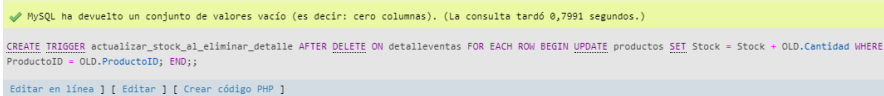


Tema: Triggers en SQL

Alumno: Emanuel Schmer

1. Crear un trigger para actualizar el stock de un producto cuando se elimine un detalle de venta..

```
CREATE TRIGGER actualizar_stock_al_eliminar_detalle  
  
AFTER DELETE ON detalleventas  
  
FOR EACH ROW  
  
BEGIN  
  
    UPDATE productos  
  
    SET Stock = Stock + OLD.Cantidad  
  
    WHERE ProductoID = OLD.ProductoID;  
  
END;
```



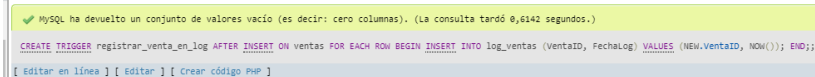
MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,7991 segundos.)

```
CREATE TRIGGER actualizar_stock_al_eliminar_detalle AFTER DELETE ON detalleventas FOR EACH ROW BEGIN UPDATE productos SET Stock = Stock + OLD.Cantidad WHERE ProductoID = OLD.ProductoID; END;;
```

[Editar en línea](#)] [Editar](#)] [Crear código PHP](#)]

2. Crear un trigger para registrar en un log cada vez que se inserta una venta.

```
CREATE TRIGGER registrar_venta_en_log  
  
AFTER INSERT ON ventas  
  
FOR EACH ROW  
  
BEGIN  
  
    INSERT INTO log_ventas (VentaID, FechaLog)  
  
    VALUES (NEW.VentaID, NOW());  
  
END;
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,6142 segundos.)

```
CREATE TRIGGER registrar_venta_en_log AFTER INSERT ON ventas FOR EACH ROW BEGIN INSERT INTO log_ventas (VentaID, FechaLog) VALUES (NEW.VentaID, NOW()); END;;
```

[Editar en línea](#)] [Editar](#)] [Crear código PHP](#)]

3. Crear un trigger para evitar la eliminación de un cliente si tiene ventas registradas.

```
CREATE TRIGGER evitar_eliminar_cliente  
  
BEFORE DELETE ON clientes  
  
FOR EACH ROW  
  
BEGIN  
  
    DECLARE venta_count INT;  
  
    SELECT COUNT(*) INTO venta_count  
  
    FROM ventas  
  
    WHERE ClienteID = OLD.ClienteID;  
  
    IF venta_count > 0 THEN  
  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede eliminar el cliente porque tiene  
ventas registradas.';  
  
    END IF;  
  
END;
```

```
MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,7299 segundos.)  
  
CREATE TRIGGER evitar_eliminar_cliente BEFORE DELETE ON clientes FOR EACH ROW BEGIN DECLARE venta_count INT; SELECT COUNT(*) INTO venta_count FROM ventas WHERE ClienteID =  
OLD.ClienteID; IF venta_count > 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede eliminar el cliente porque tiene ventas registradas.'; END IF; END;;  
  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

