

# Práctica 19

## Procesos Batch

### Objetivo

*En esta práctica se realizan ejercicios de programación y ejecución de transacciones dentro de procesos batch.*

### Introducción

Además de ser interactivo, **psql** puede procesar comandos por lotes, ejecutando las instrucciones que se desean ejecutar escritas en un archivo de texto, desde el sistema operativo mediante la siguiente sintaxis:

```
psql -d mibase -f demo.txt
```

El siguiente comando también funciona en el mismo sentido:

```
psql -d mibase < demo.txt
```

Si es necesario especificar la conexión se debe poner:

```
psql -h host -U user -p < demo.txt
```

Se puede solicitar la ejecución de un solo comando y terminar inmediatamente mediante la siguiente forma:

```
psql -d mibase -c "comando sql"
```

Se puede omitir la referencia a la base de datos, si dentro del script se emplea el comando **|c mibase**.

Si se requiere, se puede guardar la salida en un archivo de texto:

```
psql < demo.psql > salida.txt
```

Estos son algunos casos en que se aconseja el uso los scripts:

- Cuando se ejecutan repetidamente las mismas operaciones.
- Se pueden hacer nuevos scripts a partir de otros similares.
- Cuando se ejecutan pruebas, que se repiten varias veces con ciertas correcciones, solamente se edita la parte a modificar en el script.
- Se pueden repartir los scripts entre distintos usuarios para que todos ejecuten lo

mismo.

## Equipo necesario

Una computadora con sistema operativo **Windows** que cuente con **psql**.

## Metodología

1. Ejecute la herramienta **SQL Shell (psql)**.
2. Deje el usuario **postgres** y proporcione la contraseña definida en la primera práctica.
3. Ahora cree una nueva base de datos llamada banco y conéctese a ella:

```
postgres=# CREATE DATABASE banco;
CREATE DATABASE
postgres=# \c banco;
Ahora está conectado a la base de datos «banco» con el usuario «postgres».
```

4. A continuación, cree una tabla llamada **cliente** con los atributos siguientes: **clavecte** (entero, llave principal), **nombre** (varchar de 15), **apellido** (varchar de 15) y **teléfono** (varchar de 12).

5. Inserte los registros que se muestran en la siguiente tabla:

clavecte	nombre	apellido	teléfono
12	Juan	Gómez	443 312 5720
26	Lucy Dora	Lara	443 342 5287
35	Jaime	Nava	443 212 4194
41	Blanca	Luna	443 276 3197
59	Lucrecia	Borja	443 314 5729

6. Revise el contenido de la tabla:

```
banco=# SELECT * FROM cliente;
```

7. Obtenga una imagen completa de la pantalla, mostrando el resultado de esta consulta y consérvela como evidencia.

8. Ahora cree otra tabla llamada **sucursal**, con los atributos siguientes: **clavesuc** (entero, llave principal), **nombresuc** (varchar de 20), **gerente** (varchar de 25) y **ciudad** (varchar de 15).

9. Inserte los registros que se muestran en la siguiente tabla:

clavesuc	nombresuc	gerente	ciudad
124	La Villa	Manuel Villa	Morelia
256	Lomas Altas	Lucas Becerra	Uruapan
275	Montecito	Omar Bravo	Zamora
341	Altozano	Alma Torres	Morelia

10. Revise el contenido de la tabla:



```
banco=# SELECT * FROM sucursal;
```

11. Obtenga una imagen completa de la pantalla, mostrando el resultado de esta consulta y consérvela como evidencia.
12. Cree una tercera tabla, llamada **cuentas**, con los atributos siguientes: número (entero, llave principal), clavecte (entero), clavesuc (entero) y saldo (numérico).
13. Ahora, usando ALTER TABLE deberá crear una restricción de llave ajena llamada **fk\_cliente**, que vincule el atributo **clavecte** con la llave de la tabla **cliente**, permitiendo actualización en cascada.
14. Luego deberá crear otra llave ajena llamada **fk\_sucursal** que vincule **clavesuc** con la llave de la tabla **sucursal**, permitiendo actualización en cascada.
15. Ahora, inserte en **cuentas** los registros que se muestran en la siguiente tabla:

numero	clavecte	clavesuc	saldo
3491	35	256	1155.23
5179	41	124	5341.64
6331	12	275	452.76
6993	59	124	500.00

16. Revise el contenido de la tabla:

```
banco=# SELECT * FROM cuentas;
```

17. Obtenga una imagen completa de la pantalla, mostrando el resultado de esta consulta y consérvela como evidencia.
18. Para verificar los nombres de las restricciones que se crearon, ejecute la siguiente consulta:  

```
banco=# SELECT table_name, constraint_name, column_name FROM
information_schema.key_column_usage WHERE table_schema=
'public';
```
19. A continuación, utilizando el **bloc de notas**, escriba las instrucciones de la siguiente transición:

```
\c banco;
START TRANSACTION;
INSERT INTO cliente VALUES (45,'Dora','Bustos','433 312 1570');
UPDATE cliente SET nombre='Rosa' WHERE apellido='Luna';
INSERT INTO cuentas VALUES (4323,45,341,650.15);
UPDATE sucursal SET clavesuc=244 WHERE ClaveSuc=124;
DELETE FROM cliente WHERE nombre='Lucrecia';
UPDATE cuentas SET saldo= ROUND(saldo*1.10,2) WHERE número IN
(6331,3491,4323);
COMMIT;
```

20. Guarde el archivo con el nombre de **prueba1.txt** y con codificación **ANSI**, en la carpeta **DATA**.

21. Ejecute la aplicación **Símbolo del Sistema como administrador**, sin cerrar el Shell **psql**.

22. Cambie a la dirección:

```
cd C:\Archivos de Programa\PostgreSQL\16\bin
```

23. Ahora va a ejecutar la transacción en modo **Batch**, ejecutando todas las instrucciones contenidas en el archivo **prueba1.txt**, mediante la siguiente opción:

```
C:\Archivos de Programa\PostgreSQL\16\bin>psql -U postgres -f  
\DATA\prueba1.txt  
Contraseña para usuario postgres:
```

24. Proporcione su contraseña.

25. Intencionalmente existe un error en una sentencia de la transacción, por lo que ésta es abortada y los cambios no se realizaron.

26. Obtenga una imagen completa de la pantalla, mostrando los mensajes mostrados por esta ejecución y consérvela como evidencia.

27. Edite el archivo **prueba1.txt** y reemplace el nombre de **Lucrecia** por el de **Lucy Dora** en la línea 7.

28. Vuelva a realizar la instrucción anterior.

```
C:\Archivos de Programa\PostgreSQL\16\bin>psql -U postgres -f  
\DATA\prueba1.txt  
Contraseña para usuario postgres:
```

29. Proporcione su contraseña y observe que ahora todo procede con normalidad.

30. Obtenga una imagen completa de la pantalla, mostrando los mensajes mostrados por esta ejecución y consérvela como evidencia.

31. Pase a la ventana del Shell **psql** y verifique los cambios en las tres tablas, realizados por esta transacción.

```
banco=# SELECT * FROM cliente;  
banco=# SELECT * FROM sucursal;  
banco=# SELECT * FROM cuentas;
```

32. Obtenga una imagen completa de la pantalla, mostrando el resultado de estas consultas y consérvela como evidencia.

33. En un nuevo documento del bloc de notas, escriba las instrucciones para una nueva transacción:

```
\c banco;  
START TRANSACTION;  
INSERT INTO cliente VALUES (16,'Homero','Rosas','433 324  
7709');  
UPDATE cliente SET apellido ='Medina' WHERE nombre='Dora';  
INSERT INTO sucursal VALUES (183,'Rio Grande','Miguel
```



```
Flores','Zacapu');  
UPDATE cuentas SET clavesuc=341 WHERE clavesuc=244;  
DELETE FROM sucursal WHERE gerente='Manuel Villa';  
UPDATE cuentas SET saldo= saldo*1.10 WHERE clavesuc =341;  
COMMIT;
```

34. Guarde el archivo con el nombre de **prueba2.txt** y con codificación **ANSI**, en su carpeta de **DATA**.
35. En la ventana del símbolo de sistema ejecute el archivo **prueba2.txt** como se indica:

```
C:\Archivos de Programa\PostgreSQL\16\bin>psql -U postgres -f  
\DATA\prueba2.txt  
Contraseña para usuario postgres:
```

36. Proporcione su contraseña y observe la ejecución de la transacción con normalidad.
37. Obtenga una imagen completa de la pantalla, mostrando los mensajes mostrados por esta ejecución y consérvela como evidencia.
38. Pase a la ventana del Shell **psql** y verifique los cambios en las tres tablas, realizados por esta transacción.

```
banco=# SELECT * FROM cliente;  
banco=# SELECT * FROM sucursal;  
banco=# SELECT * FROM cuentas;
```

39. Obtenga una imagen completa de la pantalla, mostrando el resultado de estas consultas y consérvela como evidencia.
40. Guarde el resultado de una consulta de las tres tablas, ejecutando la siguiente instrucción:

```
C:\Archivos de Programa\PostgreSQL\16\bin>psql -U postgres -d  
banco -c "SELECT * FROM cliente; SELECT * FROM sucursal; SELECT  
* FROM cuentas" > \DATA\prueba3.txt  
Contraseña para usuario postgres:
```

41. Verifique que se creó el archivo **prueba3.txt** en la carpeta DATA.
42. Por último, genere un respaldo de su base de datos:

```
C:\Archivos de Programa\PostgreSQL\16\bin>pg_dump -U postgres  
banco > \DATA\banco.sql  
Contraseña:
```

43. Verifique que se creó el archivo **banco.sql** en la carpeta **DATA**.

44. Anexe como evidencia, los cuatro archivos creados en esta práctica.

45. Cierre todas las aplicaciones que tenga abiertas.

46. Fin de la Práctica.

## Evidencias

El alumno deberá enviar al instructor las evidencias requeridas durante la realización de la práctica.

## Sugerencias didácticas

El instructor deberá atender a los alumnos que tengan dificultades en la interpretación y la realización de las instrucciones de la práctica.

## Resultados

Se aprendió a realizar procesos batch mediante la herramienta ***PostgreSQL***, creando transacciones que modifican diversos registros de distintas tablas.

## Bibliografía

- <https://www.postgresql.org/docs/15/backup-dump.html>
- <https://www.postgresql.org/docs/15/app-pgdump.html>