

Trabajo Práctico: Triggers en MySQL

**Ejercicio 1: Crear un Trigger para Auditar Inserciones en la Tabla Clientes.
Implementar un trigger que registre cada vez que se inserta un nuevo cliente en una tabla de auditoría llamada AuditoriaClientes.**

```
CREATE TABLE AuditoriaClientes (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ClienteID INT,  
    Nombre VARCHAR(50),  
    Apellido VARCHAR(50),  
    Email VARCHAR(100),  
    Telefono VARCHAR(15),  
    FechaInsercion TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- atencion  
    Tipo_Operacion VARCHAR(20)  
);
```

```
CREATE TRIGGER auditarInsercionCliente  
AFTER INSERT ON Clientes  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaClientes (ClienteID, Nombre, Apellido, Email, Telefono,  
    FechaInsercion, Tipo_Operacion)  
    VALUES (NEW.ClienteID, NEW.Nombre, NEW.Apellido, NEW.Email, NEW.Telefono,  
    NOW(), 'INSERT'); -- acá dejamos registrado el evento en la auditoria  
END
```

**Ejercicio 2: Crear un Trigger para Auditar Actualizaciones en la Tabla Clientes.
Implementar un trigger que registre cada vez que se actualiza un cliente en la tabla Clientes.**

```
CREATE TABLE AuditoriaActualizacionClientes (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ClienteID INT,  
    NombreAntiguo VARCHAR(50),  
    NombreNuevo VARCHAR(50),  
    FechaActualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    Tipo_Operacion VARCHAR(20)  
);
```

```
CREATE TRIGGER auditarActualizacionCliente  
AFTER UPDATE ON Clientes  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaActualizacionClientes (ClienteID, NombreAntiguo,  
    NombreNuevo, Tipo_Operacion)  
    VALUES (OLD.ClienteID, OLD.Nombre, NEW.Nombre, 'UPDATE');  
END
```

**Ejercicio 3: Crear un Trigger para Auditar Eliminaciones en la Tabla Clientes
Implementar un trigger que registre cada vez que se elimina un cliente de la
tabla Clientes.**

```
CREATE TABLE AuditoriaEliminacionClientes (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ClienteID INT,  
    Nombre VARCHAR(50),  
    Apellido VARCHAR(50),  
    FechaEliminacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    Tipo_Operacion VARCHAR(20)  
);
```

```
CREATE TRIGGER auditarEliminacionCliente  
AFTER DELETE ON Clientes  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaEliminacionClientes (ClienteID, Nombre, Apellido,  
    Tipo_Operacion)  
    VALUES (OLD.ClienteID, OLD.Nombre, OLD.Apellido, 'DELETE');  
END
```

**Ejercicio 4: Crear un Trigger para Actualizar el Precio Total en la Tabla Pedidos.
Implementar un trigger que actualice el precio total de un pedido cada vez que se
inserte un nuevo detalle de pedido.**

```
CREATE TRIGGER actualizarPrecioTotalPedido  
AFTER INSERT ON DetallesPedido  
FOR EACH ROW  
BEGIN  
    UPDATE Pedidos  
    SET PrecioTotal = (  
        SELECT SUM(p.Precio * dp.Cantidad)  
        FROM DetallesPedido dp  
        JOIN Productos p ON dp.ProductoID = p.ProductoID  
        WHERE dp.PedidoID = NEW.PedidoID  
    )  
    WHERE PedidoID = NEW.PedidoID;  
END;
```

**Ejercicio 5: Crear un Trigger para Validar la Cantidad de Productos en la
Tabla DetallesPedido. Implementar un trigger que valide que la cantidad de
productos en un detalle de pedido no sea negativa.**

```
CREATE TRIGGER validarCantidadProducto  
BEFORE INSERT ON DetallesPedido  
FOR EACH ROW  
BEGIN  
    IF NEW.Cantidad < 0 THEN -- atención: generamos una excepción en este y otros ejer.  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La cantidad de productos no puede  
ser negativa';  
    END IF;  
END
```

Ejercicio 6: Crear un Trigger para Actualizar el Precio de un Producto en la Tabla Productos. Implementar un trigger que actualice el precio de un producto cada vez que se actualice su costo.

```
CREATE TRIGGER actualizarPrecioProducto
BEFORE UPDATE ON Productos
FOR EACH ROW
BEGIN
    IF NEW.Costo <> OLD.Costo THEN
        SET NEW.Precio = NEW.Costo * 1.2;
    END IF;
END
```

Ejercicio 7: Crear un Trigger para Auditar Inserciones en la Tabla Pedidos. Implementar un trigger que registre cada vez que se inserta un nuevo pedido en una tabla de auditoría llamada AuditoriaPedidos.

```
CREATE TABLE AuditoriaPedidos (
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,
    PedidoID INT,
    ClienteID INT,
    FechaPedido DATE,
    FechaInsercion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TRIGGER auditarInsercionPedido
AFTER INSERT ON Pedidos
FOR EACH ROW
BEGIN
    INSERT INTO AuditoriaPedidos (PedidoID, ClienteID, FechaPedido)
    VALUES (NEW.PedidoID, NEW.ClienteID, NEW.FechaPedido);
END
```