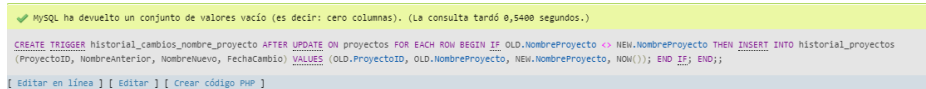
4. Crear un trigger para enviar un mensaje de advertencia cuando un proyecto se asigna a más de 5 empleados.

```
CREATE TRIGGER advertir_asignacion_empleados  
  
BEFORE INSERT ON asignaciones  
  
FOR EACH ROW  
  
BEGIN  
  
    DECLARE empleado_count INT;  
  
    SELECT COUNT(*) INTO empleado_count  
  
    FROM asignaciones  
  
    WHERE ProyectoID = NEW.ProyectoID;  
  
    IF empleado_count >= 5 THEN  
  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Advertencia: El proyecto tiene más de 5  
empleados asignados.';  
  
    END IF;  
  
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). ((La consulta tardó 0,3712 segundos.)  
  
CREATE TRIGGER advertir_asignacion_empleados BEFORE INSERT ON asignaciones FOR EACH ROW BEGIN DECLARE empleado_count INT; SELECT COUNT(*) INTO empleado_count FROM asignaciones WHERE  
ProyectoID = NEW.ProyectoID; IF empleado_count >= 5 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Advertencia: El proyecto tiene más de 5 empleados asignados.'; END IF; END;;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

5. Crear un trigger que registre el historial de cambios en el nombre de un proyecto.

```
CREATE TRIGGER historial_cambios_nombre_proyecto  
  
AFTER UPDATE ON proyectos  
  
FOR EACH ROW  
  
BEGIN  
  
    IF OLD.NombreProyecto <> NEW.NombreProyecto THEN  
  
        INSERT INTO historial_proyectos (ProyectoID, NombreAnterior, NombreNuevo, FechaCambio)  
  
        VALUES (OLD.ProyectoID, OLD.NombreProyecto, NEW.NombreProyecto, NOW());  
  
    END IF;  
  
END;
```




MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,5400 segundos.)

```
CREATE TRIGGER historial_cambios_nombre_proyecto AFTER UPDATE ON proyectos FOR EACH ROW BEGIN IF OLD.NombreProyecto <> NEW.NombreProyecto THEN INSERT INTO historial_proyectos (ProyectoID, NombreAnterior, NombreNuevo, FechaCambio) VALUES (OLD.ProyectoID, OLD.NombreProyecto, NEW.NombreProyecto, NOW()); END IF; END;
```

[Editar en línea] [Editar] [Crear código PHP]

6. Crear un trigger para que el monto total de una venta no pueda ser negativo.

```
CREATE TRIGGER validar_monto_total_venta  
  
BEFORE INSERT ON ventas  
  
FOR EACH ROW  
  
BEGIN  
  
    IF NEW.MontoTotal < 0 THEN  
  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El monto total de la venta no puede ser negativo';  
  
    END IF;  
  
END;
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,4692 segundos.)

```
CREATE TRIGGER validar_monto_total_venta BEFORE INSERT ON ventas FOR EACH ROW BEGIN IF NEW.MontoTotal < 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El monto total de la venta no puede ser negativo'; END IF; END;
```

[Editar en línea] [Editar] [Crear código PHP]

7. Crear un trigger para evitar que el salario de un empleado sea mayor que el salario del gerente de su departamento.

```
CREATE TRIGGER trigger_salario_empleado

BEFORE INSERT ON empleados

FOR EACH ROW

BEGIN

    DECLARE salario_gerente INT;

    SELECT Salario INTO salario_gerente

    FROM empleados

    WHERE DepartamentoID = NEW.DepartamentoID

    AND EmpleadoID = (SELECT EmpleadoID FROM departamentos WHERE DepartamentoID =

NEW.DepartamentoID);

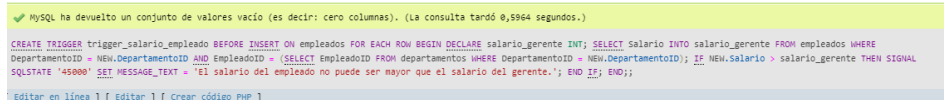
    IF NEW.Salario > salario_gerente THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El salario del empleado no puede ser mayor

que el salario del gerente.';

    END IF;

END;
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,5964 segundos.)

