Tarea BD06

Actividad 1

**La creación del procedimiento:**

CREATE OR REPLACE PROCEDURE CambiarAgentesFamilia(id\_FamiliaOrigen NUMBER, id\_FamiliaDestino NUMBER) AS

v\_familia\_origen VARCHAR(255);

v\_familia\_destino VARCHAR(255);

v\_numero\_agentes NUMBER := 0;

v\_NumAgentes NUMBER := 0;

v\_hay\_errores BOOLEAN := FALSE;

v\_FamiliaOrigenCount NUMBER := 0;

v\_FamiliaDestinoCount NUMBER := 0;

BEGIN

--comprobamos que las familias existen

SELECT COUNT(\*) INTO v\_FamiliaOrigenCount FROM familias WHERE identificador = id\_FamiliaOrigen;

SELECT COUNT(\*) INTO v\_FamiliaDestinoCount FROM familias WHERE identificador = id\_FamiliaDestino;

-- Comprobar la existencia de la familia de origen

IF v\_FamiliaOrigenCount = 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'La familia de origen especificada no existe.');

END IF;

-- Comprobar la existencia de la familia de destino

IF v\_FamiliaDestinoCount = 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'La familia de destino especificada no existe.');

END IF;

--y no son iguales

IF id\_FamiliaOrigen = id\_FamiliaDestino THEN

v\_hay\_errores := TRUE;

RAISE\_APPLICATION\_ERROR(-20001, 'Las familias no pueden ser iguales');

END IF;

--buscamos nombres de las familias ya que existen

SELECT nombre INTO v\_familia\_origen FROM familias WHERE identificador = id\_FamiliaOrigen;

SELECT nombre INTO v\_familia\_destino FROM familias WHERE identificador = id\_FamiliaDestino;

-- si no tengo errores Cambio de los agentes de la familia origen a la familia destino

IF v\_hay\_errores = FALSE THEN

UPDATE agentes SET familia = id\_FamiliaDestino WHERE familia = id\_FamiliaOrigen;

v\_NumAgentes := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE('Se han trasladado '||v\_NumAgentes||' agentes de la familia '||v\_familia\_origen||' a la familia '||v\_familia\_destino);

END IF;

END;

/

**Ejecución del procedimiento:**

DECLARE

BEGIN

CambiarAgentesFamilia(11, 111);

