

INICIACIÓN PYTHON

Anexo I

Acceso a datos externos

Curso 2018

Germán Alonso Lascurain
germanalonso@opendeusto.es



ACCESO A DATOS EXTERNOS

- Listado de las interfaces disponibles para operar contra BBDD

<https://wiki.python.org/moin/DatabaseInterfaces>

ACCESO A DATOS EXTERNOS - MYSQL

- Acceso a BBDD MySQL

Necesario el módulo MySQLdb (hacen falta permisos de administrador para ello).

Windows → `pip install MySQL-python`

Linux → `sudo apt-get install python-pip python-dev libmysqlclient-dev`

ACCESO A DATOS EXTERNOS - MYSQL

Acceso a la BBDD

```
import MySQLdb
```

- Abrir un conexión con la BBDD

```
db = MySQLdb.connect("localhost", "testuser", "test123", "TESTDB")
```

- Preparar un "cursor object" usando el método `cursor()`.

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

- Ejecutar una consulta SQL mediante el método `execute()`.

```
cursor.execute("SELECT VERSION()")
```

- Obtener un registro usando `fetchone()`.

```
data = cursor.fetchone()
```

```
print "Database version : %s " % data
```

- Desconectar del servidor

```
db.close()
```

ACCESO A DATOS EXTERNOS - MYSQL

Creación de una tabla

- A través del método **execute()**

```
import MySQLdb
```

- Abrir conexión con la BBDD

```
db = MySQLdb.connect("localhost", "testuser", "test123", "TESTDB" )
```

- Preparar el objeto cursor

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

- Eliminar la tabla, si ya existe usando execute().

```
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
```

- Crear tabla

```
sql = """CREATE TABLE EMPLOYEE (  
    FIRST_NAME CHAR(20) NOT NULL,  
    LAST_NAME CHAR(20),  
    AGE INT,  
    SEX CHAR(1),  
    INCOME FLOAT ) """
```

```
cursor.execute(sql)
```

ACCESO A DATOS EXTERNOS - MYSQL

Creación de una tabla

- Desconectar del servidor
`db.close()`

ACCESO A DATOS EXTERNOS - MYSQL

Insertar registros en tabla

```
import MySQLdb
db = MySQLdb.connect("localhost", "testuser", "test123", "TESTDB" )
cursor = db.cursor()
```

- Sentencia SQL para insertar un registro

```
sql = """INSERT INTO EMPLOYEE (FIRST_NAME,
    LAST_NAME, AGE, SEX, INCOME)
    VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""
```

```
try:
    # Ejecuta el comando sql
    cursor.execute(sql)
    # Aplica los cambios en la BBDD
    db.commit()
except:
    # En caso de error volver atrás
    db.rollback()
```

Las inserciones se ejecutan a través del cursor que ejecuta la consulta

- Desconectar del servidor
- ```
db.close()
```

# ACCESO A DATOS EXTERNOS - MYSQL

## Insertar registros en tabla a través de consultas SQL dinámicas

- Sentencia SQL para insertar un registro

```
sql = "INSERT INTO EMPLOYEE(FIRST_NAME, \
 LAST_NAME, AGE, SEX, INCOME) \
 VALUES ('%s', '%s', '%d', '%c', '%d')" % \
 ('Mac', 'Mohan', 20, 'M', 2000)
```

# ACCESO A DATOS EXTERNOS - MYSQL

## Lectura de registros

`fetchone()` → recupera un registro

`fetchall()` → recupera todos los registros

`rowcount` → Devuelve el número de filas de la tabla.