```
CREATE TRIGGER trigger_salario_empleado BEFORE INSERT ON empleados FOR EACH ROW BEGIN DECLARE salario_gerente INT; SELECT salario INTO salario_gerente FROM empleados WHERE DepartamentoID = NEW.DepartamentoID AND EmpleadoID = (SELECT EmpleadoID FROM departamentos WHERE DepartamentoID = NEW.DepartamentoID); IF NEW.Salario > salario_gerente THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El salario del empleado no puede ser mayor que el salario del gerente.'; END IF; END;
```

[Editar en línea](#) | [Editar](#) | [Crear código PHP](#)

8. Crear un trigger para registrar el historial de cambios en los métodos de pago.

```
CREATE TRIGGER trg_historial_cambio_metodo_pago

AFTER UPDATE ON mediosdepago

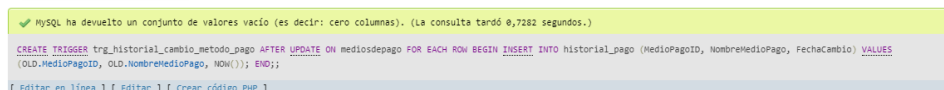
FOR EACH ROW

BEGIN

    INSERT INTO historial_pago (MedioPagoID, NombreMedioPago, FechaCambio)

    VALUES (OLD.MedioPagoID, OLD.NombreMedioPago, NOW());

END;
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,7282 segundos.)

```
CREATE TRIGGER trg_historial_cambio_metodo_pago AFTER UPDATE ON mediosdepago FOR EACH ROW BEGIN INSERT INTO historial_pago (MedioPagoID, NombreMedioPago, FechaCambio) VALUES (OLD.MedioPagoID, OLD.NombreMedioPago, NOW()); END;
```

[Editar en línea](#) | [Editar](#) | [Crear código PHP](#)

9. Crear un trigger que impida la asignación de un proyecto si el presupuesto total excede el presupuesto disponible del departamento.

```
CREATE TRIGGER trg_presupuesto_proyecto
BEFORE INSERT ON proyectos
FOR EACH ROW
BEGIN
    DECLARE presupuesto_actual INT;
    DECLARE presupuesto_disponible INT;
    SELECT SUM(Presupuesto) INTO presupuesto_actual
    FROM proyectos
    WHERE DepartamentoID = NEW.DepartamentoID;
    SELECT presupuesto_disponible INTO presupuesto_disponible
    FROM departamentos
    WHERE DepartamentoID = NEW.DepartamentoID;
    IF presupuesto_actual + NEW.Presupuesto > presupuesto_disponible THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Presupuesto total del departamento excedido.';
    END IF;
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,2569 segundos.)

CREATE TRIGGER trg_presupuesto_proyecto BEFORE INSERT ON proyectos FOR EACH ROW BEGIN DECLARE presupuesto_actual INT; DECLARE presupuesto_disponible INT; SELECT SUM(Presupuesto) INTO presupuesto_actual FROM proyectos WHERE DepartamentoID = NEW.DepartamentoID; SELECT presupuesto_disponible INTO presupuesto_disponible FROM departamentos WHERE DepartamentoID = NEW.DepartamentoID; IF presupuesto_actual + NEW.Presupuesto > presupuesto_disponible THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Presupuesto total del departamento excedido.'; END IF; END;;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

10. Crear un trigger para registrar la fecha y hora de la última actualización en un proyecto.

