



Taller de Triggers en MySQL

Caso 1: Control de Stock de Productos

```
DELIMITER //
```

```
CREATE TRIGGER before_control_stock  
BEFORE INSERT ON ventas  
FOR EACH ROW  
BEGIN  
    DECLARE stock_actual INT;  
  
    SELECT stock INTO stock_actual  
    FROM productos  
    WHERE id = NEW.id_producto;  
  
    IF NEW.cantidad > stock_actual THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Error: No hay suficiente stock para esta venta.';  
    END IF;  
END//
```

```
DELIMITER ;
```

```
INSERT INTO ventas (id_producto, cantidad) VALUES (2, 10) ;  
ERROR 1644 (45000): Error: No hay suficiente stock para esta venta.
```

```
mysql> INSERT INTO ventas (id_producto, cantidad) VALUES (2, 10) ;  
ERROR 1644 (45000): Error: No hay suficiente stock para esta venta.  
mysql> 
```

◆ Caso 2: Registro Automático de Cambios en Salarios

```

DELIMITER//

CREATE TRIGGER registro_automatico_salarios
BEFORE UPDATE ON empleados
FOR EACH ROW
BEGIN

    INSERT INTO historial_salarios ( id_empleado,salario_anterior,salario_nuevo)
    VALUES (OLD.id, OLD.salario,NEW.salario);
END//

DELIMITER ;

```

```

UPDATE empleados SET salario = 1700.00 WHERE id = 1;

```

```

mysql> SELECT * FROM historial_salarios;
+----+-----+-----+-----+-----+
| id | id_empleado | salario_anterior | salario_nuevo | fecha                |
+----+-----+-----+-----+-----+
|  1 |           1 |          1600.00 |          1700.00 | 2025-07-08 12:34:44 |
+----+-----+-----+-----+-----+
1 row in set (0,00 sec)

```

◆ Caso 3: Registro de Eliminaciones en Auditoría

```

DELIMITER //

CREATE TRIGGER after_delete_auditorias
AFTER DELETE ON clientes
FOR EACH ROW
BEGIN

    INSERT INTO clientes_auditoria(id_cliente, nombre, email, fecha_eliminacion)
    VALUES (OLD.id, OLD.nombre, OLD.email, NOW());
END//

DELIMITER ;

```

```

mysql> SELECT * FROM clientes_auditoria;
+----+-----+-----+-----+-----+
| id | id_cliente | nombre      | email                | fecha_eliminacion    |
+----+-----+-----+-----+-----+
|  1 |           1 | Juan Perez | juan@example.com    | 2025-07-08 12:48:01 |
+----+-----+-----+-----+-----+
1 row in set (0,00 sec)

```

◆ Caso 4: Restricción de Eliminación de Pedidos Pendientes

```
CREATE TRIGGER bloquear_eliminacion_pedidos_pendientes
BEFORE DELETE ON pedidos
FOR EACH ROW
BEGIN
    IF OLD.estado = 'pendiente' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: No se puede eliminar
un pedido pendiente.';
    END IF;
END//

DELIMITER ;
```

```
mysql> DELETE FROM pedidos WHERE id = 1;
ERROR 1644 (45000): Error: No se puede eliminar un pedido pendiente.
mysql>
```

```
mysql> SELECT * FROM pedidos;
+----+-----+-----+
| id | cliente   | estado   |
+----+-----+-----+
|  1 | Juan Perez | pendiente |
+----+-----+-----+
1 row in set (0,00 sec)
```