<u>Ing. Ubaldo Acosta</u> <u>Universidad Python</u>



★ Cómo manejar transacciones en PostgreSQL utilizando Python

Introducción

En esta guía aprenderás cómo manejar **transacciones** en una base de datos PostgreSQL utilizando Python. Las transacciones permiten asegurar que **todas las operaciones se ejecuten correctamente o ninguna se aplique**, evitando inconsistencias en la base de datos.

🌔 Paso 1: Conexión y apertura de cursor sin with

Archivo: transacciones.py

Descripción:

Primero establecemos la conexión y abrimos el cursor de forma manual, sin usar with, para tener control total de la transacción.

import psycopg

conexion = psycopg.connect(

<u>Ing. Ubaldo Acosta</u> <u>Universidad Python</u>

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

Explicación:

Abrimos la conexión y el cursor manualmente para manejar **commit** y **rollback** de manera explícita.



Paso 2: Desactivar el autocommit

Descripción:

Desactivamos el autocommit para que los cambios no se guarden automáticamente.

```
conexion.autocommit = False
```

Explicación:

Con autocommit = False indicamos que los cambios no se guardarán hasta que ejecutemos manualmente commit().



Paso 3: Definir y ejecutar la sentencia insert

Descripción:

Definimos una sentencia SQL para insertar un nuevo registro en la tabla **persona**.

```
sentencia = (
    "INSERT INTO persona (nombre, apellido, email) "
    "VALUES (%s, %s, %s)"
)
valores = ("Maria", "Esparza", "mesparza@mail.com")
cursor.execute(sentencia, valores)
```

Explicación:

Insertamos el registro con nombre Maria, apellido Esparza y su email correspondiente.



Paso 4: Confirmar o deshacer la transacción

<u>Ing. Ubaldo Acosta</u> <u>Universidad Python</u>

Descripción:

Usamos **try-except-finally** para controlar si confirmamos (**commit**) o deshacemos (**rollback**) la transacción según ocurra o no una excepción.

```
try:
    sentencia = (
        "INSERT INTO persona (nombre, apellido, email) "
        "VALUES (%s, %s, %s)"
)
    valores = ("Maria", "Esparza", "mesparza@mail.com")
    cursor.execute(sentencia, valores)
    conexion.commit()
    print("Transacción completada exitosamente")
except Exception as e:
    conexion.rollback()
    print("Ocurrió un error. Se hizo rollback de la transacción:", e)
finally:
    cursor.close()
    conexion.close()
```

Explicación:

- Si todo sale bien \rightarrow ejecutamos commit() y guardamos los cambios.
- Si ocurre un error \rightarrow ejecutamos rollback () para deshacer los cambios.
- Finalmente cerramos el cursor y la conexión.

Código completo por archivo

- Archivo completo: transacciones.py
- Aquí tienes el **código completo actualizado**:

```
import psycopg

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

cursor = conexion.cursor()
conexion.autocommit = False

try:
    sentencia = (
        "INSERT INTO persona (nombre, apellido, email) "
```

<u>Universidad Python</u>

```
"VALUES (%s, %s, %s)"
    )
    valores = ("Maria", "Esparza", "mesparza@mail.com")
    cursor.execute(sentencia, valores)
    conexion.commit()
    print("Transacción completada exitosamente")
except Exception as e:
    conexion.rollback()
    print("Ocurrió un error. Se hizo rollback de la transacción:", e)
   cursor.close()
    conexion.close()
```

K Conclusión

En esta guía aprendiste a manejar **transacciones manualmente** en PostgreSQL usando Python. Configuramos autocommit = False para controlar cuándo guardar o deshacer los cambios, y utilizamos commit() y rollback() según el resultado de la operación.

Con este enfoque tendrás mayor control sobre las operaciones que modifican el estado de la base de datos.



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

¡Saludos! 🤏

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de GlobalMentoring.com.mx