MySQL y el lenguaje Python

MySQL y el lenguaje Python

Las bases de datos permiten que, de manera simultánea, varios usuarios tengan acceso y consulta de información de una manera rápida y segura, utilizando una lógica y un lenguaje altamente complejos (Oracle, s.f).

MySQL organiza grandes cantidades de información, sin embargo, ¿cómo es el manejo de estas bases de datos? ¿Cuáles son los procedimientos para conectar una base de datos en Python?

De acuerdo con el sitio Comunidad Python Argentina (s.f.), se recomiendan los siguientes pasos para el uso de bases de datos en Python:

1. Importar el conector.
2. Conectarse a la base de datos (función connect del módulo conector).
3. Abrir un curso (método cursor de la conexión).
4. Ejecutar una consulta (método *execute* del cursor).
5. Obtener los datos (método *fetch* sobre el cursor).
6. Cerrar el cursor (método *close* sobre del cursor).

Con base en lo anterior, un ejemplo elemental para conectar a una base de datos con MySQL es el siguiente:

```
>>> import MySQLdb
>>> db = MySQLdb.connect(host="localhost", user= "root",
. . . passwd= "mypassword", db= "PythonU")
```

Una vez constituida la conexión, se crea un cursor. El método para crear uno se llama cursor ():

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

Realizados los pasos anteriores, se procede a ejecutar algunos comandos SQL:

```
>>> cursor.execute("SELECT * FROM Students")
5L
```

El método **execute** es utilizado para ejecutar comandos SQL. Para obtener un solo elemento utilizamos *fetchone()* y *fetchall()* para todos los elementos:

```
>>> cursor. fetchone()
(1L, 'Joe' , 'Campbell', datetime.date(2006, 2, 10), 'N')

>>> cursor. fetchall()
((1L, 'Joe' , 'Campbell', datetime.date(2006, 2, 10), 'N'),
(2L, 'Joe', 'Doe', datetime.date(2004, 2, 16), 'N'),
(3L, 'Rick', 'Hunter', datetime.date(2005, 3, 20, 'N'),
(4L, 'Laura', 'Ingalls', datetime.date(2001, 3, 15), 'Y'),
(5L, 'Virginia', ' Gonzalez' , datetime.date(2003, 4, 2,), 'N'))
```

El método por utilizar dependerá de la cantidad de datos disponibles, la memoria disponible en la computadora, y cómo deseamos realizarlo.

Ejemplo de cómo conectarte a una base de datos en Python a través de la librería SQLAlchemy. Es importante considerar que, en caso de que no esté instalada, podemos llevarlo a cabo de la siguiente manera. Es recomendable que practiques programando al mismo tiempo.

```
! pip install sqlalchemy
```

Enseguida mandar llamar la base de datos:

```
from sqlalchemy import create_engine
Import pandas as pd
Engine= create_engine('base_de_datos", "usuario", "contraseña")
Informacion=pd.read_sql_query("Select", "usuario", "contraseña")
```

Ejemplo de cómo conectarte a una base de datos MySQL en Python:

Requerimientos

1. Contar con el nombre del servidor donde se encuentra la base de datos.
2. Contar con el nombre de la base de datos (se recomienda una de prueba para cuidar la integridad de la información).
3. Descargar e instalar el driver de MySQL para Python.
4. Importar el conector dentro el entorno de Python.
5. Generar un diccionario o un dataframe con todos los datos de nuestra conexión, deben ser por lo menos 4:

   - El host
   - El usuario
   - La contraseña
   - El nombre de la base de datos

Ahora observa un ejemplo del código:

```
import mysql.connector

dbConnect = {
        'host' : 'localhost',
        'user': 'root',
        'password' : '',
        'database' : 'python'
}
```

Con ayuda de Python, puedes llegar a crear infinitas aplicaciones. Finalmente, es importante saber que los datos son la materia prima dentro del Big Data, lo que hace fundamental y necesario el utilizar correctamente una base de datos.

Referencias bibliográficas

- Python Argentina. (s.f.). *Acceso A Bases De Datos Desde Python: Interfaz Db-Api*. Recuperado de: https://wiki.python.org.ar/dbapi/

- Oracle. (s.f.). *¿Qué es una base de datos?* Recuperado de: https://www.oracle.com/mx/database/what-is-database.html