

Implementación de Triggers en MariaDB

Actividades realizadas

1. Creación de tablas

Se crearon dos tablas en la base de datos:

- **Tabla `estudiantes`**: Para registrar información básica de los estudiantes.
- **Tabla `auditoria`**: Para registrar las acciones realizadas en la tabla `estudiantes`.

Código SQL utilizado:

-- Crear tabla estudiantes

```
CREATE TABLE estudiantes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    edad INT NOT NULL,  
    carrera VARCHAR(50)  
);
```

-- Crear tabla auditoria

```
CREATE TABLE auditoria (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    accion VARCHAR(50),  
    fecha DATETIME,  
    usuario VARCHAR(50)  
);
```

2. Creación del trigger

Se implementó un trigger llamado `after_insert_estudiantes`. Este trigger se activa automáticamente después de que se inserta un registro en la tabla `estudiantes`, registrando la acción en la tabla `auditoria`.

Código SQL utilizado:

```
DELIMITER //

CREATE TRIGGER after_insert_estudiantes

AFTER INSERT ON estudiantes

FOR EACH ROW

BEGIN

    INSERT INTO auditoria (accion, fecha, usuario)

    VALUES ('INSERT en estudiantes', NOW(), 'admin');

END;

//

DELIMITER ;
```

3. Pruebas realizadas

Para verificar el correcto funcionamiento del trigger, se insertaron registros en la tabla `estudiantes` y se revisaron los resultados en la tabla `auditoria`.

Prueba 1: Insertar un registro en la tabla `estudiantes`

```
INSERT INTO estudiantes (nombre, edad, carrera)

VALUES ('Juan Pérez', 20, 'Ingeniería');
```

Resultado esperado en la tabla `auditoria`:

```
SELECT * FROM auditoria;
```

Contenido de la tabla `auditoria`:

| id | accion | fecha | usuario |
|----|-----------------------|---------------------|---------|
| 1 | INSERT en estudiantes | 2025-06-09 20:30:00 | admin |

4. Observaciones

- El trigger funcionó correctamente, registrando automáticamente las acciones realizadas en la tabla **estudiantes**.
- La tabla **auditoria** permite llevar un registro de auditoría básico, útil para monitorear cambios en los datos.

Preguntas respondidas

Pregunta 1: ¿Qué utilidad tiene un trigger (ventajas)?

- Automatización de tareas repetitivas.
- Garantizar la integridad de los datos.
- Implementar reglas de negocio directamente en la base de datos.
- Realizar auditorías y seguimiento de cambios.
- Reducir errores humanos en la manipulación de datos.

Pregunta 2: ¿Tipos de triggers?

- **Por evento:**
 - **INSERT:** Se activa al insertar un registro.
 - **UPDATE:** Se activa al actualizar un registro.
 - **DELETE:** Se activa al eliminar un registro.
- **Por momento:**
 - **BEFORE:** Se ejecuta antes del evento.
 - **AFTER:** Se ejecuta después del evento.

Pregunta 3: ¿En qué casos NO son de utilidad?

- Cuando los triggers son demasiado complejos y difíciles de mantener.
 - Si afectan negativamente el rendimiento de la base de datos.
 - Cuando la lógica del negocio cambia constantemente (es mejor manejarla en la aplicación).
 - Si los desarrolladores no están al tanto de los triggers, pueden generar comportamientos inesperados.
 - En sistemas que requieren control explícito sobre las operaciones.
-