Actividad 1.

Crear un procedimiento que permita cambiar a todos los agentes de una familia determinada (familia origen) a otra familia (familia destino).

El procedimiento tendrá la siguiente cabecera **CambiarAgentesFamilia(id_FamiliaOrigen, id_FamiliaDestino),** donde cada uno de los argumentos corresponde a un identificador de Familia. Cambiará la columna Identificador de Familia de todos los agentes, de la tabla AGENTES, que pertenecen a la Familia con código **id_FamiliaOrigen** por el código **id_FamiliaDestino**

Previamente comprobará que ambas familias existen y que no son iguales.

Para la comprobación de la existencia de las familias se puede utilizar un cursor variable, o contar el número de filas y en caso de que no exista, se visualizará el mensaje correspondiente mediante una excepción del tipo RAISE_APPLICATION_ERROR. También se mostrará un mensaje en caso de que ambos argumentos tengan el mismo valor.

El procedimiento visualizará el mensaje "Se han trasladado XXX agentes de la familia XXXXXX a la familia ZZZZZZ" donde XXX es el número de agentes que se han cambiado de familia, XXXXXX es el nombre de la familia origen y ZZZZZZ es el nombre de la familia destino.

```
create or replace procedure CambiarAgentesFamilia( id_FamiliaOrigen familias.identificador%TYPE, id_FamiliaDestino familias.identificador%TYPE)
is
    type c_compruebaFamilia is ref cursor return FAMILIAS%ROWTYPE; --Cursor variable
    c_familia c_compruebaFamilia;
    filaFamilia FAMILIAS%rowtype;
    nombreFamiliaOrigen familias.nombre%TYPE;    -- Para guardar el nombre de la familia de Origen
    nombreFamiliaDestino familias.nombre%TYPE;   -- Para guardar el nombre de la familia de Destino
    contadorCambios number;                      -- Para guardar el numero de cambios que se hacen
    error_numero number;
    error_mensaje varchar2(200);
begin
    if id_FamiliaOrigen = id_FamiliaDestino then
        raise_application_error(-20000, 'Los parametros son iguales');
    end if;
    open c_familia for select * from FAMILIAS where identificador=id_familiaorigen;
        fetch c_familia into filaFamilia;
        if c_familia%notfound then
            raise_application_error(-20001, 'Id Origen no existe');
        end if;
        nombreFamiliaOrigen := filaFamilia.nombre; -- Obtener el nombre de la Familia Origen
    close c_familia;
    open c_familia for select * from FAMILIAS where identificador=id_FamiliaDestino;
        fetch c_familia into filaFamilia;
        if c_familia%notfound then
            raise_application_error(-20002, 'Id Destino no existe');
        end if;
        nombreFamiliaDestino  := filaFamilia.nombre; -- Obtener el nombre de la Familia Destino
    close c_familia;
    select count(*) into contadorCambios from AGENTES where familia=id_FamiliaOrigen;
    if contadorcambios > 0 then
        -- Cambiamos la familia de la tabla agentes
        update AGENTES set familia=id_FamiliaDestino where familia=id_FamiliaOrigen;
        dbms_output.put_line('--------------------------------------------------------------------------------');
        dbms_output.put_line('Se han trasladado ' || contadorCambios ||
                             ' agentes de la familia ' || nombrefamiliaorigen ||
                             ' a la familia ' || nombrefamiliadestino);
        dbms_output.put_line('--------------------------------------------------------------------------------');
    else
        raise_application_error(-20003, 'En la tabla Agentes no hay ning?n registro de la familia origen');
    end if;
    commit;
exception
    when others then
        error_numero := SQLCODE;
        error_mensaje := substr(SQLERRM, 1, 200);
        dbms_output.put_line('ERROR: Mensaje: ' || error_mensaje || 'Codigo:' || to_char(error_numero));
        rollback;
end;
```

```
create or replace procedure CambiarAgentesFamilia( id_FamiliaOrigen familias.identificador%TYPE,
id_FamiliaDestino familias.identificador%TYPE)
is
   -- Declaraci�n de variables y cursores --
   type c_compruebaFamilia is ref cursor return FAMILIAS%ROWTYPE; --Cursor variable
   c_familia c_compruebaFamilia;
   filaFamilia FAMILIAS%rowtype;

   nombreFamiliaOrigen familias.nombre%TYPE;   -- Para guardar el nombre de la familia de Origen
   nombreFamiliaDestino familias.nombre%TYPE;  -- Para guardar el nombre de la familia de Destino
   contadorCambios number;                     -- Para guardar el numero de cambios que se hacen

   error_numero number;
   error_mensaje varchar2(200);

begin
   -- Comprobamos que son diferentes los ID
   if id_FamiliaOrigen = id_FamiliaDestino then
      raise_application_error(-20000, 'Los parametros son iguales');
   end if;
   -- Comprobamos que el id_familiaOrigen existe
   open c_familia for select * from FAMILIAS where identificador=id_familiaorigen;
      fetch c_familia into filaFamilia;
      if c_familia%notfound then
         raise_application_error(-20001, 'Id Origen no existe');
      end if;
      nombreFamiliaOrigen := filaFamilia.nombre; -- Obtener el nombre de la Familia Origen
   close c_familia;
    -- Comprobamos que el id_familiaDestino existe
   open c_familia for select * from FAMILIAS where identificador=id_FamiliaDestino;
      fetch c_familia into filaFamilia;
      if c_familia%notfound then
         raise_application_error(-20002, 'Id Destino no existe');
      end if;
      nombreFamiliaDestino  := filaFamilia.nombre; -- Obtener el nombre de la Familia Destino
   close c_familia;
   -- Contamos los registros que tiene Agentes que familia son igual al parametro Origen
   select count(*) into contadorCambios from AGENTES where familia=id_FamiliaOrigen;
   if contadorcambios > 0 then
      -- Cambiamos la familia de la tabla agentes
      update AGENTES set familia=id_FamiliaDestino where familia=id_FamiliaOrigen;
      dbms_output.put_line('--------------------------------------------------------------------------------');
      dbms_output.put_line('Se han trasladado ' || contadorCambios ||
                           ' agentes de la familia ' || nombrefamiliaorigen ||
                           ' a la familia ' || nombrefamiliadestino);
      dbms_output.put_line('--------------------------------------------------------------------------------');
   else
      raise_application_error(-20003, 'En la tabla Agentes no hay ning�n registro de la familia origen');
   end if;

   commit;
```

```
exception
   when others then
      error_numero := SQLCODE;
      error_mensaje := substr(SQLERRM, 1, 200);
      dbms_output.put_line('ERROR: Mensaje: ' || error_mensaje || 'Codigo:' || to_char(error_numero));
      rollback;
end;
```