Consulta que devuelva los registros donde el salario > 1000

```
sql = "SELECT * FROM EMPLOYEE \
 WHERE INCOME > '%d'" % (1000)
```

```
try:
```

```
 # Ejecución de la sentencia SQL
```

```
 cursor.execute(sql)
```

```
 # Recuperar todos los datos en una lista de listas.
```

```
 results = cursor.fetchall()
```

```
 for row in results:
```

```
 fname = row[0]
```

```
 lname = row[1]
```

```
 age = row[2]
```

```
 sex = row[3]
```

```
 income = row[4]
```

```
 # Imprimir datos recuperados
```

```
 print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \
 (fname, lname, age, sex, income)
```

```
except:
```

```
 print "Error: No puedo recuperar los datos"
```

# ACCESO A DATOS EXTERNOS - MYSQL

## Actualización de registros

- Actualización de todos los registros de la BBDD cuyo sexo sea M, de manera que su edad se incremente en 1

```
sql = "UPDATE EMPLOYEE SET AGE = AGE + 1
 WHERE SEX = '%c'" % ('M')
```

```
try:
 # Ejecución de la sentencia SQL
 cursor.execute(sql)
 # Aplicar cambios
except:
 # Deshacer en caso de error
 db.rollback()
```

# ACCESO A DATOS EXTERNOS - MYSQL

## Borrar registros

- Borrado de los registros cuya edad sea mayor que 20.

```
sql = "DELETE FROM EMPLOYEE WHERE AGE > '%d'" % (20)
try:
 # Ejecución de la sentencia SQL
 cursor.execute(sql)
 # Aplicar cambios
except:
 # Deshacer en caso de error
 db.rollback()
```

# ACCESO A DATOS EXTERNOS - ORACLE

## Acceso a BBDD Oracle

- <http://www.orafaq.com/wiki/Python>
- <http://www.oracle.com/technetwork/articles/dsl/prez-python-queries-101587.html>
- <https://community.toadworld.com/platforms/oracle/w/wiki/11723.using-python-and-cx-oracle-part-1>

Necesario el módulo `cx_Oracle`.

### **Conectar a la BBDD**

```
import cx_Oracle
connection = cx_Oracle.connect('user/pwd@bbdd')
```

```
desconectar de la BBDD
connection.close()
```

# ACCESO A DATOS EXTERNOS - ORACLE

## Lectura de registros

```
import sys, cx_Oracle
```

- Crear la conexión

```
connection = cx_Oracle.connect('user/pwd@bbdd')
```

- Crear el cursor

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

- Ejecutar la consulta

```
cursor.execute("SELECT 'Hello world!' FROM tabla")
```

- Leer el contenido de cursor.

```
for row in cursor:
```

```
 print (row[0])
```

```
except cx_Oracle.DatabaseError, e:
```

```
 error, = e.args
```

```
 print >> sys.stderr, "Código Error Oracle:", error.code
```

```
 print >> sys.stderr, "Mensaje Error Oracle:", error.message
```

```
finally:
```

```
 # Close cursor and connection.
```

```
 cursor.close()
```

```
 db.close()
```

# ACCESO A DATOS EXTERNOS - ORACLE

## Lectura de registros 2

Ejecutar la consulta

```
cursor.execute("SELECT item_title " +
 ", item_rating " +
 "FROM item " +
 "WHERE item_type = "
 " (SELECT common_lookup_id " +
 " FROM common_lookup " +
 " WHERE common_lookup_type = 'DVD_WIDE_SCREEN')")

#Leer el contenido de cursor.
for row in cursor:
 print (row[0], row[1])
except cx_Oracle.DatabaseError, e:
 error, = e.args
 print >> sys.stderr, "Código Error Oracle:", error.code
 print >> sys.stderr, "Mensaje Error Oracle:", error.message

finally:
 # Cerrar la conexión con la BBDD
 cursor.close()
 db.close()
```

# ACCESO A DATOS EXTERNOS - ORACLE

## Insertar registros

```
connection = cx_Oracle.connect("uid/pwd@database")
cursor = connection.cursor()
cursor.execute("INSERT INTO User_Tables(login,first_name,last_name,age,date_of_birth)
 VALUES (:login,:first,:last,:age,to_date(:dob,'YYYY-MM-DD
HH24:MI:SS'))",
 {
 'login' : 'some_user_login',
 'first' : 'some_first_name',
 'last' : 'some_last_name',
 'age' : 42,
 'dob' : '1970-01-01 23:52:00',
 }
)
count = cursor.fetchall()[0][0]
cursor.close()
connection.commit()
connection.close()
```

