

Cómo usar MySQL para auditoría y control de accesos

Cuando se habla de bases de datos, mucha gente piensa únicamente en almacenar información.

Pero en entornos reales, MySQL también puede utilizarse para:

- auditoría
- trazabilidad
- control de accesos
- monitorización
- detección de comportamientos sospechosos

Y lo interesante es que se puede enseñar todo esto con ejemplos muy sencillos.

1██ Levantar MySQL con Docker

```
docker run --name mysql-demo \  
-e MYSQL_ROOT_PASSWORD=root \  
-p 3306:3306 \  
-d mysql:8
```

2██ Crear una base de datos

de ejemplo

Vamos a simular una empresa con empleados y salarios.

```
DROP DATABASE IF EXISTS demo_seguridad;  
CREATE DATABASE demo_seguridad;  
USE demo_seguridad;
```

3██ Crear tabla de empleados

```
CREATE TABLE empleados (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(50),  
puesto VARCHAR(50),  
salario INT  
);
```

Insertamos algunos datos:

```
INSERT INTO empleados(nombre,puesto,salario) VALUES  
('Ana','Analista',32000),  
('Carlos','Administrador Sistemas',38000),  
('Marta','Marketing',42000),  
('Luis','Director General',150000);
```

4██ Registrar accesos sensibles

Ahora creamos una tabla de auditoría.

```
CREATE TABLE log_accesos_salarios (  
id INT AUTO_INCREMENT PRIMARY KEY,  
usuario_mysql VARCHAR(100),
```

```
fecha DATETIME,  
evento VARCHAR(200)  
);
```

Aquí se almacenarán intentos de acceso sensibles.

5.2 Crear un procedimiento con control de acceso

Este es el punto interesante.

```
DELIMITER $$
```

```
CREATE PROCEDURE ver_salario(IN nombre_empleado VARCHAR(50))  
BEGIN
```

```
IF nombre_empleado = 'Luis' THEN
```

```
INSERT INTO log_accesos_salarios(usuario_mysql, fecha, evento)  
VALUES(  
CURRENT_USER(),  
NOW(),  
'Intento de consulta del salario del Director General'  
);
```

```
SIGNAL SQLSTATE '45000'  
SET MESSAGE_TEXT='ACCESO DENEGADO: salario protegido';
```

```
ELSE
```

```
SELECT nombre, puesto, salario  
FROM empleados  
WHERE nombre = nombre_empleado;
```

```
END IF;
```

```
END$$
```

DELIMITER ;

6██ Probar el sistema

Consulta permitida:

```
CALL ver_salario('Ana');
```

Consulta bloqueada:

```
CALL ver_salario('Luis');
```

Ver eventos registrados:

```
SELECT * FROM log_accesos_salarios;
```

██ ¿Qué se aprende realmente con esto?

Aunque el ejemplo es pequeño, aquí aparecen conceptos reales de seguridad:

- ██ control de acceso
- ██ auditoría
- ██ trazabilidad
- ██ protección de información sensible
- ██ registros de eventos
- ██ alertas ante accesos indebidos

Y además el alumnado entiende algo importante:

La seguridad no solo está en el firewall o el antivirus. También debe existir dentro de la propia base de datos.

Ese es precisamente el tipo de cosas que en muchos cursos técnicos casi nunca se enseñan.