

PARTE 1: CREACIÓN DE ROLES Y ASIGNACIÓN DE PRIVILEGIOS

Escribe un script SQL que cree los cinco usuarios Asegúrate de agregar comentarios que expliquen qué puede hacer cada usuario. Usa contraseñas seguras y personalízalas si es necesario en la creación de usuarios

```
2  -- CREACIÓN DE ROLES
3  -- Super Administrador: Puede crear y eliminar bases de datos
4  • CREATE ROLE 'SuperAdmin';
5  • GRANT CREATE, DROP ON *.* TO 'SuperAdmin';
6
7  -- Administrador: Puede crear usuarios y procesos
8  • CREATE ROLE 'Administrador';
9  • GRANT CREATE USER, PROCESS ON *.* TO 'Administrador';
10
11 -- CRUD: Permite insertar, actualizar y eliminar datos
12 • CREATE ROLE 'CRUD';
13 • GRANT INSERT, UPDATE, DELETE ON *.* TO 'CRUD';
14
15 -- CRU: Permite insertar y actualizar, pero sin eliminar
16 • CREATE ROLE 'CRU';
17 • GRANT INSERT, UPDATE ON *.* TO 'CRU';
18
19 -- Solo Lectura: Permite realizar consultas a las tablas
20 • CREATE ROLE 'SoloLectura';
21 • GRANT SELECT ON *.* TO 'SoloLectura';
```

```
23 -- CREACIÓN DE USUARIOS
24 -- Usuario Super Administrador
25 • CREATE USER 'superadmin'@'localhost' IDENTIFIED BY 'ContraseñaSeguraSuperAdmin!';
26 • GRANT 'SuperAdmin' TO 'superadmin'@'localhost';
27
28 -- Usuario Administrador
29 • CREATE USER 'admin'@'localhost' IDENTIFIED BY 'ContraseñaSeguraAdmin!';
30 • GRANT 'Administrador' TO 'admin'@'localhost';
31
32 -- Usuario CRUD
33 • CREATE USER 'cruduser'@'localhost' IDENTIFIED BY 'ContraseñaSeguraCRUD!';
34 • GRANT 'CRUD' TO 'cruduser'@'localhost';
35
36 -- Usuario CRU
37 • CREATE USER 'cruuser'@'localhost' IDENTIFIED BY 'ContraseñaSeguraCRU!';
38 • GRANT 'CRU' TO 'cruuser'@'localhost';
39
40 -- Usuario Solo Lectura
41 • CREATE USER 'readuser'@'localhost' IDENTIFIED BY 'ContraseñaSeguraRead!';
42 • GRANT 'SoloLectura' TO 'readuser'@'localhost';
```

```
46 • SHOW GRANTS FOR 'admin'@'localhost';
47
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Grants for admin@localhost
▶ GRANT USAGE ON *.* TO 'admin'@'localhost'
GRANT 'Administrador'@'%' TO 'admin'@'l...

```
44 -- Verificación de privilegios de los usuarios:
45 -- Usar el siguiente comando para verificar los privilegios
46 • SHOW GRANTS FOR 'cruduser'@'localhost';
47
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Grants for cruduser@localhost
▶ GRANT USAGE ON *.* TO 'cruduser'@'localho...
GRANT 'CRUD'@'%' TO 'cruduser'@'localh...

```
44 -- Verificación de privilegios de los usuarios:
45 -- Usar el siguiente comando para verificar los privilegios
46 • SHOW GRANTS FOR 'cruuser'@'localhost';
47
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Grants for cruduser@localhost
▶ GRANT USAGE ON *.* TO 'cruduser'@'localho...
GRANT 'CRUD'@'%' TO 'cruduser'@'localh...

```

44 -- Verificación de privilegios de los usuarios:
45 -- Usar el siguiente comando para verificar los privilegios
46 • SHOW GRANTS FOR 'readuser'@'localhost';
47

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Grants for readuser@localhost			
GRANT USAGE ON *.* TO 'readuser'@'localh...			
GRANT 'SoloLectura'@'%' TO 'readuser'@'...			

