Evidencias

- Será necesario añadir las sentencias ejecutadas así como consultas de comprobación.

La empresa TechData S.L., dedicada a la venta y soporte de equipos informáticos, ha decidido implantar una base de datos en PostgreSQL para gestionar la información de clientes y pedidos.

Como administrador de bases de datos, tu tarea consiste en configurar los usuarios, roles y privilegios del sistema, creando vistas personalizadas y aplicando medidas de seguridad y control de acceso.

El objetivo es que cada perfil (administrador de ventas, empleado y auditor) acceda solo a la información necesaria, garantizando la protección de datos y el principio de mínimo privilegio.

Actividad 1 – Creación de usuarios y roles

1.- Crea una base de datos llamada `empresa`.
2.- Crea tres usuarios con las opciones definidas:

- `admin_ventas`
  - El usuario puede iniciar sesión con la contraseña 'Audit$2025'
  - Puede crear bases de datos
  - Puede crear y gestionar roles
  - Hereda privilegios de roles asignados
  - Su cuenta expira el 31/12/2026

- `empleado_ventas`
  - El usuario puede iniciar sesión con la contraseña 'Empleado#2025'
  - Tiene un límite de 3 conexiones simultáneas
  - No puede crear roles ni bases de datos

- `auditor`
  - El usuario puede iniciar sesión con la contraseña 'Audit#2025'
  - No hereda permisos de otros roles
  - Solo puede tener una sesión activa

3.- Crea un rol llamado `ventas_grupo`.

- No puede iniciar sesión
- Puede heredar privilegios
- No puede crear bases de datos ni roles
- No es superusuario ni tiene permisos de replicación
- No puede omitir políticas de seguridad por filas

4.- Crea un rol llamado `ventas_acceso`

- Puede **iniciar sesión**
- Tiene una **contraseña cifrada** `'Ventas#2025'`
- Puede realizar hasta **10 conexiones simultáneas**
- **Hereda privilegios** de otros roles
- No puede crear bases de datos ni roles
- No es superusuario ni tiene permisos de replicación
- No puede omitir políticas de seguridad

5.- Asocia `empleado_ventas` y `admin_ventas` al rol `ventas_grupo`. `admin_ventas` tendrá permisos de revocación y asignación al role, `empleado_ventas` no.

6.- Crea una tabla con el usuario `empleado_ventas`, y luego intenta eliminar el usuario. ¿Qué ocurre? ¿Cómo podrías eliminar el usuario? ¿Qué consecuencias tendría?

Actividad 2 - Creación de vistas personalizadas

Creación de tablas y datos base

1. Crea una base de datos para el área de ventas llamada ventas_db
2. Crea las tablas e inserta algunos registros de ejemplo:
    - clientes (id, nombre, dni, telefono, email, saldo)
    - pedidos (id, id_cliente, fecha, total, estado)
3. Inserta algunos registros de ejemplo:
4. Verifica el contenido:

Creación de vistas personalizadas

5. Los administradores deben tener acceso total a los datos de clientes y pedidos, con el número de pedidos y total de todos los pedidos
6. Los empleados solo deben ver información de contacto y saldo, sin DNI ni email.
7. El auditor puede consultar datos pero sin información personal identificable.

Asignación de permisos a usuarios

8. Concede permisos de lectura sobre las vistas a cada role o usuarios:
9. Revoca permisos directos sobre las tablas base, para que solo puedan acceder a través de las vistas
10. Comprueba que cada usuario solo puede acceder a su vista correspondiente:

Ampliación

11. Crea una nueva vista vista_clientes_negativos que muestre solo clientes con saldo menor que 0.

12. Asigna esta vista al rol ventas_acceso para que todos los usuarios de ventas puedan consultarla.

Actividad 3 – Sinónimos y alias

Creación de sinónimos simulados

Recuerda: PostgreSQL no tiene el comando `CREATE SYNONYM` (como Oracle o SQL Server).

- Crea una vista simple que actúe como alias `vista_clientes_admin` de la tabla `clientes`:

- Crea otra vista que funcione como sinónimo de la vista `vista_clientes_admin`:

- Crea un alias más complejo para la vista de empleados, que renombre columnas

Permisos y pruebas de acceso

- Concede permisos de lectura sobre los alias a los usuarios o roles que consideres.
- Comprueba la diferencia entre un usuario con permisos y sin permisos

Actividad 4 – Gestión de privilegios (criterios d–g)

Comprobación previa

- Lista los privilegios actuales sobre todas las tablas y vistas:

- Comprueba también los roles y sus pertenencias:

Asignación y prueba de privilegios

- Sobre la tabla clientes
    - Concede privilegios de manipulación a `ventas_grupo` sobre `clientes`
    - Da permisos de eliminación **solo** al usuario `admin_ventas`:
    - Revoca explícitamente el permiso de eliminación a `empleado_ventas`:
    - Verifica el efecto práctico:

Agrupación de privilegios y rol de solo lectura

- Crea un rol de solo lectura:
- Concede privilegios de lectura sobre todas las tablas y vistas actuales:
- Haz que los **nuevos objetos creados** también sean legibles por este rol:
- Asigna el rol al usuario `auditor`:
- Comprueba que el usuario `auditor` hereda los permisos:
- Inicia sesión como `auditor` y prueba:

Gestión dinámica de privilegios

Sobre la tabla clientes:

- Elimina los permisos del grupo de ventas:
- Observa el efecto
- Añade privilegios de consulta
- Comprueba si el permiso vuelve a estar disponible.

Privilegios sobre esquemas

- Crea un nuevo esquema llamado pruebas:
- Concede a `ventas_grupo` permiso para usar el esquema pero no para crear objetos
- Intenta crear una tabla dentro del esquema con `empleado_ventas` y verifica el resultado.
- Da permisos para crear
- Vuelve a probar y observa la diferencia.

Auditoría final de roles y privilegios

- Consulta todos los privilegios otorgados a cada usuario:

Actividad 5 – Seguridad y cumplimiento

1. **Configuración de políticas de seguridad**
    - Activa el registro (`logging_collector = on`) y revisa el archivo `postgresql.conf`.
    - Cambia la política de autenticación de `md5` a `scram-sha-256` en `pg_hba.conf`.
2. **Auditoría de accesos**

- Conéctate con distintos usuarios y revisa los logs.
- Identifica intentos fallidos de conexión. Explica que ves

3. **Comprobación de seguridad**
    - Verifica los privilegios de cada usuario (`\du` y `\z`).
    - Elabora un informe final indicando qué medidas garantizan la seguridad.