

Python Programming

Módulo 4

MySQL en Python

El módulo pymysql

[PyMySQL](#) es un módulo para comunicarse con bases de datos MySQL, que ya sabes utilizar por el simple hecho de haber leído el apartado anterior sobre el módulo `sqlite3`. Ambos adhieren a la DB-API por lo que su estructura es casi idéntica.

PyMySQL puede ser instalado vía pip ejecutando:

```
pip install pymysql
```

Para conectarse a MySQL, usamos la función `connect()`, indicando la dirección y el puerto del servidor, nombre de la base de datos, usuario y contraseña.

```
import pymysql

conn = pymysql.connect(
    host="localhost",
    port=3306,
    user="usuario",
    passwd="clave",
    db="nombredb"
)
```

(Cuando llamamos a una función -y, en general, para cualquier expresión encerrada entre paréntesis- es posible especificar los argumentos en distintas líneas para mejorar la legibilidad. También podemos indicarle a Python que seguiremos un código en la próxima línea vía el carácter `\` —una barra invertida—).

El resto de las funciones son similares a las de SQLite. `conn.cursor()` retorna un cursor a partir del cual podemos llamar a `execute()` para ejecutar consultas.

Nótese que esta última funciona con los caracteres `%s` en lugar del signo de interrogación para incluir objetos de Python en una consulta:

```
for nombre, edad in personas:
    cursor.execute("INSERT INTO personas
VALUES (%s, %s)", (nombre, edad))
```

En el caso de PyMySQL, la excepción lanzada ante un error de sintaxis es: `pymysql.ProgrammingError`.

Por ejemplo:

```
cursor.execute("SELECT FROM personas")
```

provoca lo siguiente:

```
Traceback (most recent call last):  
  [...]  
pymysql.err.ProgrammingError: (1064, "You  
have an error in your SQL syntax; check the  
manual that corresponds to your MySQL server  
version for the right syntax to use near  
'FROM personas' at line 1")
```

También si intentamos acceder a una tabla inexistente:

```
# Lanza `ProgrammingError` porque la tabla  
`profesores` no existe.  
cursor.execute("SELECT * FROM profesores")
```

No obstante, los siguientes casos arrojan otra excepción, `pymysql.err.InternalError`:

```
# La fila `apellido` no existe.  
cursor.execute("SELECT apellido FROM personas")
```

```
# La tabla `personas` ya existe.  
cursor.execute("CREATE TABLE personas (nombre  
VARCHAR(50), edad INT)")
```

Que asimismo podemos capturarla del siguiente modo:

```
try:  
    cursor.execute("SELECT apellido FROM  
personas")  
except pymysql.err.InternalError:  
    print("La consulta no se ejecutó  
correctamente.")
```

¡Sigamos trabajando!