Actividad 2.

Queremos controlar algunas restricciones a la hora de trabajar con 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.
- Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
- Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.
- Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.

Se pide crear un disparador para asegurar estas restricciones. El disparador deberá lanzar todos los errores que se puedan producir en su ejecución mediante errores que identifiquen con un mensaje adecuado por qué se ha producido dicho error.

C: > Users > JRBlanco > Desktop > Actividad 2 JRBlanco v2.sql > ...

```
create or replace trigger res_agentes before insert or update on AGENTES
for each row
begin
    -- ------------------------------------------------------------
    -- La longitud de la clave de un agente no puede ser inferior a 6.
    if (length(:new.clave) < 6) then
        raise_application_error(-20101,'{La clave no puede ser inferior de 6 caracteres}');
    end if;
    -- ------------------------------------------------------------------------
    --La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
    if (:new.habilidad < 0 or :new.habilidad > 9) then
        raise_application_error(-20102,'{La habilidad tiene que estar entre 0 y 9}');
    end if;
    -- ------------------------------------------------------------
    -- La categor?a de un agente s?lo puede ser igual a 0, 1 o 2.
    if (:new.categoria < 0 or :new.categoria > 2) then
        raise_application_error(-20103,'{La categoria solo puede ser 0, 1 o 2}');
    end if;
    -- Si un agente tiene categor?a 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
    if (:new.categoria=2) then
        if (:new.familia is not null) then
            raise_application_error(-20104,'{Si la categoria es 2, NO puede tener familia}');
        end if;
        if (:new.oficina is null) then
            raise_application_error(-20105, '{Si la categoria es 2 debe de pertener a una ofinina}');
        end if;
    end if;
    -- ------------------------------------------------------------------------------------------------
    -- Si un agente tiene categor?a 1 no puede pertenecer a ninguna oficina y debe pertenecer  a una familia.
    if (:new.categoria=1) then
        if (:new.oficina is not null) then
            raise_application_error(-20106,'{Si la categoria es 1 no pude de pertener a una ofinina}');
        end if;
        if (:new.familia is null) then
            raise_application_error(-20107,'{Si la categoria es 1 debe de pertener a una familia}');
        end if;
    end if;
    -- ------------------------------------------------------------------------------------------------
    -- Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez.
    if (:new.oficina is null) and (:new.familia is null) then
        raise_application_error(-20108,'{Un agente tiene que pertenecer a una familia y a una oficina}');
    elsif (:new.oficina is not null) and (:new.familia is not null) then
        raise_application_error(-20109,'{Un agente no puede pertenecer a una familia y a una oficina a la vez}');
    end if;

end;
```

```
create or replace trigger res_agentes before insert or update on AGENTES
for each row

begin
   -- ---------------------------------------------------------------
   -- La longitud de la clave de un agente no puede ser inferior a 6.
   -- ---------------------------------------------------------------
   if (length(:new.clave) < 6) then
      raise_application_error(-20101,'{La clave no puede ser inferior de 6 caracteres}');
   end if;

   -- ------------------------------------------------------------------------------
   --La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
   -- ------------------------------------------------------------------------------
   if (:new.habilidad < 0 or :new.habilidad > 9) then
      raise_application_error(-20102,'{La habilidad tiene que estar entre 0 y 9}');
   end if;

   -- ------------------------------------------------------------
   -- La categoría de un agente sólo puede ser igual a 0, 1 o 2.
   -- ------------------------------------------------------------
   if (:new.categoria < 0 or :new.categoria > 2) then
