

DocTic

Objetos almacenados

1. **Verificar que un comentario no tenga una autoreferencia. Por ejemplo, que un comentario con id 1 no tenga el mismo subIdComentario, ya que implica que no se pueda borrar el registro sin perder la integridad referencial.**

DELIMITER \$\$

```
CREATE TRIGGER verificarSubIdComentario
BEFORE INSERT ON comentario
FOR EACH ROW
BEGIN
    IF NEW.idComentario = NEW.subIdComentario THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El idComentario no puede ser igual
al subIdComentario';
    END IF;
END $$
```

DELIMITER ;

2. **Asegurar que un usuario no pueda valorar el mismo documento más de una vez.**

```
select * from valora;
DELIMITER $$
CREATE TRIGGER validarCantidadValoraciones
BEFORE INSERT ON valora
FOR EACH ROW
BEGIN
    DECLARE existe INT;

    SELECT count(idValora) INTO existe
    FROM valora
    WHERE idUsuario = NEW.idUsuario
    AND idDocumento = NEW.idDocumento;

    IF existe > 0 THEN
        SIGNAL SQLSTATE '45000' SET
        MESSAGE_TEXT = "El usuario ya ha valorado el documento";
    END IF;
```

```
END $$
```

```
DELIMITER ;
```

- 3. Se garantiza que no se utilicen contraseñas ya utilizadas anteriormente por el mismo usuario.**

```
DELIMITER $$
CREATE TRIGGER validarGuardarContrasena
BEFORE INSERT ON contrasena
FOR EACH ROW
BEGIN
    DECLARE existe INT;
    SELECT count(idHistorial) INTO existe
    FROM contrasena AS c
    WHERE c.contrasena = NEW.contrasena
    AND idUsuario = NEW.idUsuario;

    IF existe > 0 THEN
        SIGNAL SQLSTATE '45000' SET
        MESSAGE_TEXT = "No puede usar una contraseña antigua";
    END IF;
END $$
DELIMITER ;
```

- 4. Actualiza el estado de las contraseñas a 'inactiva' para un usuario dado, asegurándose de que solo se modifiquen aquellas contraseñas que están actualmente activas.**

```
DELIMITER $$

CREATE PROCEDURE actualizarContrasenaInactiva(IN id_usuario INT, IN
id_historial INT)
BEGIN
    UPDATE contrasena
    SET estado = 'inactiva'
    WHERE idUsuario = id_usuario
    AND estado = 'activa'
    AND idHistorial != id_historial;
END $$

DELIMITER ;
```

5. Se garantiza que el estado de una nueva contraseña debe ser “activa”.

```
DELIMITER $$
```

```
CREATE TRIGGER validarEstadoContrasena  
BEFORE INSERT ON contrasena  
FOR EACH ROW
```

```
    BEGIN
```

```
        IF NEW.estado = 'inactiva' THEN
```

```
            SIGNAL SQLSTATE '45000' SET
```

```
            MESSAGE_TEXT = "El estado de una nueva contraseña no puede  
ser inactivo";
```

```
        END IF;
```

```
    END $$
```

```
DELIMITER ;
```