1. Crea **una tabla** llamada alumnos con las siguientes columnas.

Tabla alumnos:

* id (number(6)) PRIMARY KEY
* nombre (varchar2(20))
* apellido1 (varchar2(20))
* apellido2 (varchar2(20))
* nota (number(2))

CREATE TABLE ALUMNOS(

ID NUMBER(6) PRIMARY KEY,

NOMBRE VARCHAR2(20),

APELLIDO1 VARCHAR2(20),

APELLIDO2 VARCHAR2(20),

NOTA NUMBER(2)

);

Una vez creada la tabla escriba **dos triggers** con las siguientes características:

* Trigger 1: trigger\_check\_nota\_before\_insert
  + Se ejecuta sobre la tabla alumnos.
  + Se ejecuta *antes* de una operación de *inserción*.
  + Si el nuevo valor de la nota que se quiere insertar es **negativo, se guarda como 0.**
  + Si el nuevo valor de la nota que se quiere insertar es **mayor que 10, se guarda como 10.**
  + Si el nuevo valor de la nota que se quiere insertar tiene **números decimales, se debe redondear hacia abajo.**
* Trigger 2 : trigger\_check\_nota\_before\_update
  + Se ejecuta sobre la tabla alumnos.
  + Se ejecuta *antes* de una operación de *actualización*.
  + Si el nuevo valor de la nota que se quiere actualizar es negativo, se guarda como 0.
  + Si el nuevo valor de la nota que se quiere actualizar es mayor que 10, se guarda como 10.
  + Si el nuevo valor de la nota que se quiere actualizar tiene números decimales, se debe redondear hacia abajo.

CREATE OR REPLACE TRIGGER SEGURIDAD\_INSERT

BEFORE INSERT ON ALUMNOS FOR EACH ROW

BEGIN

IF (:NEW.NOTA<0) THEN :NEW.NOTA:=0;

ELSIF (:NEW.NOTA>10) THEN :NEW.NOTA:=10;

ELSE :NEW.NOTA:=FLOOR(:NEW.NOTA-0.1);

END IF;

END SEGURIDAD;

/

CREATE OR REPLACE TRIGGER SEGURIDAD\_UPDATE

BEFORE UPDATE ON ALUMNOS FOR EACH ROW

BEGIN

IF (:NEW.NOTA<0) THEN :NEW.NOTA:=0;

ELSIF (:NEW.NOTA>10) THEN :NEW.NOTA:=10;

ELSE :NEW.NOTA:=FLOOR(:NEW.NOTA-0.1);

END IF;

END SEGURIDAD\_UPDATE;

/

INSERT INTO ALUMNOS VALUES (1, NOMBRE, APELLIDO1, APELLIDO2, -2);

Una vez creados los triggers escriba UN ÚNICO PROCEDIMIENTO que inserte y actualice registros sobre la tabla alumnos y verifica que los *triggers* se están ejecutando correctamente.

CREATE OR REPLACE PROCEDURE P\_ALUMNOS

(V\_ID IN ALUMNO.ID%TYPE)

IS

V\_NOMBRE ALUMNO.NOMBRE%TYPE,

V\_APELLIDO1 ALUMNO.APELLIDO1%TYPE,

V\_APELLIDO2 ALUMNO.APELLIDO2%TYPE,

V\_NOTA ALUMNO.NOTA%TYPE

BEGIN

IF EXISTS (

SELECT FROM ALUMNOS

WHERE ID=&V\_ID

)

BEGIN

UPDATE ALUMNOS SET

NOTA = &V\_NOTA

WHERE ID= V\_ID;

DBMS\_OUTPUT.PUT\_LINE( 'ALUMNO ' || V\_ID || ' ACTUALIZADO' );

END

ELSE

BEGIN

DBMS\_OUTPUT.PUT\_LINE( 'EL ALUMNO NO EXISTE, INGRESE SUS DATOS' );

INSERT INTO ALUMNOS VALUES (&V\_ID , &V\_NOMBRE , &V\_APELLIDO1 , &V\_APELLIDO2 , &V\_NOTA );

DBMS\_OUTPUT.PUT\_LINE( 'ALUMNO ' || V\_ID || ' AGREGADO' );

END

END P\_ALUMNOS;

/