      raise_application_error(-20103,'{La categoria solo puede ser 0, 1 o 2}');
   end if;

   -- Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
   if (:new.categoria=2) then
      if (:new.familia is not null) then
         raise_application_error(-20104,'{Si la categoria es 2, NO puede tener familia}');
      end if;
      if (:new.oficina is null) then
         raise_application_error(-20105, '{Si la categoria es 2 debe de pertener a una ofinina}');
      end if;
   end if;

   -- ---------------------------------------------------------------------------------------------------
   -- Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer  a una familia.
   -- ---------------------------------------------------------------------------------------------------
   if (:new.categoria=1) then
      if (:new.oficina is not null) then
         raise_application_error(-20106,'{Si la categoria es 1 no pude de pertener a una ofinina}');
      end if;
      if (:new.familia is null) then
         raise_application_error(-20107,'{Si la categoria es 1 debe de pertener a una familia}');
      end if;
   end if;

   -- ---------------------------------------------------------------------------------------------
   -- Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez.
   -- ---------------------------------------------------------------------------------------------
   if (:new.oficina is null) and (:new.familia is null) then
```

```
    raise_application_error(-20108,'{Un agente tiene que pertenecer a una familia y a una oficina}');
  elsif (:new.oficina is not null) and (:new.familia is not null) then
    raise_application_error(-20109,'{Un agente no puede pertenecer a una familia y a una oficina a la vez}');
  end if;

end;
```

Algunas de las restricciones implementadas con el disparador se pueden incorporar a la definición del esquema de la tabla utilizando el Lenguaje de Definición de Datos (Check,Unique,..).Identifica cuáles son y con qué tipo de restricciones las implementarías.

```
-- ------------------------------------------------------------
-- La longitud de la clave de un agente no puede ser inferior a 6.
-- ------------------------------------------------------------
alter table AGENTES add constraint chk_tam_clave check(length(ltrim(rtrim(clave)))>6);
```

DDL

alter table AGENTES add constraint chk_tam_clave check(length(ltrim(rtrim(clave)))>6);

Explicación: Obliga a que las claves tengan mas de 6 caracteres.

```
-- ------------------------------------------------------------------------------
--La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
-- ------------------------------------------------------------------------------
alter table AGENTES add constraint chk_habilidad check(habilidad between 0 and 9);
```

DDL

alter table AGENTES add constraint chk_habilidad check(habilidad between 0 and 9);

Explicación: Obliga a que en el campo habilidad estén entre 0 y 9.

```
-- ------------------------------------------------------------------------------------------------
-- La categoría de un agente sólo puede ser igual a 0, 1 o 2.
-- Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
-- Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer  a una familia.
-- Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez.
-- ------------------------------------------------------------------------------------------------
alter table AGENTES add constraint chk_cat_ofi_fam check((categoria=2 and familia is null and oficina is not null)
                                                      or (categoria=1 and familia is not null and oficina is null)
                                                      or (categoria=0) and ((oficina is not null and familia is null) or
                                                                            (oficina is null and familia is not null)));
```

DDL

alter table AGENTES add constraint chk_cat_ofi_fam check(
(categoria=2 and familia is null and oficina is not null)
or (categoria=1 and familia is not null and oficina is null)
or (categoria=0) and ((oficina is not null and familia is null) or
(oficina is null and familia is not null)));

Explicación: Para el resto de las restricciones como están vinculadas unas con otras he creado un único CHECK.