EXCEPTION

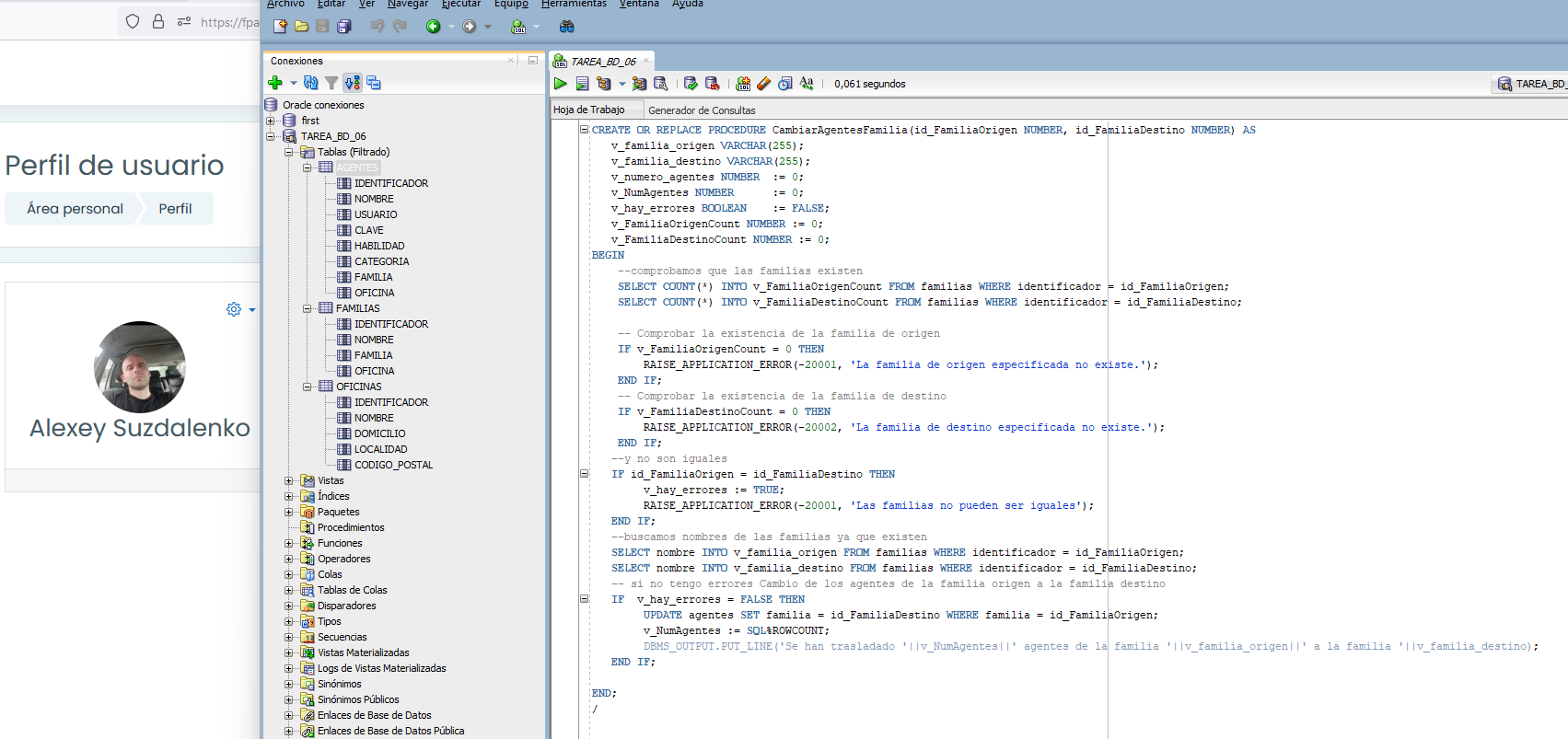
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLCODE || ' - ' || SQLERRM);

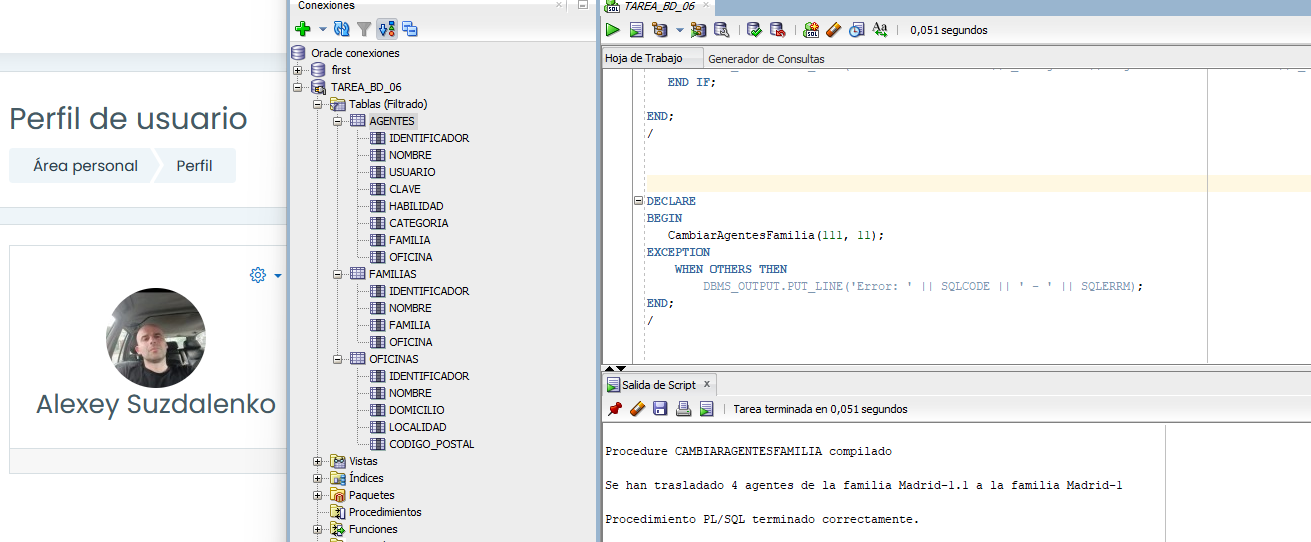
END;

