



```
CREATE DATABASE IF NOT EXISTS bd_triggers;
```

```
USE bd_triggers;
```

```
-- Tabla de usuarios
```

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    rol VARCHAR(20)  
);
```

```
-- Tabla de logs para registrar inserciones en usuarios
```

```
CREATE TABLE logs (  
    id_log INT AUTO_INCREMENT PRIMARY KEY,  
    id_usuario INT,  
    fecha DATETIME  
);
```

```
-- Tabla de cuentas
```

```
CREATE TABLE cuentas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100),  
    rol VARCHAR(20)
```

```
id_cuenta INT AUTO_INCREMENT PRIMARY KEY,  
saldo DECIMAL(10,2)  
);
```

-- Tabla de productos

```
CREATE TABLE productos (  
id_producto INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100),  
precio DECIMAL(10,2)  
);
```

-- Tabla de auditoría de eliminaciones de productos

```
CREATE TABLE auditoria (  
id_auditoria INT AUTO_INCREMENT PRIMARY KEY,  
id_producto INT,  
nombre VARCHAR(100),  
fecha DATETIME  
);
```

-- Tabla de auditoría de precios

```
CREATE TABLE auditoria_precios (  
id_auditoria INT AUTO_INCREMENT PRIMARY KEY,
```

```
id_producto INT,  
precio_antiguo DECIMAL(10,2),  
precio_nuevo DECIMAL(10,2),  
fecha DATETIME  
);
```

-- Tabla de pedidos

```
CREATE TABLE pedidos (  
    id_pedido INT AUTO_INCREMENT PRIMARY KEY,  
    fecha_vencimiento DATE  
);
```

-- Tabla de inventario

```
CREATE TABLE inventario (  
    id_producto INT PRIMARY KEY,  
    cantidad INT  
);
```

-- Tabla de ventas

```
CREATE TABLE ventas (  
    id_venta INT AUTO_INCREMENT PRIMARY KEY,  
    id_producto INT,
```

```
    cantidad_vendida INT
);

-- Tabla de clientes
CREATE TABLE clientes (
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    telefono VARCHAR(20)
);

-- Historial de cambios de teléfono
CREATE TABLE historial_clientes (
    id_historial INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    telefono_anterior VARCHAR(20),
    telefono_nuevo VARCHAR(20),
    fecha DATETIME
);

-- Tabla de empleados
CREATE TABLE empleados (
    id_empleado INT AUTO_INCREMENT PRIMARY KEY,
```

nombre VARCHAR(100),
edad INT
);

Problemas

1 Creación de un Trigger:

Crea un *trigger* que registre en una tabla `logs` cada vez que se inserta un nuevo usuario en la tabla `usuarios`. El log debe incluir el `id` del usuario y la fecha de inserción.

2 Trigger para Evitar Valores Negativos:

Crea un *trigger* que evite que los valores negativos sean insertados en la columna `saldo` de la tabla `cuentas`. Si se intenta insertar un valor negativo, debe cambiarse a 0.

3 Registro de Eliminaciones:

Crea un *trigger* que almacene en una tabla `auditoria` los datos de cada registro eliminado de la tabla `productos`. Debe registrar `id_producto`, `nombre` y la fecha de eliminación.

4 Control de Modificaciones en Precio:

Crea un *trigger* que verifique si el precio de un producto en la tabla `productos` aumenta en más del 50% en una actualización. Si es así, que registre el cambio en la tabla `auditoria_precios`.

5 Restricción en Actualización de Fechas:

Crea un *trigger* que evite que la fecha de vencimiento (`fecha_vencimiento`) de un pedido en la tabla `pedidos` sea modificada a una fecha anterior a la actual.

6 Mantenimiento de Stock:

Crea un *trigger* que, cada vez que se realiza una venta en la tabla `ventas`, disminuya automáticamente la cantidad disponible del producto en la tabla `inventario`.

7 Prevención de Eliminación de Usuarios Administradores:

Crea un *trigger* que impida eliminar usuarios con rol "admin" de la tabla `usuarios`.

8 Registro de Cambios en Teléfono de Clientes:

Crea un *trigger* que almacene en la tabla `historial_clientes` el `id_cliente`, el

teléfono anterior y el nuevo teléfono cada vez que se actualiza este dato en la tabla `clientes`.

9 Validación de Edad Mínima:

Crea un *trigger* que evite que se inserten registros en la tabla `empleados` si la edad es menor a 18 años.

10 Conversión de Texto a Mayúsculas:

Crea un *trigger* que convierta automáticamente a mayúsculas los valores ingresados en la columna `nombre` de la tabla `clientes`.

Respuestas

1 Trigger para registrar inserciones en `logs`

