Universidad Nacional Autónoma De México

Facultad de Ingeniería

Grupo: 01

Curso: Base de Datos

Alumno: Ortiz Valles Joaquín Rafael

Profesor: Ing. Fernando Arreola Franco

Tarea: 3

ÍNDICE

Crear un usuario, con límite de conexiones, contraseña y 1 mes de vigencia

1. Creación del usuario

Define un nombre único.
Usa el comando o instrucción correspondiente para crear el usuario.

Ejemplo en SQL Server:

```
CREATE LOGIN usuario_demo WITH PASSWORD = 'ContraseñaSegura123!';
CREATE USER usuario_demo FOR LOGIN usuario_demo;
```

2. Asignación de contraseña

Establece una contraseña segura.
En algunos sistemas puedes definir políticas como longitud mínima, complejidad o caducidad.

3. Límite de conexiones

En sistemas como PostgreSQL puedes limitar el número de conexiones simultáneas:

```
ALTER ROLE usuario_demo CONNECTION LIMIT 3;
```

En sistemas operativos tipo Linux, puedes usar ulimit o configurar en /etc/security/limits.conf.

4. Vigencia temporal (1 mes)

Puedes usar funciones de expiración de cuenta o programar tareas que desactiven al usuario.

Ejemplo en Oracle:

```
CREATE USER usuario_demo IDENTIFIED BY "Contraseña123"
PASSWORD EXPIRE
ACCOUNT LOCK AFTER 30 DAYS;
```

O bien, en sistemas sin soporte directo, puedes usar un script programado (cron en Linux, SQL Agent en SQL Server) que desactive el usuario tras 30 días.

Crear un role, asignar permisos de lectura, actualización y borrado en una tabla de nombre “estudiante”. Asignar dicho role al usuario del paso anterior.

1. Crear el rol con permisos específicos

```
CREATE ROLE rol_estudiante;
```

Este rol no tiene permisos aún. Lo siguiente es asignarle privilegios sobre la tabla estudiante.

2. Asignar permisos sobre la tabla

```
GRANT SELECT, UPDATE, DELETE ON TABLE estudiante TO rol_estudiante;
```

SELECT: permite leer los datos.
UPDATE: permite modificarlos.
DELETE: permite eliminarlos.

3. Asignar el rol al usuario creado previamente

Supongamos que el usuario se llama usuario_demo:

```
GRANT rol_estudiante TO usuario_demo;
```

Esto le otorga al usuario los permisos definidos en el rol, sin necesidad de asignarlos directamente.

Si quieres que el usuario asuma automáticamente el rol al iniciar sesión, puedes usar:

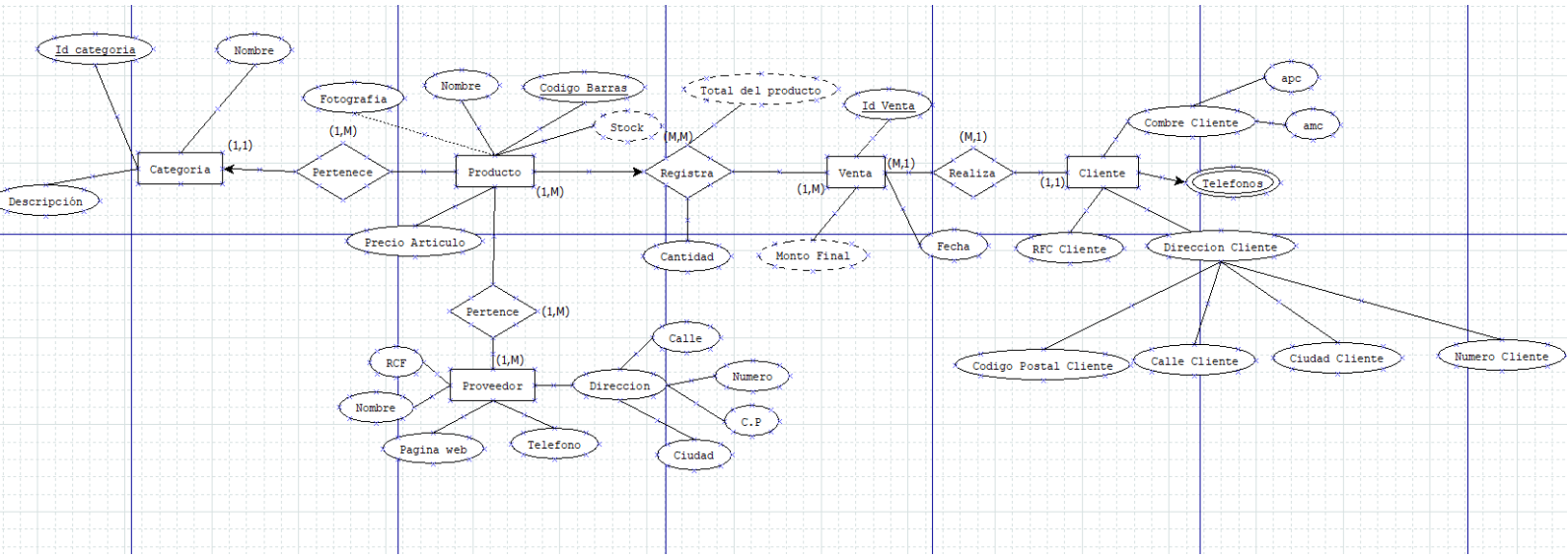
```
ALTER USER usuario_demo SET ROLE rol_estudiante;
```

Si la tabla estudiante no existe aún, deberías crearla antes de asignar permisos:

```
CREATE TABLE estudiante (
  id SERIAL PRIMARY KEY,
```

```
  nombre VARCHAR(100),
  carrera VARCHAR(100),
  semestre INT
);
```

Implementar el ejercicio 2_2 en algún software de diseño.



Bibliografía

[1] Microsoft, “Roles de nivel de base de datos - SQL Server,” Microsoft Learn, 31-Jul-2025. [En línea]. Disponible en: https://learn.microsoft.com/es-es/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-ver16

[2] S. Robles Nuñez, “USUARIOS, PRIVILEGIOS Y ROLES DE LAS BASES DE DATOS,” Prezi, 18-Sep-2015. [En línea]. Disponible en: https://prezi.com/lgtxy00p1gwa/usuarios-privilegios-y-roles-de-las-bases-de-datos/

[3] IBM, “Privilegios de bases de datos SQL Server,” IBM Docs, 2023. [En línea]. Disponible en: https://www.ibm.com/docs/es/baw/23.0.x?topic=privileges-sql-server-database