sterge inregistrare"
    print "======"
    campCautare="titlu"
    dvdStergere = raw_input("\nIntroduceti titlul DVD-ului pe care doriti sa-l
stergeti:\t")
    dvdStergere = "\"%s\""% (dvdStergere)
    querySelect = "SELECT * FROM dvd WHERE %s = %s"% (campCautare,dvdStergere)
    try:
        db = dbConnection.CreeazaConexiune()
        c = db.cursor()
        c.execute(querySelect)
        searchResult = c.fetchall()
        # if searchResult[0] == ():
        #     raise
    except:
        print "Eroare la accesarea inregistrarii in baza de date!"
        raw_input("Apasati 'Enter' pentru a continua.")
        return

    print "======"
    print "Informatii despre DVD care va fi sters"
    print "======"
    print "Titlu:\t", searchResult[0][0]
    print "Actor principal:\t", searchResult[0][1]
    print "Actor secundar:\t", searchResult[0][2]
    print "An:\t", searchResult[0][3]
    print "Gen:\t:", searchResult[0][4]
    print "======"
    print ""
    Sunteti sigur ca doriti sa stergeti DVD-ul? \n
    Introduceti optiunea si apasati 'Enter' (D/d = da, Orice altceva = Nu)
    ""
    opt = raw_input("\t")
    if (opt == "D" or opt == "d"):
        StergeDVD(dvdStergere)
    else:
        c.close()
        db.close()
        raw_input("Inregistrarea NU a fost stearsa! Apasati 'Enter' pentru a
continua.")

```

Exportul în format CSV:

```
import MySQLdb, csv, os, dbConnection
```

```
#export.py

import dbConnection
import csv, os
```

```

#Exporta baza de date in format CSV
def ExportCSV():
    querySelect = "SELECT * FROM dvd"
    try:
        db = dbConnection.CreeazaConexiune()
        c = db.cursor()
        c.execute(querySelect)
        output = c.fetchall()
        c.close()
        db.close()
    except:
        print "Eroare la conexiunea cu baza de date!"
        raw_input("Apasati 'Enter' pentru a reveni la meniu.")
        return

    try:
        os.system('cls')
        print "======"
        print "Exporta baza de date in format CSV:"
        print "======"
        filename = raw_input("Introduceti numele fisierului (fara extensia .csv):
")
        filename = filename + ".csv"
        writer = csv.writer(open(filename, "w"))
        writer.writerow(("TITLU", "ACTOR PRINCIPAL", "ACTOR SECUNDAR", "AN",
"GEN"))
        writer.writerows(output)
        print filename, "Baza de date a fost exportata cu succes, apasati 'Enter'
pentru a continua."
        raw_input("")
        return
    except:
        print "Eroare la scrierea fisierului!"
        raw_input("Apasati 'Enter' pentru a reveni la meniu")

```