

Tarea BD06

Antonio Jiménez Sevilla

ACTIVIDAD 1

```
CREATE OR REPLACE
PROCEDURE MUEVE_FAMILIA(ID_FAMILIA_ORIGEN NUMBER, ID_FAMILIA_DESTINO
NUMBER)
IS
--DECLARACIÓN DE VARIABLES

    RESULTADO BOOLEAN;
    PADRE NUMBER;
    TYPE CURSOR_FAMILIAS IS REF CURSOR RETURN FAMILIAS%ROWTYPE;
    CFAMILIAS CURSOR_FAMILIAS;
    FAMILIA CFAMILIAS%ROWTYPE;
    EXISTE_F_ORIGEN BOOLEAN;
    EXISTE_F_DESTINO BOOLEAN;
    FAMILIA_PERTENECE NUMBER;

--FUNCIÓN QUE COMPRUEBA SI LA FAMILIA DESTINO ES HIJA DE LA FAMILIA ORIGEN

    FUNCTION COMPRUEBA_SI_ES_HIJA(ID_F_ORIGEN NUMBER,
ID_F_DESTINO NUMBER)
        RETURN BOOLEAN IS ES_HIJA BOOLEAN;
    BEGIN
        SELECT FAMILIA INTO PADRE FROM FAMILIAS WHERE IDENTIFICADOR =
ID_F_DESTINO;
        IF (PADRE IS NULL) OR (PADRE = ID_F_DESTINO) THEN

            RETURN FALSE;

        ELSE

            IF PADRE = ID_F_ORIGEN THEN

                RETURN TRUE;
            ELSE

                RETURN COMPRUEBA_SI_ES_HIJA (ID_F_ORIGEN, PADRE);

            END IF;

        END IF;

    END FUNCTION;

--HABILITO SALIDA PARA DEBUG

    DBMS_OUTPUT.ENABLE( 1000000 );
```

--COMPRUEBO SI EXISTE LA FAMILIA ORIGEN

```
        EXISTE_F_ORIGEN := TRUE;
        OPEN CFAMILIAS FOR SELECT * FROM FAMILIAS WHERE IDENTIFICADOR
= ID_FAMILIA_ORIGEN;
        FETCH CFAMILIAS INTO FAMILIA;
        IF CFAMILIAS%NOTFOUND THEN
            DBMS_OUTPUT.PUT_LINE('LA FAMILIA ORIGEN NO EXISTE');
            EXISTE_F_ORIGEN := FALSE;
        END IF;
        CLOSE CFAMILIAS;
```

--COMPRUEBO SI EXISTE LA FAMILIA DESTINO

```
        EXISTE_F_DESTINO := TRUE;
        OPEN CFAMILIAS FOR SELECT * FROM FAMILIAS WHERE IDENTIFICADOR =
ID_FAMILIA_DESTINO;
        FETCH CFAMILIAS INTO FAMILIA;
        IF CFAMILIAS%NOTFOUND THEN
            DBMS_OUTPUT.PUT_LINE('LA FAMILIA DESTINO NO EXISTE');
            EXISTE_F_DESTINO := FALSE;
        END IF;
        CLOSE CFAMILIAS;
```

--SI EXISTE TANTO LA FAMILIA ORIGEN COMO LA DESTINO CONTINUÓ CON EL RESTO DE ACCIONES

```
        IF EXISTE_F_ORIGEN = TRUE AND EXISTE_F_DESTINO = TRUE THEN
```

--COMPRUEBO QUE LA FAMILIA DESTINO NO SEA YA HIJA DE LA FAMILIA ORIGEN

```
        RESULTADO := COMPRUEBA_SI_ES_HIJA(ID_FAMILIA_ORIGEN,
ID_FAMILIA_DESTINO);
        IF RESULTADO = TRUE THEN
            DBMS_OUTPUT.PUT_LINE('FAMILIA DESTINO ES HIJA DE FAMILIA
ORIGEN. NO SE PODRÁ REALIZAR EL PROCESO');
        ELSE
```

--CONTINUÓ CON EL PROCESO

```
            DBMS_OUTPUT.PUT_LINE('REALIZADO EL PROCESO DE REASIGNACIÓN. ');
            UPDATE FAMILIAS SET FAMILIA = ID_FAMILIA_DESTINO,
OFICINA = NULL WHERE IDENTIFICADOR = ID_FAMILIA_ORIGEN;
            COMMIT;
        ELSE
            DBMS_OUTPUT.PUT_LINE('NO SE REALIZA EL PROCESO DE REASIGNACIÓN, YA
QUE LA FAMILIA DESTINO ES PADRE DE LA ORIGEN. ');
```

```
        END IF;
```

```
    END IF;
```

```
END IF;
```

```
END;
```

ACTIVIDAD 2

```
CREATE OR REPLACE TRIGGER restricciones_agentes
BEFORE INSERT OR UPDATE ON agentes
FOR EACH ROW
DECLARE
    mensaje VARCHAR2(1000);
    fallo BOOLEAN;
BEGIN
    mensaje := '';
    fallo := false;
    IF (:new.usuario = :new.clave) THEN
        mensaje := mensaje || 'El usuario y la clave de un agente no pueden ser iguales';
        fallo := true;

    END IF;
    IF (:new.habilidad < 0 or :new.habilidad > 9) THEN
        mensaje := mensaje || 'La habilidad de un agente debe estar comprendida entre 0
y 9 (ambos inclusive)';
        fallo := true;

    END IF;
    IF (:new.categoria < 0 or :new.categoria > 2) THEN
        mensaje := mensaje || 'La categoría de un agente sólo puede ser igual a 0, 1 o 2';
        fallo := true;

    END IF;
    IF (:new.oficina is not null and :new.categoria != 2) THEN
        mensaje := mensaje || 'Si un agente pertenece a una oficina directamente, su
categoría debe ser igual 2';
        fallo := true;

    END IF;
    IF (:new.oficina is null and :new.categoria = 2) THEN
        mensaje := mensaje || 'Si un agente no pertenece a una oficina directamente, su
categoría no puede ser 2';
        fallo := true;

    END IF;
    IF (:new.oficina is null and :new.familia is null) THEN
        mensaje := mensaje || 'No puede haber agentes que no pertenezcan a una oficina
o a una familia';
        fallo := true;

    END IF;
```

```
IF (:new.oficina is not null and :new.familia is not null) THEN
mensaje := mensaje || 'No puede haber agentes que pertenezcan a una oficina y a
una familia a la vez';
fallo := true;
```

```
END IF;
```

```
IF (fallo = true) THEN
```

```
RAISE_APPLICATION_ERROR(-20100, mensaje);
```

```
END IF;
```

```
END;
```

