

Examen Parcial Práctico: Administración y Seguridad en MySQL

Contexto del Caso Práctico: "Proyecto ERP NovaTech"

La empresa NovaTech está implementando un nuevo sistema ERP (Planificación de Recursos Empresariales). Como futuros Ingenieros de Sistemas y Administradores de Bases de Datos (DBA), han sido contratados para diseñar e implementar la arquitectura de seguridad, usuarios y roles del motor de base de datos MySQL. Se requiere aplicar el principio del "menor privilegio", garantizando que cada empleado tenga acceso estricto a lo que necesita.

Tareas a Desarrollar:

Parte 1: Preparación del Entorno y Administrador Principal

1. Crea la base de datos central llamada *erp_novatech*.
2. El director del proyecto necesita una cuenta de superadministrador. Crea el usuario *director_dba* que solo pueda conectarse desde el servidor local (localhost) con la contraseña *AdminNova2026**.
3. Otórgale a *director_dba* todos los privilegios sobre todo el servidor y dale la capacidad de delegar o transferir permisos a otros usuarios.
4. Aplica los cambios.

Parte 2: Estandarización mediante Roles Para facilitar la administración, NovaTech gestiona sus permisos mediante roles.

5. Crea tres roles: *rol_auditor*, *rol_desarrollador* y *rol_ventas*.
6. Asigna los siguientes privilegios a cada rol sobre la base de datos *erp_novatech*:
 - **rol_auditor:** Solo lectura de datos y permiso para ver la definición de las vistas (SHOW VIEW).
 - **rol_desarrollador:** Operaciones DML completas (seleccionar, insertar, actualizar, eliminar), además de permisos para crear tablas temporales y modificar la estructura de las tablas (ALTER).
 - **rol_ventas:** Solo debe poder insertar y consultar datos.

Parte 3: Creación de Usuarios y Asignación Llegan nuevos empleados a la empresa que necesitan accesos específicos:

7. Crea al usuario `carlos_dev` para que trabaje desde cualquier dirección IP remota (%), con la contraseña `Dev#2026`.
 - Comprueba conectándolo a MySQL Workbench
8. Crea a la usuaria `ana_ventas`, que trabaja desde su casa con una IP fija (La IP de tu WSL - Ubuntu, ejemplo: 192.168.1.50), con la contraseña `Ventas#2026`.
9. Asigna el `rol_desarrollador` a *Carlos* y el `rol_ventas` a *Ana*.
10. ¡Muy importante! Configura el sistema para que estos roles se activen automáticamente cuando *Carlos* y *Ana* inicien sesión por defecto.

Parte 4: Seguridad Granular (Nivel de Columna) y Auditoría

11. La empresa tiene una tabla llamada `empleados` dentro de `erp_novatech`. El equipo de Recursos Humanos requiere un usuario llamado `rrhh_user` (conexión local, contraseña `Rh2026*`).
12. Por políticas de seguridad, `rrhh_user` **solo** puede actualizar (UPDATE) las columnas `salario` y `cargo` de la tabla `empleados`, y no debe poder modificar ninguna otra columna.
13. Escribe la consulta para auditar y verificar qué usuarios existen actualmente en el sistema y desde qué hosts pueden conectarse.
14. Escribe el comando para comprobar los privilegios y roles combinados del usuario `carlos_dev`.

Parte 5: Lenguaje de Definición de Datos (DDL) - Estructura Los privilegios DDL permiten modificar la estructura de la base de datos.

15. Inicia sesión en MySQL con el usuario `carlos_dev` (quien posee el `rol_desarrollador`).
16. Utilizando los permisos DDL de tu rol, crea 4 tablas relacionadas para el ERP: *departamentos*, *empleados*, *clientes* y *ventas*. Ingresarle unos 5 registros en cada una de las tablas.
17. El director del proyecto solicita que los clientes tengan un correo de contacto. Como `carlos_dev`, utiliza la sentencia que permite modificar la estructura de tablas existentes para agregar la columna `email` a la tabla `clientes`.

Parte 6: Lenguaje de Manipulación de Datos (DML) - Inserción y Borrado Los privilegios DML controlan las operaciones sobre los datos existentes.

18. Inicia sesión con la usuaria *ana_ventas*.
19. Aprovechando el privilegio que permite agregar nuevas filas a una tabla, inserta 2 registros en la tabla clientes y 2 registros en la tabla ventas.
20. *ana_ventas* se da cuenta de que ingresó mal una venta e intenta borrarla usando el comando para eliminar filas de una tabla. ¿Qué sucede y por qué? Documenta el error.
21. Para solucionar el problema de Ana, inicia sesión con *carlos_dev* y ejecuta la eliminación del registro incorrecto en la tabla ventas.

Parte 7: Actualización Granular de Registros (UPDATE)

22. Inicia sesión de manera local con el usuario *rrhh_user*.
23. *carlos_dev* ha insertado previamente un registro en la tabla empleados. Como *rrhh_user*, utiliza el comando que permite modificar datos existentes para cambiar el salario y el cargo de ese empleado.
24. Intenta modificar el nombre de ese mismo empleado. Documenta por qué el sistema bloquea esta acción, basándote en la configuración de nivel de granularidad aplicada en la primera parte del taller.