/

Captura procedimiento:



Captura ejecución:



Actividad 2.

De las restricciones pedidas se implementan en el esquema a la hora de crear tablas agentes:

La longitud de la clave de un agente no puede ser inferior a 6.

La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).

La categoría de un agente sólo puede ser igual a 0, 1 o 2.

CREATE TABLE agentes (

identificador NUMBER(6) NOT NULL PRIMARY KEY,

nombre VARCHAR2(60) NOT NULL,

usuario VARCHAR2(20) NOT NULL UNIQUE,

clave VARCHAR2(20) NOT NULL CHECK (LENGTH(clave) >= 6),

habilidad NUMBER(1) NOT NULL CHECK (habilidad BETWEEN 0 AND 9),

categoria NUMBER(1) NOT NULL CHECK (categoria IN (0, 1, 2)),

familia NUMBER(6) REFERENCES familias,

oficina NUMBER(6) REFERENCES oficinas

);

Y el en triger se implementan todas las restricciones:

CREATE OR REPLACE TRIGGER restricciones\_agentes

BEFORE INSERT OR UPDATE ON agentes

FOR EACH ROW

DECLARE

v\_oficina\_exist NUMBER := 0;

v\_familia\_exist NUMBER := 0;

BEGIN

-- Verificar longitud de la clave

IF LENGTH(:NEW.clave) < 6 THEN

RAISE\_APPLICATION\_ERROR(-20005, 'La longitud de la clave debe ser al menos 6 caracteres.');

END IF;

-- Verificar habilidad

IF :NEW.habilidad NOT BETWEEN 0 AND 9 THEN

RAISE\_APPLICATION\_ERROR(-20006, 'La habilidad del agente debe estar entre 0 y 9.');

END IF;

-- Verificar categoría

IF :NEW.categoria NOT IN (0, 1, 2) THEN

RAISE\_APPLICATION\_ERROR(-20007, 'La categoría del agente solo puede ser 0, 1 o 2.');

END IF;

-- Verificar categoría y pertenencia

IF :NEW.categoria = 2 THEN

IF :NEW.familia IS NOT NULL THEN

RAISE\_APPLICATION\_ERROR(-20008, 'Un agente de categoría 2 no puede pertenecer a ninguna familia.');

END IF;

IF :NEW.oficina IS NULL THEN

RAISE\_APPLICATION\_ERROR(-20009, 'Un agente de categoría 2 debe pertenecer a una oficina.');

END IF;

ELSIF :NEW.categoria = 1 THEN

IF :NEW.oficina IS NOT NULL THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Un agente de categoría 1 no puede pertenecer a ninguna oficina.');

END IF;

IF :NEW.familia IS NULL THEN

RAISE\_APPLICATION\_ERROR(-20011, 'Un agente de categoría 1 debe pertenecer a una familia.');

END IF;

END IF;

-- Verificar pertenencia a oficina o familia pero no a ambas

SELECT COUNT(\*) INTO v\_oficina\_exist FROM oficinas WHERE identificador = :NEW.oficina;

SELECT COUNT(\*) INTO v\_familia\_exist FROM familias WHERE identificador = :NEW.familia;

IF v\_oficina\_exist = 1 AND v\_familia\_exist = 1 THEN

RAISE\_APPLICATION\_ERROR(-20012, 'Un agente no puede pertenecer a una oficina y a una familia al mismo tiempo.');

END IF;

END;

/

Captura de creación trigger:

