

CIERRE DE TEMA

AUTOMATIZACIÓN DE TAREAS MEDIANTE USO DEL SHELL DE POSTGRESQL

Por Martínez de la Cruz José Julián

Fecha: 28/06/2025

Palabras clave: Automatización, Shell PostgreSQL, Privilegios, Exportación de datos, Importación de datos, Automatización de tareas administrativas en PostgreSQL.

1. Objetivo del Tema

El propósito de esta práctica es lograr un Shell personalizado que permita la automatización de tareas clave en la administración de Bases de Datos de PostgreSQL mediante la implementación de scripts que ejecuten comandos tanto de respaldo, restauración, exportación, importación y registro de usuarios con privilegios para mantener la base de datos de Mi Financiera Demo perfectamente gestionada.

2. Resumen

Para mejorar la administración de la base de datos de Mi Financiera Demo, se realizó un Shell con scripts personalizados que permiten automatizar tareas clave para la misma. Todo esto se organiza y se ejecuta por medio de un menú principal en el Shell de Windows que permite al usuario acceder fácilmente a las distintas funciones sin interactuar directamente con PostgreSQL con el objetivo de reducir errores manuales y facilitando la gestión.

```
C:\WINDOWS\system32\cmd.  X  +  v

*****
*   Menu de Respaldo y Restauracion   *
*****
1. Respaldar la base de datos
2. Restaurar la base de datos existente
3. Crear nueva base de datos y restaurar respaldo
4. Borrar una base de datos
5. Crear usuario y asignar permisos
6. Modificar permisos de usuario
7. Ver usuarios y sus privilegios
8. Exportar Script
9. Importar Script
0. Salir
*****
Selecciona una opcion (1/2/3/4/5/6/7/8/9): |
```

Este menú general, llamado *main_menú*, actúa como un punto de control desde donde se puede mandar a llamar a cada uno de los diferentes scripts los cuales permiten:

1. **Registrar Usuarios con Privilegios:** Este script permite registrar nuevos usuarios en PostgreSQL y asignarles privilegios (crud) en una base de datos y tabla determinada según sus necesidades, lo que permite tener controlada y asegurada la base de datos.

```

ca C:\Windows\system32\cmd.exe
*****
*      Crear Usuario de PostgreSQL      *
*****
Ingresa el nombre de usuario de PostgreSQL [por defecto: postgres]:

Ingresa el nombre del nuevo usuario: JulianPractica
Ingresa la contraseña para el nuevo usuario: 123

Creando usuario JulianPractica, Confirme la contraseña de postgres...

Contraseña para usuario postgres:
CREATE ROLE
Usuario JulianPractica creado exitosamente.

*****
*      BD, TABLA Y PERMISOS PARA EL NUEVO USUARIO      *
*****
Listando bases de datos disponibles, Confirme la contraseña de postgres...

Contraseña para usuario postgres:
1. postgres
2. mi_financiera_demo_II
3. mi_financiera_demo
4. test
Selecciona la base de datos a la que se le otorgaran permisos al nuevo usuario: 3

Listando tablas disponibles en la base de datos mi_financiera_demo...
Contraseña para usuario postgres:
1. sucursal
2. cliente
3. campana_fidelidad_cliente
4. dw_historico_saldo
5. movimiento_tarjeta_credito
6. pago_credito
7. seguimiento_cobro
8. empleado
9. prestamo
10. calendario_pago
11. tarjeta_credito
Selecciona la tabla: 1

Selecciona los permisos a otorgar:
1. ALL PRIVILEGES
2. SELECT
3. INSERT
4. UPDATE
5. DELETE
6. SELECT, INSERT, UPDATE
7. Personalizado
Opción: 6

Otorgando permisos SELECT, INSERT, UPDATE al usuario JulianPractica en la tabla public.sucursal...
Contraseña para usuario postgres:
GRANT

Permisos otorgados exitosamente para el usuario JulianPractica sobre public.sucursal.
Presione una tecla para continuar . . .

```

2. **Modificar Privilegios:** Permite modificar los privilegios ya asignados a un usuario, ya sea para restringir o ampliar lo que puede hacer en la base de datos y tabla en específico. De este modo, se mantiene un control total sobre lo que se puede hacer y quién lo puede hacer dentro del sistema.

```

C:\Windows\system32\cmd.exe
*****
*   Modificar Permisos en PostgreSQL   *
*****

Bases de datos disponibles:
Contraseña para usuario postgres:
    mi_financiera_demo
    mi_financiera_demo_II
    postgres
    template0
ECHO está desactivado.
    template1
ECHO está desactivado.
    test

Escriba el nombre de la base de datos [por defecto: mi_financiera_demo]:

Usuarios disponibles en la base de datos mi_financiera_demo:
Contraseña para usuario postgres:
    1.postgres
    2.julian
    3.admin
    4.jesus
    5.julianprueba
    6.julianpractica

Selecciona la tabla por número para ver sus privilegios: 6

===== Permisos disponibles =====
1. SELECT      - Leer datos de tablas
2. INSERT      - Insertar datos en tablas
3. UPDATE      - Modificar datos en tablas
4. DELETE      - Eliminar datos de tablas
5. TRUNCATE    - Truncar tablas
6. REFERENCES  - Crear claves foraneas
7. TRIGGER     - Crear triggers
8. TODOS LOS PERMISOS
=====

Ingrese el/los permisos a asignar (ej. 1,3,4 o 0 para todos): 1

Listando tablas disponibles en la base de datos mi_financiera_demo...
Contraseña para usuario postgres:
1. cliente
2. campana_fidelidad_cliente
3. sucursal
4. dw_historico_saldo
5. movimiento_tarjeta_credito
6. pago_credito
7. seguimiento_cobro
8. empleado
9. prestamo
10. calendario_pago
11. tarjeta_credito

Selecciona la tabla por número para ver sus privilegios: 1

Revocando permisos existentes para julianpractica en cliente...
Contraseña para usuario postgres:
REVOKE
Otorgando permisos [SELECT] sobre cliente a julianpractica...
Contraseña para usuario postgres:
GRANT

Permisos actualizados correctamente para el usuario julianpractica.
Presione una tecla para continuar . . .