PARTE 2 TRIGGERS

• Investigación sobre Triggers:

- **Definición:** Un trigger es un conjunto de instrucciones SQL que se ejecutan automáticamente en respuesta a ciertos eventos en una tabla o vista.
- **Tipos de Triggers:**
 - **BEFORE:** Se ejecuta antes de la acción (INSERT, UPDATE, DELETE).
 - **AFTER:** Se ejecuta después de la acción.
 - **INSTEAD OF:** Reemplaza la acción que se habría ejecutado.
- **Eventos que activan un Trigger:**
 - **INSERT:** Cuando se inserta un nuevo registro.
 - **UPDATE:** Cuando se actualiza un registro existente.
 - **DELETE:** Cuando se elimina un registro.
- **Contexto de los triggers:**
 - **NEW:** Para referirse a los valores nuevos que van a insertarse o actualizarse.
 - **OLD:** Para referirse a los valores antiguos antes de la modificación o eliminación.

• Importancia de los Triggers:

- **Beneficios:**
 - Automatización de tareas.
 - Mantenimiento de la integridad referencial.
 - Control de cambios y auditoría.
- **Desventajas:**
 - Pueden generar sobrecarga en el rendimiento.
 - Complejidad en el manejo de errores.

• Aplicaciones comunes de Triggers:

- **Auditoría:** Registrar cambios en los datos.
- **Reglas de negocio:** Asegurar que se apliquen ciertas reglas de negocio automáticamente (por ejemplo, validación de datos).
- **Control de integridad referencial:** Mantener la coherencia entre tablas relacionadas.

• Trigger para registrar operaciones en la tabla de empleados:

Creación de las tablas:

```

50 • CREATE DATABASE TriggerSS;
51 • USE TriggerSS;
52
53 -- Tabla de empleados
54 • CREATE TABLE Empleados (
55     EmpID INT AUTO_INCREMENT PRIMARY KEY,
56     Nombre VARCHAR(100),
57     Departamento VARCHAR(100),
58     Salario DECIMAL(10,2)
59 );
60
61 -- Tabla de auditoría
62 • CREATE TABLE Auditoria (
63     AudID INT AUTO_INCREMENT PRIMARY KEY,
64     Accion VARCHAR(10), -- INSERT, UPDATE, DELETE
65     EmpID INT,
66     Nombre VARCHAR(100),
67     Departamento VARCHAR(100),
68     Salario DECIMAL(10,2),
69     Fecha DATETIME
70 );

```

Crear el trigger:

El trigger debe activarse después de realizar cualquier operación (INSERT, UPDATE o DELETE) sobre la tabla de empleados. El trigger debe insertar un nuevo registro en la tabla de auditoría cada vez que se realice una de estas operaciones

```

73 #1. Trigger para INSERT:
74 DELIMITER $$
75 • CREATE TRIGGER AuditoriaInsert
76 AFTER INSERT ON Empleados
77 FOR EACH ROW
78 BEGIN
79     INSERT INTO Auditoria (Accion, EmpID, Nombre, Departamento, Salario, Fecha)
80     VALUES ('INSERT', NEW.EmpID, NEW.Nombre, NEW.Departamento, NEW.Salario, NOW());
81 END$$
82 DELIMITER ;

```

```

84 #2. Trigger para DELETE:
85 DELIMITER $$
86
87 • CREATE TRIGGER AuditoriaDelete
88 AFTER DELETE ON Empleados
89 FOR EACH ROW
90 BEGIN
91     INSERT INTO Auditoria (Accion, EmpID, Nombre, Departamento, Salario, Fecha)
92     VALUES ('DELETE', OLD.EmpID, OLD.Nombre, OLD.Departamento, OLD.Salario, NOW());
93 END$$
94
95 DELIMITER ;

```

```

97 #3. Trigger para UPDATE:
98 DELIMITER $$
99
100 • CREATE TRIGGER AuditoriaUpdate
101 AFTER UPDATE ON Empleados
102 FOR EACH ROW
103 BEGIN
104     INSERT INTO Auditoria (Accion, EmpID, Nombre, Departamento, Salario, Fecha)
105     VALUES ('UPDATE', OLD.EmpID, OLD.Nombre, OLD.Departamento, OLD.Salario, NOW());
106 END$$
107
108 DELIMITER ;

```

```
111 • SELECT * FROM Auditoria;
```

112

413

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

AudID	Accion	EmpID	Nombre	Departamento	Salario	Fecha
1	1	1	1	1	1	1