```
ALTER TABLE proyectos ADD COLUMN FechaUltimaActualizacion DATETIME DEFAULT NULL;
```

```
CREATE TRIGGER trg_actualizahora_proyectos
```

```
BEFORE UPDATE ON proyectos
```

```
FOR EACH ROW
```

```
BEGIN
```

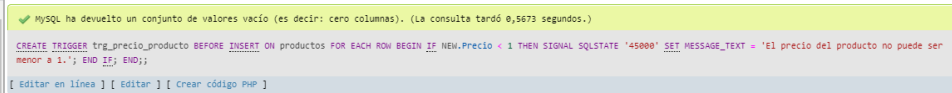
```
    SET NEW.FechaUltimaActualizacion = NOW();
```

```
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,7299 segundos.)  
CREATE TRIGGER trg_actualizahora_proyectos BEFORE UPDATE ON proyectos FOR EACH ROW BEGIN SET NEW.FechaUltimaActualizacion = NOW(); END;;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

11. Crear un trigger para controlar que un producto no pueda tener un precio menor a 1.

```
CREATE TRIGGER trg_precio_producto  
BEFORE INSERT ON productos  
FOR EACH ROW  
BEGIN  
    IF NEW.Precio < 1 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'El precio del producto no puede ser menor a 1!';  
    END IF;  
END;
```



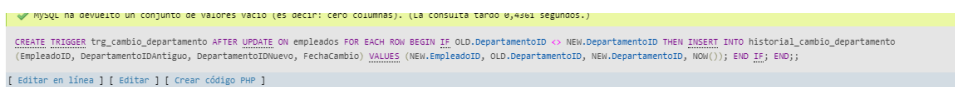
MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,5673 segundos.)

```
CREATE TRIGGER trg_precio_producto BEFORE INSERT ON productos FOR EACH ROW BEGIN IF NEW.Precio < 1 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El precio del producto no puede ser menor a 1.'; END IF; END;;
```

[Editar en línea] [Editar] [Crear código PHP]

12. Crear un trigger para registrar cuando un empleado cambia de departamento.

```
CREATE TRIGGER trg_cambio_departamento  
AFTER UPDATE ON empleados  
FOR EACH ROW  
BEGIN  
    IF OLD.DepartamentoID <> NEW.DepartamentoID THEN  
        INSERT INTO historial_cambio_departamento (EmpleadoID, DepartamentoIDAntiguo,  
DepartamentoIDNuevo, FechaCambio)  
VALUES (NEW.EmpleadoID, OLD.DepartamentoID, NEW.DepartamentoID, NOW());  
    END IF;  
END;
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,4401 segundos.)

```
CREATE TRIGGER trg_cambio_departamento AFTER UPDATE ON empleados FOR EACH ROW BEGIN IF OLD.DepartamentoID <> NEW.DepartamentoID THEN INSERT INTO historial_cambio_departamento (EmpleadoID, DepartamentoIDAntiguo, DepartamentoIDNuevo, FechaCambio) VALUES (NEW.EmpleadoID, OLD.DepartamentoID, NEW.DepartamentoID, NOW()); END IF; END;;
```

[Editar en línea] [Editar] [Crear código PHP]

13. Crear un trigger que evite la eliminación de un proyecto si tiene empleados asignados.

```
CREATE TRIGGER trg_evitar_eliminacion_proyecto
BEFORE DELETE ON proyectos
FOR EACH ROW
BEGIN
    DECLARE num_asignaciones INT;

    SELECT COUNT(*) INTO num_asignaciones
    FROM asignaciones
    WHERE ProyectoID = OLD.ProyectoID;

    IF num_asignaciones > 0 THEN
        SIGNAL SQLSTATE '45000'

        SET MESSAGE_TEXT = 'No se puede eliminar un proyecto con empleados asignados.';
    END IF;
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,5801 segundos.)
CREATE TRIGGER trg_evitar_eliminacion_proyecto BEFORE DELETE ON proyectos FOR EACH ROW BEGIN DECLARE num_asignaciones INT; SELECT COUNT(*) INTO num_asignaciones FROM asignaciones WHERE ProyectoID = OLD.ProyectoID; IF num_asignaciones > 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede eliminar un proyecto con empleados asignados.'; END IF; END;;
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