Ejercicio 8: Crear un Trigger para Auditar Actualizaciones en la Tabla Pedidos. Implementar un trigger que registre cada vez que se actualiza un pedido en la tabla Pedidos.

```
CREATE TABLE AuditoriaActualizacionPedidos (
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,
    PedidoID INT,
    ClienteIDAntiguo INT,
    ClienteIDNuevo INT,
    FechaActualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TRIGGER auditarActualizacionPedido
AFTER UPDATE ON Pedidos
FOR EACH ROW
BEGIN
    INSERT INTO AuditoriaActualizacionPedidos (PedidoID, ClienteIDAntiguo,
    ClienteIDNuevo)
    VALUES (OLD.PedidoID, OLD.ClienteID, NEW.ClienteID);
END
```

Ejercicio 9: Crear un Trigger para Auditar Eliminaciones en la Tabla Pedidos. Implementar un trigger que registre cada vez que se elimina un pedido de la tabla Pedidos.

```
CREATE TABLE AuditoriaEliminacionPedidos (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    PedidoID INT,  
    ClienteID INT,  
    FechaEliminacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TRIGGER auditarEliminacionPedido  
AFTER DELETE ON Pedidos  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaEliminacionPedidos (PedidoID, ClienteID)  
    VALUES (OLD.PedidoID, OLD.ClienteID);  
END
```

Ejercicio 10: Crear un Trigger para Actualizar el Stock de Productos en la Tabla Productos. Implementar un trigger que actualice el stock de un producto cada vez que se inserte un nuevo detalle de pedido.

```
CREATE TRIGGER actualizarStockProducto  
AFTER INSERT ON DetallesPedido  
FOR EACH ROW  
BEGIN  
    UPDATE Productos  
    SET Stock = Stock - NEW.Cantidad  
    WHERE ProductoID = NEW.ProductoID;  
END
```

Ejercicio 11: Crear un Trigger para Validar el Email en la Tabla Clientes. Implementar un trigger que valide que el email de un cliente tenga un formato correcto antes de insertarlo.

```
CREATE TRIGGER validarEmailCliente  
BEFORE INSERT ON Clientes  
FOR EACH ROW  
BEGIN  
    IF NEW.Email NOT LIKE '%_@_%._%' THEN  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El formato del email es incorrecto';  
    END IF;  
END
```

Ejercicio 12: Crear un Trigger para Auditar Inserciones en la Tabla Productos. Implementar un trigger que registre cada vez que se inserta un nuevo producto en una tabla de auditoría llamada AuditoriaProductos.

```
CREATE TABLE AuditoriaProductos (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductoID INT,  
    NombreProducto VARCHAR(100),  
    Precio DECIMAL(10, 2),  
    FechaInsercion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TRIGGER auditarInsercionProducto  
AFTER INSERT ON Productos  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaProductos (ProductoID, NombreProducto, Precio)  
    VALUES (NEW.ProductoID, NEW.NombreProducto, NEW.Precio);  
END
```

Ejercicio 13: Crear un Trigger para Auditar Actualizaciones en la Tabla Productos. Implementar un trigger que registre cada vez que se actualiza un producto en la tabla Productos.

```
CREATE TABLE AuditoriaActualizacionProductos (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductoID INT,  
    NombreProductoAntiguo VARCHAR(100),  
    NombreProductoNuevo VARCHAR(100),  
    FechaActualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TRIGGER auditarActualizacionProducto  
AFTER UPDATE ON Productos  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaActualizacionProductos (ProductoID, NombreProductoAntiguo,  
    NombreProductoNuevo)  
    VALUES (OLD.ProductoID, OLD.NombreProducto, NEW.NombreProducto);  
END
```

Ejercicio 14: Implementar un trigger que registre cada vez que se elimina un producto de la tabla Productos.

```
CREATE TABLE AuditoriaEliminacionProductos (  
    AuditoriaID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductoID INT,  
    NombreProducto VARCHAR(100),  
    FechaEliminacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TRIGGER auditarEliminacionProducto  
AFTER DELETE ON Productos  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditoriaEliminacionProductos (ProductoID, NombreProducto)  
    VALUES (OLD.ProductoID, OLD.NombreProducto);  
END
```

Ejercicio 15: Implementar un Trigger para Actualizar el Stock de Productos en la Tabla Productos al Eliminar un Detalle de Pedido

```
CREATE TRIGGER actualizarStockProductoAlEliminar  
AFTER DELETE ON DetallesPedido  
FOR EACH ROW  
BEGIN  
    UPDATE Productos  
    SET Stock = Stock + OLD.Cantidad  
    WHERE ProductoID = OLD.ProductoID;  
END
```