# ACCESO A DATOS EXTERNOS - ORACLE

## Ejecutar sentencias CREATE TABLE, CREATE INDEX, ALTER Y DROP

```
connection = cx_Oracle.connect("uid/pwd@database")
cursor = connection.cursor()
cursor.execute("CREATE TABLE x(a DATE)")
cursor.close()
```

# ACCESO A DATOS EXTERNOS - SQLITE

## Acceso a BBDD Sqlite

<https://www.pythoncentral.io/introduction-to-sqlite-in-python/>

Necesario el módulo splite3

```
import sqlite3
```

## Conexión a una BBDD SQLite

2 opciones, crear la BBDD en memoria o en un archivo

```
Crear BBDD en memoria
db = sqlite3.connect(':memory:')
Crea o abre un fichero llamado mydb con SQLite3 DB
db = sqlite3.connect('data/mydb')
```

```
Cerrar BBDD
db.close()
```

# ACCESO A DATOS EXTERNOS - SQLITE

## Crear tablas

```
Obtener el objeto cursor
cursor = db.cursor()
```

```
cursor.execute('' 'CREATE TABLE users(id INTEGER PRIMARY
KEY, name TEXT, phone TEXT, email TEXT unique, password
TEXT) ' '')
```

```
db.commit()
```

# ACCESO A DATOS EXTERNOS - SQLITE

## Insertar datos en tablas

### A través de tuplas

```
cursor = db.cursor()
nombre1 = 'Andres'
tlfn1 = '3366858'
email1 = 'user@example.com'
password1 = '12345'

nombre2 = 'John'
tlfn2 = '5557241'
email2 = 'johndoe@example.com'
password2 = 'abcdef'

Insertar usuario1
cursor.execute(''INSERT INTO users(nombre, tlfn, email, password)
 VALUES(?,?,?,?)'' , (nombre1,tlfn1, email1, password1))
print('Primer usuario introducido')

Insertar usuario2
cursor.execute(''INSERT INTO users(nombre, tlfn, email, password)
 VALUES(?,?,?,?)'' , (nombre2,tlfn2, email2, password2))
print('Segundo usuario introducido')

db.commit()
```

# ACCESO A DATOS EXTERNOS - SQLITE

## Insertar datos en tablas

### A través de diccionarios

```
cursor.execute(''INSERT INTO users(nombre, tlf, email, password)
VALUES(:nombre, :tlf, :email, :password)'' , {'nombre':nombre1,
'tlf':tlf1, 'email':email1, 'password':password1})
```

# ACCESO A DATOS EXTERNOS - SQLITE

## Borrar tablas

```
Obtener el objeto cursor
cursor = db.cursor()

cursor.execute('' 'DROP TABLE users'' ')
db.commit()
```

# ACCESO A DATOS EXTERNOS – MSSQL SERVER

- Acceso a BBDD MSSQL Server

- 1) Descargar e instalar el controlador ODBC de MS

<https://docs.microsoft.com/es-es/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-2017>

- 2) Instalar el módulo **pyodbc** desde una consola Anaconda

<https://docs.microsoft.com/es-es/sql/connect/python/pyodbc/python-sql-driver-pyodbc?view=sql-server-2017>

```
pip install pyodbc
```

# ACCESO A DATOS EXTERNOS – MSSQL SERVER

## Acceso a la BBDD

```
import pyodbc
```

- Abrir un conexión con la BBDD

```
db = pyodbc.connect('Driver={SQL
Server}; ' 'Server=localhost; 'Database=testdb; ' 'uid=sa;pwd=Password')
```

- Preparar un “cursor object” usando el método `cursor()` .

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

- Ejecutar una consulta SQL mediante el método `execute()` .

```
cursor = connection.cursor()
SQLCommand = ("sentencia_sql")
cursor.execute(SQLCommand, Values)
```

- Obtener un registro usando `fetchone()` .

```
data = cursor.fetchone()
print "Database version : %s " % data
```

- Desconectar del servidor

```
db.close()
```

## Copyright (c) 2018 Germán Alonso Lascurain

This work (but the quoted images, whose rights are reserved to their owners\*) is licensed under the Creative Commons “Attribution-ShareAlike” License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Germán Alonso Lascurain  
germanalonso@opendeusto.es