14. Crear un trigger que registre la eliminación de una venta en un log de auditoría.

```
CREATE TRIGGER trg_eliminacion_venta
AFTER DELETE ON ventas
FOR EACH ROW
BEGIN
    INSERT INTO log_auditoria (VentaID, FechaEliminacion)
    VALUES (OLD.VentaID, NOW());
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,1295 segundos.)
CREATE TRIGGER trg_eliminacion_venta AFTER DELETE ON ventas FOR EACH ROW BEGIN INSERT INTO log_auditoria (VentaID, FechaEliminacion) VALUES (OLD.VentaID, NOW()); END;;
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

15. Crear un trigger para controlar que no se pueda asignar un empleado a más de 3 proyectos simultáneamente.

```
CREATE TRIGGER trg_control_proyectos_empleado  
  
BEFORE INSERT ON asignaciones  
  
FOR EACH ROW  
  
BEGIN  
  
    DECLARE num_proyectos INT;  
  
    SELECT COUNT(*) INTO num_proyectos  
  
    FROM asignaciones  
  
    WHERE EmpleadoID = NEW.EmpleadoID;  
  
    IF num_proyectos >= 3 THEN  
  
        SIGNAL SQLSTATE '45000'  
  
        SET MESSAGE_TEXT = 'El empleado ya está asignado a 3 proyectos.';  
  
    END IF;  
  
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,6118 segundos.)  
  
CREATE TRIGGER trg_control_proyectos_empleado BEFORE INSERT ON asignaciones FOR EACH ROW BEGIN DECLARE num_proyectos INT; SELECT COUNT(*) INTO num_proyectos FROM asignaciones WHERE  
EmpleadoID = NEW.EmpleadoID; IF num_proyectos >= 3 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El empleado ya está asignado a 3 proyectos.'; END IF; END;;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

16. Crear un trigger para evitar que se cambie el gerente de un departamento si el gerente actual tiene proyectos asignados.

```
ALTER TABLE `departamentos`  
  
ADD COLUMN `GerenteID` INT(10) NULL;  
  
DELIMITER //  
  
CREATE TRIGGER trigerevitar_actualizar_gerente  
  
BEFORE UPDATE ON `departamentos`  
  
FOR EACH ROW  
  
BEGIN  
  
    IF (SELECT COUNT(*) FROM proyectos WHERE DepartamentoID = OLD.DepartamentoID AND  
EXISTS  
  
        (SELECT 1 FROM asignaciones WHERE EmpleadoID = OLD.GerenteID AND ProyectoID =  
proyectos.ProyectoID)) > 0 THEN  
  
        SIGNAL SQLSTATE '45000'  
  
        SET MESSAGE_TEXT = 'No se puede cambiar el gerente porque tiene proyectos asignados.';  
  
END IF;  
  
END //  
  
DELIMITER ;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,7895 segundos.)  
  
CREATE TRIGGER trigerevitar_actualizar_gerente BEFORE UPDATE ON `departamentos` FOR EACH ROW BEGIN IF (SELECT COUNT(*) FROM proyectos WHERE DepartamentoID = OLD.DepartamentoID AND  
EXISTS (SELECT 1 FROM asignaciones WHERE EmpleadoID = OLD.GerenteID AND ProyectoID = proyectos.ProyectoID)) > 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede cambiar el  
gerente porque tiene proyectos asignados.'; END IF; END;  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

17. Crear un trigger para registrar cada vez que un cliente realice una compra por más de \$500.

