Universidad Nacional Autónoma de México
Facultad de Ingeniería
Semestre 2026-2

Bases de Datos

Grupo 01

Ing. Fernando Arreola Franco

Tarea 3

Fecha de entrega: 20 de febrero de 2026

Parra Vera Héctor Jesús

Calificación: _____

1. Creación de usuario con restricciones de conexión y vigencia

En PostgreSQL, la creación de un usuario con características específicas como un límite de conexiones simultáneas, una contraseña segura y una fecha de expiración se realiza mediante el comando `CREATE ROLE` o su alias `CREATE USER` [1]. La sintaxis permite definir todos estos atributos en una sola instrucción [2].
Para cumplir con el requerimiento de crear un usuario con límite de conexiones, contraseña y vigencia de un mes, se utiliza el siguiente bloque de código SQL:

```
-- Se asume un mes a partir de la fecha de creación
-- (ejemplo: hasta el 20 de marzo de 2026)
CREATE USER usuario_app
    WITH PASSWORD 'SuperSecreta123'
    CONNECTION LIMIT 5
    VALID UNTIL '2026-03-20 00:00:00';
```

El parámetro `CONNECTION LIMIT` restringe cuántas conexiones concurrentes puede abrir este rol específico, lo cual es fundamental para evitar el agotamiento de recursos en el servidor [3]. Por su parte, el atributo `VALID UNTIL` establece una marca de tiempo tras la cual la contraseña del rol deja de ser válida para iniciar sesión, deshabilitando el acceso de forma automática [1].

2. Creación de rol, asignación de permisos y vinculación

El control de acceso basado en roles (RBAC) es una práctica de seguridad que separa la definición de permisos de las cuentas de usuario individuales [4]. El proceso consiste en crear un rol lógico base, otorgarle los privilegios necesarios sobre los objetos mediante el comando `GRANT`, y finalmente asignar este grupo de permisos al usuario [5].
Para otorgar permisos de lectura (`SELECT`), actualización (`UPDATE`) y borrado (`DELETE`) sobre la tabla `estudiante`, e integrarlo con el usuario creado en el paso anterior, se ejecutan las siguientes instrucciones:

```
-- 1. Crear el rol que agrupará los permisos (sin atributo de login)
CREATE ROLE rol_gestion_estudiantes;

-- 2. Asignar los permisos DML solicitados sobre la tabla "estudiante"
GRANT SELECT, UPDATE, DELETE ON estudiante TO rol_gestion_estudiantes;

-- 3. Asignar dicho rol al usuario creado en el paso anterior
GRANT rol_gestion_estudiantes TO usuario_app;
```

Es importante destacar que en la arquitectura de PostgreSQL, para que el rol pueda acceder efectivamente a la tabla `estudiante`, también debe tener permiso de uso sobre el esquema que la contiene (generalmente el esquema `public`) [1]. Si las políticas de seguridad del clúster son estrictas, se debería añadir una instrucción previa: `GRANT USAGE ON SCHEMA public TO rol_gestion_estudiantes;`.

Referencias

[1] The PostgreSQL Global Development Group, *PostgreSQL 16.0 Documentation*, 2023.

[2] A. Beaulieu, *Learning SQL: Generate, Manipulate, and Retrieve Data*. O'Reilly Media, 3rd ed., 2020.

[3] C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*. Cengage Learning, 13th ed., 2018.

[4] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*. McGraw-Hill Education, 7th ed., 2020.

[5] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*. Pearson, 7th ed., 2015.