

En este laboratorio trabajaremos triggers en PL/SQL.

Trigger

Un trigger permite definir un programa que el motor de base de datos ejecuta implícitamente cuando ocurre un evento DML en los datos

```
CREATE [OR REPLACE] TRIGGER nombre_trigger
momento evento ON tabla
[FOR EACH ROW [WHEN condición]]
bloque de PL/SQL
-- El bloque comienza con DECLARE o BEGIN
```

Ejemplos:

- Mantener actualizada la tabla totalSalarios

```
CREATE OR REPLACE TRIGGER trCambioSalarios
AFTER INSERT OR UPDATE OR DELETE ON Empleado
BEGIN
    UPDATE totalSalarios
    SET    total = (SELECT SUM(salario) FROM empleado);
END;
```

- Evitar salarios menores al mínimo actual

```
CREATE OR REPLACE TRIGGER trValor
BEFORE INSERT ON Empleado FOR EACH ROW
DECLARE minSal NUMBER(10) DEFAULT 0;
BEGIN
    SELECT MIN(salario) INTO minSal FROM empleado;
    IF :NEW.salario < minSal THEN
        Raise_application_error(-20000, 'Salario fuera de rango: ' || :NEW.salario);
    END IF;
END;
```

- Tener registro del usuario que realiza las actualizaciones de los datos

```
CREATE OR REPLACE TRIGGER trUser
BEFORE INSERT OR UPDATE ON Empleado
FOR EACH ROW
BEGIN
    SELECT USER INTO :NEW.Usuario FROM DUAL;
END;
```

- Evitar que se asigne un salario menor que el actual

```
CREATE OR REPLACE TRIGGER trUser
BEFORE UPDATE ON Empleado
FOR EACH ROW
```

```
BEGIN
  IF :NEW.salario < :OLD.salario THEN
    Raise_application_error(-20100, 'El nuevo salario (':NEW.salario:') debe
    ser mayor que el anterior (':OLD.salario:')');
  END IF;
END;
```

Ejercicios

Usando el esquema de base de datos creado y modificado en los laboratorios anteriores, escriba los comandos para dar respuesta a los requerimientos especificados en los siguientes ejercicios. En la respuesta de cada punto incluya en el script el código para crear el trigger y las sentencias que permiten hacer las pruebas de su funcionamiento. En esas sentencias debe haber al menos una que permite verificar que el trigger hace la función solicitada, y en los puntos 1 y 2, sentencias que muestren que los casos válidos operan sin problema.

1. Cree un trigger que asegure que cuando se **inserta** un registro en la tabla **matricula** la fecha de expedición quede registrada con la **fecha y hora actual**.
Hint: Recuerde que se pueden asignar valores a los atributos de **:NEW** y los valores de **:NEW** son los que quedarán guardados en la tabla.
2. Cree un trigger que evite que se actualicen **los datos y las infracciones** de los partes que ya tienen pagos realizados.
3. Agregue a la tabla **Persona** el atributo **SaldoMultas**. Cree los triggers necesarios para mantener actualizado este valor en cada persona.
4. Se requiere un trigger que evite registrar un pago por un valor mayor que el saldo de la deuda del parte. El saldo de la deuda se calcula encontrando el total de la multa del parte (la suma de los valores de sus infracciones) y restando el total de los pagos ya realizados.

Al finalizar la sesión, cada estudiante debe enviar el script a mcpabon@javerianacali.edu.co.