```
CREATE TRIGGER trg_compra_cliente  
  
AFTER INSERT ON ventas  
  
FOR EACH ROW  
  
BEGIN  
  
    DECLARE monto_total INT;  
  
    SELECT SUM(Cantidad * Precio) INTO monto_total  
  
    FROM detalleventas  
  
    WHERE VentaID = NEW.VentaID;  
  
    IF monto_total > 500 THEN  
  
        INSERT INTO historial_compras (ClienteID, MontoTotal, FechaCompra)  
  
        VALUES (NEW.ClienteID, monto_total, NOW());  
  
    END IF;  
  
END;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,5204 segundos.)

```
CREATE TRIGGER trg_compra_cliente AFTER INSERT ON ventas FOR EACH ROW BEGIN DECLARE monto_total INT; SELECT SUM(Cantidad * Precio) INTO monto_total FROM detalleventas WHERE VentaID =  
NEW.VentaID; IF monto_total > 500 THEN INSERT INTO historial_compras (ClienteID, MontoTotal, FechaCompra) VALUES (NEW.ClienteID, monto_total, NOW()); END IF; END;
```

[Editar en línea] [Editar] [Crear código PHP]

18. Crear un trigger que actualice el presupuesto disponible del departamento cuando se agregue un nuevo proyecto.

```
CREATE TRIGGER trg_actualizar_presupuesto_departamento  
  
AFTER INSERT ON proyectos  
  
FOR EACH ROW  
  
BEGIN  
  
    UPDATE departamentos  
  
    SET presupuesto_disponible = presupuesto_disponible - NEW.Presupuesto  
  
    WHERE DepartamentoID = NEW.DepartamentoID;  
  
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,6784 segundos.)  
  
CREATE TRIGGER trg_actualizar_presupuesto_departamento AFTER INSERT ON proyectos FOR EACH ROW BEGIN UPDATE departamentos SET presupuesto_disponible = presupuesto_disponible -  
NEW.Presupuesto WHERE DepartamentoID = NEW.DepartamentoID; END;  
  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

19. Crear un trigger que impida que se reduzca el presupuesto de un proyecto a menos del 50% del presupuesto original.

```
CREATE TRIGGER trg_presupuesto_proyecto_minimo  
  
BEFORE UPDATE ON proyectos  
  
FOR EACH ROW  
  
BEGIN  
  
    DECLARE presupuesto_original INT;  
  
    SELECT Presupuesto INTO presupuesto_original  
  
    FROM proyectos  
  
    WHERE ProyectoID = OLD.ProyectoID;  
  
    IF NEW.Presupuesto < presupuesto_original * 0.5 THEN  
  
        SIGNAL SQLSTATE '45000'  
  
        SET MESSAGE_TEXT = 'No se puede reducir el presupuesto a menos del 50% del original!';  
  
    END IF;  
  
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,2560 segundos.)  
  
CREATE TRIGGER trg_presupuesto_proyecto_minimo BEFORE UPDATE ON proyectos FOR EACH ROW BEGIN DECLARE presupuesto_original INT; SELECT Presupuesto INTO presupuesto_original FROM  
proyectos WHERE ProyectoID = OLD.ProyectoID; IF NEW.Presupuesto < presupuesto_original * 0.5 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se puede reducir el presupuesto a menos  
del 50% del original!'; END IF; END;  
  
[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

20. Crear un trigger para registrar la fecha y hora cuando se actualiza el salario de un empleado.

```
CREATE TRIGGER trg_actualizacion_salario
```

```
AFTER UPDATE ON empleados
```

```
FOR EACH ROW
```

```
BEGIN
```

```
IF OLD.Salario <> NEW.Salario THEN
```

```
INSERT INTO historial_salario (EmpleadoID, SalarioAnterior, SalarioNuevo, FechaActualizacion)
```

```
VALUES (NEW.EmpleadoID, OLD.Salario, NEW.Salario, NOW());
```

```
END IF;
```

```
END;
```

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,3079 segundos.)

CREATE TRIGGER trg_actualizacion_salario AFTER UPDATE ON empleados FOR EACH ROW BEGIN IF OLD.Salario <> NEW.Salario THEN INSERT INTO historial_salario (EmpleadoID, SalarioAnterior, SalarioNuevo, FechaActualizacion) VALUES (NEW.EmpleadoID, OLD.Salario, NEW.Salario, NOW()); END IF; END;;

[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```