



Taller de Triggers en MySQL

◆ Caso 1: Control de Stock de Productos

```
DELIMITER //
```

```
CREATE TRIGGER bloquear_venta  
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 = 'La cantidad solicitada supera el stock disponible';  
    END IF;  
END;  
  
DELIMITER ;
```

```
INSERT INTO ventas (cantidad)
```

```
INSERT INTO ventas (id, id_producto, cantidad) VALUES (1,1,100);
```

```
mysql> INSERT INTO productos (id, nombre, stock)  
-> VALUES (1, 'Mouse inalámbrico', 50);  
Query OK, 1 row affected (0,01 sec)
```

```
mysql> DESC ventas;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
id_producto	int	YES	MUL	NULL	
cantidad	int	YES		NULL	

```
3 rows in set (0,00 sec)
```

```
mysql> INSERT INTO ventas (id, id_producto, cantidad) VALUES (1,1,100);  
ERROR 1644 (45000): La cantidad solicitada supera el stock disponible
```

◆ Caso 2: Registro Automático de Cambios en Salarios

```
DELIMITER //
CREATE TRIGGER actualizacion_salario
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 = 5000.00 WHERE id = 1;
SELECT * FROM historial_salarios;
```

```
+-----+-----+-----+-----+-----+
| id | id_empleado | salario_anterior | salario_nuevo | fecha                |
+-----+-----+-----+-----+-----+
| 1 | 1 | 2900.00 | 5000.00 | 2025-07-08 12:59:08 |
+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)
```

◆ Caso 3: Registro de Eliminaciones en Auditoría

```
DELIMITER //
CREATE TRIGGER registro_eliminacion
AFTER DELETE ON clientes
FOR EACH ROW
BEGIN
    INSERT INTO clientes_auditoria (id_cliente, nombre, email)
    VALUES (OLD.id, OLD.nombre, OLD.email);
END//

DELIMITER ;

SELECT * FROM clientes_auditoria;
```

```
mysql> DESC clientes;
-> //
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| nombre| varchar(50)   | YES  |     | NULL    |                |
| email | varchar(50)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)

mysql> SELECT * FROM clientes//
+-----+-----+-----+
| id | nombre          | email                      |
+-----+-----+-----+
| 1  | Ana Pérez      | ana.perez@example.com     |
| 2  | Carlos Gómez   | carlos.gomez@example.com  |
+-----+-----+-----+
2 rows in set (0,00 sec)

mysql> DELETE FROM clientes
-> WHERE id = 1;
-> //
Query OK, 1 row affected (0,01 sec)

mysql> SELECT * FROM clientes_auditoria;
-> //
+-----+-----+-----+-----+-----+
| id | id_cliente | nombre          | email                      | fecha_eliminacion |
+-----+-----+-----+-----+-----+
| 1  | 1          | Ana Pérez      | ana.perez@example.com     | 2025-07-08 13:13:17 |
+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)

mysql>
```

◆ Caso 4: Restricción de Eliminación de Pedidos Pendientes

```
DELIMITER //
```

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

```
DELIMITER ;
DELETE FROM pedidos WHERE cliente = 'Ana Pérez';
```