```

3. **Ver Usuarios:** Este script lista todos los usuarios registrados en PostgreSQL junto con sus respectivos privilegios, permitiendo una visualización rápida del estado actual de los accesos.

```

C:\Windows\system32\cmd.exe
*****
* Ver privilegios de tablas *
*****
Ingresa el nombre de usuario de PostgreSQL [por defecto: postgres]:
Ingresa el nombre de la base de datos [por defecto: mi_financiera_demo]:

Listando tablas disponibles en la base de datos mi_financiera_demo...
Contraseña para usuario postgres:
1. cliente
2. campana_fidelidad_cliente
3. sucursal
4. dw_historico_saldo
5. movimiento_tarjeta_credito
6. pago_credito
7. seguimiento_cobro
8. empleado
9. prestamo
10. calendario_pago
11. tarjeta_credito

Selecciona la tabla por número para ver sus privilegios: 1

=====
Privilegios sobre la tabla cliente en mi_financiera_demo
=====
Contraseña para usuario postgres:

          Privilegios
Esquema | Nombre | Tipo | Privilegios | Privilegios de acceso a columnas | Políticas
-----+-----+-----+-----+-----+-----
public | cliente | tabla | postgres=arwdDxt/postgres+ | | 
      |         |      | julian=w/postgres          + | | 
      |         |      | jesus=r/postgres           + | | 
      |         |      | julianpractica=r/postgres  | | 
(1 fila)

Presione una tecla para continuar . . .

```

4. **Exportación de Datos:** En cuanto a la gestión de datos, se realizó un script que permita exportar todos los datos de una tabla en múltiples formatos: TEXT; CSV, JSON y XML. Estos formatos podrían cubrir bastantes necesidades como puede ser migrar de Modelo de relacional a no relacional o permitir realizar reportes.
5. **Importación de Datos:** Se integró otro script que permita importar los datos desde los archivos TEXT y CSV del sistema directamente hacia la base de datos, facilitando la carga de datos.

Finalmente, cada uno de estos scripts están diseñados para que se puedan operar de manera intuitiva, autónoma y facilitándole la interacción con los mismos al usuario mediante opciones numéricas. También se procura que se acceda a los scripts desde el menú general para permitir una operación más segura y organizada para administrar de manera eficiente las bases de datos de PostgreSQL.

3. Conclusión

La práctica permite cumplir correctamente el objetivo ya que con el uso de este Shell personalizado se puede administrar estructurada y organizadamente de forma más eficiente la base de datos Mi Financiera Demo. Al contar con un menú principal se simplifica demasiado el trabajo y reduce el riesgo de errores que pueden ser causados por personal que no cuenta con la suficiente experiencia pues todo está pensado para que el administrador no tenga que ejecutar

comandos sueltos o recordar la sintaxis de los comandos de PostgreSQL, de esta manera de cumple todo lo esencial para mantener la base de gestionada correctamente.

4. Anexos

Para complementar el desarrollo de la práctica, a continuación, se comparte un enlace a Google Drive, el cual contiene los archivos .bats utilizados en la automatización.

https://drive.google.com/file/d/1t3_YjCTHaGaSmt8Hu2JfQk1AUswUChq/view?usp=sharing

A su vez, en el siguiente enlace se proporciona un vídeo explicando a detalle el código y funcionamiento del Shell personalizado.

<https://youtu.be/zvE0V3EtH2Y>

5. Referencias

- Presentación: *Tabla de referencia de comandos*, MTI Jesús E. Romero, UTTEC.
- PostgreSQL (s.f). *SQL Syntax Copy*. PostgreSQL Documentation.

8. Reflexiones Finales

1. **Automatizar ahorra tiempo y evita errores:** Tener scripts listos para importar/exportar o dar permisos específicos a usuarios es mucho más eficiente que hacer todo a mano o mediante PgAdmin.
2. **El menú completo hace la diferencia:** Tener un solo punto de entrada que organiza todas las tareas da claridad, estructura y facilita al flujo de trabajo.
3. **Los formatos de exportación/importación amplían la utilidad:** Permitir trabajar con CSV, JSON, XML y TXT facilita migrar datos entre sistemas o documentar de mejor manera la información.