```
CREATE TRIGGER tr_insert_usuario
AFTER INSERT ON usuarios
FOR EACH ROW
BEGIN
    INSERT INTO logs (id_usuario, fecha) VALUES (NEW.id, NOW());
END;
```

2 Trigger para evitar valores negativos en `saldo`

```
CREATE TRIGGER tr_evitar_saldo_negativo
BEFORE INSERT ON cuentas
FOR EACH ROW
BEGIN
    IF NEW.saldo < 0 THEN
        SET NEW.saldo = 0;
    END IF;
END;
```

3 Trigger para registrar eliminaciones en `auditoria`

```
CREATE TRIGGER tr_eliminar_producto
AFTER DELETE ON productos
FOR EACH ROW
BEGIN
    INSERT INTO auditoria (id_producto, nombre, fecha)
    VALUES (OLD.id_producto, OLD.nombre, NOW());
END;
```

4 Trigger para control de precios en auditoria_precios

```
CREATE TRIGGER tr_control_precio
BEFORE UPDATE ON productos
FOR EACH ROW
BEGIN
  IF NEW.precio > OLD.precio * 1.5 THEN
    INSERT INTO auditoria_precios (id_producto, precio_antiguo, precio_nuevo, fecha)
    VALUES (OLD.id_producto, OLD.precio, NEW.precio, NOW());
  END IF;
END;
```

5 Trigger para evitar modificación de fecha de vencimiento a una fecha anterior

```
CREATE TRIGGER tr_restriccion_fecha
BEFORE UPDATE ON pedidos
FOR EACH ROW
BEGIN
  IF NEW.fecha_vencimiento < CURDATE() THEN
    SET NEW.fecha_vencimiento = OLD.fecha_vencimiento;
  END IF;
END;
```

6 Trigger para disminuir stock en inventario

```
CREATE TRIGGER tr_actualizar_stock
AFTER INSERT ON ventas
FOR EACH ROW
BEGIN
  UPDATE inventario
  SET cantidad = cantidad - NEW.cantidad_vendida
  WHERE id_producto = NEW.id_producto;
END;
```

7 Trigger para evitar eliminación de administradores

```
CREATE TRIGGER tr_prevenir Eliminacion_admin
BEFORE DELETE ON usuarios
FOR EACH ROW
BEGIN
  IF OLD.rol = 'admin' THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede eliminar un
administrador';
  END IF;
END;
```

8 Trigger para registrar cambios de teléfono en **historial_clientes**

```
CREATE TRIGGER tr_historial_telefono
AFTER UPDATE ON clientes
FOR EACH ROW
BEGIN
  IF OLD.telefono <> NEW.telefono THEN
    INSERT INTO historial_clientes (id_cliente, telefono_anterior, telefono_nuevo, fecha)
    VALUES (OLD.id_cliente, OLD.telefono, NEW.telefono, NOW());
  END IF;
END;
```

9 Trigger para evitar empleados menores de 18 años

```
CREATE TRIGGER tr_validar_edad
BEFORE INSERT ON empleados
FOR EACH ROW
BEGIN
  IF NEW.edad < 18 THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se pueden registrar empleados
menores de 18 años';
  END IF;
END;
```

10 Trigger para convertir nombres a mayúsculas

```
CREATE TRIGGER tr_nombre_mayusculas
BEFORE INSERT ON clientes
FOR EACH ROW
BEGIN
  SET NEW.nombre = UPPER(NEW.nombre);
END;
```

Estos problemas y respuestas te servirán para reforzar el tema de *Triggers* con ejemplos prácticos en clases. 🚀