Como solo pueden existe tres tipos de categorías (0, 1 y 2) pues le digo al CHECK que si categoria es el

numero 2 el campo familia tiene que ser NULL y el de oficina tiene que tener dato. Para la categoria 1 exactamente igual pero al reves, es decir que si es 1 familia tiene que tener datos y oficina tiene que ser NULL.
Pero en cambio para la categoría 0 es a la única que se le permite o que familia tenga datos y oficina no, o al reves que familia no tenga datos pero oficina si.

Fichero completo SQL

```
-- ------------------------------------------------------------
-- La longitud de la clave de un agente no puede ser inferior a 6.
-- ------------------------------------------------------------
alter table AGENTES add constraint chk_tam_clave check(length(ltrim(rtrim(clave)))>6);
-- ------------------------------------------------------------------------
--La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
-- ------------------------------------------------------------------------
alter table AGENTES add constraint chk_habilidad check(habilidad between 0 and 9);

-- ------------------------------------------------------------------------------------------------------------
-- La categor?a de un agente s?lo puede ser igual a 0, 1 o 2.
-- Si un agente tiene categor?a 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
-- Si un agente tiene categor?a 1 no puede pertenecer a ninguna oficina y debe pertenecer  a una familia.
-- Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez.
-- ------------------------------------------------------------------------------------------------------------
alter table AGENTES add constraint chk_cat_ofi_fam check((categoria=2 and familia is null and oficina is not null)
                                                      or (categoria=1 and familia is not null and oficina is null)
                                                      or (categoria=0) and ((oficina is not null and familia is null) or
                                                                            (oficina is null and familia is not null)));

-- ------------------------------------------------------------------------------------------------------------------------------------
-- EST?N TODOS AGRUPADOS EN EL CHECK ANTERIOR
-- ------------------------------------------------------------
-- La categor?a de un agente s?lo puede ser igual a 0, 1 o 2.
-- ------------------------------------------------------------
-- alter table AGENTES add constraint chk_categoria check(categoria between 0 and 2);
-- ------------------------------------------------------------------------------------------------------------
-- Si un agente tiene categor?a 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
-- ------------------------------------------------------------------------------------------------------------
-- alter table AGENTES add constraint chk_cat2 check(categoria=2 and familia is null and oficina is not null);
-- ------------------------------------------------------------------------------------------------------------
-- Si un agente tiene categor?a 1 no puede pertenecer a ninguna oficina y debe pertenecer  a una familia.
-- ------------------------------------------------------------------------------------------------------------
-- alter table AGENTES add constraint chk_cat1 check(categoria=1 and familia is not null and oficina is null);
-- ------------------------------------------------------------------------------------------------------------
-- Todos los agentes deben pertenecer  a una oficina o a una familia pero nunca a ambas a la vez.
-- ------------------------------------------------------------------------------------------------------------
-- alter table AGENTES add constraint chk_pertene check ((oficina is not null and familia is null) or (oficina is null and familia is not null));
```

-- --
-- La longitud de la clave de un agente no puede ser inferior a 6.
-- --
alter table AGENTES add constraint chk_tam_clave check(length(ltrim(rtrim(clave)))>6);
-- --
--La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
-- --
alter table AGENTES add constraint chk_habilidad check(habilidad between 0 and 9);

-- --
-- La categoría de un agente sólo puede ser igual a 0, 1 o 2.
-- Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
-- Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.
-- Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.
-- --
alter table AGENTES add constraint chk_cat_ofi_fam check(
(categoria=2 and familia is null and oficina is not null)
or (categoria=1 and familia is not null and oficina is null)
or (categoria=0) and ((oficina is not null and familia is null) or

```
                                                    (oficina is null and familia is not null)));
```

NOTA: Ajunto a este documento los tres ficheros SQL