



FACULTAD DE CIENCIAS DE LA VIDA Y TECNOLOGÍAS

CARRERA INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

TEMA:

GUÍA 3: AUTOMATIZACIÓN CON TRIGGERS Y AUDITORÍA

AUTORES:

Juan Marco Proaño Chele

Diego Miguel Arteaga Saltos

DIRECTOR DE TEMA:

Ing. Edison Almeida Zambrano, Mg.

MANTA – MANABÍ – ECUADOR

2025 – 2026

CONTENIDO

1.	Introducción	3
1.1	Importancia de la automatización	3
2.	Creación de la Tabla de Auditoría	3
2.1	Propósito de la auditoría.....	3
2.2	Definición de la tabla	4
3.	Creación del Trigger.....	5
3.1	Definición del trigger	5
3.2	Implementación en SQL.....	5
4.	Prueba del Trigger	6
4.1	Ejemplo de actualización	6
4.2	Verificación en la auditoría	6
5.	Conclusión de la guía:	7

1. Introducción

Esta guía tiene como propósito enseñar el uso de triggers en bases de datos como una herramienta para automatizar procesos y asegurar el control de la información. A través de un ejemplo práctico, se muestra cómo implementar un sistema de auditoría de sueldos, en el que cada modificación realizada por un profesor queda registrada automáticamente en una tabla de historial. De esta manera, se garantiza un seguimiento confiable de cambios sensibles, como sueldos o calificaciones, sin necesidad de depender de registros manuales.

1.1 Importancia de la automatización

El uso de triggers es fundamental en entornos donde se requiere mantener un historial de cambios o garantizar la integridad de la información sin depender de que los usuarios realicen registros manuales. Un ejemplo típico es la auditoría de cambios en sueldos, calificaciones o datos financieros.

2. Creación de la Tabla de Auditoría

2.1 Propósito de la auditoría

La auditoría permite llevar un historial de modificaciones sobre campos sensibles, en este caso los sueldos de los profesores. Así se puede rastrear quién, cuándo y cómo se produjo un cambio.

2.2 Definición de la tabla

Se crea una tabla específica para registrar los cambios en los sueldos de los profesores:

```
CREATE TABLE auditoria_sueldos (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  profesor VARCHAR(50),  
  sueldo_anterior DECIMAL(4,2),  
  sueldo_nuevo DECIMAL(4,2),  
  fecha_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Ilustración 1. Resultado a la definición de las tablas.



Fuente. Elaborado por los autores de la guía.

3. Creación del Trigger

3.1 Definición del trigger

El trigger se configura para ejecutarse **antes de cada actualización** sobre el campo sueldo_final de la tabla profesores.

3.2 Implementación en SQL

```
DELIMITER $$
```

```
CREATE TRIGGER trg_actualiza_sueldo
```

```
BEFORE UPDATE ON profesores
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF NEW.sueldo_final <> OLD.sueldo_final THEN
```

```
INSERT INTO auditoria_sueldos (profesor, sueldo_anterior, sueldo_nuevo)
```

```
VALUES (OLD.nombre, OLD.sueldo_final, NEW.sueldo_final);
```

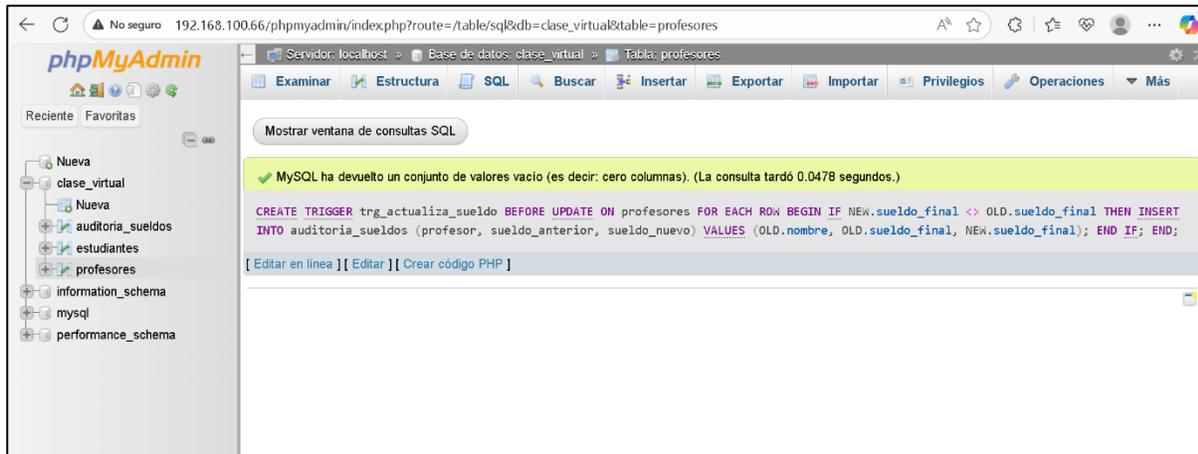
```
END IF;
```

```
END$$
```

```
DELIMITER ;
```

El trigger compara el valor antiguo del sueldo con el nuevo (OLD.sueldo_final vs NEW.sueldo_final). Si existe una diferencia, se registra automáticamente en la tabla de auditoría junto con la fecha y hora del cambio.

Ilustración 2. Implementación del trigger en SQL.



Fuente. Elaborado por los autores de la guía.

4. Prueba del Trigger

4.1 Ejemplo de actualización

Cuando actualizamos el sueldo de un profesor, automáticamente se genera un registro en la tabla de auditoría:

```
UPDATE profesores SET sueldo_final = 10 WHERE nombre = 'Luis Mendoza';
```

4.2 Verificación en la auditoría

Para comprobar que el trigger funciona correctamente:

```
SELECT * FROM auditoria_sueldos;
```

5. Conclusión de la guía:

Has implementado un **sistema automático de auditoría**. Esto es vital en entornos reales porque permite **rastrear cambios sensibles** (como sueldos, calificaciones o datos financieros) sin depender de que un usuario los registre manualmente.