

1. **¿Para qué sirven los “disparadores” en SQL?**

**R/** - Realizar comprobaciones de valores que se insertarán en una tabla.  
- Realizar cálculos sobre valores implicados en una actualización  
- Invocar un procedimiento almacenado usando la instrucción CALL  
IMPORTANTE → El desencadenante no puede utilizar la sentencia CALL para invocar procedimientos almacenados que devuelven datos al cliente o que utilizan SQL dinámico. (Los procedimientos almacenados pueden devolver datos al disparador a través de los parámetros OUT o INOUT).

2. **¿Cuál es la ventaja de usar disparadores?**

**R/** - Ahorra tiempos de ejecución y automatiza procedimientos en la base de datos.

3. **¿Cómo puedo definir el tiempo de la acción en un disparador?**

**R/** Es el momento en el que las sentencias SQL que se encuentran en el disparador se ejecutara, siendo estas: La palabra clave BEFORE indica el tiempo de acción del disparador. En este caso, el disparador se activa antes de cada fila insertada en la tabla. La otra palabra clave permitida aquí es AFTER.

4. **¿Cómo puedo definir el evento que dispara la acción del disparador?**

**R/** Es la acción a ejecutar sobre una tabla de la base de datos, se dividen en: La palabra clave INSERT indica el evento desencadenante; es decir, el tipo de operación que activa el gatillo. También puede crear activadores para las operaciones DELETE y UPDATE.

5. **¿Qué componentes pueden ir en el cuerpo de un disparador?**

**R/** - Las palabras clave OLD y NEW le permiten acceder a las columnas en las filas afectadas por un disparador.  
- INSERT, DELETE, UPDATE.  
- IF, ELSE, ELSE IF.  
- LOOP, SET.

6. **¿Cómo puedo hacer referencia a los elementos o distinguir los elementos de las columnas a afectar, dentro de un disparador?**

**R/** OLD y NEW le permiten acceder a las columnas en las filas afectadas por un disparador. OLD y NEW son extensiones de MySQL para disparadores; no

distinguen entre mayúsculas y minúsculas. En un disparador INSERT, solo se puede usar NEW.col\_name; no hay fila antigua. En un disparador DELETE, solo se puede usar OLD.col\_name; no hay nueva fila. En un desencadenador UPDATE, puede usar OLD.col\_name para hacer referencia a las columnas de una fila antes de que se actualice y NEW.col\_name para hacer referencia a las columnas de la fila después de que se actualice. Una columna denominada OLD es de solo lectura. Puede consultarlo (si tiene el privilegio SELECT), pero no modificarlo.

**7. ¿Cuáles son las limitaciones de código que se pueden implementar en el cuerpo de un trigger de acuerdo con la lectura del manual de MySQL en compatibilidad con MariaDB? R/**

- El desencadenante no puede utilizar la sentencia CALL para invocar procedimientos almacenados que devuelven datos al cliente o que utilizan SQL dinámico. (Los procedimientos almacenados pueden devolver datos al disparador a través de los parámetros OUT o INOUT).

- El desencadenante no puede utilizar declaraciones que inicien o finalicen una transacción de forma explícita o implícita, como START TRANSACTION, COMMIT o ROLLBACK. (ROLLBACK a SAVEPOINT está permitido porque no finaliza una transacción).

**8. ¿Cómo un trigger realiza el manejo de errores? R/**

- Si falla un disparador ANTES, no se realiza la operación en la fila correspondiente.

- Un disparador ANTES se activa al intentar insertar o modificar la fila, independientemente de si el intento tiene éxito posteriormente.

- Un disparador DESPUÉS se ejecuta solo si se dispara ANTES y la operación de fila se ejecuta correctamente.

- Un error durante un activador ANTES o DESPUÉS da como resultado el fallo de toda la declaración que provocó la invocación del activador.

- En el caso de tablas transaccionales, la falla de una declaración debería provocar la reversión de todos los cambios realizados por la declaración. La falla de un disparador hace que la declaración falle, por lo que la falla del disparador también causa una reversión. Para las tablas no transaccionales, dicha reversión no se puede realizar, por lo que, aunque la instrucción falla, cualquier cambio realizado antes del punto del error permanece en vigor.

**9. ¿Cómo puedo visualizar los triggers creados en el SGBD?**

**R/** SHOW TRIGGERS [{FROM | IN} db\_name] | [LIKE 'pattern' | WHERE expr]