



Conexión a PostgreSQL desde Python utilizando psycopg


Introducción

En esta guía aprenderemos cómo conectarnos a una base de datos de PostgreSQL utilizando Python. Crearemos un proyecto, instalaremos el módulo necesario para la conexión, y realizaremos una consulta básica para recuperar los registros de una tabla. Al final también aprenderemos a cerrar correctamente la conexión.

Paso 1: Crear el proyecto y archivo principal

1. Dentro del proyecto que creamos en la lección anterior, creamos un archivo nuevo de Python llamado:

`prueba_bd.py`

 Este será nuestro archivo principal donde escribiremos el código.

Paso 2: Importar el módulo psycopg

2. En el archivo `prueba_bd.py` agregamos la línea para importar el módulo que nos permitirá conectarnos a la base de datos.

Código:

```
import psycopg
```

Explicación:

Esta línea importa el módulo `psycopg`, que nos permite realizar conexiones y ejecutar consultas en PostgreSQL desde Python.

Paso 3: Crear el objeto de conexión

3. Ahora vamos a crear un objeto de conexión utilizando los parámetros de acceso a nuestra base de datos.

Código:

```
conexion = psycopg.connect(  
    user="postgres",  
    password="admin",  
    host="localhost",  
    port="5432",  
    dbname="test_db"  
)
```

Explicación:

Creamos la conexión con los datos de usuario, contraseña, host, puerto y nombre de la base de datos (`test_db`). Este objeto `conexion` será utilizado para comunicarnos con PostgreSQL.

Paso 4: Probar la conexión

4. Para verificar que la conexión fue exitosa, agregamos una línea para imprimir el objeto `conexion`.

Código:

```
print(conexion)
```

💬 **Explicación:**

Esto nos muestra en consola la información de la conexión establecida, confirmando que nos conectamos correctamente.

🔥 Paso 5: Crear el cursor

5. Luego creamos un cursor a partir de la conexión, que será el encargado de ejecutar sentencias SQL.

👉 **Código:**

```
cursor = conexion.cursor()
```

💬 **Explicación:**

El cursor es un objeto que nos permite enviar y recibir comandos SQL en la base de datos.

🔥 Paso 6: Definir la sentencia SQL

6. Definimos la sentencia SQL que vamos a ejecutar: una consulta para obtener todos los registros de la tabla `persona`.

👉 **Código:**

```
sentencia = "SELECT * FROM persona"
```

💬 **Explicación:**

La variable `sentencia` contiene la consulta SQL que selecciona todos los registros de la tabla `persona`.

🔥 Paso 7: Ejecutar la sentencia

7. Ejecutamos la sentencia SQL utilizando el cursor.

👉 **Código:**

```
cursor.execute(sentencia)
```

💬 **Explicación:**

Esta línea ejecuta la consulta en la base de datos a través del cursor.

🔥 Paso 8: Obtener los resultados

8. Recuperamos todos los registros obtenidos de la consulta y los almacenamos en una variable.

👉 **Código:**

```
registros = cursor.fetchall()
```

💬 **Explicación:**

La variable `registros` es una lista que contiene las filas resultantes de la consulta. Cada fila es una tupla con los datos de cada columna.

🔥 Paso 9: Imprimir los registros

9. Imprimimos los registros obtenidos en consola.

👉 **Código:**

```
print(registros)
```

💬 **Explicación:**

Mostramos en pantalla el contenido de la lista `registros`, lo que nos permitirá ver los datos recuperados de la base de datos.

🔥 Paso 10: Cerrar cursor y conexión

10 Finalmente, cerramos el cursor y la conexión para liberar los recursos.

👉 **Código:**

```
cursor.close()  
conexion.close()
```

🗨 Explicación:

Estas líneas cierran correctamente la comunicación con la base de datos.

📁 Sección de código final completo

✅ Archivo: prueba_bd.py

```
import psycopg
conexion = psycopg.connect(
    user="postgres",
    password="admin",
    host="localhost",
    port="5432",
    dbname="test_db"
)
# print(conexion)
cursor = conexion.cursor()
sentencia = "SELECT * FROM persona"
cursor.execute(sentencia)
registros = cursor.fetchall()
print(registros)

cursor.close()
conexion.close()
```

🎯 Conclusión

En esta guía aprendimos a realizar una conexión desde Python hacia una base de datos de PostgreSQL utilizando el módulo `psycopg`. Logramos conectarnos, ejecutar una consulta para obtener registros de la tabla `persona`, imprimir los resultados y cerrar adecuadamente la conexión. Este es el primer paso para manipular bases de datos desde Python, permitiéndonos interactuar con los datos de forma programática. 🙌

💡 Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 🙌

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)