📌 Cómo manejar múltiples operaciones dentro de una transacción en PostgreSQL usando Python

---

✨ Introducción

En esta guía aprenderás cómo ejecutar **múltiples sentencias SQL dentro de una transacción** en una base de datos PostgreSQL usando Python. Veremos cómo insertar y actualizar registros en la misma transacción, garantizando que **o se ejecutan todas las operaciones o ninguna se guarda**.

🔥 Paso 1: Conexión y apertura de cursor sin `with`

📁 **Archivo:** `transacciones.py`

Descripción:
Establecemos la conexión y el cursor de forma manual para controlar la transacción explícitamente.

```
import psycopg

conexion = psycopg.connect(
```

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

👉 **Explicación:**
Abrimos la conexión y el cursor de forma manual, y desactivamos el **autocommit** para tener control total de la transacción.

🔥 Paso 2: Ejecutar la sentencia `INSERT`

Descripción:
Definimos y ejecutamos la primera sentencia SQL para insertar un nuevo registro en la tabla **persona**.

```
    sentencia_insert = (
        "INSERT INTO persona (nombre, apellido, email) "
        "VALUES (%s, %s, %s)"
    )
    valores_insert = ("Rocio", "Mijares", "rmijares@mail.com")
    cursor.execute(sentencia_insert, valores_insert)
```

👉 **Explicación:**
Insertamos un nuevo registro con nombre **Maria**, apellido **Esparza** y su email correspondiente.

🔥 Paso 3: Ejecutar la sentencia `UPDATE`

Descripción:
Definimos y ejecutamos una segunda sentencia SQL para actualizar un registro existente.

```
    sentencia_update = (
        "UPDATE persona "
        "SET nombre=%s, apellido=%s, email=%s "
        "WHERE id_persona=%s"
    )
    valores_update = ("Juan Carlos", "Juarez", "jcjuarez@mail.com", 1)
    cursor.execute(sentencia_update, valores_update)
```

👉 **Explicación:**
Actualizamos el registro con **id_persona = 1**, cambiando su nombre a **Juan Carlos**, apellido a **Juarez** y email a **jcjuarez@mail.com**.

🔥 Paso 4: Confirmar o deshacer la transacción

Descripción:
Usamos **try-except-finally** para controlar si confirmamos (**commit**) o deshacemos (**rollback**) la transacción según ocurra o no un error.

```
try:
    # Las sentencias ya ejecutadas aquí
    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 es correcto → **commit**
❌ Si ocurre error → **rollback**

✅ Código completo por archivo

📂 **Archivo completo:** `transacciones.py`

➡️ **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 = (
        "INSERT INTO persona (nombre, apellido, email) "
        "VALUES (%s, %s, %s)"
    )
    valores_insert = ("Rocio", "Mijares", "rmijares@mail.com")
    cursor.execute(sentencia_insert, valores_insert)
    sentencia_update = (
        "UPDATE persona "
        "SET nombre=%s, apellido=%s, email=%s "
        "WHERE id_persona=%s"
    )
    valores_update = ("Juan Carlos", "Juarez", "jcjuarez@mail.com", 1)
    cursor.execute(sentencia_update, valores_update)
    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()
```

🎉 Conclusión

En esta guía aprendiste cómo manejar **múltiples operaciones dentro de una transacción** en PostgreSQL usando Python. Lograste ejecutar varias sentencias SQL que modifican la base de datos, asegurando que **todas se ejecuten correctamente o ninguna cambie el estado de la base de datos**.

Esto garantiza la integridad de los datos y el control total de las modificaciones. 🚀

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

¡Saludos! 👋

Ing. Marcela Gamiño e Ing. Ubaldo Acosta

Fundadores de GlobalMentoring.com.mx